SHARIFAH FOOD INVENTORY MANAGEMENT SYSTEM

TAN YUAN JIE

UNIVERSITI TUNKU ABDUL RAHMAN

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name : Tan Yuan Jie

ID No. : 1802629

Date : 21/4/2022

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"SHARIFAH FOOD INVENTORY MANAGEMENT SYSTEM"** was prepared by **TAN YUAN JIE** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature :

Supervisor : Dr. Sugumaran a/l Nallusamy

Date : 22/4/2022

Signature :

Co-Supervisor :

Date :

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Sugumaran a/l Nallusamy for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement. This project would not be possible to complete without all of them.

ABSTRACT

Sharifah Food is a food processing company that specialises in the ready meals market. Over the years, Sharifah Food had to manually take orders from its customer through WhatsApp and record the sales data in Excel sheets. Nevertheless, it was discovered that it became difficult to keep track of all sales, orders, and inventory as the company grew. Therefore, this project aims to implement a web-based computerised inventory management system that helps the company streamline the inventory management process. Moreover, the e-commerce platform will be built using the Progressive Web Application (PWA) to provide the customers with a quick and straightforward way to place orders.

The basic functionalities of the inventory management system include product stock, promotional packages, orders, and discount management. On the other hand, the basic functionalities of the e-commerce platform include products and promotions browsing, viewing an item, adding to the shopping cart, search items, checkout and payments.

The methodology adopted in this project is the Rapid Application Development (RAD) prototyping methodology, and the development process is divided into three main phases. After completing the system implementation, the systems were evaluated through unit, integration, system usability and user acceptance tests. The system usability and user acceptance tests were conducted remotely. The test results showed that the inventory management system and the e-commerce platform are easy to use and not complex. However, the users would require a little time to be familiar with the inventory management system. In conclusion, the project has achieved all the planned objectives and the systems were successfully developed and deployed.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

ABC	Activity-Based Costs
API	Application Programming Interface
CLI	Command-Line Interface
CSS	Cascading Style Sheets
EOQ	Economic Order Quantity
HML	High-Medium-Low
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JWT	JSON Web Token
MLM	Multi-level Marketing
MTV	Model-Template-Views
MVC	Model View Controller
POS	Point of Sale
PaaS	Platform as a Service
PWA	Progressive Web Application
RAD	Rapid Application Development
RDBMS	Relational Database Management System
REST	Representational State Transfer
SDLC	Software Development Life Cycle
SKU	Stock Keeping Unit
SME	Small-to-medium Enterprise
SQL	Structured Query Language
SS	Safety Stock
SUS	System Usability Scale
UI	User Interface
UX	User Experience
WBS	Work Breakdown Structure
XSS	Cross-Site Scripting

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

In today's highly crowded and competitive business world, all companies strive to drive their business effectively and efficiently. When it comes to the effectiveness and efficiency of the business, data plays a crucial role as it can help the business perform analytics on marketing trends and customer behaviour. Often, inventory lies at the heart of the business operation, especially for the retail industry. With proper inventory management, we can collect data such as stock availability, sales demand, customer data, etc to gain more insight into the business. Therefore, an inventory management system is crucial to streamline the business of the company effectively and efficiently.

To fulfill the customers' demand, most of the companies maintain and manage their inventories. Nonetheless, it is the biggest challenge for businesses, especially those that use the manual inventory management system. Inaccurate inventory management can lead to backorders and loss of profits for the company. Thus, a computerised inventory management system was introduced to overcome this problem.

This project aims to develop a web and mobile computerised inventory management system for the company Sharifah Food. It also aims to integrate the system with an e-commerce platform to enable the company to take orders through the e-commerce platform.

This chapter discusses the project's preliminary planning, including the problem background, problem statement, project objectives, proposed solution, proposed approach, and project scope.

1.2 Problem Background

Sharifah Food is a food processing company that specialises in the ready meals market. Its aim is to provide quick, delicious and halal ready meals that can be prepared in just 3 minutes (Sharifah Food, n.d.). The company has more than 20 products, including “Nasi Briyani”, “Rendang Dendeng Daging”, “Nasi Goreng”, and other ready-made meals of local Malaysian cuisine. There are four main categories of its products: “Ready-To-Eat”, “Ready-To-Cook”, “Paste”, and “Others”. Figure 1.1 shows the sample of the “Ready-To-Cook” products sold by Sharifah Food.

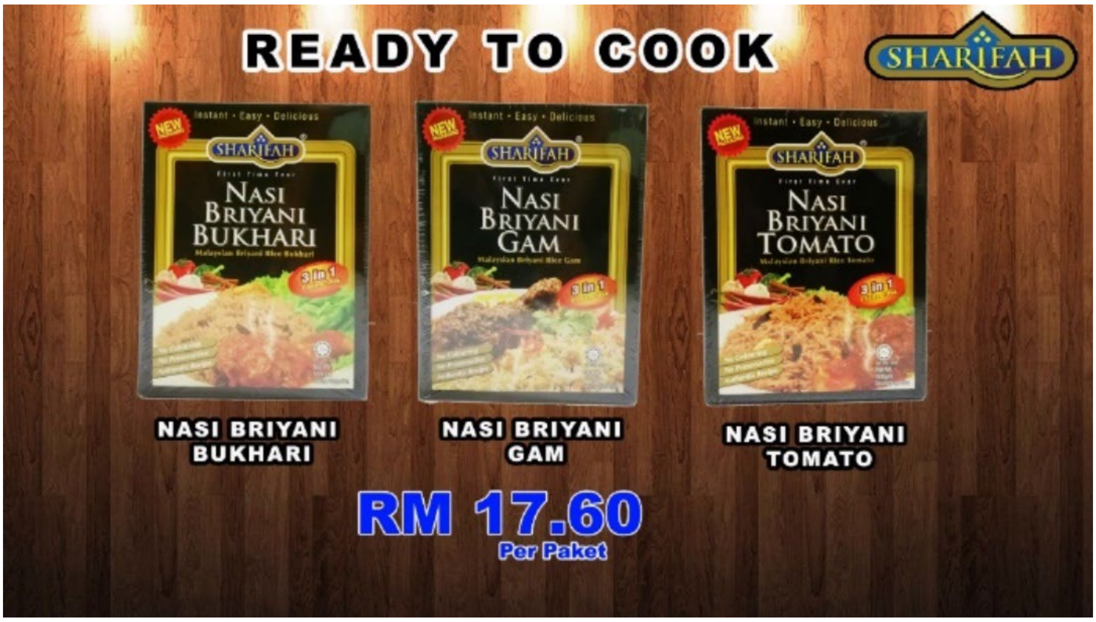


Figure 1.1: Sample Products Sold by Sharifah Food

Sharifah Food applies Multi-level Marketing (MLM) system to sell its products. The customers can apply as agents or dropshippers to promote and sell the company's products by purchasing the products at discounts. For instance, an agent and a dropshipper can receive 30 percent and 20 percent discounts, respectively, if they order a large quantity of products. Sharifah Food currently uses a manual inventory management system, specifically WhatsApp and Excel, to take orders and store the sales data. However, as the company grows, the number of orders to be processed increases, and the company struggles to keep track of all sales and inventory.

Nowadays, many SMEs, including Sharifah Food, still employ the manual inventory management system. Nevertheless, manual inventory management system has many drawbacks. According to Chan (2017), poor inventory management has become the most prevalent problem for SMEs in Malaysia. Most SMEs cannot gain an advantage over their competitors because of the deficiency of inventory management skills. In addition, a company with poor inventory management may face underproduction, overproduction, and inaccurate inventory (Mat & Kadir, 2016).

James Ng'ang'a (2013) proposed four factors affecting inventory management effectiveness: the bureaucratic procurement procedures, funding, skills and knowledge acquired by staff, and documentation and store records. Regarding the situation of Sharifah Food, the most critical factor is documentation and store records. Proper documentation is significant, especially to avoid inaccuracy in the inventory records. Although the manual inventory management system can help to document inventory and sales data, it requires an extended processing time and is very likely to cause the discrepancy of records. Therefore, it is not practical for a growing company to adopt a manual inventory management system.

In fact, to improve inventory management of Sharifah Food, a computerised inventory management system that can take orders, manage orders from customers, agents and dropshippers, manage inventory stocks, as well as sales reporting is indeed required. Furthermore, since the company mainly uses WhatsApp to take orders, the system will be integrated with an e-commerce platform to provide a quick and straightforward way to view the company's products catalogue and place orders from the e-commerce platform.

In 2015, there were nearly 14.5 million smartphone users in Malaysia, which is predicted to increase to beyond 20 million by 2020 (Müller, 2021). Moreover, behavioural studies have found that 40 percent of Malaysians look over their smartphones at 30-minute intervals and that they spend an average of USD 2000 per year online (Kaur, et al., 2016). Therefore, a mobile e-commerce application will be developed to link to the computerised inventory management system to facilitate the process of taking orders, as it makes it easier for the customers to place orders from their mobile devices. The development of this system will undoubtedly free the company from the problems of manual inventory management system.

1.3 Problem Statement

The current problems with manual inventory management systems can be classified into two main areas. In the problem statements listed below, the first three statements address the problems in the inventory management system used by Sharifah Food currently. On the other hand, the last problem statement addresses the limitation of the general-purpose computerised inventory management system.

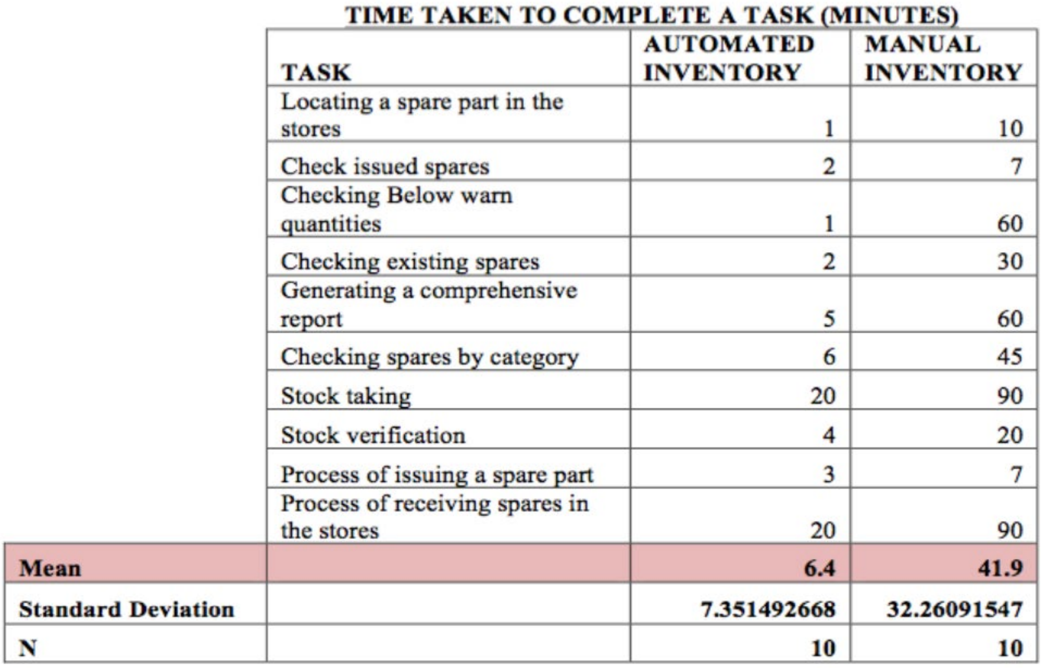
1.3.1 Work-concentrated and time-consuming to record every transaction

Anulika, et al. (2020) and Shinde, et al. (2018) claimed that the manual inventory management system requires constant observation to ensure that every transaction is clarified and recorded. Besides, It would also require monitoring of the inventory to ensure that the goods are maintained at an appropriate level. So, it requires a lot of work to manage the inventory manually.

Furthermore, manually creating backup data is incredibly difficult and time-consuming (Khobragade, et al., 2018). If the system that stores the spreadsheet data fails, all the data could be lost. Therefore, a regular manual backup process is required, which would cost a lot of time.

One of the critical problems of the manual inventory management system is the repetitive and tedious data entry (Zoho Inventory Team, n.d.). When the order volume is large, it is tedious to enter the order information repeatedly. For example, if the customer has bought the goods before, the stockist has to search for the customer's information and enter the order information manually each time. Also, one of Sharifah Food's problems is that they need to calculate the price for each order. This undoubtedly consumes a lot of unnecessary time that could be used for other tasks beneficial to the company.

Based on the efficiency analysis presented by Muyumba and Phiri (2017), the computerised inventory management system takes much less time (6.4 minutes on average) than the manual inventory management system, which takes 41.9 minutes on average. Figure 1.2 shows the efficiency analysis between automated and manual inventories presented by Muyumba and Phiri (2017).



TIME TAKEN TO COMPLETE A TASK (MINUTES)

	TASK	AUTOMATED INVENTORY	MANUAL INVENTORY
	Locating a spare part in the stores	1	10
	Check issued spares	2	7
	Checking Below warn quantities	1	60
	Checking existing spares	2	30
	Generating a comprehensive report	5	60
	Checking spares by category	6	45
	Stock taking	20	90
	Stock verification	4	20
	Process of issuing a spare part	3	7
	Process of receiving spares in the stores	20	90
Mean		**6.4**	**41.9**
Standard Deviation		**7.351492668**	**32.26091547**
N		**10**	**10**

Figure 1.2: Time Taken to Complete a Task Between Automated and Manual Inventories (Muyumba & Phiri, 2017)

1.3.2 Inaccurate inventory records ensue from human errors

When dealing with all the sales manually, it is very likely to make incautious errors. The problem of inaccurate inventory records may arise due to poor communication among staff (Abisoye, et al., 2013). For instance, if a staff processed an order and took the items from inventory without the others knowing, the entry of that order will not be noted. Others will then assume that the items are still available, resulting in inaccurate inventory records. In addition, the person in charge may ignore recording some transactions that he or she deems unnecessary (Anulika, et al., 2020). However, trivial transaction records may affect the entire inventory.

Chuang and Oliva (2015) reveal that a firm with 29 percent inaccurate inventory records results in an estimated 10 percent profit loss. In addition, the study by Shabani, et al. (2021) found that a loss in revenue of more than 1 percent of sales and more than 3 percent of gross profit can result from inaccurate inventory records. These studies indicate that documentation of accurate inventory records is crucial to maintain or escalate the company's profit.

1.3.3 Difficult to perform sales analysis

Sales analysis is critical to understanding business performance and predicting demand for goods. Moreover, it is becoming popular due to the advent of Big Data. Therefore, it is important to perform sales analysis to gain insights into customer segmentation, performance, popularity of products, etc. However, since Sharifah Food uses the Excel spreadsheet to record each transaction, it cannot generate the sales report immediately. f the company wants to analyse its sales, it has to go through various types of data from the spreadsheet, perform calculations and generate sales reports (Zoho Inventory Team, n.d.). This process, in turn, is related to the problem of work-concentrated and time-consuming as the manual preparation of sales reports often takes time. In addition, the manual preparation of sales reports also carries the risk of errors in the calculations.

1.3.4 Lack of feature to manage agents and dropshippers

Since Sharifah Food implements an MLM system consisting of agents and dropshippers, it is essential to manage them. However, it has been found that the popular general-purpose inventory management systems such as Shopify (Shopify, 2021), Odoo (Odoo, 2021), Ecwid (Ecwid, 2021) and WooCommerce (WooCommerce, 2021) do not support the feature to manage agents and dropshippers. Therefore, the computerised inventory management system for Sharifah Food should include this feature so that the client can easily manage the orders and requests of agents and dropshippers as well as view the sales records of each agent and dropshipper.

1.4 Project Objectives

1.4.1 Main Objective

To implement a web and mobile food inventory management system that helps Sharifah Food to manage and keep track of its inventory and sales data.

1.4.2 Specific Objectives

a. To analyse and determine the most appropriate software development tools and methodology in developing the inventory management system that runs on both web and mobile platforms by 30 July 2021.

b. To design a prototype of a web-based computerised food inventory management system integrated with an e-commerce platform by 22 August 2021.

c. To develop a web and mobile computerised food inventory management system integrated with an e-commerce platform by April 2022.

d. To evaluate the system's functionalities through performing unit, integration, system usability, and user acceptance testing by achieving a 95 percent pass rate on all tests before May 2022.

1.5 Proposed Solution

The development of a computerised inventory management system could overcome the problems stated in the **Problem Statement**. The primary purpose of the computerised inventory management system is to replace the manual process of the inventory management process. This project specifically develops a computerised inventory management system for Sharifah Food to solve the problem of manual inventory management of the company.

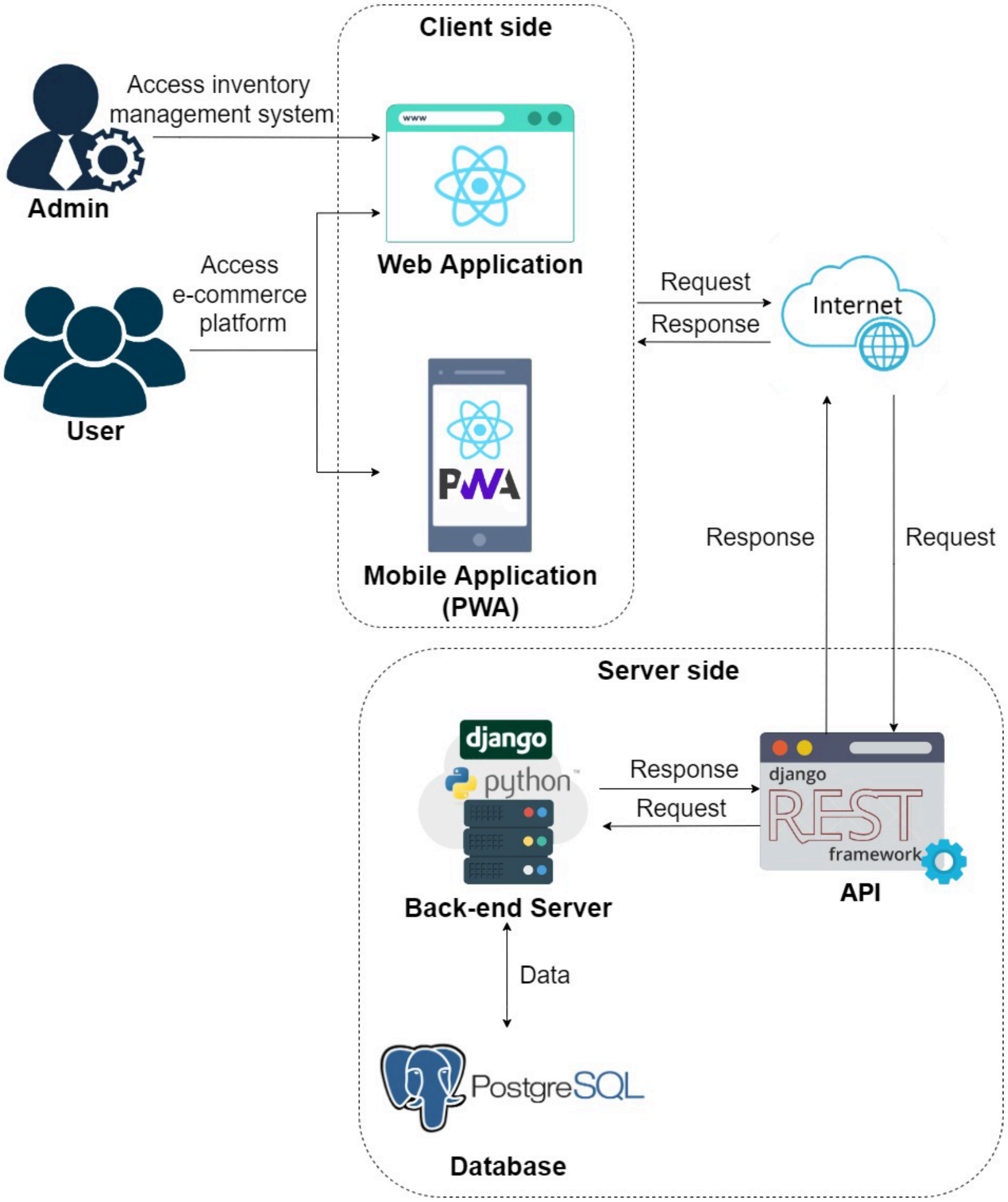


Figure 1.3: System Overview

Figure 1.3 shows the overview of the computerised inventory management system being developed in this project. This system is based on client-server architecture. Besides, the system can be divided into two parts: the e-commerce platform and the inventory management system. The user can access the e-commerce platform through web and mobile applications, while the admin can access the inventory management system through the web application. The workflow of the computerised inventory management system starts with the user or admin visiting the web or mobile applications. Then, the user and admin can access the features listed in the **Scope of Project**.

When a user or an admin performs any tasks or functionalities that require a back-end server, the application sends the request to the API through an internet connection. The API, which acts as an intermediary between the user and the server, allows the systems to communicate with each other for accessing and exchanging data. The primary purpose of the API is to hide the underlying procedures and implementation of the application and only expose the necessary actions or objects (Meng, et al., 2018). After the API receives the request, it sends the request to the back-end server, and the back-end server will process the request. In addition, the back-end server might access the database to retrieve the stored data, modify or add new data, if necessary. Once the back-end server completes processing the request, it sends a response to the client to indicate that the action has been performed or the resource requested by the client has been returned (Singh & Kumar, 2016).

To illustrate the whole process in a real scenario, the application sends a request to the web API with the order information when a user places his order on the e-commerce platform. Then, the web API forwards the request to the back-end server, provided that the authentication of the request is passed. When the back-end server receives the request, it performs some operations on the data and stores it in the database. If we implement this system, we can automatically store the data in the database without the need of humans to work on it after the users place their orders. Besides, we can easily retrieve the data from the database whenever it is needed.

Undoubtedly, this system can solve the problems associated with the manual inventory management system. When Sharifah Food employs this computerised inventory management system in its day-to-day business, it is no longer necessary to manually record each transaction after the order is placed. Instead of taking orders through Whatsapp, Sharifah Food can provide its customers with an e-commerce website where they can place orders. The order information is automatically stored in the database after the order is placed. Although computerised inventory management minimises the likelihood of inaccurate inventory records, this problem can still exist if the stockist inadvertently delivers the wrong quantity of goods to customers.

1.6 Proposed Approach

The proposed software development methodology for the project is the **prototyping methodology**. Figure 1.4 shows the software development process with prototyping methodology for this project.

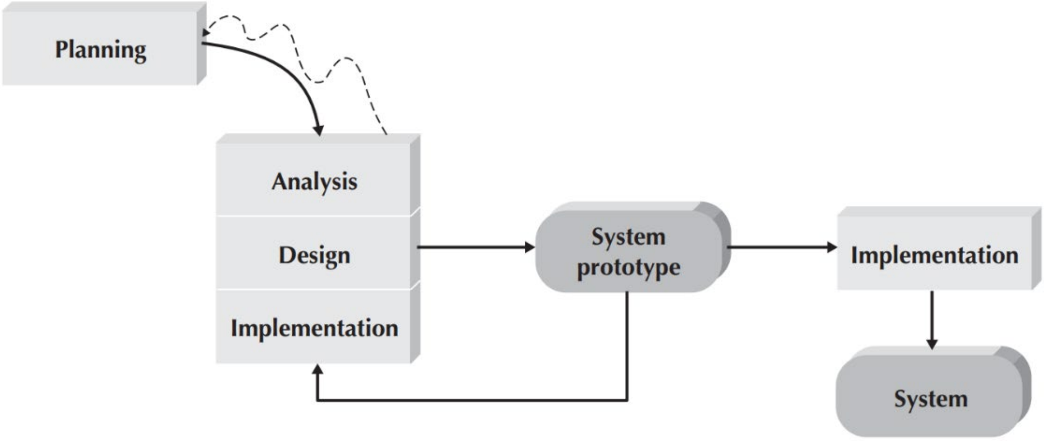


Figure 1.4: Prototyping methodology (Dennis, et al., 2015)

The prototyping methodology focuses on the development of a system prototype, a rapid and functional system model that is created based on user requirements to allow the users to evaluate the system's functionality (Dennis, et al., 2015; Dora & Dubey, 2013; Egwoh & Nonyelum, 2017; Gajalakshmi, 2016). The primary reason for selecting prototyping methodology as the proposed approach for this project is the refinement iterations that can be used to improve the system in each iteration. Bermudez and Cantos (2012), Irene (2011), Tapado and Delluza (2016) as well as Sutanto (2019) also adopted the prototyping methodology in the development of their inventory management systems. The results of these previous inventory management system projects support the development of the inventory management system using the prototyping methodology as it contributes to a better and more complete system specification.

There are four main phases in the prototyping methodology: Planning, Analysis, Design, and Implementation. In the planning phase, a feasibility study is conducted on the project topic, and a preliminary report is prepared which includes the project background, problem statement, project objective, proposed solution, proposed approach, and project scope will be produced. A Gantt chart is also prepared as a work plan to guide the progress of the project.

In the initial analysis phase, an extensive literature review is conducted on the system development process of the inventory management system. Also, existing similar applications are analysed to understand how the inventory management system works. The functional and non-functional requirements are also collected in the analysis phase. The design plan, such as a use case diagram, is created based on the user requirements identified in the analysis phase.

In the design phase, the architecture and interfaces of the system are planned and designed. Also, the system design is created, such as the software architecture design, entity-relationship diagram, activity diagram, etc. In the implementation phase, a prototype is developed and shown to the user to check whether the system prototype meets the requirements. Once the user has evaluated the prototype, the user's feedback and comments are forwarded to the developer. With the user's feedback and comments, the conflicting or ambiguous requirements can be easily resolved.

The feedback and comments are then brought to the next iteration for refinement. Next, the developer will reanalyse, redesign and reimplement a subsequent prototype with more detailed features. The user is then asked again to evaluate and provide feedback on the subsequent prototype. This iteration process repeats until the user is satisfied with the final prototype. After the user approves the final prototype, the complete inventory management system is implemented, marking the completion of the project.

In this project, the user who evaluates the prototype will be the project supervisor. Also, the development process of this project will be divided into three iterations. The first iteration will focus on the web application, the second iteration will focus on the back-end server, and the last iteration will focus on the mobile application.

1.7 Scope of Project

This section defines the platform used, the target users, the features covered, and the features not covered for this project.

1.7.1 Platform

This project aims to develop the Sharifah Food inventory management system on two platforms, the web platform and the mobile platform. The web platform covers both the user and admin sides, while the mobile platform covers only the user side. The mobile platform should be compatible with mobiles that have browser support.

1.7.2 Target Users

1.7.2.1 Direct Customers, Agents and Dropshippers

Direct customers, agents and dropshippers are the users of the system. They can browse and view the products and promotions. They can also place and track orders through the e-commerce website. Direct customers, agents, and dropshippers can use both web and mobile platforms for the tasks mentioned.

1.7.2.2 Stockist

The stockist is the admin of the system. The stockist can manage inventory stocks, manage orders from the direct customers, agents and dropshippers, manage agents and dropshippers and view sales reports. The stockist can only use the web platform to perform the tasks mentioned.

1.7.3 Features Covered

The features of the system can be divided into features in the inventory management system and e-commerce platform. The mobile platform includes only the e-commerce platform for user, while the web platform includes both the e-commerce platform for user and the inventory management system for admin.

1.7.3.1 E-commerce Platform (Direct Customer, Agent and Dropshipper)

a) Register as Customer

Although users can use the system without registering, they can register as customers to use all available features, such as managing addresses and viewing purchase history. Registration requires details such as email and password.

b) Register as Agent or Dropshipper

The users can register as agents or dropshipper by filling the registration form on the website. However, users who have submitted the registration form are not compulsory becoming as agents or dropshippers. The decision lies with the stockist. Details such as name, email, contact number, gender, address, occupation and other relevant information are required.

c) Login

The users who have registered can log in to the system using their emails and passwords.

d) Profile Management

The registered users can manage their profiles by changing their current password, email, gender, adding and saving addresses for future orders, etc.

e) Browse Products and Promotions

The users can view all the products and promotions through the website and mobile application. Besides, when a user selects a particular product, the details of the product will be displayed to the user. Additionally, users can also filter or sort the item catalogue by specific criteria.

f) View Item

The users can view the details of the item by selecting the item while browsing. All the information about the item is displayed to the user.

g) **Add to Shopping Cart**

The users can add the items to their shopping carts. The shopping cart can contain as many items as the user wants, but it cannot store out-of-stock items. Also, the cart can only be kept for unregistered users until the cache of their devices is cleared.

h) **Search Items**

The users can search for items by entering the keywords of the products.

i) **Order History**

The users can search for the order by entering their email and order number, which can be found in the order confirmation email. In case of the registered users, They can view the list of their order history on the order history page.

j) **Checkout**

The users can check out the items in their shopping cart. Users can also confirm their delivery address, order summary and payment methods before paying their orders. Unregistered or registered users without shipment details will have to enter order shipment details.

k) **Payment**

The users can pay for their orders during the checkout process. Two methods of payment are available: Credit or debit card and FPX online banking.

l) **About Us**

The users can learn about the company, such as the company's background, goals, culture, activities, etc.

m) **Contact Us**

The users can view all contact information of the company. Besides, users can send messages to the company's email to make inquiries.

n) **Notification**

The system automatically notifies the users of the order updates via email.

1.7.3.2 Inventory Management System (Admin)

a) Login

The admin is able to log in to the inventory management system by entering their username and password.

b) Profile Management

The admins can manage their profiles and change their current password, phone number, email, gender, etc.

c) Product Stock Management

The admin can manage the product inventory, such as add new products, add new stocks to the products, modify the product information, remove the existing product, and check the products' information.

d) Promotional Packages Management

The admin can manage the promotions by adding new promotional packages as well as modifying and deleting the existing promotional packages.

e) Order Management

The admin can view all orders from direct customers, agents and dropshippers. In addition, the admin can update the status of each order or even cancel the customers' orders.

f) Agent and Dropshipper Management

The admin can view the lists of agents and dropshippers and their respective orders to confirm their sales performance. In addition, the admin can also view, accept and reject the applications of the agents and dropshippers.

g) Shipping Fee Management

The admin can add, edit, delete and view the shipping fee in different states. Besides, the admin can customize the shipping fee for different weight ranges. The shipping fees added will be applied to the total order amount when the customers place an order.

h) Pickup Location Management

The admin can add the pickup location for the placed orders. This allows the customers to choose whether to ship their orders to their addresses or pick up the order at the pickup location.

i) Business Insights

The admin can view the sales reports such as daily sales, monthly sales, number of new customers, product rankings, etc., to gain more insight into the company's business performance. Sales and inventory analyses are crucial for the business as they help cut unnecessary costs and predict the demand for products. There are many inventory analyses such as ABC analysis, HML analysis, SS analysis, EOQ analysis, etc. This system uses these analyses to provide the information for better inventory control on Sharifah Food. For instance, the system calculates the safety stock for each product and notifies the user when the product quantity falls below the safety stock level.

j) Invoice, Sales, and Inventory Reports Generator

The admin can generate an invoice for each order placed by a customer. The system should also generate the sales and inventory reports at the admin’s request.

k) Discount Management

The admin can manage and create discount codes that can be applied while making payments on the e-commerce platform. The discount can be specified either as a percentage or a fixed amount and is customizable for various criteria such as minimum order spend, a maximum discount, customer eligibility, etc.

l) Notification

Email notifications automatically alert the admin when new orders are placed, products are in low stock, or pending approvals for the registrations for agent and dropshipper.

1.7.4 Features Not Covered

Due to the time constraint and the broad scope of this system, some additional features may not be delivered in a timely manner. The following features will not be covered under this project.

a) **Barcode or RFID System**

A barcode or RFID system is handy for recording new inventory as well as POS records. However, Sharifah Food currently focuses on online sales. Therefore, it is not mandatory to develop a barcode or RFID system in this project.

b) **Chatbot**

A chatbot could be important to increase customer satisfaction and improve the efficiency of the company's customer service. However, developing a chatbot requires a lot of time to train and test the chatbot. Since the time frame of this project is limited, the features of the chatbot will not be covered.

c) **Automated Order Tracking System**

An automated order tracking system could inform the customers about the current order status. This function is also not covered due to the limited time frame of the project. Nevertheless, the tracking number and courier information will be included in each order so that customers can track their orders by the tracking number provided.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will discuss the inventory management system concept to understand the rationale behind such project implementation better. Besides, several existing applications related to the project will be reviewed and analysed to recognise the common features of the computerised inventory management system. Next, this chapter will also study different inventory analysis techniques and database transaction management and concurrency control techniques. Lastly, various Software Development Life Cycle (SDLC) methodologies will be examined and evaluated to determine the most suitable SDLC methodologies to be employed in this project.

2.2 Inventory Management System

Inventory is a valuable asset of a company; most companies gross with their inventories. To streamline a business process, the company needs to monitor and manage its inventory. Nonetheless, it is of the utmost challenge to monitor and manage the company's inventory. There are many factors to consider, especially in this swiftly changing business environment. Factors such as the risk of inventory obsolescence, extensive inventory proliferation, the cost of holding inventory should be controlled by the company (Imeokparia, 2013). Hence, the term inventory management has been introduced to ease the companies in managing their inventories.

Inventory management refers to a process of efficiently monitoring and managing the flow of the inventory units in and out from the existing inventory stocks so that the company can meet the demand of the goods at a minimum investment. (Abisoye, et al., 2013; Imeokparia, 2013). It is an essential aspect of the company's business process, especially in minimising the costs and maximising its profits. In general, the process of inventory management is associated with monitoring the movement of the inventory units to prevent overstock and understock problems (Abisoye, et al., 2013). Nowadays, from small to large scale, almost every business implemented different kinds of inventory management to handle and keep track of their daily business process (Khabbazi, et al., 2013). With proper inventory management, all the stocks can be managed, controlled, and tracked easily.

Two types of inventory management systems have been introduced to make inventory management more efficient and effective: manual inventory management and computerised inventory management systems.

2.2.1 Manual Inventory Management System

The manual inventory management system is an inventory management system that updates, maintains, and controls manually. In a manual inventory management system, the daily sales and inventory records are dependent upon dynamic individuals where the individual manually performs documentation on the business sales and inventory records (Anulika, et al., 2020). In a manual inventory management system, the tools such as Excel spreadsheets or even paperwork are used to record the inventory data.

A manual inventory management system is still a prevalent option to be employed, especially for SMEs. Nevertheless, since the manual inventory management system requires humans to work on, several studies have discovered several problems associated with the manual inventory management system. For instance, Chan, et al. (2017) claimed that manual documentation took a relatively long time and had a comparatively higher chance of recording the information, causing inaccurate inventory records wrongly. In addition to this, the study of Chuang and Oliva (2015) on a retail store shows that a total of 10 percent profits were lost due to the 29 percent of inaccurate inventory record. Besides, Shinde, et al. (2018) pointed out that the manual inventory management system employed in the company they were working for had faced many limitations, including the lack of centralised information and the time-consuming of updating the records.

Additionally, the **Problem Statement** can be referred to for a deeper insight into the problem of the manual inventory management system.

2.2.2 Computerised Inventory Management System

The computerised inventory management system, also known as an automated inventory management system, is a computer-based inventory management system that assists in automating inventory management processes, such as tracking inventory records, product level, order, and sales (Anulika, et al., 2020). More companies, especially large-scale companies, have adopted the computerised inventory management system in this highly competitive business world. Most companies transform from manual to computerised inventory management systems because the computerised inventory management system overcomes the problems faced by the manual inventory management system. For example, a computerised inventory management system's automated process dramatically reduces the time required to record the sales and inventory data (Adhena, 2020; Hamadi, 2018). Moreover, the computerised inventory management system reduces the inaccuracy of the inventory records because the inventory process is automated without human labour.

Although many companies changed their inventory management system to be computerised, some still use manual systems instead of automated systems. According to the study by Laar, et al. (2015), the factors that some companies prefer manual systems are the cost of accessing the full package and technical functions as well as the difficulty in using the computerised inventory management system.

The computerised inventory management system can be divided into different types such as Barcode, Vendor Managed Inventory (VMI), Materials Requirement Planning (MRP), Enterprise Resource Planning (ERP), Point of Sale (POS), E-Procurement as well as Radio Frequency Identification (RIFD) systems (Samuel & Ondiek, 2014; Shinde, et al., 2018). Each type of computerised inventory management system performs different tasks to improve the inventory management process and performance in various aspects, as shown in Figure 2.1.

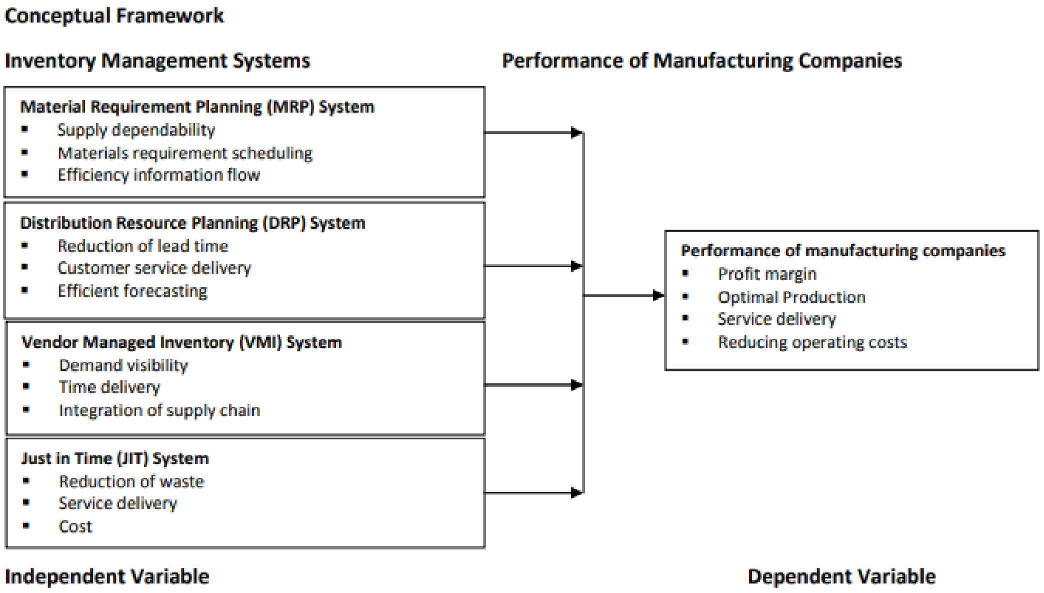

Conceptual Framework
Inventory Management Systems
Performance of Manufacturing Companies
Material Requirement Planning (MRP) System
Supply dependability
Materials requirement scheduling
Efficiency information flow
Distribution Resource Planning (DRP) System
Reduction of lead time
Customer service delivery
Efficient forecasting
Vendor Managed Inventory (VMI) System
Demand visibility
Time delivery
Integration of supply chain
Just in Time (JIT) System
Reduction of waste
Service delivery
Cost
Performance of manufacturing companies
Profit margin
Optimal Production
Service delivery
Reducing operating costs
Independent Variable
Dependent Variable

Figure 2.1: Different Types of Computerized Inventory Management System (Ngugi, et al., 2019)

2.2.3 Existing Computerized Inventory Management Systems

In this section, a total of 4 existing computerised inventory management systems will be reviewed. The systems to be studied include Lazada, senangKIRA, Shopify, and WeNiaga.

2.2.3.1 Lazada

Adopted from:

< https://sellercenter.lazada.com.my/>

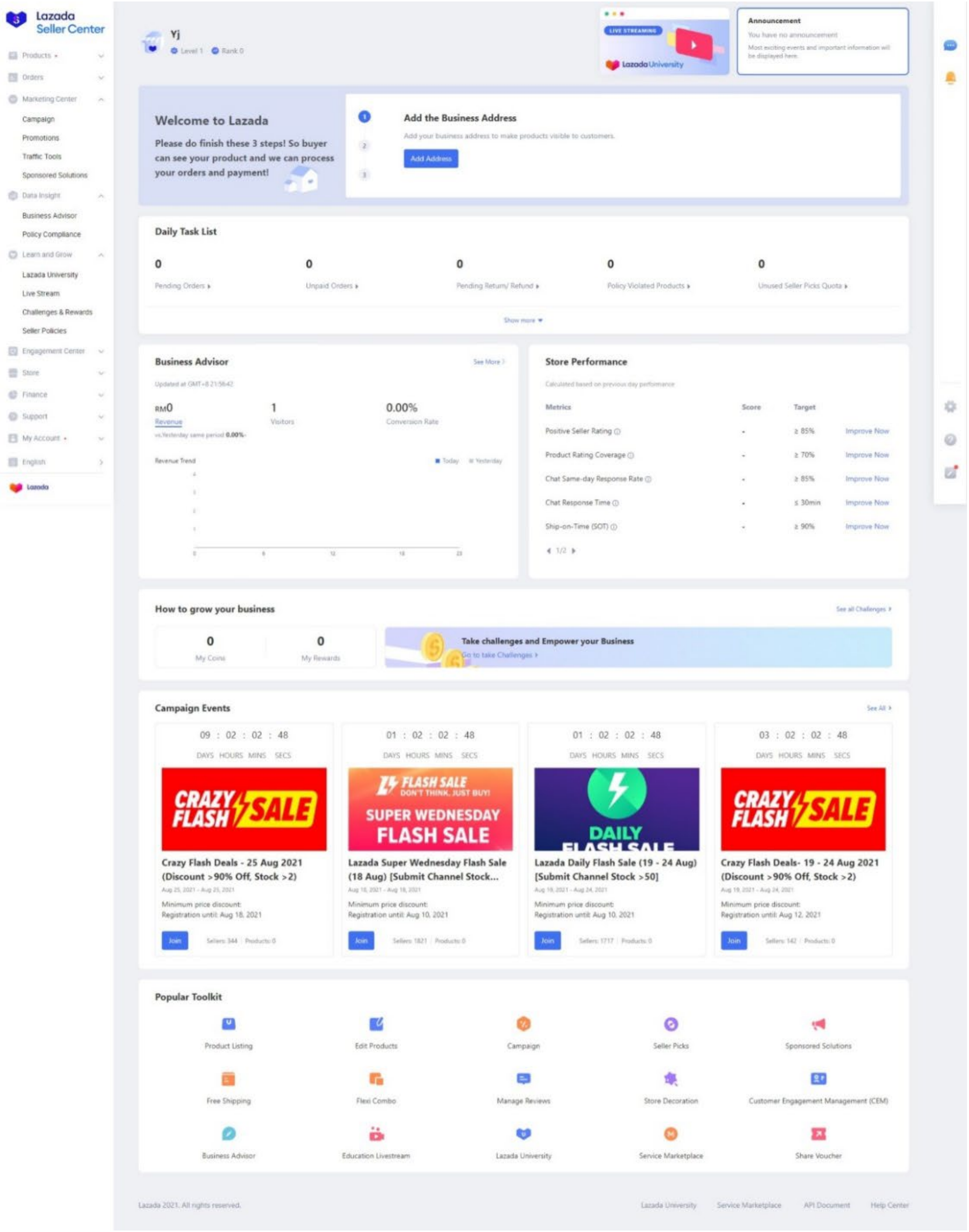
< https://www.lazada.com.my/>

Lazada is a crowd-pleasing e-commerce platform for online shopping. It was founded in 2012 and has become one of the leading e-commerce platforms nowadays. Up until now, Lazada has expanded its markets to Malaysia, Thailand, Singapore, the Philippines, and Vietnam. Moreover, Lazada is also one of the regional flagships of a Chinese multinational technology company, Alibaba Group. Besides providing the e-commerce platform to its users, Lazada owns its seller centre for the sellers to manage their sales and inventory. Lazada comes with Web and mobile applications, and both applications support the e-commerce platform and seller centre. This section will discuss the features available in Lazada's web and mobile e-commerce platform as well as its seller centre on web applications. The major features of Lazada are as follows:

Lazada Seller Center Website (Inventory management system)

a) Dashboard

   Once the seller logs into the seller centre, a dashboard will be shown. The dashboard contains the daily task list, business advisor, store performance, campaign events, and popular toolkit. It provides an overview of the seller store.



Announcement

You have no announcement

Most exciting events and important information will be displayed here.

Welcome to Lazada

Please do finish these 3 steps! So buyer can see your product and we can process your orders and payment!

1 Add the Business Address

Add your business address to make products visible to customers.

Daily Task List

Pending Orders	Unpaid Orders	Pending Return/ Refund	Policy Violated Products	Unused Seller Picks Quota
0	0	0	0	0

Business Advisor

Updated at GMT+8 21:56:42

Revenue	Visitors	Conversion Rate
RM0	1	0.00%

vs.Yesterday same period 0.00%-

Store Performance

Calculated based on previous day performance

Metrics	Score	Target	
Positive Seller Rating ⓘ	-	≥ 85%	Improve Now
Product Rating Coverage ⓘ	-	≥ 70%	Improve Now
Chat Same-day Response Rate ⓘ	-	≥ 85%	Improve Now
Chat Response Time ⓘ	-	≤ 30min	Improve Now
Ship-on-Time (SOT) ⓘ	-	≥ 90%	Improve Now

How to grow your business

My Coins	My Rewards
0	0

Take challenges and Empower your Business

Campaign Events

09 : 02 : 02 : 48 DAYS HOURS MINS SECS

Crazy Flash Deals - 25 Aug 2021 (Discount >90% Off, Stock >2)

Aug 25, 2021 - Aug 25, 2021

Minimum price discount:

Registration until: Aug 18, 2021

Sellers: 344 | Products: 0

01 : 02 : 02 : 48 DAYS HOURS MINS SECS

Lazada Super Wednesday Flash Sale (18 Aug) [Submit Channel Stock...

Aug 18, 2021 - Aug 18, 2021

Minimum price discount:

Registration until: Aug 10, 2021

Sellers: 1821 | Products: 0

01 : 02 : 02 : 48 DAYS HOURS MINS SECS

Lazada Daily Flash Sale (19 - 24 Aug) [Submit Channel Stock >50]

Aug 19, 2021 - Aug 24, 2021

Minimum price discount:

Registration until: Aug 10, 2021

Sellers: 1717 | Products: 0

03 : 02 : 02 : 48 DAYS HOURS MINS SECS

Crazy Flash Deals- 19 - 24 Aug 2021 (Discount >90% Off, Stock >2)

Aug 19, 2021 - Aug 24, 2021

Minimum price discount:

Registration until: Aug 12, 2021

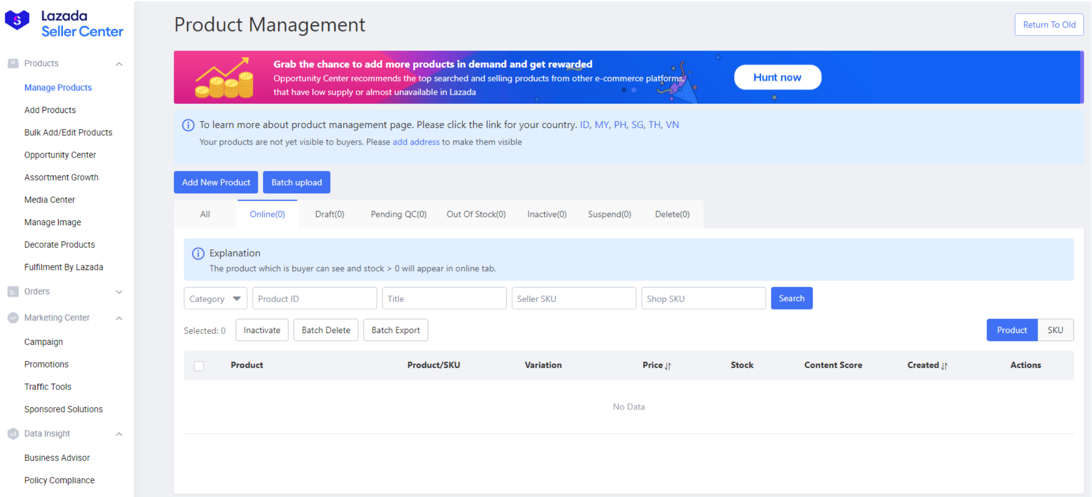
Sellers: 142 | Products: 0

Lazada 2021. All rights reserved.

Figure 2.2: Dashboard of Lazada Seller Center

b) Product Management

The sellers can manage their products easily from the Lazada Seller Center website. There are different options for product stock management. For example, the sellers can add and edit their products, manage product images, and decorating products. In addition, Lazada provides an extra feature of bulk-adding and editing the products. However, the users must download and modify the CSV file template provided by Lazada to use bulk-adding and editing products function.



Grab the chance to add more products in demand and get rewarded
Opportunity Center recommends the top searched and selling products from other e-commerce platforms, that have low supply or almost unavailable in Lazada

To learn more about product management page. Please click the link for your country. ID, MY, PH, SG, TH, VN
Your products are not yet visible to buyers. Please add address to make them visible

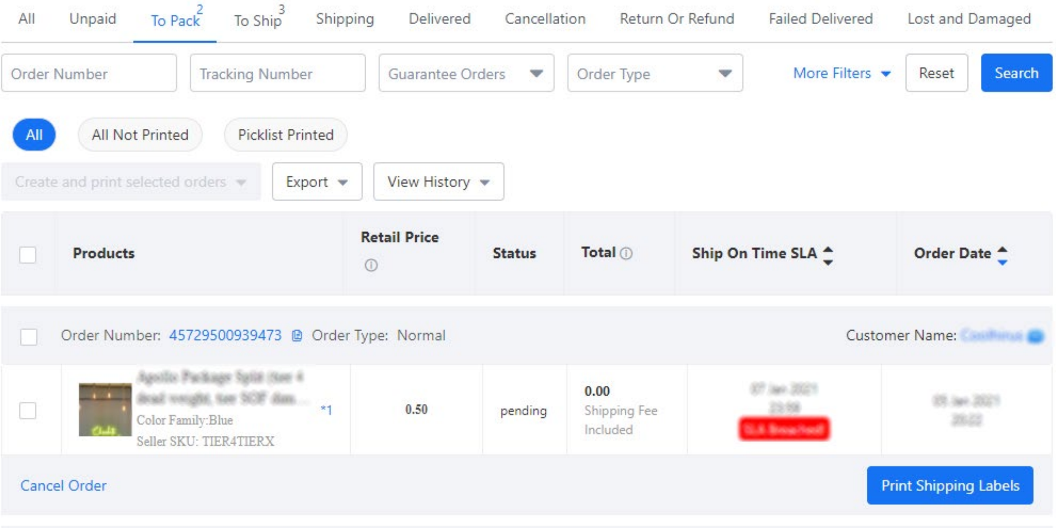
Explanation
The product which is buyer can see and stock > 0 will appear in online tab.

□	Product	Product/SKU	Variation	Price	Stock	Content Score	Created	Actions
				No Data				

Figure 2.3: Product Management in Lazada Seller Center

c) Order Management

The sellers can manage the orders placed by the customers in the Lazada Seller Center. With this feature, the sellers are allowed to observe all the details of their orders. Different categories of the order conditions, such as to pack, ship, return or refund, failed delivered, as well as lost or damaged, enable the sellers to notice and take action on the orders.

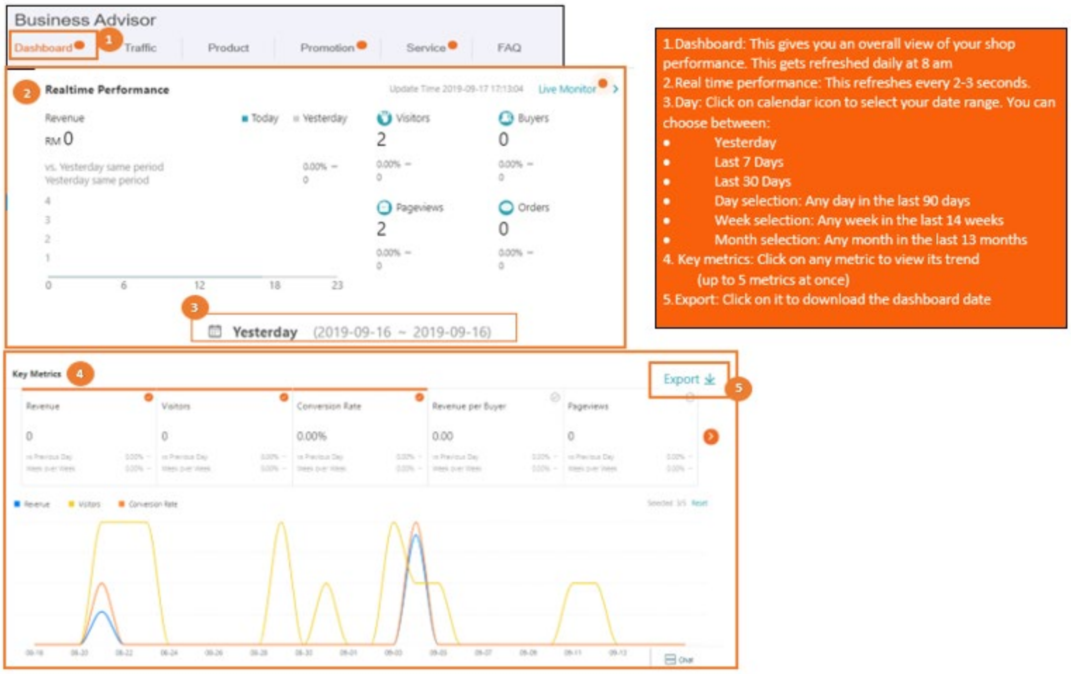


□	Products	Retail Price ⓘ	Status	Total ⓘ	Ship On Time SLA	Order Date
□	Order Number: 45729500939473 Order Type: Normal					Customer Name: [redacted]
□	[redacted] Color Family:Blue Seller SKU: TIER4TIERX *1	0.50	pending	0.00 Shipping Fee Included	[redacted]	[redacted]

Figure 2.4: Order Management in Lazada Seller Center

d) Business Advisor

When the sellers want to gain insight into their business performance, they can utilise the feature of the business advisor provided by the Lazada Seller Center. In the business advisor feature, sellers can inspect basic analyses such as real-time performance, ranking, key metrics, and product dashboards. The real-time performance provides the overview of the total visitors, pageviews, buyers, orders, and revenue. The real-time ranking offers the function to sort the products based on different criteria such as units sold, add to cart units, revenue and visitors. The key metrics visualise the different business performances such as conversion rate, revenue, units sold, average within a date interval. The product dashboards provide the summary of the sellers' inventory goods.



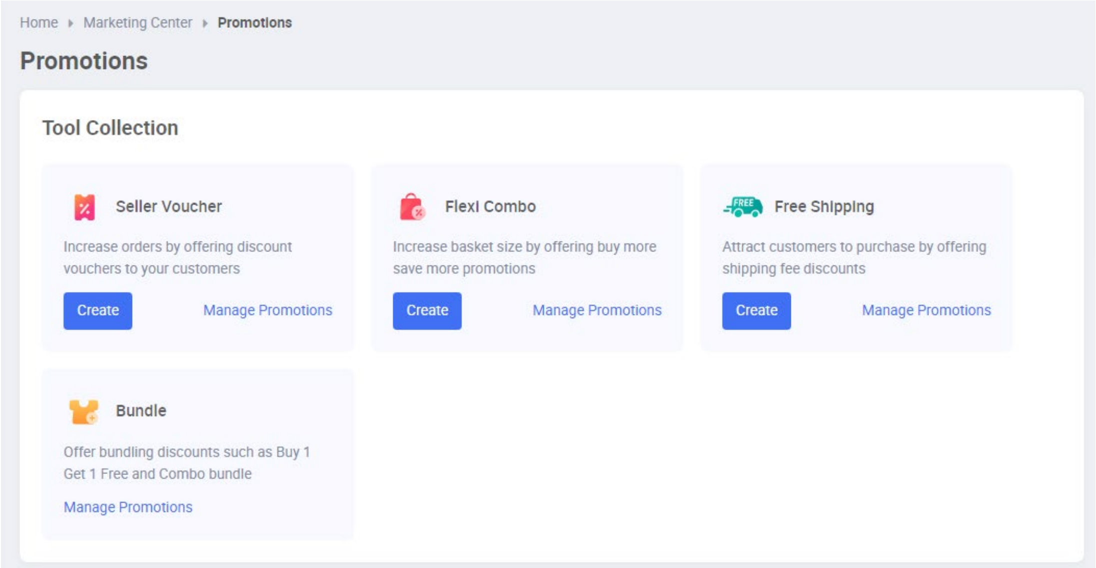
1.Dashboard: This gives you an overall view of your shop performance. This gets refreshed daily at 8 am
2.Real time performance: This refreshes every 2-3 seconds.
3.Day: Click on calendar icon to select your date range. You can choose between:
- Yesterday
- Last 7 Days
- Last 30 Days
- Day selection: Any day in the last 90 days
- Week selection: Any week in the last 14 weeks
- Month selection: Any month in the last 13 months

4. Key metrics: Click on any metric to view its trend (up to 5 metrics at once)
5.Export: Click on it to download the dashboard date

Figure 2.5: Business Advisor in Lazada Seller Center

e) Promotions Management

The sellers can manage their promotions with different types of vouchers and bundles. Besides, the sellers can create a new voucher by specifying the voucher's criteria such as voucher name, voucher redemption period, discount type, voucher limit per customer, etc.



Seller Voucher

Increase orders by offering discount vouchers to your customers

Flexi Combo

Increase basket size by offering buy more save more promotions

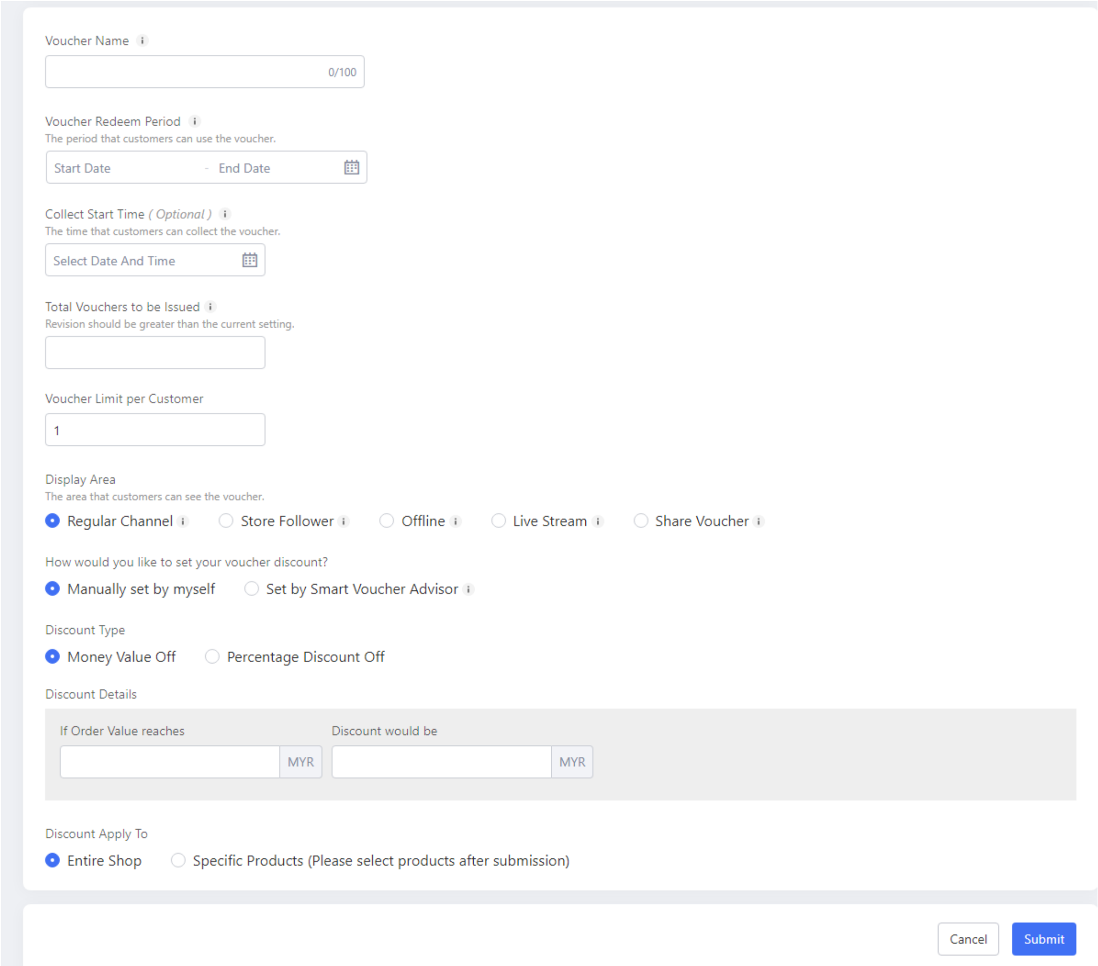
Free Shipping

Attract customers to purchase by offering shipping fee discounts

Bundle

Offer bundling discounts such as Buy 1 Get 1 Free and Combo bundle

Figure 2.6: Promotions Management in Lazada Seller Center



Voucher Name

0/100

Voucher Redeem Period

The period that customers can use the voucher.

Start Date - End Date

Collect Start Time *(Optional)*

The time that customers can collect the voucher.

Select Date And Time

Total Vouchers to be Issued

Revision should be greater than the current setting.

Voucher Limit per Customer

1

Display Area

The area that customers can see the voucher.

● Regular Channel ○ Store Follower ○ Offline ○ Live Stream ○ Share Voucher

How would you like to set your voucher discount?

● Manually set by myself ○ Set by Smart Voucher Advisor

Discount Type

● Money Value Off ○ Percentage Discount Off

Discount Details

If Order Value reaches	Discount would be
MYR	MYR

Discount Apply To

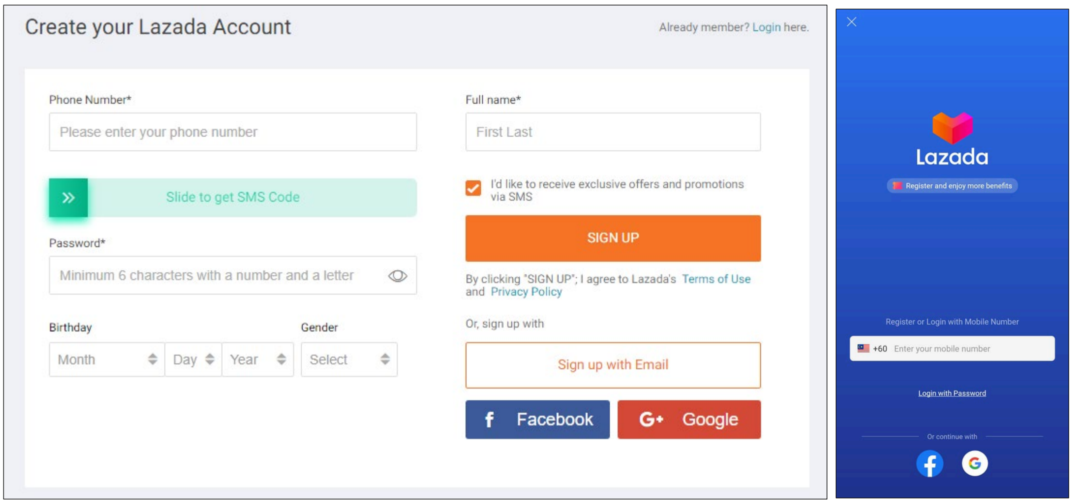
● Entire Shop ○ Specific Products (Please select products after submission)

Figure 2.7: Create Voucher in Lazada Seller Center

Lazada Web and Mobile E-commerce Platform

a) Register and Login

The users can either choose to log in or continue as a guest. The user should have an account if they want to log in. Otherwise, the users will be required to sign up for an account. In addition, the users can log in or sign up with Facebook or Google accounts.



Create your Lazada Account

Already member? Login here.

Phone Number*: Please enter your phone number

Full name*: First Last

☑ I'd like to receive exclusive offers and promotions via SMS

Password*: Minimum 6 characters with a number and a letter

By clicking "SIGN UP"; I agree to Lazada's Terms of Use and Privacy Policy

Birthday: Month Day Year

Gender: Select

Or, sign up with

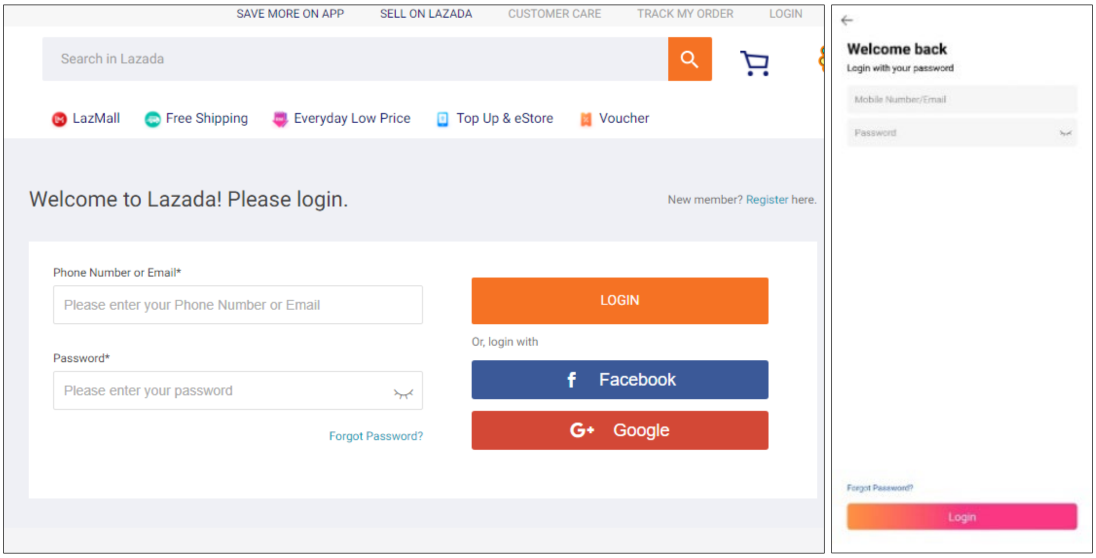
Register and enjoy more benefits

Register or Login with Mobile Number

+60 Enter your mobile number

Or continue with

Figure 2.8: Lazada Sign Up Page



Welcome to Lazada! Please login.

New member? Register here.

Phone Number or Email*: Please enter your Phone Number or Email

Password*: Please enter your password

Or, login with

Welcome back

Login with your password

Mobile Number/Email

Password

Figure 2.9: Lazada Login Page

b) View catalogue of stores and products

The users can view all the stores and products available in Lazada. Besides, the users can filter the products by different criteria and categories. Moreover, searching products by keywords is also available in Lazada.

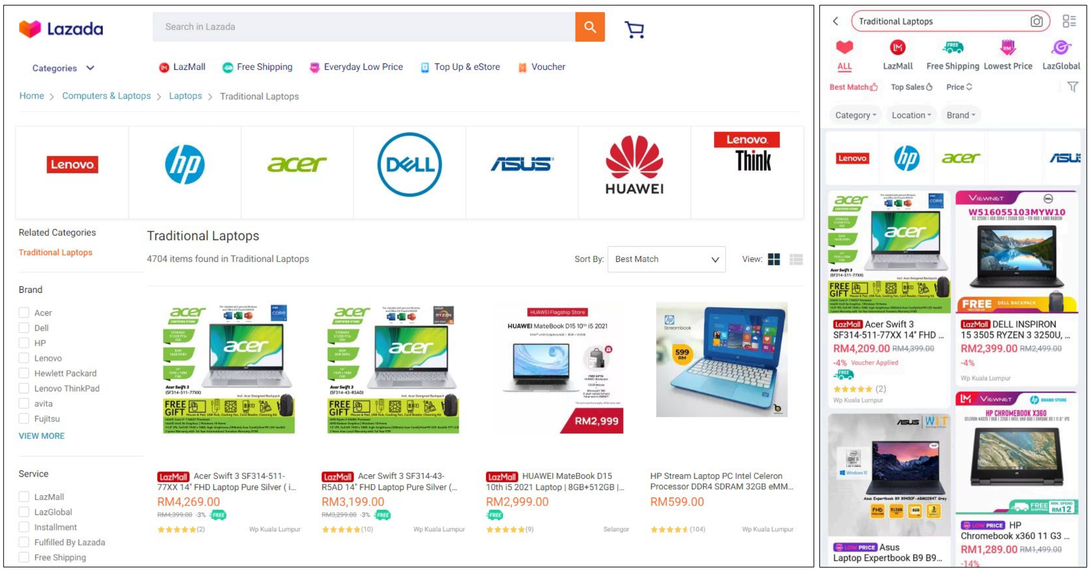


Figure 2.10: Lazada Catalogue of Stores and Products

When users select a product by clicking the product, the user will be directed to the product details page, as shown in Figure 2.11.

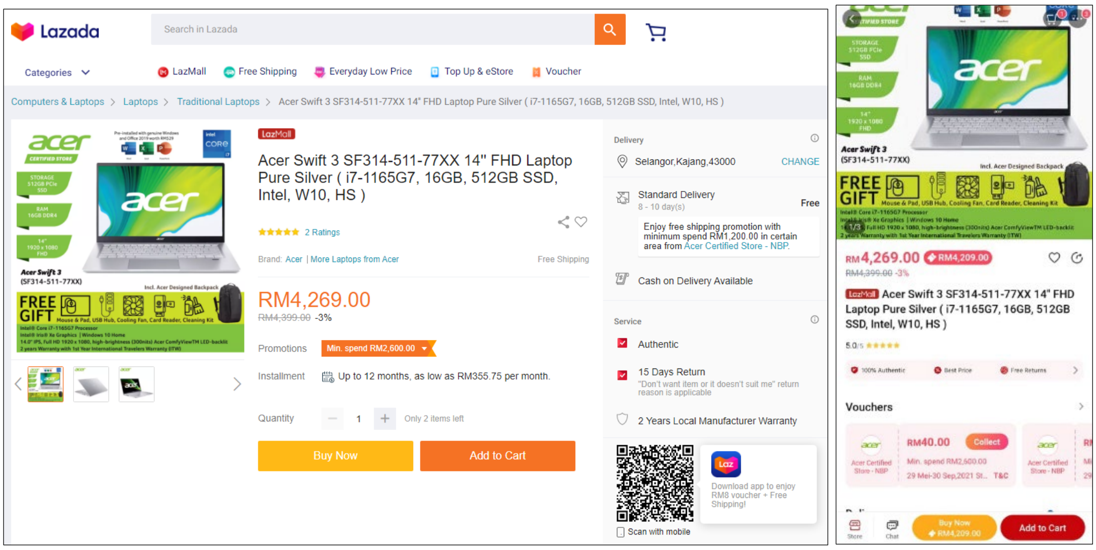
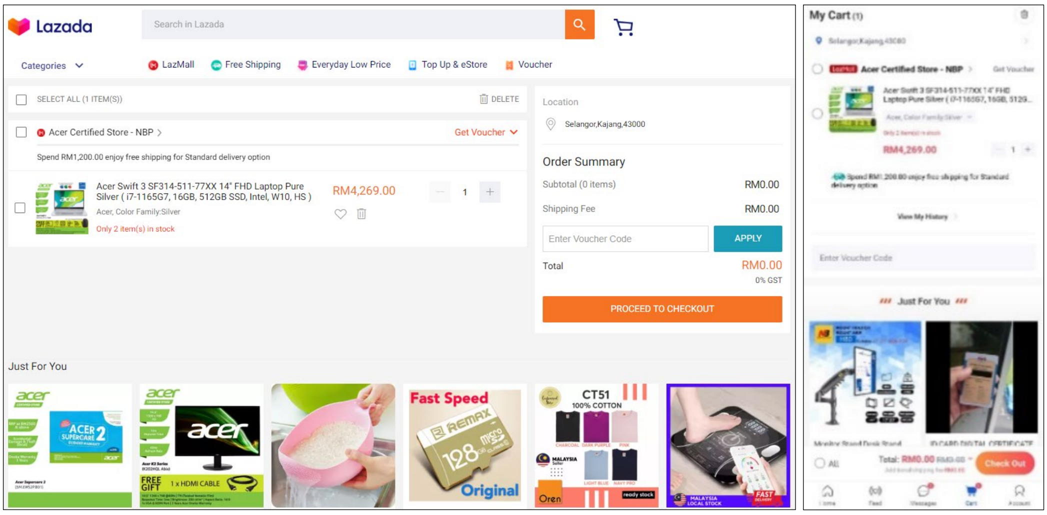

Lazada
Search in Lazada
Categories
LazMall
Free Shipping
Everyday Low Price
Top Up & eStore
Voucher
Computers & Laptops > Laptops > Traditional Laptops > Acer Swift 3 SF314-511-77XX 14" FHD Laptop Pure Silver (i7-1165G7, 16GB, 512GB SSD, Intel, W10, HS)
LazMall
Acer Swift 3 SF314-511-77XX 14" FHD Laptop Pure Silver (i7-1165G7, 16GB, 512GB SSD, Intel, W10, HS)
2 Ratings
Brand: Acer | More Laptops from Acer
Free Shipping
RM4,269.00
RM4,399.00 -3%
Promotions Min. spend RM2,600.00
Installment Up to 12 months, as low as RM355.75 per month.
Quantity 1 Only 2 items left
Buy Now
Add to Cart
Delivery
Selangor,Kajang,43000 CHANGE
Standard Delivery 8 - 10 day(s) Free
Enjoy free shipping promotion with minimum spend RM1,200.00 in certain area from Acer Certified Store - NBP.
Cash on Delivery Available
Service
Authentic
15 Days Return
"Don't want item or it doesn't suit me" return reason is applicable
2 Years Local Manufacturer Warranty
Download app to enjoy RM8 voucher + Free Shipping!
Scan with mobile
RM4,269.00
RM4,209.00
Acer Swift 3 SF314-511-77XX 14" FHD Laptop Pure Silver (i7-1165G7, 16GB, 512GB SSD, Intel, W10, HS)
Vouchers
RM40.00
Collect
Min. spend RM2,600.00
Buy Now
Add to Cart

Figure 2.11: Lazada Product Details Page

c) Add to Cart

The users can add the product to their cart by clicking the ‘Add to Cart’ button. After the product is added to the cart, the user can find the list of cart items by clicking the cart icon, and the shopping cart page will be shown, as in Figure 2.12.



□ SELECT ALL (1 ITEM(S))

□ Acer Certified Store - NBP

Spend RM1,200.00 enjoy free shipping for Standard delivery option

□ Acer Swift 3 SF314-511-77XX 14" FHD Laptop Pure Silver (i7-1165G7, 16GB, 512GB SSD, Intel, W10, HS) RM4,269.00 1

Acer, Color Family:Silver

Only 2 item(s) in stock

Location

Selangor,Kajang,43000

Order Summary

Subtotal (0 items) RM0.00

Shipping Fee RM0.00

Enter Voucher Code

Total RM0.00

0% GST

Selangor,Kajang,43000

Acer Swift 3 SF314-511-77XX 14" FHD Laptop Pure Silver (i7-1165G7, 16GB, 512G...

Acer, Color Family:Silver

RM4,269.00

Spend RM1,200.00 enjoy free shipping for Standard delivery option

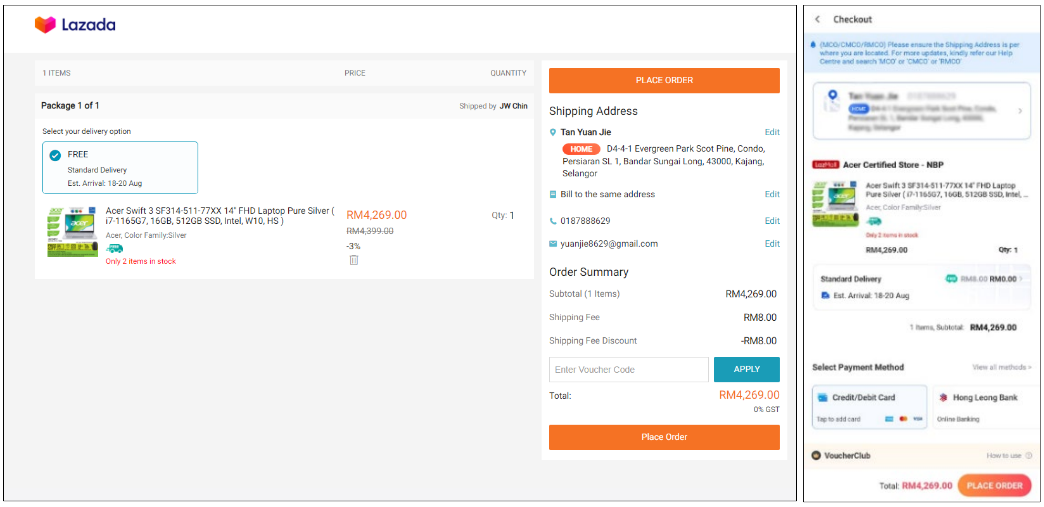
Enter Voucher Code

○ All Total: RM0.00

Figure 2.12: Lazada Shopping Cart Page

d) Checkout

The users can check out their products by selecting the products to check out and clicking the ‘Check Out’ or ‘Buy Now’ buttons. After the users click the button, the order summary will be displayed for confirmation as shown in Figure 2.13.



1 ITEMS PRICE QUANTITY

Package 1 of 1 Shipped by JW Chin

Select your delivery option

FREE

Standard Delivery

Est. Arrival: 18-20 Aug

Acer Swift 3 SF314-511-77XX 14" FHD Laptop Pure Silver (i7-1165G7, 16GB, 512GB SSD, Intel, W10, HS) RM4,269.00 ~~RM4,399.00~~ -3% Qty: 1

Acer, Color Family:Silver

Only 2 items in stock

Shipping Address

Tan Yuan Jie

HOME D4-4-1 Evergreen Park Scot Pine, Condo, Persiaran SL 1, Bandar Sungai Long, 43000, Kajang, Selangor

Bill to the same address

0187888629

yuanjie8629@gmail.com

Order Summary

Subtotal (1 Items) RM4,269.00

Shipping Fee RM8.00

Shipping Fee Discount -RM8.00

Enter Voucher Code

Total: RM4,269.00

0% GST

(MCO/CMCO/RMCO) Please ensure the Shipping Address is per where you are located. For more updates, kindly refer our Help Centre and search 'MCO' or 'CMCO' or 'RMCO'

Acer Certified Store - NBP

Acer Swift 3 SF314-511-77XX 14" FHD Laptop Pure Silver (i7-1165G7, 16GB, 512GB SSD, Intel, ...

Acer, Color Family:Silver

Only 2 items in stock

RM4,269.00 Qty: 1

Standard Delivery ~~RM8.00~~ RM0.00

Est. Arrival: 18-20 Aug

1 Items, Subtotal: RM4,269.00

Select Payment Method

Credit/Debit Card Tap to add card

Hong Leong Bank Online Banking

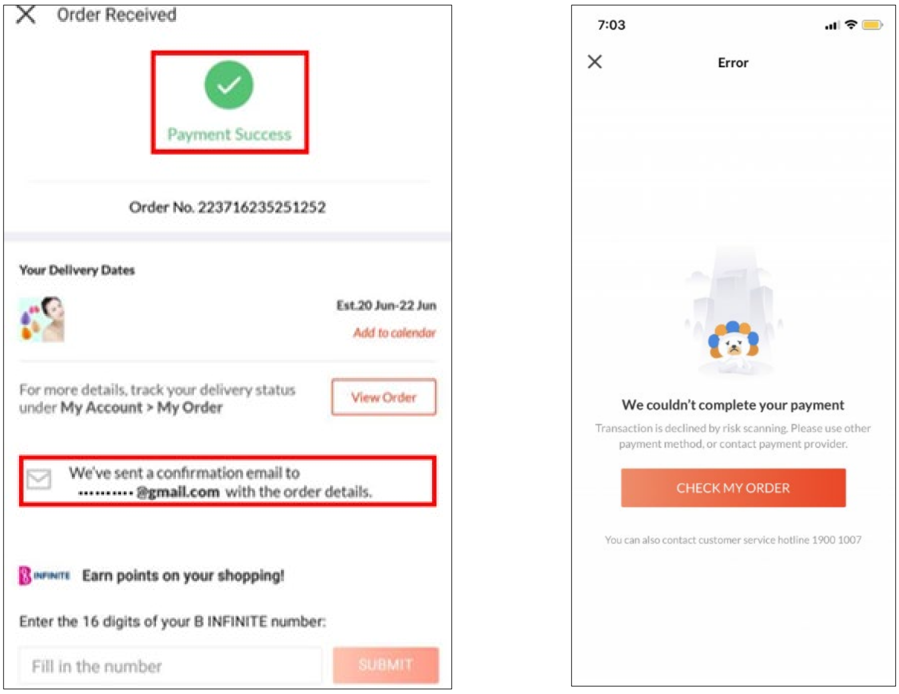
VoucherClub

Total: RM4,269.00

Figure 2.13: Lazada Checkout Page

e) Payment

Once the users confirm the order by clicking the ‘Place Order’ button, Lazada will redirect the users to the specific payment gateway for the payment process. Then, the users will be redirected back to Lazada applications after the payment gateway completed its process. The users will be shown the order’s status, such as successful payment or failed payment, as in Figure 2.14.



Order Received

Payment Success

Order No. 223716235251252

Your Delivery Dates

Est.20 Jun-22 Jun

Add to calendar

For more details, track your delivery status under **My Account > My Order**

View Order

We've sent a confirmation email to **••••••••••@gmail.com** with the order details.

Earn points on your shopping!

Enter the 16 digits of your B INFINITE number:

Fill in the number

SUBMIT

7:03

Error

We couldn't complete your payment

Transaction is declined by risk scanning. Please use other payment method, or contact payment provider.

CHECK MY ORDER

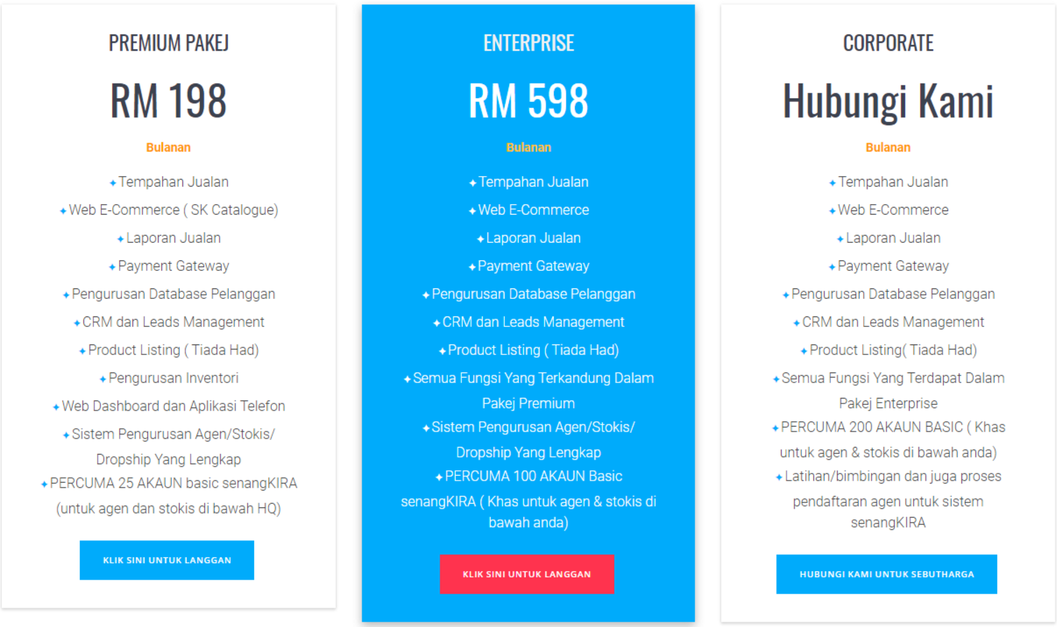
You can also contact customer service hotline 1900 1007

Figure 2.14: Order Status After Payment

2.2.3.2 senangKIRA

Adopted from <https://senangkira.my/>

senangKIRA is a stockist and agent inventory management system. This application assists the stockist and agent inventory management process by managing the orders from agents, monitoring agents’ sales, and providing complete and organised customer sales data management. This system comes with different packages, and each package charges with different prices. There are three packages available: premium, enterprise, and corporate. Each of the packages provides distinct functionalities. Figure 2.15 shows the packages and relative functionalities provided by the senangKIRA system.



PREMIUM PAKEJ

RM 198

Bulanan

- Tempahan Jualan
- Web E-Commerce (SK Catalogue)
- Laporan Jualan
- Payment Gateway
- Pengurusan Database Pelanggan
- CRM dan Leads Management
- Product Listing (Tiada Had)
- Pengurusan Inventori
- Web Dashboard dan Aplikasi Telefon
- Sistem Pengurusan Agen/Stokis/ Dropship Yang Lengkap
- PERCUMA 25 AKAUN basic senangKIRA (untuk agen dan stokis di bawah HQ)

KLIK SINI UNTUK LANGGAN

ENTERPRISE

RM 598

Bulanan

- Tempahan Jualan
- Web E-Commerce
- Laporan Jualan
- Payment Gateway
- Pengurusan Database Pelanggan
- CRM dan Leads Management
- Product Listing (Tiada Had)
- Semua Fungsi Yang Terkandung Dalam Pakej Premium
- Sistem Pengurusan Agen/Stokis/ Dropship Yang Lengkap
- PERCUMA 100 AKAUN Basic senangKIRA (Khas untuk agen & stokis di bawah anda)

KLIK SINI UNTUK LANGGAN

CORPORATE

Hubungi Kami

Bulanan

- Tempahan Jualan
- Web E-Commerce
- Laporan Jualan
- Payment Gateway
- Pengurusan Database Pelanggan
- CRM dan Leads Management
- Product Listing(Tiada Had)
- Semua Fungsi Yang Terdapat Dalam Pakej Enterprise
- PERCUMA 200 AKAUN BASIC (Khas untuk agen & stokis di bawah anda)
- Latihan/bimbingan dan juga proses pendaftaran agen untuk sistem senangKIRA

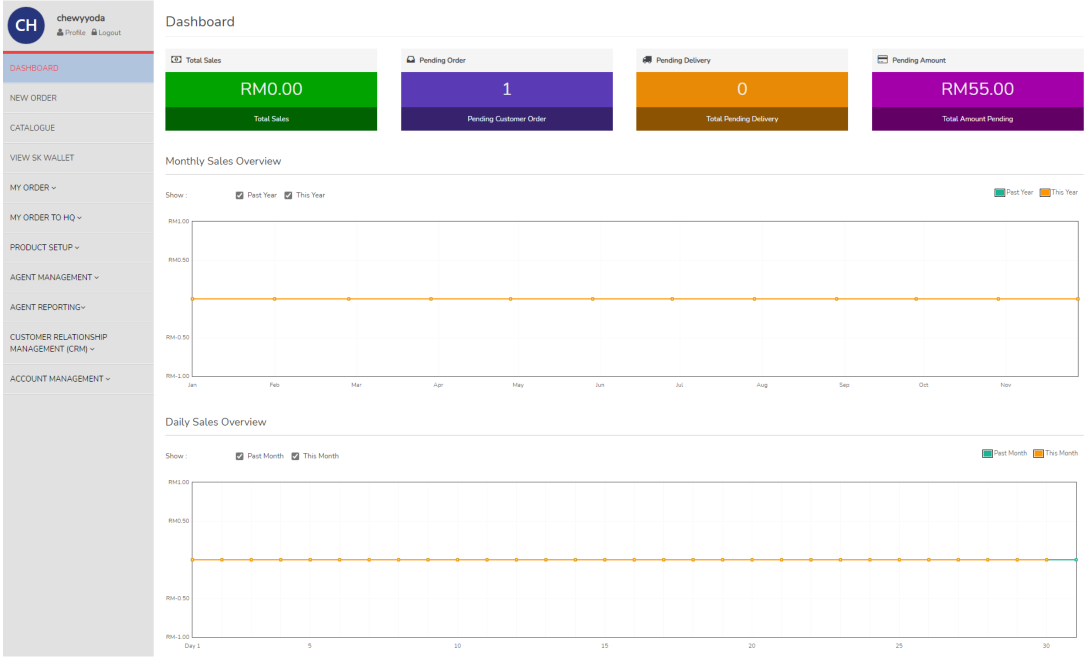
HUBUNGI KAMI UNTUK SEBUTHARGA

Figure 2.15: Packages provided by senangKIRA System

As the system is not freely accessible, there will be some limitations when testing the system's functionalities. Nonetheless, I have received free trial access of 14 days by signing up for an account. Therefore, I can access some functionalities provided by the 14-day free trial access. This section will discuss the features available in the senangKIRA system. The significant features of the senangKIRA are as follows:

a) Dashboard overview

senangKIRA system provides a dashboard when the users logged in to the website and mobile application. The users can view the summary of their sales, such as the total sales earnings as well as the number of pending orders, pending delivery, and total pending amount. Besides, the users can also view the daily and monthly sales overviews with the line graph.



Dashboard

Total Sales: RM0.00 (Total Sales)

Pending Order: 1 (Pending Customer Order)

Pending Delivery: 0 (Total Pending Delivery)

Pending Amount: RM55.00 (Total Amount Pending)

Monthly Sales Overview

Show : ☑ Past Year ☑ This Year

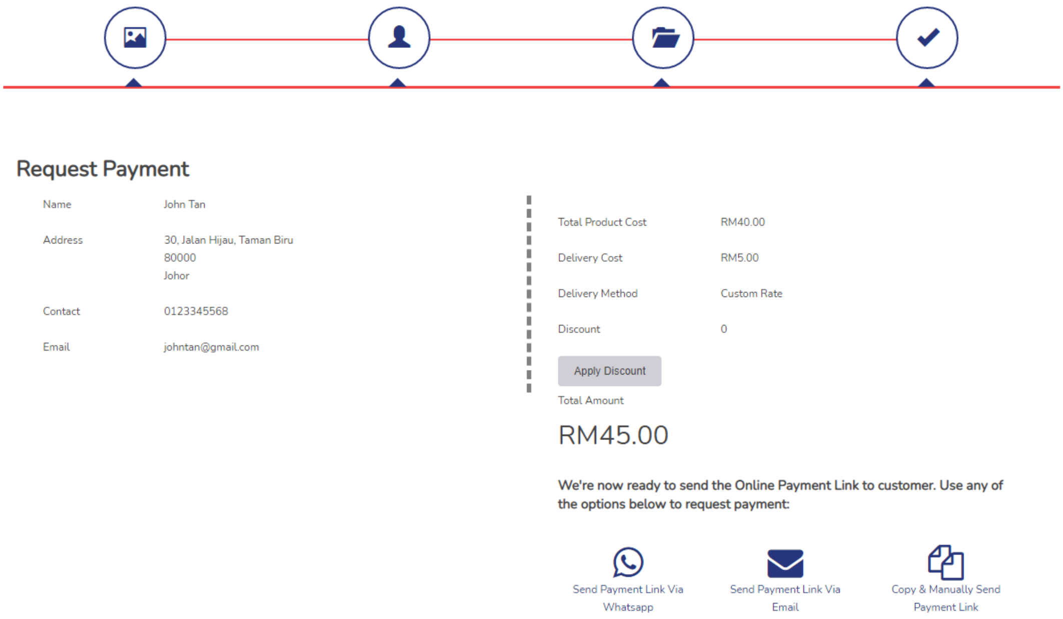
Daily Sales Overview

Show : ☑ Past Month ☑ This Month

Figure 2.16: senangKIRA Dashboard Overview

b) Create New Order

The users are allowed to create a new order by filling in the order information, such as the information about products to be ordered, customer information, and delivery method. After the order information is filled up, the users can apply the discount to the orders by discount percentage or discount amount. Once everything is settled, the users can send the payment request to their customers through WhatsApp, email, or URL link. With the payment request link, the customers can pay their orders with the payment gateway.



Request Payment

Name	John Tan
Address	30, Jalan Hijau, Taman Biru 80000 Johor
Contact	0123345568
Email	johntan@gmail.com

Total Product Cost	RM40.00
Delivery Cost	RM5.00
Delivery Method	Custom Rate
Discount	0

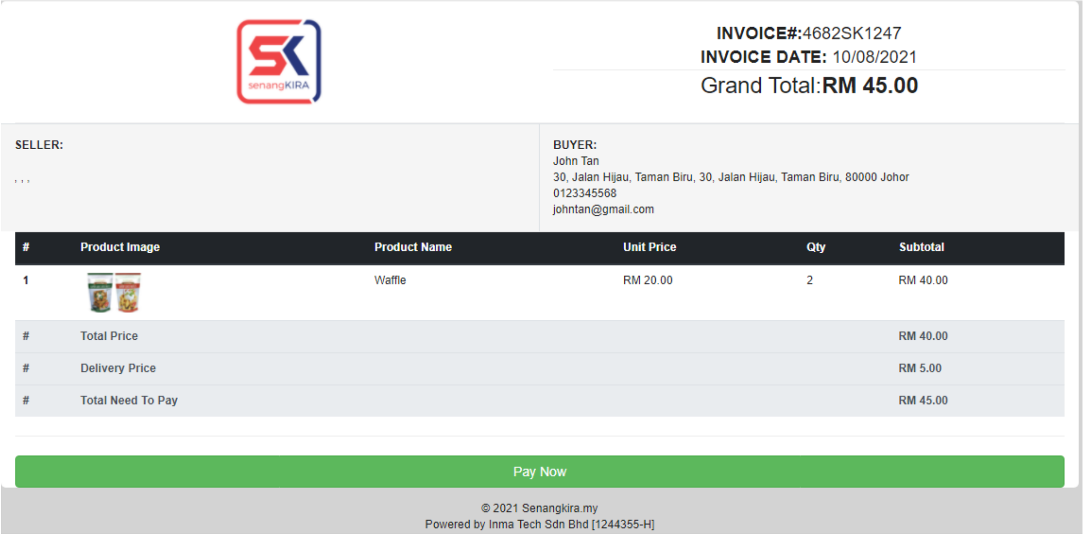
Total Amount

RM45.00

We're now ready to send the Online Payment Link to customer. Use any of the options below to request payment:

Send Payment Link Via Whatsapp
Send Payment Link Via Email
Copy & Manually Send Payment Link

Figure 2.17: Create New Order in senangKIRA System



INVOICE#:4682SK1247
INVOICE DATE: 10/08/2021
Grand Total:**RM 45.00**

SELLER:
, , ,

BUYER:
John Tan
30, Jalan Hijau, Taman Biru, 30, Jalan Hijau, Taman Biru, 80000 Johor
0123345568
johntan@gmail.com

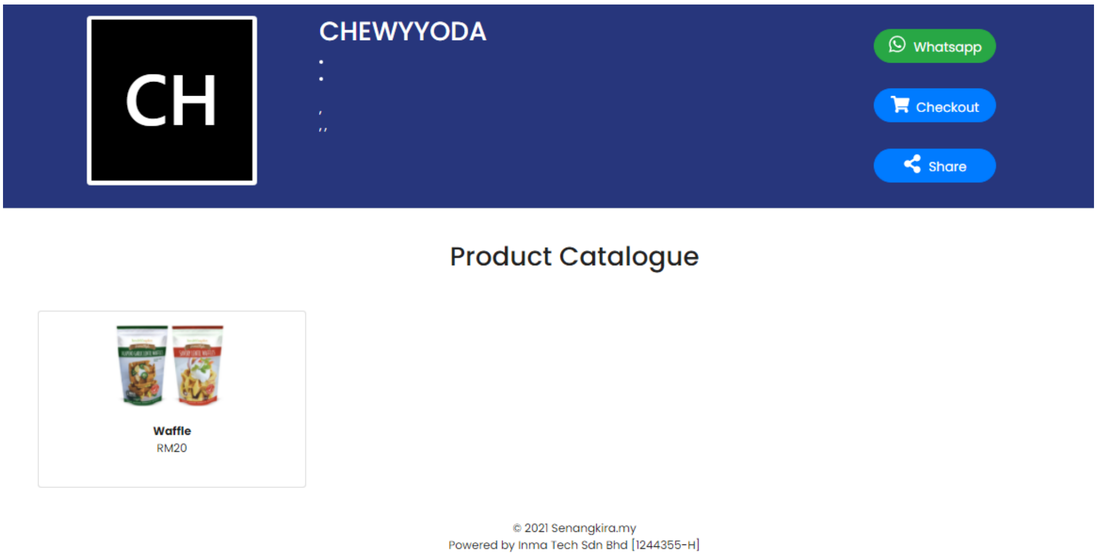
#	Product Image	Product Name	Unit Price	Qty	Subtotal
1		Waffle	RM 20.00	2	RM 40.00
#	Total Price				RM 40.00
#	Delivery Price				RM 5.00
#	Total Need To Pay				RM 45.00

© 2021 Senangkira.my
Powered by Inma Tech Sdn Bhd [1244355-H]

Figure 2.18: Payment Gateway in senangKIRA System

c) View catalogue of products

The users have their unique e-commerce web page maintained by senangKIRA. When the users add new products, the product catalogue on the e-commerce web page will be updated automatically. The users can send their e-commerce URL link to their customers so that their customers can select the products themselves and place orders through the web page.



CHEWYYODA

Product Catalogue

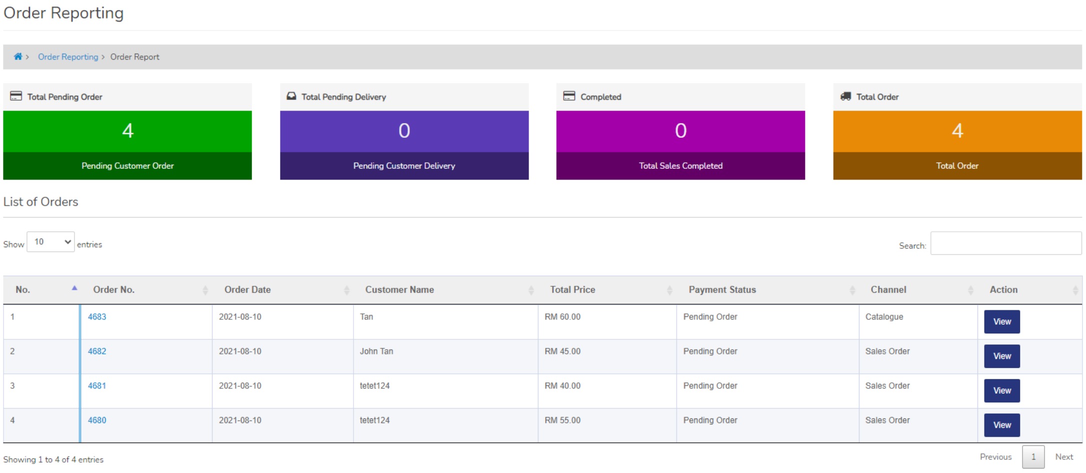
Waffle
RM20

© 2021 Senangkira.my
Powered by Inma Tech Sdn Bhd [1244355-H]

Figure 2.19: senangKIRA E-commerce Web Page

d) Order Management

The users can easily track their orders with the order management feature provided by the senangKIRA system. On the order management page, the users can view the list of all orders and the summary of the orders, such as the total pending order, total pending delivery, total completed order, and the total number of orders.



Order Reporting

Order Reporting > Order Report

Total Pending Order: 4 — Pending Customer Order

Total Pending Delivery: 0 — Pending Customer Delivery

Completed: 0 — Total Sales Completed

Total Order: 4 — Total Order

List of Orders

Show 10 entries

Search:

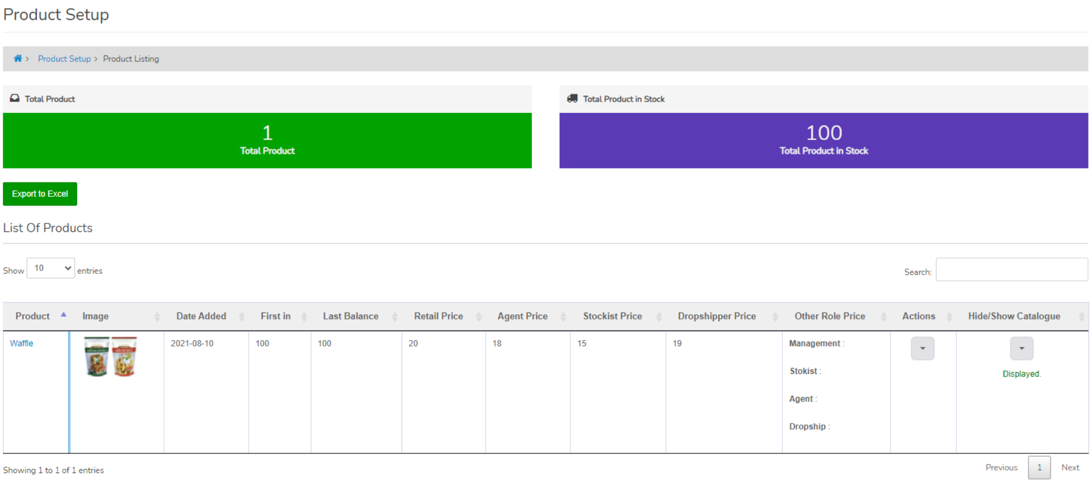
No.	Order No.	Order Date	Customer Name	Total Price	Payment Status	Channel	Action
1	4683	2021-08-10	Tan	RM 60.00	Pending Order	Catalogue	View
2	4682	2021-08-10	John Tan	RM 45.00	Pending Order	Sales Order	View
3	4681	2021-08-10	tetet124	RM 40.00	Pending Order	Sales Order	View
4	4680	2021-08-10	tetet124	RM 55.00	Pending Order	Sales Order	View

Showing 1 to 4 of 4 entries

Figure 2.20: Order Management in senangKIRA

e) Product Setup

The users can add new products to the system. When filling in the product details, the users can specify the price for different clients: agent, stockist, and dropshipper. Besides, the users can set to display or hide the product in their e-commerce catalogue.



Product Setup

> Product Setup > Product Listing

Total Product

1
Total Product

Total Product in Stock

100
Total Product in Stock

Export to Excel

List Of Products

Show 10 entries

Search:

Product	Image	Date Added	First in	Last Balance	Retail Price	Agent Price	Stockist Price	Dropshipper Price	Other Role Price	Actions	Hide/Show Catalogue
Waffle		2021-08-10	100	100	20	18	15	19	Management : Stokist : Agent : Dropship :		Displayed.

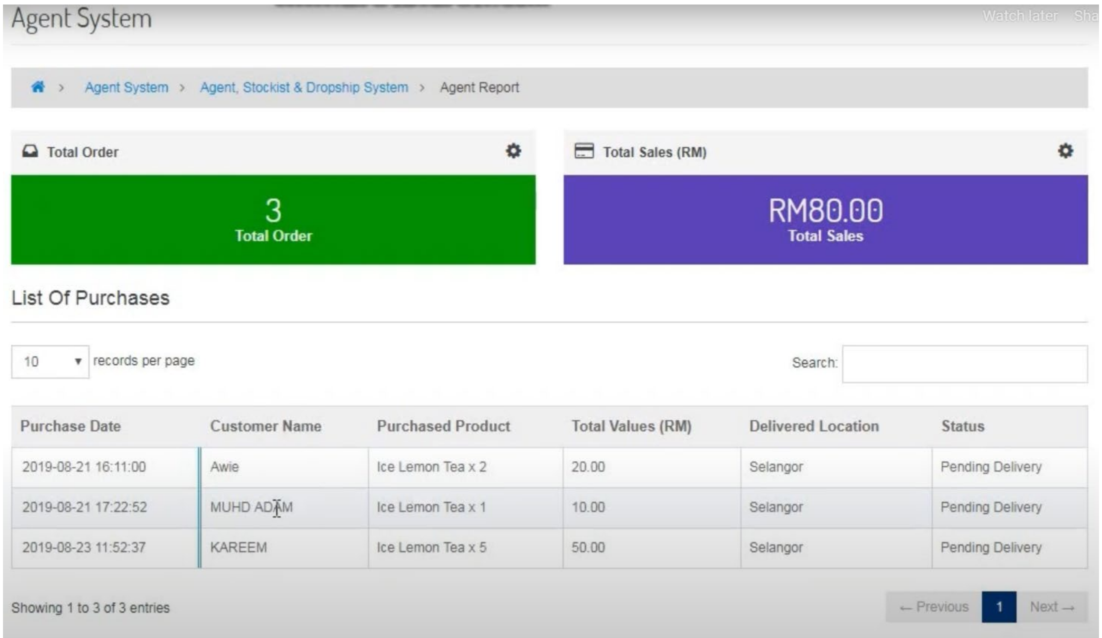
Showing 1 to 1 of 1 entries

Previous 1 Next

Figure 2.21: Product Setup in senangKIRA

f) Agent Management

The unique feature of the senangKIRA system is the agent management feature. In the agent management system, three roles are defined: agent, stockist, and dropshipper. Additionally, the users can also add new roles into the agent management system. Moreover, the users can overview each agent’s order and the order details of all the agent orders, making it easier to manage the agents, stockists, and dropshippers and perform analysis based on their sales.


Agent System
Watch later
Agent System > Agent, Stockist & Dropship System > Agent Report
Total Order
Total Sales (RM)

3
Total Order

RM80.00
Total Sales

List Of Purchases

10 records per page

Search:

Purchase Date	Customer Name	Purchased Product	Total Values (RM)	Delivered Location	Status
2019-08-21 16:11:00	Awie	Ice Lemon Tea x 2	20.00	Selangor	Pending Delivery
2019-08-21 17:22:52	MUHD ADAM	Ice Lemon Tea x 1	10.00	Selangor	Pending Delivery
2019-08-23 11:52:37	KAREEM	Ice Lemon Tea x 5	50.00	Selangor	Pending Delivery

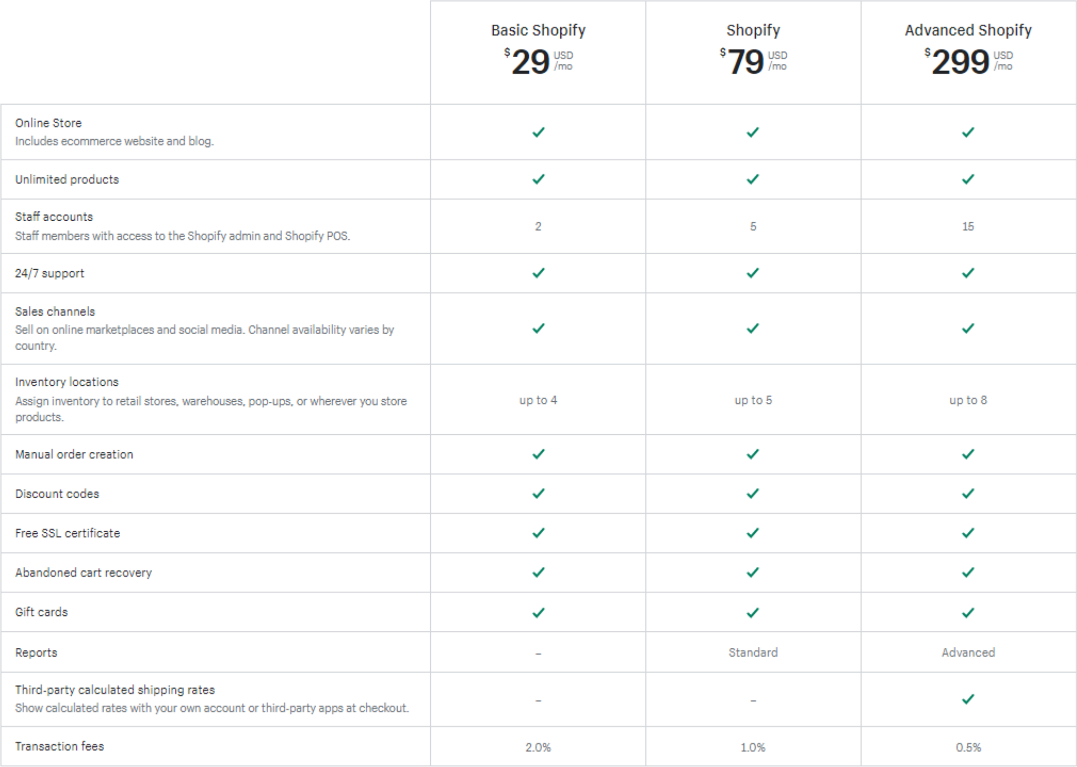
Showing 1 to 3 of 3 entries

Figure 2.22: Agent Management in senangKIRA

2.2.3.3 Shopify

Adopted from <https://www.shopify.my/>

Shopify is a subscription-based system that provides services to set up an online e-commerce store to sell products quickly. In addition, Shopify also provides POS systems services for those who run retail businesses and the inventory management system that links to the online e-commerce store. Shopify was founded in 2006 and has become one of the leading multinational e-commerce companies. One of the reasons that Shopify has become the leading e-commerce company is because it provides a highly customizable e-commerce website to its clients. Furthermore, it contains tons of apps available in the Shopify App Store. The concept of the apps in the Shopify App Store is similar to the so-called Google Chrome extensions, where the apps provide more functionalities to support the Shopify e-commerce platforms. As identical to the senangKIRA system, Shopify requires a subscription plan for its services. There are three types of subscription plans provided by Shopify: Basic Shopify, Shopify, and Advanced Shopify. Figure 2.23 shows the differences among the three different types of subscription plans.



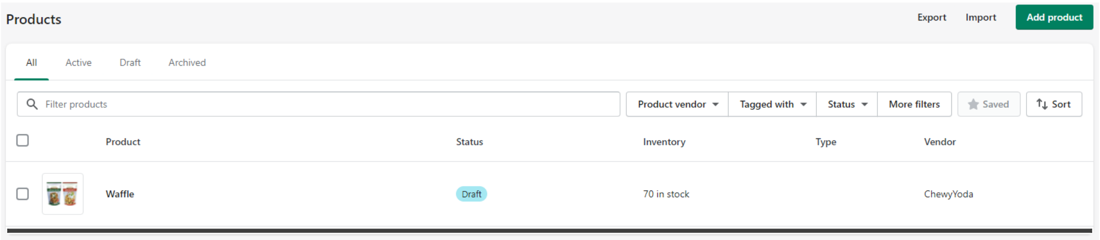
	Basic Shopify $29 USD /mo	Shopify $79 USD /mo	Advanced Shopify $299 USD /mo
Online Store Includes ecommerce website and blog.	✓	✓	✓
Unlimited products	✓	✓	✓
Staff accounts Staff members with access to the Shopify admin and Shopify POS.	2	5	15
24/7 support	✓	✓	✓
Sales channels Sell on online marketplaces and social media. Channel availability varies by country.	✓	✓	✓
Inventory locations Assign inventory to retail stores, warehouses, pop-ups, or wherever you store products.	up to 4	up to 5	up to 8
Manual order creation	✓	✓	✓
Discount codes	✓	✓	✓
Free SSL certificate	✓	✓	✓
Abandoned cart recovery	✓	✓	✓
Gift cards	✓	✓	✓
Reports	–	Standard	Advanced
Third-party calculated shipping rates Show calculated rates with your own account or third-party apps at checkout.	–	–	✓
Transaction fees	2.0%	1.0%	0.5%

Figure 2.23: Shopify Subscription Plans

Since Shopify provides 14-day free trial access for the new users, each functionality of Shopify can be tested without problems. This section will discuss the features available in the Shopify e-commerce platform and its inventory management system. The major features of Shopify are as follows:

a) Product and Inventory Management

The users can add their products quickly into the Shopify system. One of the pros of Shopify is that Shopify provides as detailed input fields as possible. For example, the users can define the cost per item, and the system will automatically compute the margin and profit per item. Besides, the system provides more configurations on the details on the product inventory side. In particular, the users can set the product’s SKU number and the product’s barcode number. Moreover, users can specify whether to continue selling when the products are out of stock.



Products

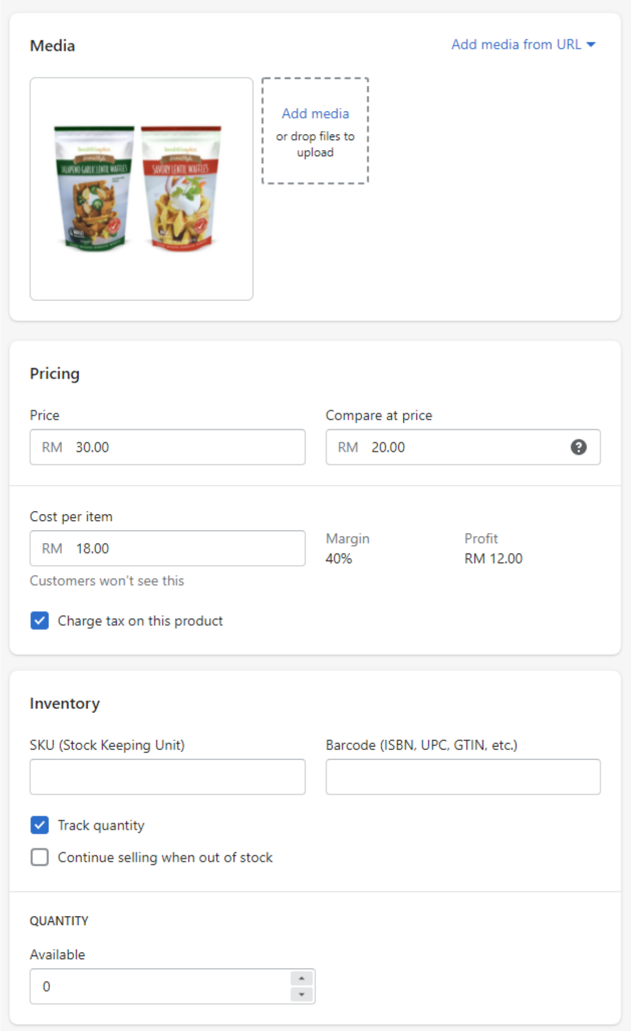
Export Import Add product

All Active Draft Archived

Filter products | Product vendor | Tagged with | Status | More filters | Saved | Sort

□	Product	Status	Inventory	Type	Vendor
□	Waffle	Draft	70 in stock		ChewyYoda

Figure 2.24: Product Management in Shopify



Media

Add media from URL

Add media
or drop files to upload

Pricing

Price: RM 30.00

Compare at price: RM 20.00

Cost per item: RM 18.00

Customers won't see this

Margin: 40%

Profit: RM 12.00

☑ Charge tax on this product

Inventory

SKU (Stock Keeping Unit):

Barcode (ISBN, UPC, GTIN, etc.):

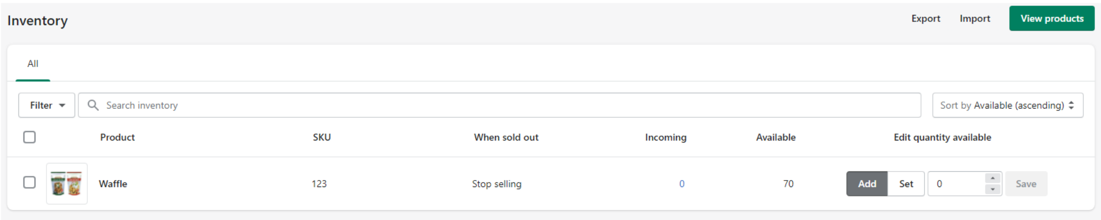
☑ Track quantity

□ Continue selling when out of stock

QUANTITY

Available: 0

Figure 2.25: Part of The Input Fields To Add Product in Shopify



Inventory

Export Import View products

All

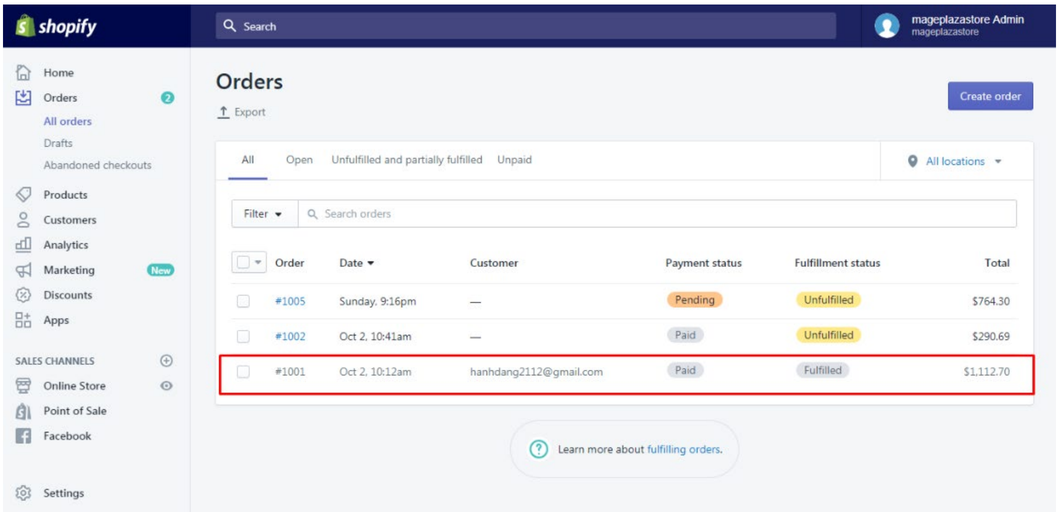
Filter | Search inventory | Sort by Available (ascending)

□	Product	SKU	When sold out	Incoming	Available	Edit quantity available
□	Waffle	123	Stop selling	0	70	Add Set 0 Save

Figure 2.26: Inventory Management in Shopify

b) Order Management

In Shopify, the users can easily manage their orders too. On the order management page, the users can view a list of all orders with information about the order id, order date, customer details, payment status, fulfilment status, and the total order amount. The fulfilment status refers to the status of the seller preparing and delivering the items. If the fulfilment status shows fulfilled, it means that the order has already been packed and dispatched to the delivery carrier and is on the way to the customer.



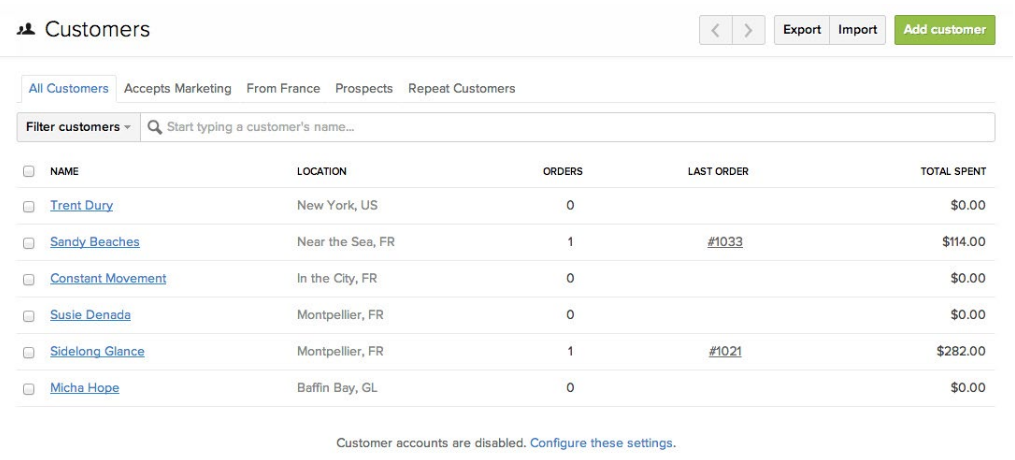
Orders

□	Order	Date	Customer	Payment status	Fulfillment status	Total
□	#1005	Sunday, 9:16pm	—	Pending	Unfulfilled	$764.30
□	#1002	Oct 2, 10:41am	—	Paid	Unfulfilled	$290.69
□	#1001	Oct 2, 10:12am	hanhdang2112@gmail.com	Paid	Fulfilled	$1,112.70

Figure 2.27: Order Management in Shopify

c) Customers management

The users can view the list of their customers on the customer management page in Shopify. On this page, the users can add a new customer by filling up the customer details. Besides, the users can view the lists of customers who accept to receive marketing and the repeat customers who have brought the goods from the users more than once.



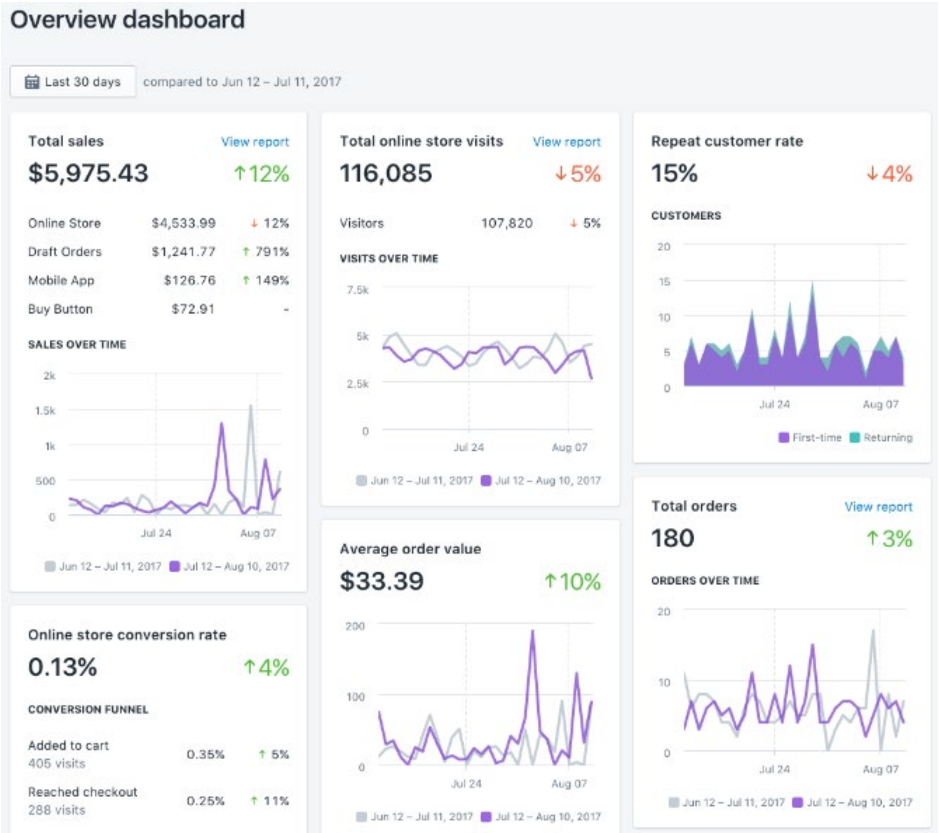
□	NAME	LOCATION	ORDERS	LAST ORDER	TOTAL SPENT
□	Trent Dury	New York, US	0		$0.00
□	Sandy Beaches	Near the Sea, FR	1	#1033	$114.00
□	Constant Movement	In the City, FR	0		$0.00
□	Susie Denada	Montpellier, FR	0		$0.00
□	Sidelong Glance	Montpellier, FR	1	#1021	$282.00
□	Micha Hope	Baffin Bay, GL	0		$0.00

Customer accounts are disabled. Configure these settings.

Figure 2.28: Customer Management

d) Analytics

There are three parts of the analytic feature of Shopify: Dashboards, Reports, and Live View. In Dashboards, the users are allowed to observe the overview of their business analyses with graphical visualisation. In Reports, the users can view the details of the analysis on different criteria they selected in various data ranges.



Overview dashboard

Last 30 days compared to Jun 12 – Jul 11, 2017

Total sales $5,975.43 ↑12%

Online Store	$4,533.99	↓ 12%
Draft Orders	$1,241.77	↑ 791%
Mobile App	$126.76	↑ 149%
Buy Button	$72.91	-

Total online store visits 116,085 ↓5%

Visitors 107,820 ↓ 5%

Repeat customer rate 15% ↓4%

Online store conversion rate 0.13% ↑4%

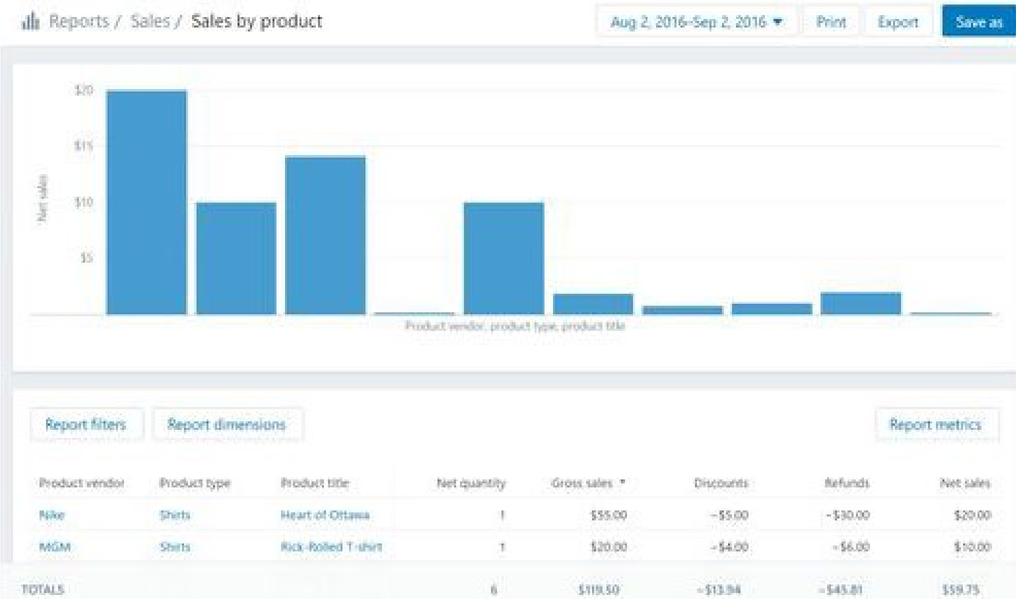
CONVERSION FUNNEL

Added to cart 405 visits	0.35%	↑ 5%
Reached checkout 288 visits	0.25%	↑ 11%

Average order value $33.39 ↑10%

Total orders 180 ↑3%

Figure 2.29: Overview Dashboard in Shopify

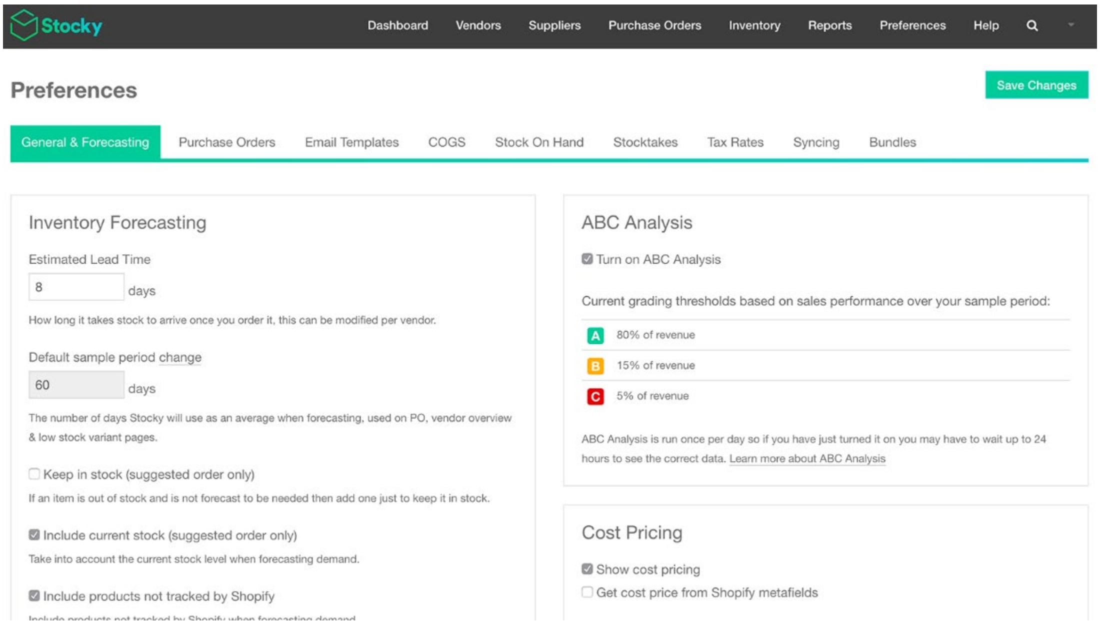

Reports / Sales / Sales by product
Aug 2, 2016–Sep 2, 2016 ▾
Print
Export
Save as
$20
$15
$10
$5
Net sales
Product vendor, product type, product title
Report filters
Report dimensions
Report metrics

Product vendor	Product type	Product title	Net quantity	Gross sales ▾	Discounts	Refunds	Net sales
Nike	Shirts	Heart of Ottawa	1	$55.00	−$5.00	−$30.00	$20.00
MGM	Shirts	Rick-Rolled T-shirt	1	$20.00	−$4.00	−$6.00	$10.00
TOTALS			6	$119.50	−$13.94	−$45.81	$59.75

Figure 2.30: Reports in Shopify

e) Inventory Management Analysis

As stated previously, Shopify App Store allows the users to add apps that provide some additional functionalities not available in Shopify. Some of the apps, such as Stocky, developed by Shopify, and Inform Analytics, developed by JM Solutions, provide further inventory management analysis like ABC analysis, low stock, etc. The users can add the apps to integrate them into their inventory management system.



Preferences

Inventory Forecasting

Estimated Lead Time: 8 days

How long it takes stock to arrive once you order it, this can be modified per vendor.

Default sample period change: 60 days

The number of days Stocky will use as an average when forecasting, used on PO, vendor overview & low stock variant pages.

□ Keep in stock (suggested order only)

If an item is out of stock and is not forecast to be needed then add one just to keep it in stock.

☑ Include current stock (suggested order only)

Take into account the current stock level when forecasting demand.

☑ Include products not tracked by Shopify

Include products not tracked by Shopify when forecasting demand.

ABC Analysis

☑ Turn on ABC Analysis

Current grading thresholds based on sales performance over your sample period:

A	80% of revenue
B	15% of revenue
C	5% of revenue

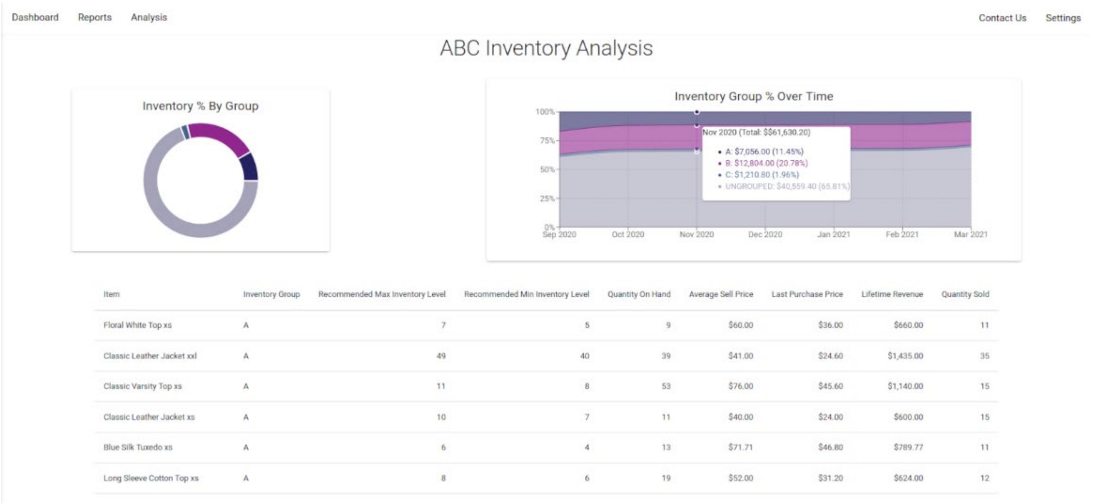
ABC Analysis is run once per day so if you have just turned it on you may have to wait up to 24 hours to see the correct data. Learn more about ABC Analysis

Cost Pricing

☑ Show cost pricing

□ Get cost price from Shopify metafields

Figure 2.31: Stocky, Developed by Shopify

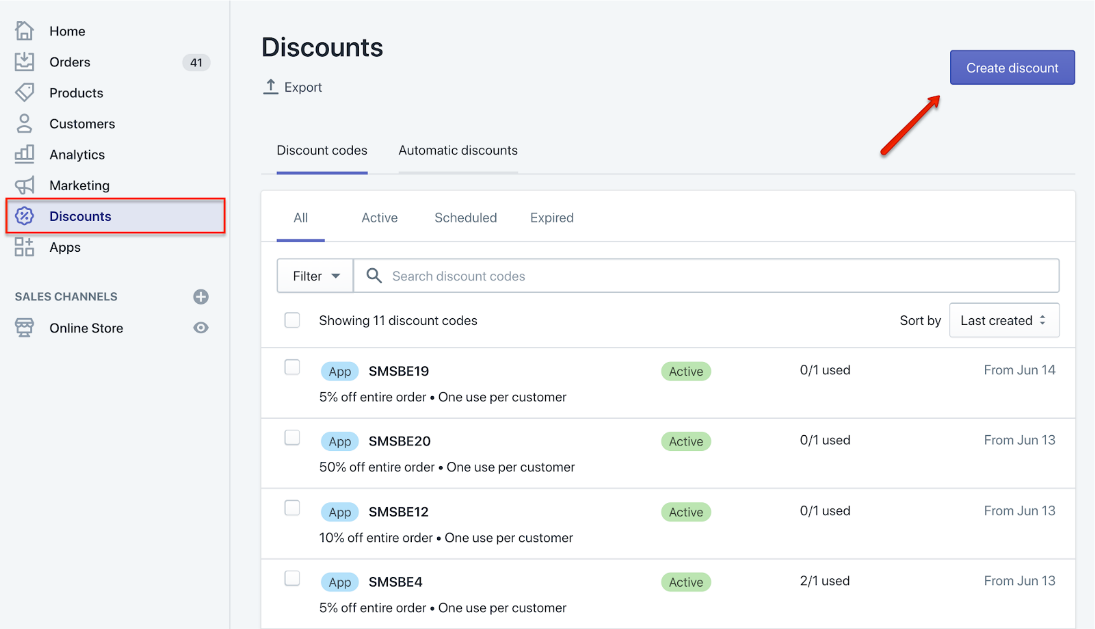


Item	Inventory Group	Recommended Max Inventory Level	Recommended Min Inventory Level	Quantity On Hand	Average Sell Price	Last Purchase Price	Lifetime Revenue	Quantity Sold
Floral White Top xs	A	7	5	9	$60.00	$36.00	$660.00	11
Classic Leather Jacket xxl	A	49	40	39	$41.00	$24.60	$1,435.00	35
Classic Varsity Top xs	A	11	8	53	$76.00	$45.60	$1,140.00	15
Classic Leather Jacket xs	A	10	7	11	$40.00	$24.00	$600.00	15
Blue Silk Tuxedo xs	A	6	4	13	$71.71	$46.80	$789.77	11
Long Sleeve Cotton Top xs	A	8	6	19	$52.00	$31.20	$624.00	12

Figure 2.32: Inform Analytics, Developed by JM Solutions

f) Discounts Management

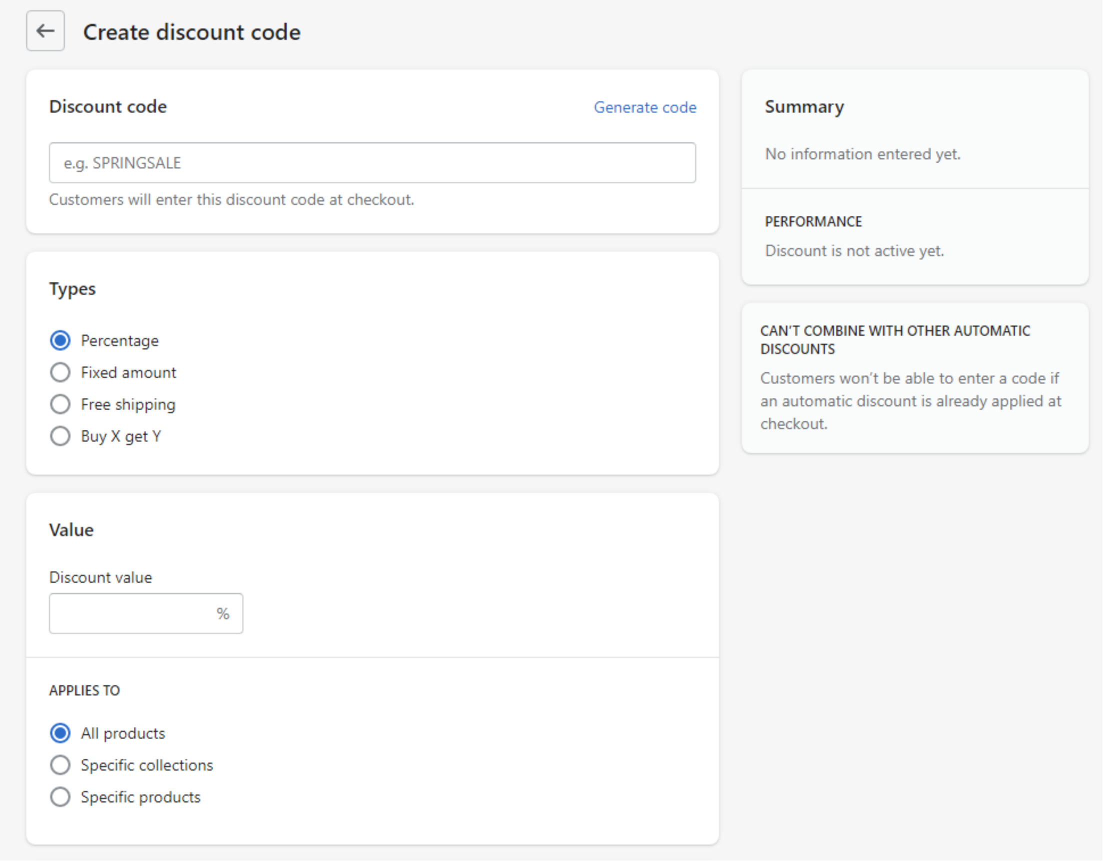
Shopify provides the feature to manage the sellers’ discount codes. The sellers are only required to click on the “Discounts” navigation on the sidebar, and the discounts management page will be shown. The sellers can view all the discounts created as well as perform different filtering and sorting to view the specific discounts. Besides, Shopify also allows the sellers to define the automatic discounts so that the customer will not require to enter the discount code for discount. The seller can also create a new discount code by entering the information of the discount code.



Showing 11 discount codes

App SMSBE19 5% off entire order • One use per customer	Active	0/1 used	From Jun 14
App SMSBE20 50% off entire order • One use per customer	Active	0/1 used	From Jun 13
App SMSBE12 10% off entire order • One use per customer	Active	0/1 used	From Jun 13
App SMSBE4 5% off entire order • One use per customer	Active	2/1 used	From Jun 13

Figure 2.33: Discounts Management in Shopify



Customers will enter this discount code at checkout.

No information entered yet.

Discount is not active yet.

CAN'T COMBINE WITH OTHER AUTOMATIC DISCOUNTS

Customers won't be able to enter a code if an automatic discount is already applied at checkout.

Types:

● Percentage
○ Fixed amount
○ Free shipping
○ Buy X get Y

APPLIES TO:

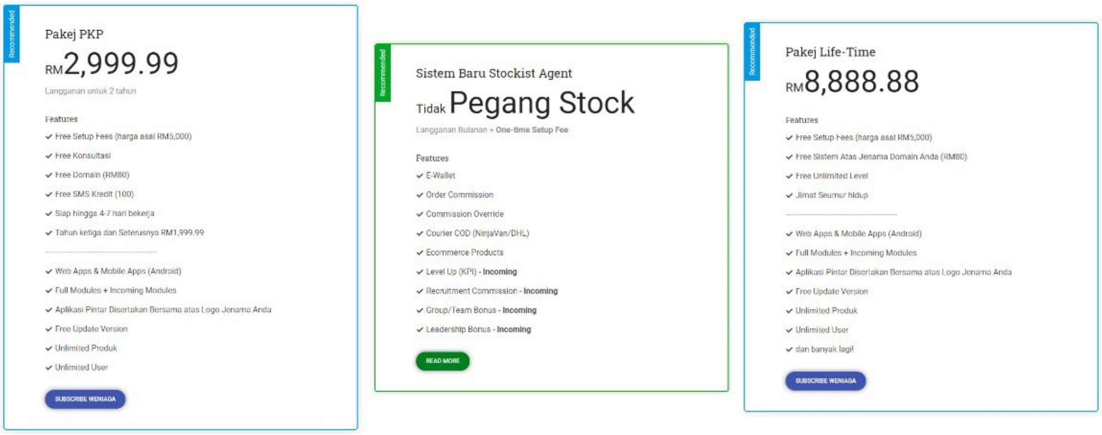
● All products
○ Specific collections
○ Specific products

Figure 2.34: Create Discount Code in Shopify

2.2.3.4 WeNiaga

Adopted from <https://weniaga.com/>

WeNiaga is a company that provides a central inventory management system service to SMEs in Malaysia. The target customer of this company is a product-based company, especially the company that encompasses the roles of stockist, agent and dropshipper. WeNiaga does not provide any free trial access. Instead, it offers some video demonstrations on the features of its inventory management system. Currently, WeNiaga provides three packages: PKP Package, Monthly Package, and Lifetime Package.



Pakej PKP

RM2,999.99

Langganan untuk 2 tahun

Features

- ✓ Free Setup Fees (harga asal RM5,000)
- ✓ Free Konsultasi
- ✓ Free Domain (RM80)
- ✓ Free SMS Kredit (100)
- ✓ Siap hingga 4-7 hari bekerja
- ✓ Tahun ketiga dan Seterusnya RM1,999.99

- ✓ Web Apps & Mobile Apps (Android)
- ✓ Full Modules + Incoming Modules
- ✓ Aplikasi Pintar Disertakan Bersama atas Logo Jenama Anda
- ✓ Free Update Version
- ✓ Unlimited Produk
- ✓ Unlimited User

Sistem Baru Stockist Agent

Tidak Pegang Stock

Langganan Bulanan + **One-time Setup Fee**

Features

- ✓ E-Wallet
- ✓ Order Commission
- ✓ Commission Override
- ✓ Courier COD (NinjaVan/DHL)
- ✓ Ecommerce Products
- ✓ Level Up (KPI) - **Incoming**
- ✓ Recruitment Commission - **Incoming**
- ✓ Group/Team Bonus - **Incoming**
- ✓ Leadership Bonus - **Incoming**

Pakej Life-Time

RM8,888.88

Features

- ✓ Free Setup Fees (harga asal RM5,000)
- ✓ Free Sistem Atas Jenama Domain Anda (RM80)
- ✓ Free Unlimited Level
- ✓ Jimat Seumur hidup

- ✓ Web Apps & Mobile Apps (Android)
- ✓ Full Modules + Incoming Modules
- ✓ Aplikasi Pintar Disertakan Bersama atas Logo Jenama Anda
- ✓ Free Update Version
- ✓ Unlimited Produk
- ✓ Unlimited User
- ✓ dan banyak lagi!

Figure 2.35: Packages Provided by WeNiaga

As there is no way to obtain free trial access without paying, the video demonstrations provided on the WeNiaga websites will be used as review materials. Since the video demonstrations only show a few features, there is no review of stock management and customer management (Agent and dropshipper management) features. This section will discuss the features available in the WeNiaga inventory management system. The main features of the WeNiaga inventory management system are as follows:

a) Dashboard

The users can overview the summary of their sales after they log in to the inventory management system. The analysis such as daily sales, weekly sales, monthly sales and yearly sales can be view in a line graph.

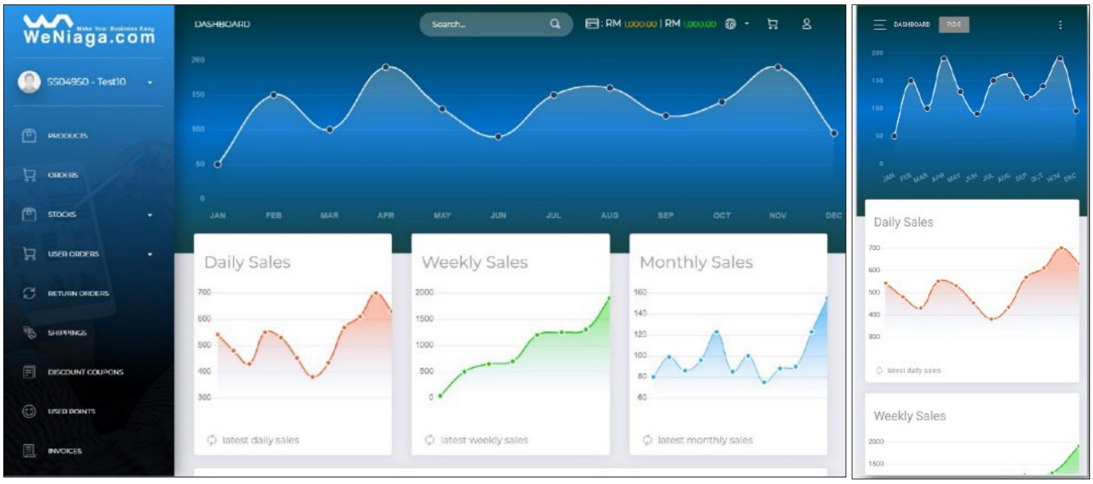
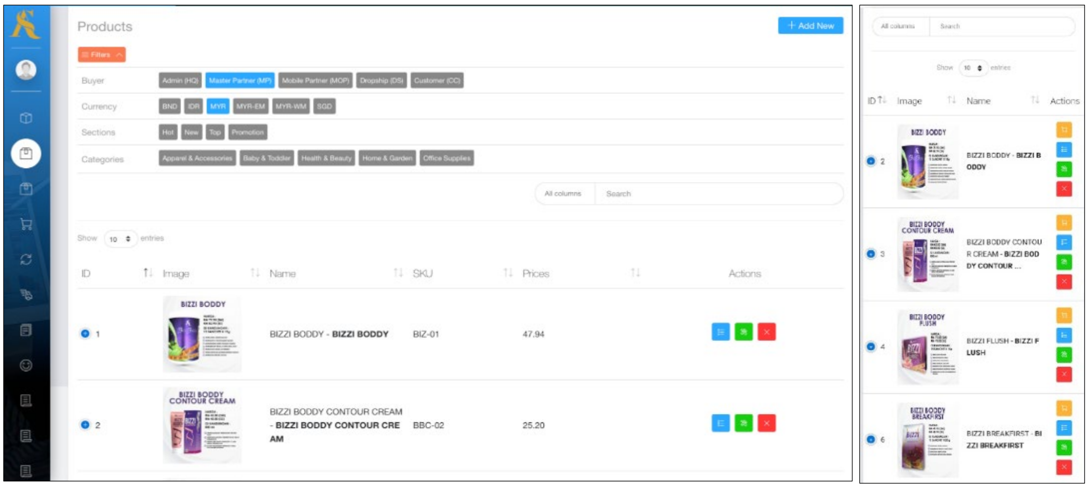


Figure 2.36: WeNiaga Dashboard

b) Product Management

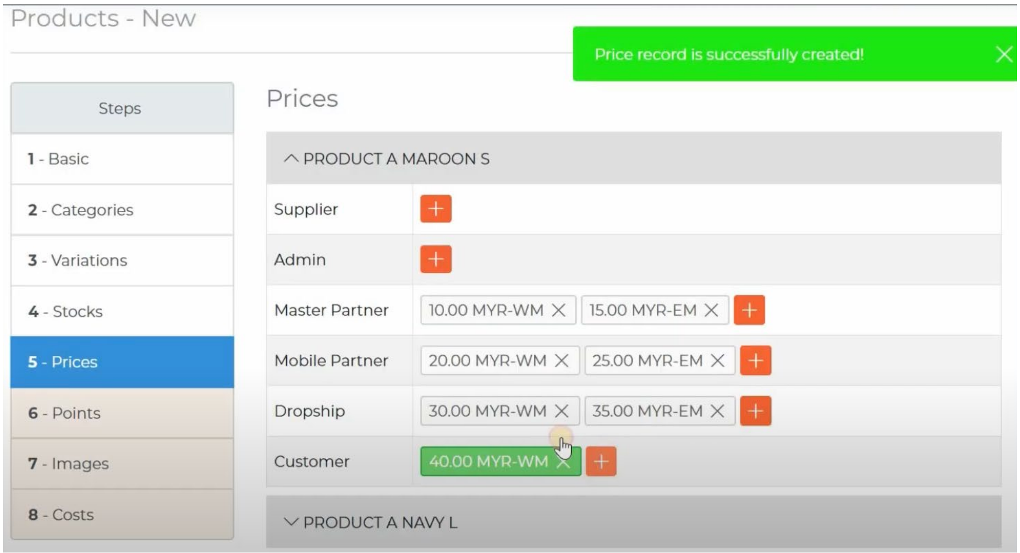
The users are able to view a list of all products on the product management page. Besides, the users can filter different criteria such as product categories, buyer type, currency and sections. The users can also add a new product, edit and delete the existing products by clicking the relevant button. In addition, the users can add different prices for different types of customers, such as a master partner, mobile partner, dropship and customer, as shown in Figure 2.38.



ID	Image	Name	SKU	Prices	Actions
1	BIZZI BODDY	BIZZI BODDY - **BIZZI BODDY**	BIZ-01	47.94	
2	BIZZI BODDY CONTOUR CREAM	BIZZI BODDY CONTOUR CREAM - **BIZZI BODDY CONTOUR CREAM**	BBC-02	25.20	

ID	Image	Name	Actions
2	BIZZI BODDY	BIZZI BODDY - **BIZZI BODDY**	
3	BIZZI BODDY CONTOUR CREAM	BIZZI BODDY CONTOUR CREAM - **BIZZI BODDY CONTOUR ...**	
4	BIZZI BODDY FLUSH	BIZZI FLUSH - **BIZZI FLUSH**	
6	BIZZI BODDY BREAKFIRST	BIZZI BREAKFIRST - **BIZZI BREAKFIRST**	

Figure 2.37: Product Management in WeNiaga



Products - New

Price record is successfully created!

Steps
1 - Basic
2 - Categories
3 - Variations
4 - Stocks
5 - Prices
6 - Points
7 - Images
8 - Costs

Prices

PRODUCT A MAROON S

Supplier	
Admin	
Master Partner	10.00 MYR-WM 15.00 MYR-EM
Mobile Partner	20.00 MYR-WM 25.00 MYR-EM
Dropship	30.00 MYR-WM 35.00 MYR-EM
Customer	40.00 MYR-WM

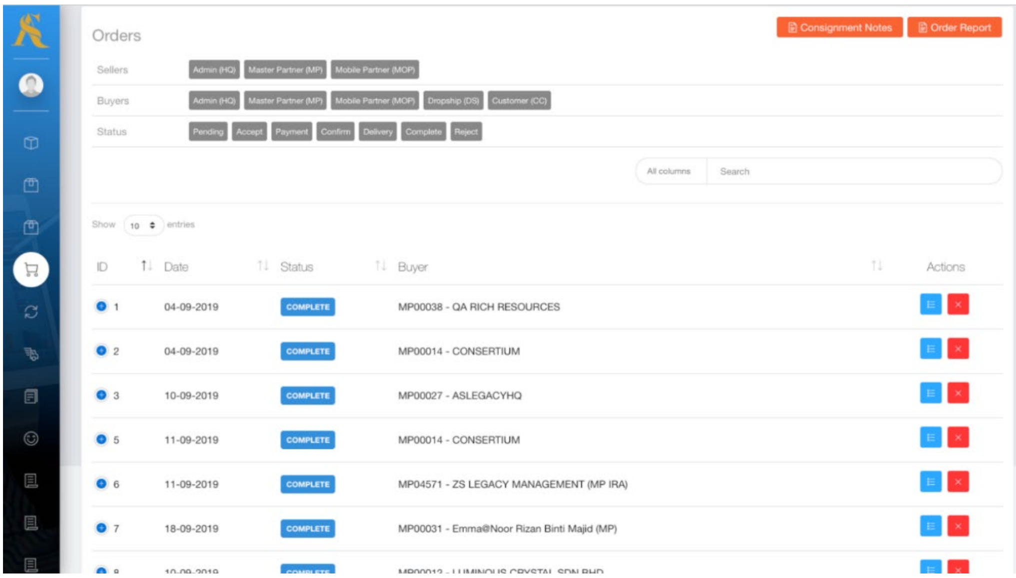
PRODUCT A NAVY L

Figure 2.38: Add Product with Different Prices in WeNiaga

c) Order Management

The users can manage their orders easily by accessing the order management page in WeNiaga. A list of all orders will be shown to the users once they enter the page. The users can filter different criteria such as seller type, buyer type and order status. Moreover, WeNiaga offers the features to create consignment notes and order reports.

Furthermore, the users can add a new order manually by adding the selected products into the cart, applying a coupon (optional), checking out the order and paying the order.



Orders

Consignment Notes | Order Report

Sellers: Admin (HQ) | Master Partner (MP) | Mobile Partner (MOP)

Buyers: Admin (HQ) | Master Partner (MP) | Mobile Partner (MOP) | Dropship (DS) | Customer (CC)

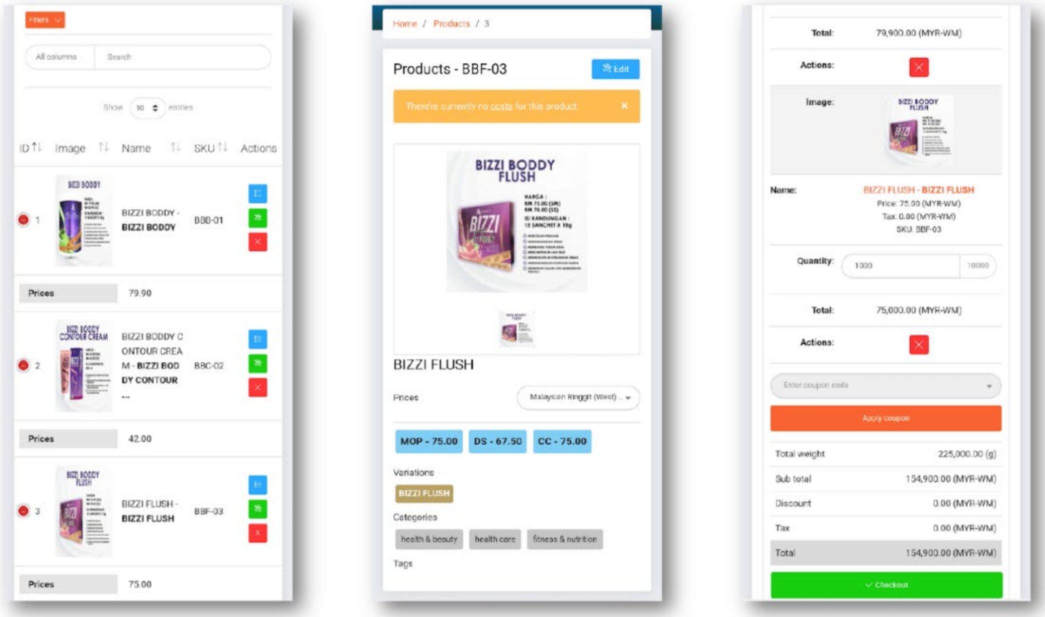
Status: Pending | Accept | Payment | Confirm | Delivery | Complete | Reject

All columns | Search

Show 10 entries

ID	Date	Status	Buyer	Actions
1	04-09-2019	COMPLETE	MP00038 - QA RICH RESOURCES	
2	04-09-2019	COMPLETE	MP00014 - CONSERTIUM	
3	10-09-2019	COMPLETE	MP00027 - ASLEGACYHQ	
5	11-09-2019	COMPLETE	MP00014 - CONSERTIUM	
6	11-09-2019	COMPLETE	MP04571 - ZS LEGACY MANAGEMENT (MP IRA)	
7	18-09-2019	COMPLETE	MP00031 - Emma@Noor Rizan Binti Majid (MP)	
8	10-09-2019	COMPLETE	MP00012 - LUMINOUS CRYSTAL SDN BHD	

Figure 2.39: Order Management in WeNiaga



ID	Image	Name	SKU	Actions
1		BIZZI BODDY - **BIZZI BODDY**	BBB-01	
Prices	79.90			
2		BIZZI BODDY CONTOUR CREAM - **BIZZI BODDY CONTOUR** ...	BBC-02	
Prices	42.00			
3		BIZZI FLUSH - **BIZZI FLUSH**	BBF-03	
Prices	75.00			

Home / Products / 3

Products - BBF-03

There're currently no costs for this product ×

BIZZI FLUSH

Prices: Malaysian Ringgit (West) ...

MOP - 75.00 DS - 67.50 CC - 75.00

Variations: BIZZI FLUSH

Categories: health & beauty, health care, fitness & nutrition

Tags

Total: 79,900.00 (MYR-WM)

Actions:

Image:

Name: BIZZI FLUSH - **BIZZI FLUSH**
Price: 75.00 (MYR-WM)
Tax: 0.00 (MYR-WM)
SKU: BBF-03

Quantity: 1000 10000

Total: 75,000.00 (MYR-WM)

Actions:

Total weight	225,000.00 (g)
Sub total	154,900.00 (MYR-WM)
Discount	0.00 (MYR-WM)
Tax	0.00 (MYR-WM)
Total	154,900.00 (MYR-WM)

Figure 2.40: Ordering process in WeNiaga

2.2.4 Conclusion

Table 2.1: Comparison of the Features in E-commerce platform Among Reviewed Existing Applications

E-commerce Platform				
Application **Features**	**Lazada**	**senangKIRA**	**Shopify**	**WeNiaga**
Platform	Web and mobile	Web	Web and mobile (mobile application is named as 'Shop')	✕
Register as Customer	✓	✕	✓	✕
Register as agent or dropshipper	✕	✕	✕	✕
Login	✓	✕	✓	✕
Profile Management	✓	✕	✓	✕
Browse products and promotions	✓	✓	✓	✕
View product	✓	✓	✓	✕
Add to shopping cart	✓	✓	✓	✕
Checkout	✓	✓	✓	✕
Payment Gateway	✓	✓	✓	✕
Contact Us	✓	✕	✓	✕
FAQ	✓	✕	✓	✕
Notification (e.g. order status)	✓	✓	✓	✕

Table 2.2: Comparison of the Features in Inventory Management System Among Reviewed Existing Applications

Inventory Management System				
Application **Features**	**Lazada**	**senangKIRA**	**Shopify**	**WeNiaga**
Platform	Web and mobile	Web and mobile	Web and mobile	Web and mobile
Register and Login	✓	✓	✓	✓
Product management	✓	✓	✓	✓
Stock management	✓	✓	✓	✓
Order management	✓	✓	✓	✓
Agent and dropshipper management	×	✓	×	✓
Business insights	✓	✓	✓	✓
Inventory management analysis	×	×	✓ (Additional app extension required)	×
Invoice, sales and inventory reports generator	✓	✓	✓	✓
Discounts Management	✓	×	✓	×

Table 2.2 (Continued)

Notification (e.g. New order placement, low stock warning)	✓	✓	✓	✓

After reviewing the four existing computerised inventory management systems, we can conclude that the features in common for the e-commerce platform are browsing the products and promotions, viewing the product, adding products to the shopping cart, checkout, payment gateway and notification on the order status. Among the reviewed applications, senangKIRA, which is more concentrated on its inventory management systems, does not include register, login, profile management, contact us and FAQ features in its e-commerce platform. On the other hand, Lazada and Shopify accommodate all the features mentioned. Therefore, their e-commerce platforms are considered complete and fully functional e-commerce platforms. WeNiaga does not support an e-commerce platform for its inventory management system.

The features in common for the inventory management systems reviewed are registration and login, product management, stock management, order management, business insights, invoice, sales, and inventory reports generator, as well as notification such as the new order placement and low stock warning notifications. The general-purpose inventory management systems like Lazada and Shopify does not provide the feature to manage the agent and dropshipper. The primary reason is that only a few companies, especially in Malaysia, implement an agent and dropshipper system. Hence, the general-purpose inventory management systems do not consider implementing this feature.

Based on all the reviewed systems, only senangKIRA and WeNiaga provide inventory management systems with the agent and dropshipper management feature. Therefore, this project will review and gain ideas on implementing the agent and dropshipper management feature based on these two inventory management systems.

Besides, there are only Lazada Seller Center and Shopify provide the discount management feature for their inventory management system. Hence, the project will also implement the discount management feature so that the any discounts to the direct customers, agents, and dropshippers can be provided with the discount codes.

Among all the reviewed inventory systems, only Shopify provides inventory management analysis such as ABC analysis. However, we must add some apps such as Stocky, developed by Shopify, and Inform Analytics, developed by JM Solutions, to Shopify for the inventory management analysis.

2.3 Inventory Analysis Techniques

As discussed previously, inventory management is critical to streamline the business process and increase the company's cost-efficiency. To further support inventory management, inventory analysis techniques could be one way to improve inventory management performance and cut unnecessary costs. Eveline, et al. (2019) claimed that the primary goal of the inventory analysis techniques is to cut the inventory costs to improve overall business efficiency. Eveline, et al. (2019) further stated that an excess inventory stock leads to a company's financial burden and increases the chances of loss and damage. On the other hand, insufficient stock leads to low customer satisfaction, loss of potential sales and injure the company's reputation.

Furthermore, Biswas, et al. (2017) claimed that a company's inventory contributes to approximately 75 to 80 percent of the total assets for the retailers and wholesalers. Hence, many companies nowadays have implemented different types of inventory analysis to ensure that they have proper control over their inventory to maximise their profit. Eveline, et al. (2019) stated that combining different inventory analysis techniques could produce more obvious improvement of the business performance. Figure 2.41 shows the conceptual framework of inventory analysis techniques, where the output from multiple inventory analysis techniques results in the advancement of procurement performance.

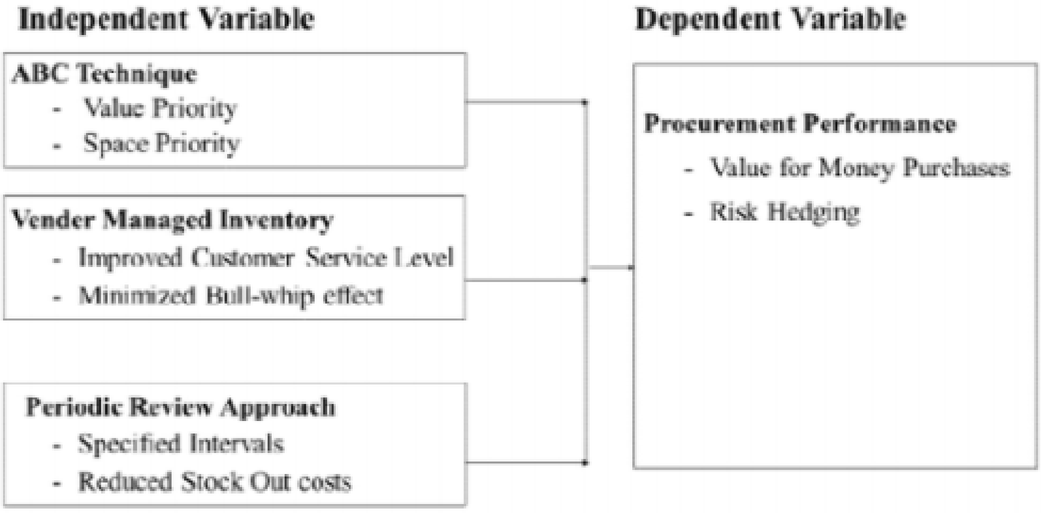


Figure 2.41: Conceptual Framework of Inventory Analysis Techniques (Eveline, et al., 2019)

There is plenty of inventory analysis techniques introduced and used in inventory management systems. In this report, a total of five inventory analysis techniques will be discussed. The five inventory analysis techniques include ABC analysis, HML analysis, VED analysis, Safety Stock (SS) analysis and Economic Order Quantity (EOQ) analysis.

2.3.1 ABC Analysis

ABC analysis, also known as Activity-Based Costs analysis, is a well-known inventory categorisation technique used by many companies to control their inventory items. ABC analysis applies the Pareto principle and works by categorising the inventory items into three sections: A, B and C (Afolabi, et al., 2017; Biswas, et al., 2017; Sporta, 2018). Each category denotes different levels of significance, and the items will be categorised based on their relevance. Category A includes items with a large investment, category B encompasses items with moderate investment, and category C consists of items with a low investment (Afolabi, et al., 2017). The formula to categorise the inventory items into ABC categories is shown as follows (Biswas, et al., 2017):

$$Annual\ Consumption = Annual\ Demand\ \times Unit\ Price \qquad (2.1)$$

where

$Annual\ Demand =\ Annual\ number\ of\ units\ sold\ per\ item$

According to Afolabi, et al. (2017), there are five steps to implement ABC analysis. The five steps of the implementation are as follows:

a) Determine the annual demand and unit price of each inventory item.

b) Calculate the annual consumption based on the above formula.

c) Sort the inventory items based on the annual consumption in descending order.

d) Calculate the percentages of the annual demand of each item and the annual consumption of each item.

e) Classify each inventory item into A, B and C categories, respectively, based on the percentage value of the annual consumption listed in Table 2.3.

Table 2.3: Rules of ABC Analysis (Biswas, et al., 2017; Nadkarni & Ghewari, 2016)

Category	**Item Ratio**	**Annual Consumption Ratio**	**Control Strictness**
A	About 20%	About 80%	Strict control
B	About 30%	About 15%	Moderate control
C	About 50%	About 5%	Lenient control

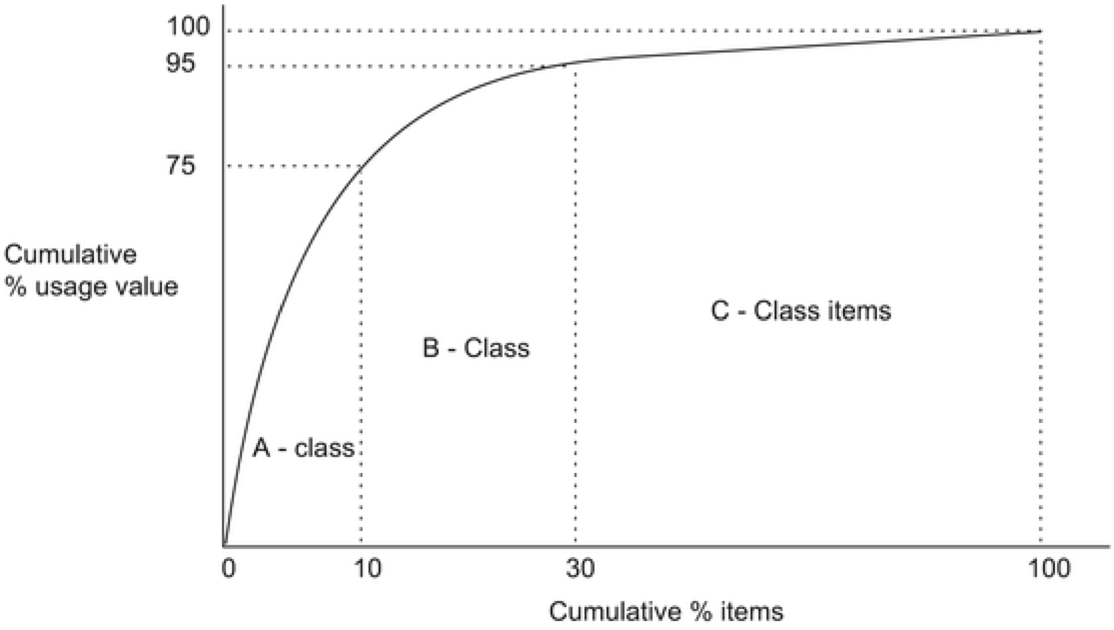


Figure 2.42: Pareto Chart of ABC Classification Analysis (Vrat, 2014)

2.3.2 HML Analysis

HML analysis, also known as High-Medium-Low analysis, is an inventory analysis technique similar to ABC analysis (Dahiwale & Sangode, 2019). The significant difference between HML analysis and ABC analysis is the managing criterion used. Instead of the use of annual consumption value as the managing criterion in ABC analysis, The HML analysis uses a cost per unit criterion (Biswas, et al., 2017; Jadhav & Jaybhaye, 2020). Besides, unlike ABC analysis which uses categories A, B, and C, the HML analysis uses the terms High, Medium, and Low for the inventory classification. The high (H) cost category includes the high unit value items, the Medium (M) cost category contains the medium unit value items, and the Low (L) cost category consists of low unit value items. The HML analysis also uses the Pareto principle, but the percentage for each category is slightly different from the ABC analysis. The formula to categorise the inventory items into ABC categories is shown as follows:

$$Percentage\ of\ Unit\ Cost = \frac{Unit\ Cost\ per\ Item}{Total\ Unit\ Cost\ of\ All\ Items} \quad (2.2)$$

According to Jadhav and Jaybhaye (2020) as well as Kumar, et al. (2016), there are four steps to implement HML analysis. The four steps of the implementation are as follows:

a) Compute unit cost for each inventory item.
b) Sort the inventory items based on their unit cost in descending order.
c) Compute the percentage of unit cost for each inventory item based on the above formula.
d) Classify each inventory item into categories H, M, and L, respectively, based on the unit cost ratio listed in Table 2.4.

Table 2.4: Rules of HML Analysis (Kumar, et al., 2016)

Category	**Item Ratio**	**Unit Cost Ratio**	**Control Strictness**
H	About 15 to 20%	About 75%	Strict control
M	About 20 to 25%	About 15%	Moderate control
L	About 60 to 70%	About 10%	Lenient control

2.3.3 Safety Stock (SS) Analysis

Safety stock is the minimum additional quantity of the inventory item to act as a safety margin to avoid the stock-out problem (Biswas, et al., 2017; Tom, et al., 2013). Since the demand could be difficult to determine, the safety stock becomes necessary to satisfy the item demand when it exceeds the expected demand. Although having safety stock can guard against stock-out problems, it will increase the holding cost of the inventory items, especially when there is too much safety stock. Therefore, it is crucial to maintain a balance between the safety stock and customer satisfaction. Biswas, et al (2017) proposed a formula to calculate the safety stock. The formula is as follows:

$$SS = Z \times D \times \sigma_L \tag{2.3}$$

where

$D = Demand\ per\ year\ (units)$

$Z = Standard\ Normal\ Value$

$\sigma_L = Standard\ Deviation\ of\ Lead\ Time$

The above formula assumes that the demand is constant. The Z value is dependent on the cycle service level. For example, if the cycle service level is assumed to be 95%, the Z value will be 1.645 (Biswas, et al., 2017). Since it is tough to determine the cycle service level, and the result might not be accurate when using assumptions, a simplified formula was found from Clarke (2021) and Stitch Labs (2019). The simplified formula is as follows:

$$SS = (S_{max} \times L_{max}) - (S_{avg} \times L_{avg}) \qquad (2.4)$$

where

$SS = Safety\ Stock$

$S_{max}\ = Maximum\ Daily\ Demand\ (Highest\ number\ of\ items\ sold)$

$L_{max}\ = Maximum\ Lead\ Time\ in\ Days$

$S_{avg}\ = Average\ Daily\ Demand\ (Average\ number\ of\ item\ sold)$

$L_{avg}\ = Average\ Lead\ Time\ in\ Days$

Using the simplified formula above, we can determine the number of stocks the company can hold as a reserved stock to prevent the items from stock-out.

2.3.4 Economic Order Quantity (EOQ) Analysis

Economic Order Quantity is an inventory analysis technique used to identify the optimum amount of inventory items to order each time (Afolabi, et al., 2017; Sporta, 2018; Tom, et al., 2013). In inventory management, the amount of the inventory items ordered will affect the inventory ordering and holding costs. Therefore, EOQ aims to minimise the inventory ordering and holding costs by calculating the optimum amount of items to be ordered to increase the company's profit. The formula to compute the EOQ of an item is shown as follows (Biswas, et al., 2017; Tom, et al., 2013):

$$EOQ = \sqrt{\frac{2DS}{H}} \qquad (2.5)$$

where

$D =\ Demand\ per\ time =\ Total\ number\ of\ units\ sold\ per\ time$

$S = Reorder\ cost =\ Fixed\ cost\ per\ purchase\ order$

$H = Holding\ cost\ per\ time\ (unit)$

According to Afolabi, et al. (2017), there are four assumptions made by the EOQ model. The assumptions are as follows:

a) Demand is constant.
b) Stock is depleted linearly and constantly.
c) No discount is implied on the quantity of the order purchases.
d) The time interval between placing and receiving the order is fixed.

2.3.5 Conclusion

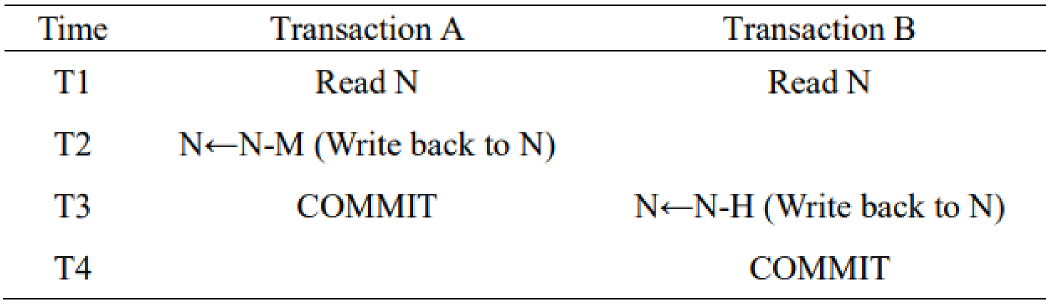
All in all, four types of inventory analysis techniques, ABC analysis, HML analysis, SS analysis and EOQ analysis, have been discussed in this section. Each of the inventory analysis techniques produces different analysis results. In my opinion, the inventory management system should be able to compute the safety stock by using SS analysis and the optimum amount of orders purchased each time. The reason is that the system can alert the admin when an item is lower than the safety stock. Besides, the admin can also know the optimum amount of inventory items to be ordered each time to reduce unnecessary costs. The ABC analysis and HML analysis both helps to classify the inventory items into different categories. These analyses allow the admin to make changes on the inventory items based on the categories of the inventory items. I think the system can compute and show the categories of the inventory items by either ABC analysis or HML analysis. However, it depends on the admin to change the control of the inventory items or set specific procedures for each category.

2.4 Database Transaction Management and Concurrency Control

The database is a collection of data stored in an electronic form. It is one of the essential parts of any software system, especially in this world of big data. In a database, a series of actions that access or change the database's contents is called a transaction (Liu & Zhou, 2015; Solanki, 2018). According to Solanki (2018), there are two purposes of the transaction: to allow rollback or recovery from the failure and to offer isolation between systems when accessing the database concurrently. Furthermore, the transaction must have ACID characteristics, including atomic, consistency, isolation and durability (Ghuman, 2016). Atomic attribute refers to 'all or nothing' property, which means that the transaction should be completed or should not start at all. The consistency attribute implies that the transaction should maintain the consistency and integrity of the database. The isolation attribute means that each transaction should not interfere with other transactions, and the effect of an incomplete transaction should not be visible to other transactions. Lastly, durability infers that the impact of a committed transaction is permanent and must not be affected due to failure.

There are times when the data in the database are accessed concurrently by several users, which would probably lead to inconsistency. That is why transaction management and concurrency control come into place. In database management, concurrency is a situation when there is more than one transaction trying to access and change the database's contents simultaneously (Gohil & Dolia, 2016). Concurrency control refers to managing the concurrent executions of the transactions, allowing the users to access the database simultaneously while maintaining the consistency and integrity of the database (Batra & Kapil, 2010; Quasim, 2013). According to Liu and Zhou (2015), three concurrency problems lead to inconsistency: lost update, dirty read, and non-repeatable read.

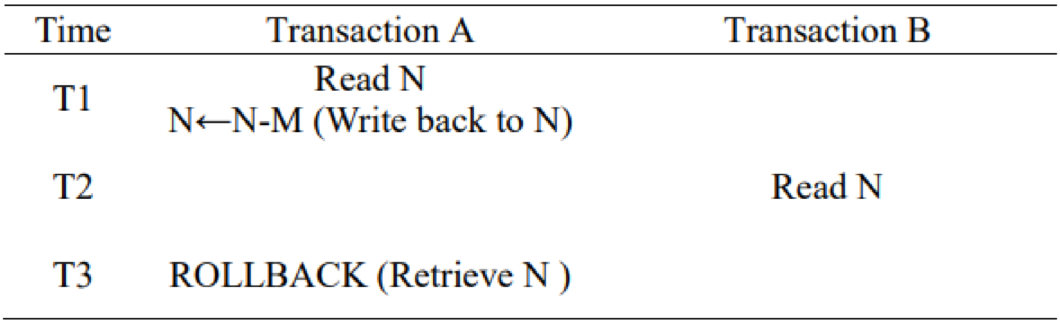
The lost update refers to when two transactions read and write on the same data from the database. In this situation, one's successful write on the database would cause another write to be lost. As shown in Figure 2.43, transaction B commits after transaction A, but the value read by transaction B is the old value. Therefore, the error of balance data occurs.



Time	Transaction A	Transaction B
T1	Read N	Read N
T2	N←N-M (Write back to N)	
T3	COMMIT	N←N-H (Write back to N)
T4		COMMIT

Figure 2.43: Lost Update

The dirty read refers to a situation when one transaction can see the intermediate result of another transaction before it has been committed. For the example illustrated in Figure 2.44, transaction B reads the same data with transaction A right after transaction A modifies the data before it commits. When transaction A rollbacks due to some errors occur, the data modified by transaction A will also be recovered back to the old value. However, the data value read by transaction B will not be recovered back to the old value, resulting in an inconsistent database.



Time	Transaction A	Transaction B
T1	Read N N←N-M (Write back to N)	
T2		Read N
T3	ROLLBACK (Retrieve N)	

Figure 2.44: Dirty Read

The non-repeatable read refers to a situation when a transaction reads the data value. Still, another transaction updates the data value during its execution, which causes the transaction that reads the data value could not reproduce the same result.

This section will introduce three types of concurrency control techniques that assist in scheduling concurrent operations in a database. The concurrency control techniques to be discussed include the lock-based, timestamp-based and optimistic concurrency controls.

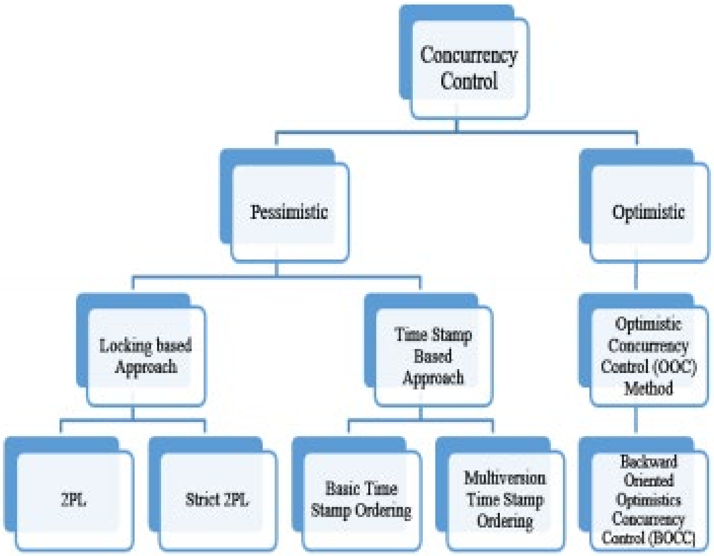
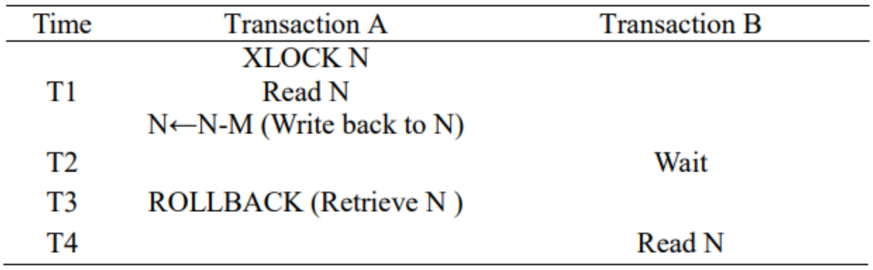

Concurrency Control
Pessimistic
Optimistic
Locking based Approach
Time Stamp Based Approach
Optimistic Concurrency Control (OOC) Method
2PL
Strict 2PL
Basic Time Stamp Ordering
Multiversion Time Stamp Ordering
Backward Oriented Optimistics Concurrency Control (BOCC)

Figure 2.45: Basic Taxonomy of the Concurrency Control (Gohil & Dolia, 2016)

2.4.1 Lock-based Concurrency Control

Lock-based concurrency control is one of the pessimistic techniques, which assumes that conflict would often occur among the transactions in a regular operation (Gohil & Dolia, 2016). Different researchers have introduced various types of lock-based concurrency control, such as two-phase locking protocol, simplistic lock protocol, and pre-claiming lock protocol. Nonetheless, this section will focus on only one lock-based concurrency control, the two-phase locking protocol (2PL).

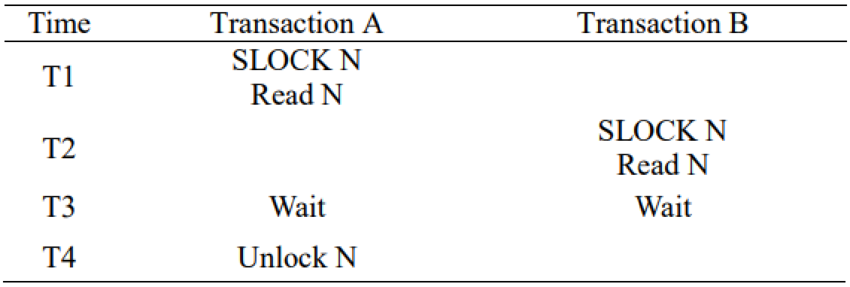
There are two lock modes in any lock-based concurrency control exclusive locks and share locks (Abbas, et al., 2016; Liu & Zhou, 2015). An exclusive lock, also known as the write lock, is the lock that offers permission to perform both read and write operations on the data. With exclusive lock, no other transactions can lock the same data to prevent them from accessing the data simultaneously and solve the dirty read problem. Figure 2.46 shows an example of an exclusive lock.



Time	Transaction A	Transaction B
T1	XLOCK N Read N N←N-M (Write back to N)	
T2		Wait
T3	ROLLBACK (Retrieve N)	
T4		Read N

Figure 2.46: Exclusive Lock

A shared lock, also known as a read lock, is a lock that offers permission only to perform a read operation. When data is locked by shared lock, no write operation will be allowed on the share-locked data. With a shared lock, lost updates problem can be resolved because the transactions can only read the data value but cannot write a new value to the data. Figure 2.47 shows the example of a shared lock.



Time	Transaction A	Transaction B
T1	SLOCK N Read N	
T2		SLOCK N Read N
T3	Wait	Wait
T4	Unlock N	

Figure 2.47: Shared Lock

2PL is a lock-based concurrency control that the data are locked by a transaction and released in two phases. It helps solve the problem of conflicting transactions because there will be no more than one transaction accessing the same data. The two phases introduced in 2PL are growing phase and shrinking phase. In the growing phase, the transaction can only receive locks. There is no release of the locks during the growing phase. The growing phase started right after the transaction acquired the first lock. When the transaction obtains all the locks, this point is called Lock Point. After that, the shrinking phase starts while the growing phase ends. In the shrinking phase, the transaction starts to release the locks obtained during the growing phase. Similarly, there are no more locks that can be acquired during shrinking phases. The transaction and the shrinking process end once all the locks are released (Abbas, et al., 2016).

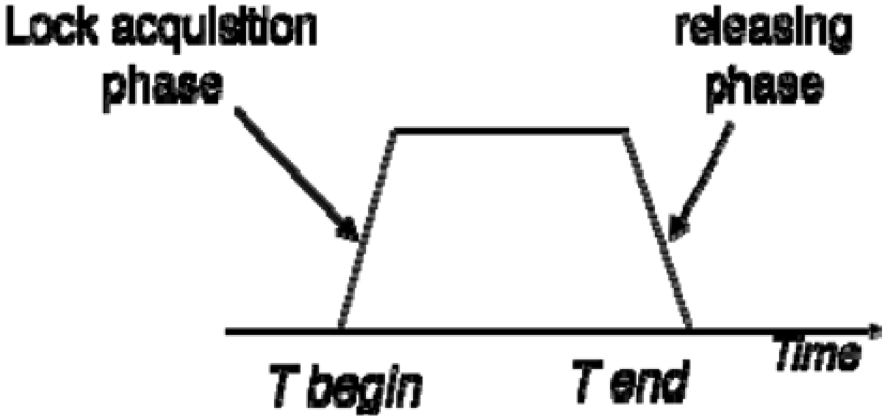


Figure 2.48: Two-Phase Locking (Abbas, et al., 2016)

Table 2.5: Advantages and Disadvantages of 2PL (Abbas, et al., 2016; Gohil & Dolia, 2016; Nasseri & Jameii, 2017)

Advantages	Disadvantages
• Only one transaction is allowed to access the data. Data consistency and integrity is guaranteed. • It performs well in the centralised environment.	• It might result in inconsistency. • It might cause a deadlock when both transactions are waiting for each other. • It might cause starvation if one transaction never commit. • It produces a higher overhead due to the presence of locks.

2.4.2 Timestamp-based Concurrency Control

The timestamp-based concurrency control is also one of the pessimistic techniques, which assumes that conflict would often occur among the transactions in a regular operation (Gohil & Dolia, 2016). The timestamp-based concurrency control does not lock the data like lock-based concurrency control. Instead, the timestamp-based concurrency control generates a unique timestamp for each transaction whenever a transaction executes using the scheduler or concurrency controller (Gupta, et al., 2018; Solanki, 2018). The timestamp can be created based on two criteria: a value generated by an incremental counter variable or the current timestamp of the system. Both methods to generate the timestamp ensure that each of the timestamps assigned to the transaction is unique. The transaction created earlier will be assigned with a lower timestamp, while the transaction created later will be assigned with a higher timestamp. By doing so, the transaction can be well-managed, and the deadlock problem faced by the lock-based concurrency problem can be solved.

In timestamp-based concurrency control, there are timestamp-ordering rules to follow (Christopher & Kabari, 2020; Gohil & Dolia, 2016). The timestamp-ordering rule can be divided into a read operation, R(x), and a write operation W(x). Besides, each data item will be assigned with a read timestamp (RTS) and a write timestamp (WTS). RTS(x) indicates the largest timestamp of any transaction executed a read operation on x data successfully. In contrast, WTS(x) indicates the largest timestamp of any transaction executed a write operation on x data successfully. The term ts indicates a timestamp, while T indicates a transaction. Whenever a transaction (T) issues read or write operations, the following rules will be checked:

1) Read operation, R(x)

- $ts(T) < WTS(x)$:

  If the transaction reads the data item x that has been updated by the later transaction, the transaction must be aborted and restarted with a new timestamp.

- $ts(T) \geqslant WTS(x)$:

  The transaction can proceed to read the data item x without problem.

2) Write operation, W(x)

- ts(T) < RTS(x):

  If the transaction updates the data item x after the later transaction has read the data, the transaction must be rolled back and restarted with a new timestamp because the current value of the data item x is used by the later transaction.

- ts(T) < WTS(x):

  Suppose the transaction updates the data item x after the later transaction updates the data item x. In this case, the write operation of the transaction can be ignored and aborted due to the ignore obsolete write rule by Thomas's write rule.

- Else, the transaction can be executed without any problem.

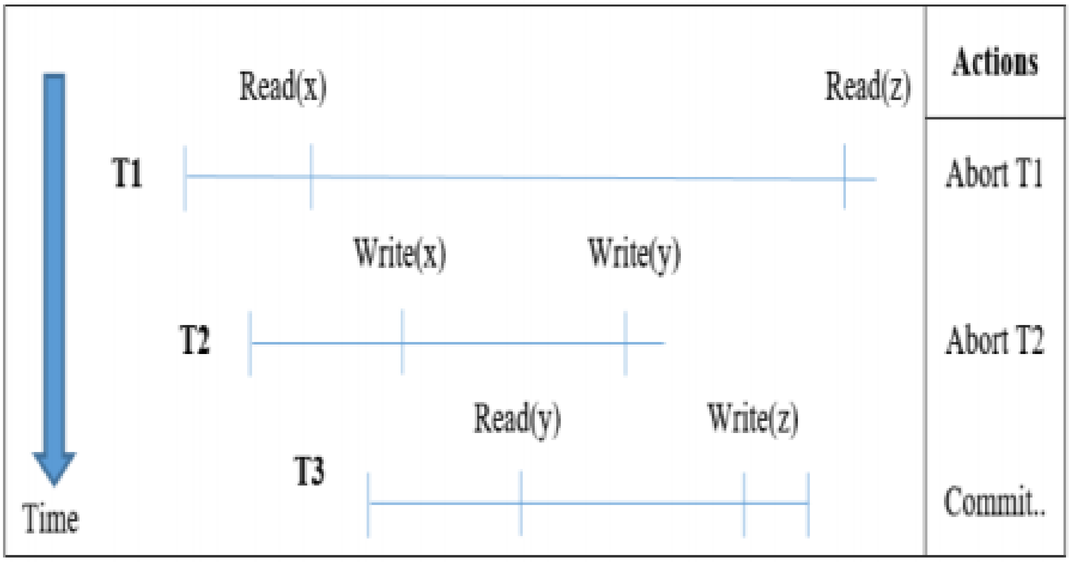


Figure 2.49: Operations of Timestamp-based concurrency control

Table 2.6: Advantages and Disadvantages of Timestamp-based Concurrency Control (Christopher & Kabari, 2020; Solanki, 2018)

Advantages	Disadvantages
• No deadlock occurs because the transactions are executed in a timestamp manner. • It has higher performance than 2PL. Based on Christopher and Kabari (2020) analysis, the average execution time of Timestamp-based concurrency control is 4.84 seconds lower than 2PL.	• Requires additional cost to manage the timestamp.

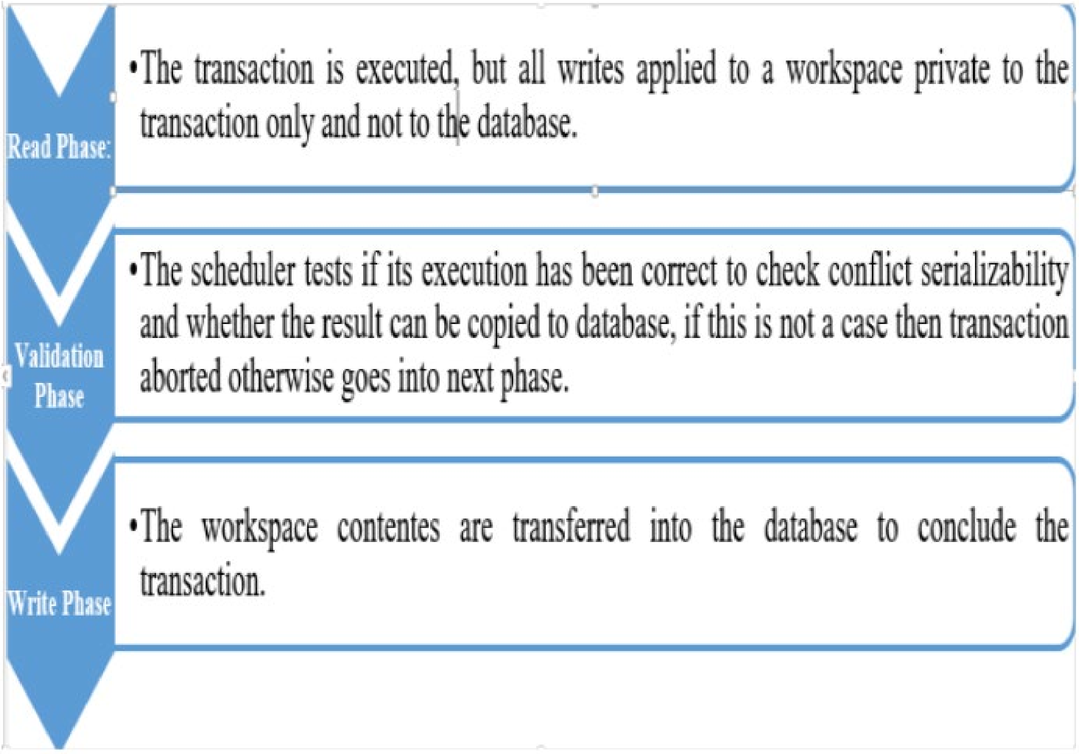
2.4.3 Optimistic Concurrency Control

The optimistic concurrency control is a concurrency control that is different from pessimistic concurrency control. Unlike pessimistic concurrency control, which assumes that conflict would often occur among the transactions in a regular operation, optimistic concurrency control considers that very few transactions will be conflict in a normal operation (Gohil & Dolia, 2016). Due to the difference in the assumptions made by optimistic concurrency control, it allows the transactions to perform their operations without any restriction, such as locking. There are three phases in optimistic concurrency control: read phase, validation phase and write phase (Gohil & Dolia, 2016; Gupta, et al., 2018; Solanki, 2018).

In the read phase, the transaction read the data value from the database, and any updates will only apply to a local copy instead of the database.

In the validation phase, the transaction will be checked to verify that the execution will not cause the loss of data integrity. If the validation fails, the transaction will be aborted.

In the write phase, the updates made by the transaction will store in the database permanently. Figure 2.50 shows the summary of the three phases of this optimistic concurrency control.



Read Phase:

- The transaction is executed, but all writes applied to a workspace private to the transaction only and not to the database.

Validation Phase

- The scheduler tests if its execution has been correct to check conflict serializability and whether the result can be copied to database, if this is not a case then transaction aborted otherwise goes into next phase.

Write Phase

- The workspace contentes are transferred into the database to conclude the transaction.

Figure 2.50: Three Phases of Optimistic Concurrency Control (Gohil & Dolia, 2016)

Table 2.7: Advantages and Disadvantages of Optimistic Concurrency Control (Gohil & Dolia, 2016; Gupta, et al., 2018; Solanki, 2018)

Advantages	Disadvantages
• There is no blocking such as locking applied to the data and transaction. • It is deadlock-free. • It provides a higher degree of concurrency.	• Only suitable for the applications that the conflict between transactions is rarely to occur. • It requires a higher storage cost to store the update information of the transactions.

2.4.4 Conclusion

As a whole, database transaction management and concurrency control are crucial, especially for the systems where the data is often accessed concurrently. The reviews conclude that the optimistic concurrency control has a better performance than the timestamp-based concurrency control and 2PL. Besides, the 2PL has the worst performance than the optimistic concurrency and timestamp-based concurrency control. Although 2PL has the worst performance, it is still an option for some applications that require high consistency guarantees. For instance, lock-based concurrency control is more suitable for booking systems such as airline booking systems, cinema booking systems, bus booking systems, etc. The primary reason is that those booking systems contain limited seats; only one seat is available for each user. Therefore, we can observe that whenever a user selects a ticket for the booking systems mentioned above, the seat will be locked in a time-bound, and other users cannot access the seat for the specified time-bound. From my point of view, different application systems would require different types of database transaction management and concurrency control. Each concurrency control technique has its advantages on different types of application systems. Therefore, the concurrency control method selection would depend on the system requirements and the developers' preference.

2.5 Software Development Methodologies

The Software Development Life Cycle (SDLC), also known as the software development process, is a process that contains a series of highly structured activities used to develop a high-quality software project. It has four fundamental phases: planning, analysis, design, and implementation. Each phase encompasses a set of steps that will produce different software project deliverables. (Barjtya, et al., 2017; Dennis, et al., 2015; Egwoh & Nonyelum, 2017; Gajalakshmi, 2016; Sommerville, 2016). Nowadays, there are many SDLC methodologies available for software development. SDLC methodology refers to a framework that employs the SDLC process. Different methodologies have different strategies to implement the SDLC process and thus suit different types of projects. Therefore, the project team should consider different criteria such as the type of project undertaken, team size, and project goals when selecting the suitable methodology for the software project development.

According to Dennis et al. (2015), the SDLC methodologies can be classified into Rapid Application Development (RAD), structured design, and agile development.

The popular methodologies found in the structured design methodology model are the waterfall methodology, parallel development methodology and V-model methodology. On the other hand, the popular models found in RAD are the phased methodology, prototyping methodology and throwaway prototyping methodology.

The methodologies covered in this section are waterfall methodology, V-model methodology, prototyping methodology, and agile methodology.

2.5.1 Waterfall Methodology

The waterfall methodology is the most fundamental SDLC model. As its name implies, it is a linear-sequential software development model. The progress is recognised as flowing steadily in a sequence like a waterfall through four phases: Planning, Analysis, Design, and Implementation (Dennis, et al., 2015; Egwoh & Nonyelum, 2017).

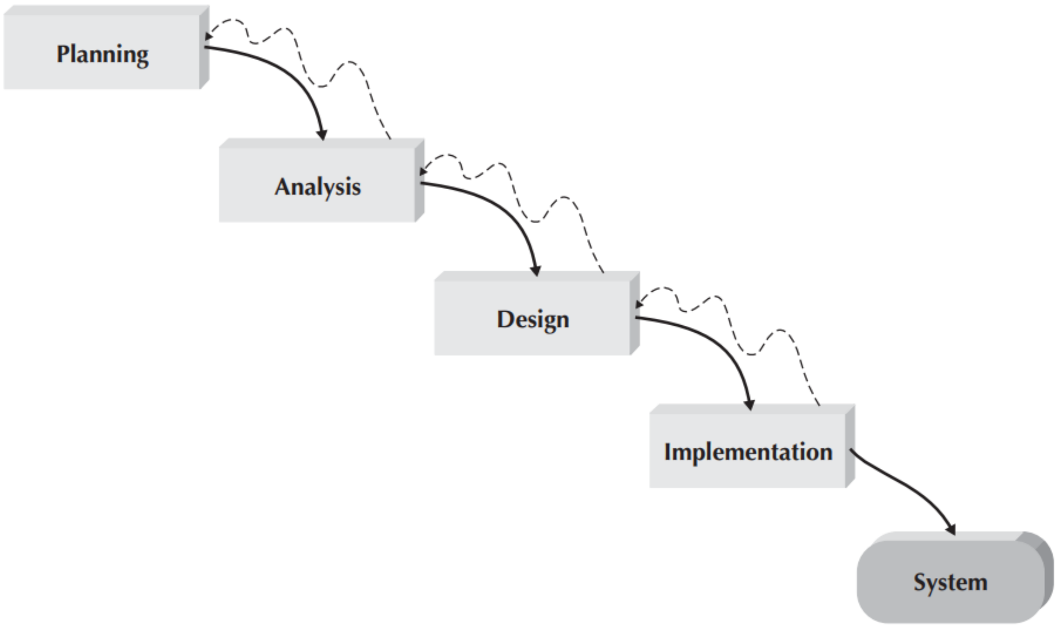


Figure 2.51: Waterfall Development Methodology (Dennis, et al., 2015)

In the waterfall methodology, each phase is completed before entering the next phase. It is of utmost difficulty to reverse back to the previous phase once entering the next phase. Besides, Dennis, et al. (2015) emphasised that each phase typically took a significant time to produce the key deliverables. Dennis, et al. further specify that the project will only move to the next phase after obtaining approval from the project sponsor. Table 2.8 shows the advantages and disadvantages of the waterfall development methodology summarised from different sources.

Table 2.8: Advantages and Disadvantages of Waterfall Development Methodology (Barjtya, et al., 2017; Gajalakshmi, 2016; Munassar & Govardhan, 2010; Salve, et al., 2018; Tarika, 2019)

Advantages	Disadvantages
• Easy to implement and understand • Development phases are processed one by one • Easy to arrange and prioritise tasks • Works well if the business requirements are clearly understood • Have key deliverables and review at the end of each phase	• Difficult to measure the progress of each phase • Difficult to go back to the previous phase • Cannot adapt to the changing business requirements • High risk and uncertainty might lead to the failure of the project • The software can only be delivered after the completion of all phases. Therefore, no early prototypes to review • Not suitable for a large or long-term project

2.5.2 V-Model Methodology

The V-model methodology is an advanced model of the waterfall development methodology. Apart from the linear-sequential flow of progress, the highlight of the V-model methodology is the involvement of different phases of testing in every phase (Barjtya, et al., 2017; Regulwar, et al., 2010).

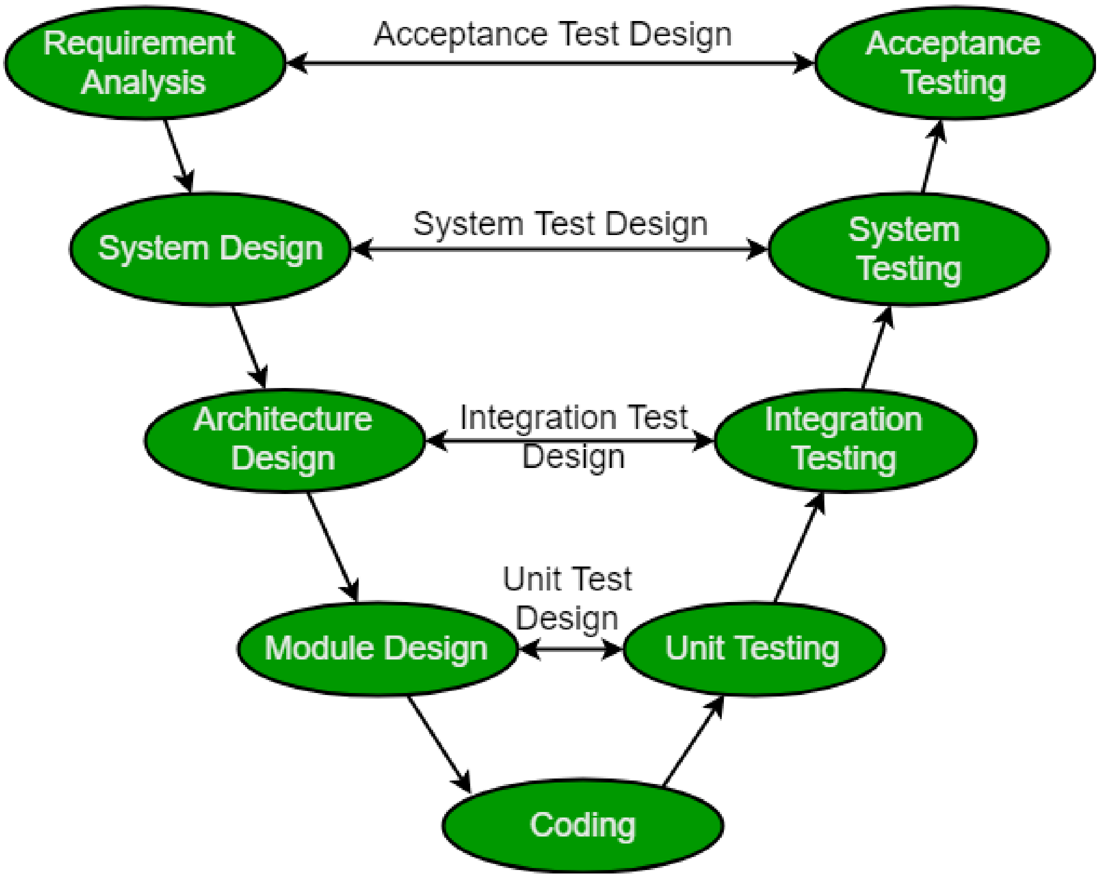


Figure 2.52: V-Model Development Methodology (Tarika, 2019)

In the V-model methodology, every phase is associated with its respective testing phase in a V-shape manner. With the association of the testing phase, the developers are allowed to start their testing early. The developers can find and rectify as many bugs as possible in the early stage. As specified by Tarika (2019), V-model is also known as the Verification and Validation model. The term verification is used for V-model because it includes static testing techniques such as review, inspection, and walkthrough. The developers can execute the testings without the need to run the working system. The verification is mainly used to analyse whether the specific requirements are met before implementing the system.

On the other hand, the term validation means that it includes dynamic testing techniques such as white-box testing and black-box testing, in which the testing is performed by executing the code. The process of validation starts only after the completion of the development phase. Based on Figure 2.52, the verification starts during the requirement analysis phase and ends on the coding phase, while the validation starts after the coding phase to the acceptance testing phase. Table 2.9 shows the advantages and disadvantages of the V-model methodology compiled from different sources.

Table 2.9: Advantages and Disadvantages of V-Model Methodology (Mathur & Malik, 2010; Munassar & Govardhan, 2010; Regulwar, et al., 2010; Tarika, 2019)

Advantages	Disadvantages
• Involves testing phases at the early stage • Easy to implement and understand • Has a higher chance of success over the waterfall methodology • It covers different types of testing, such as unit testing, integration testing, system testing and acceptance testing • Reduces the cost to fix the bugs and defects as they are found at the early stages	• The software can only be delivered after the completion of all phases. Therefore, no early prototypes to review • Rigid and less flexible methodology as like the waterfall methodology • It does not provide a clear procedure to solve the problems found during testing phases • Not suitable for small projects because it involves a lot of reviews at each phase

2.5.3 RAD Prototyping Methodology

RAD, which stands for Rapid Application Development, is a later methodology framework that loomed in the 1990s. The emergence of the RAD-based methodologies aims to address the cons of the structured methodologies such as Waterfall methodology by producing some parts of the system quickly for the user to review and evaluate. There are three RAD-based methodologies: phased methodology (also known as incremental prototyping), prototyping methodology, and throwaway prototyping (Dennis, et al., 2015; Egwoh & Nonyelum, 2017). In this section, RAD prototyping methodology will be discussed in detail.

Prototyping methodology is a model that utilises working prototypes for the users to interact. In the prototyping methodology, the phases of analysis, design, and implementation are performed repeatedly until a completed system is produced. As defined by Dennis, et al. (2015), a system prototype is a quick-and-dirty system that can visualise and express the components and features of the actual system.

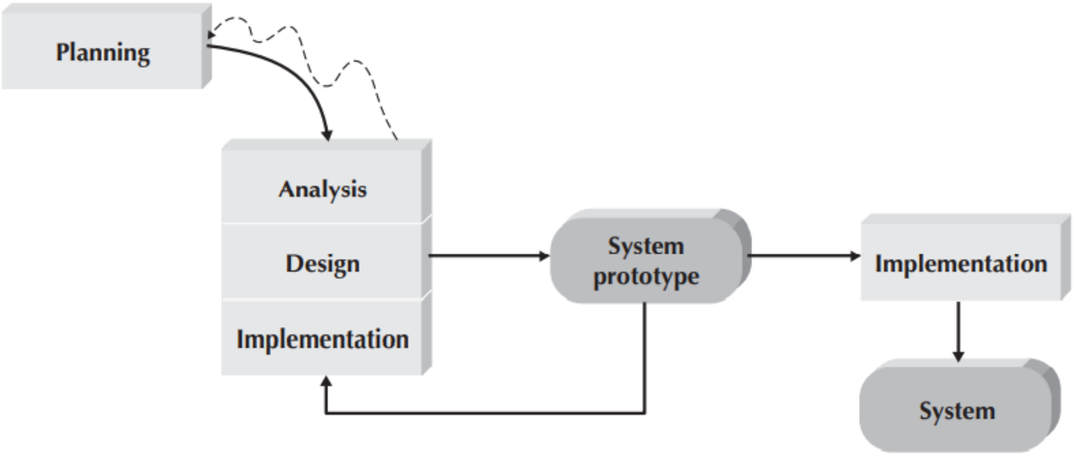


Figure 2.53: Prototyping Development Methodology (Dennis, et al., 2015)

After the system prototype is developed, it will be sent and reviewed by the system users to evaluate and check if the requirements are met. Once the system is reviewed and commented on by the users, the feedbacks obtained from the users will be used to reanalyse, redesign and reimplement a new version of the prototype. The new version of the prototype might then contain more features than the previous one and be sent to the users once it is completed. This process repeats until the approval from the stakeholders to deploy the system (Dennis, et al., 2015; Gajalakshmi, 2016).

Table 2.10 shows the findings from different research on the advantages and disadvantages of the prototyping methodology.

Table 2.10: Advantages and Disadvantages of Prototyping Methodology (Dennis, et al., 2015; Gajalakshmi, 2016)

Advantages	Disadvantages
• High involvement of stakeholders during the development process such as reviewing system prototype • Easily identify any ambiguous or conflicting requirements • Can have a working prototype to interact with the users at an early stage • Users can be reassured of the progress of the system development	• It might waste time and effort if the stakeholders reject the system prototype. • Might cause problems in the development of complex systems due to the unrecognised fundamental issues and problems • The schedule or flow might be messed up when there are too many changes in the system prototype.

2.5.4 Agile Methodology

Agile methodology is a modern software development methodology that is widely used by developers nowadays. Dennis, et al. (2015) stated that the agile methodology emphasises the agile manifesto, which signifies the conditions of working software, developers, customers, and resolving changing requirements rather than focusing on the details of the software development processes such as documentation, detailed plans, and legal contracts. Agile methodology is also based on the incremental and iterative software development model. Therefore, the system will be divided into a series of iterations or cycles, in which a working prototype of the system is delivered each cycle.

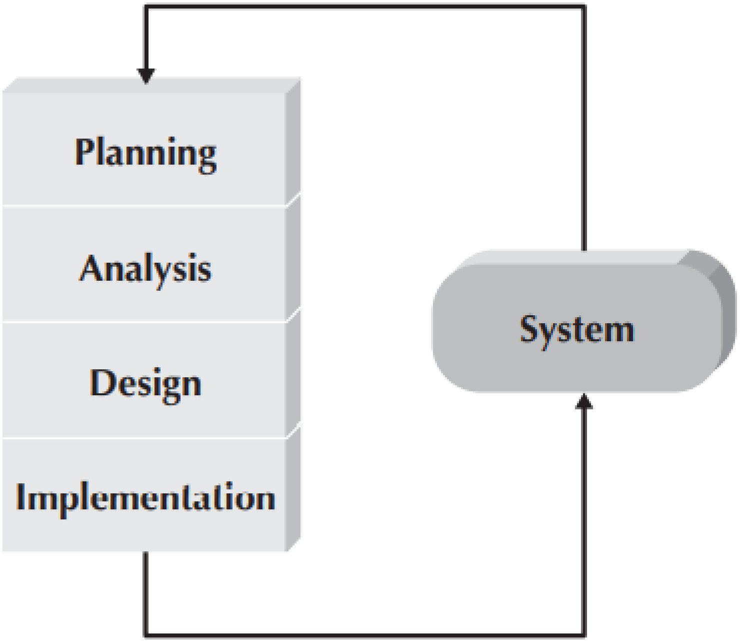


Figure 2.54: Agile Development Methodology (Dennis, et al., 2015)

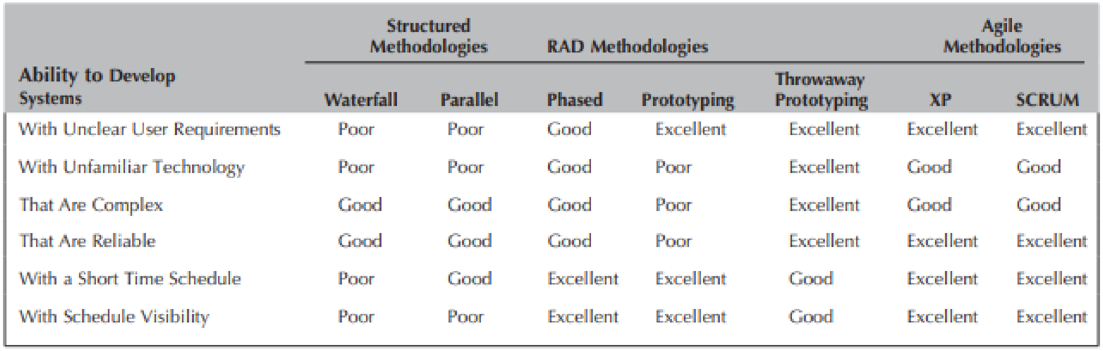
The essential aspects of a successful project are excellent communication, analysis, and flexibility in the agile development methodology. One of the highlights of the agile methodology is that it emphasises the relationship and collaboration between the development team, testers, and stakeholders (Barjtya, et al., 2017; Saqqa, et al., 2020). The priority in agile methodology is customer satisfaction. Therefore, good communication among the development team and the stakeholders is crucial. There are many agile development models, and the most prevalent agile models are Extreme Programming (XP), Feature Driven Development, and Scrum. Table 2.11 shows the advantages and disadvantages of the agile methodology compiled from different sources.

Table 2.11: Advantages and Disadvantages of Agile Development Methodology (Barjtya, et al., 2017; Malleswari, et al., 2018; Munassar & Govardhan, 2010; Tarika, 2019)

Advantages	Disadvantages
• Highly collaborative with the stakeholders • Works well with the project in which the requirements keep evolving and changing • Quick-release of the software product • Frequent customer feedback	• Challenging to scale up to large projects due to lack of documentation • An experienced, professional, and client-oriented development team is required • Conflicts might occur on the system architecture when new requirements are added

2.5.5 Conclusion

There are loads of software development methodologies available for the developers to select. The developers should be cautious and considerate when choosing the right methodology to be employed in the project. Dennis, et al. (2015) have proposed some criteria for selecting an appropriate methodology for software development, as shown in Figure 2.55. With the suggestion from Dennis, et al. (2015), we can have more knowledge on selecting software development methodology for our projects.



Ability to Develop Systems	Structured Methodologies		RAD Methodologies			Agile Methodologies	
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP	SCRUM
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Good	Good
That Are Complex	Good	Good	Good	Poor	Excellent	Good	Good
That Are Reliable	Good	Good	Good	Poor	Excellent	Excellent	Excellent
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Excellent	Excellent

Figure 2.55: Criteria for Determining a Methodology (Dennis, et al., 2015)

Some software methodologies implemented by the existing inventory management systems have been reviewed to understand better the methodologies employed by the previous inventory management system projects. Amron, et al. (2018), Choudhary and Gupta (2014), as well as Ilias, et al. (2018) reveal that they used Waterfall methodology to develop their inventory management systems mainly because the concept of the Waterfall methodology is simple to explain to the users. Besides, it has a well-defined and structured approach for each phase and helps plan and schedule the project.

Anulika, et al. (2020) and Oluwole (2019) implemented their inventory management systems using the V-model methodology. They clarified that the V-model methodology is simple and easy to use. Like Waterfall methodology, it works well for small projects where requirements can be easily understood and the advantage of early advancement of test plans.

Bales, et al. (2017), Cruz, et al. (2019) and Lirios, et al. (2018) apply agile methodology to develop their inventory management systems due to the reasons of frequent changing of business requirements, high involvement of customer feedback, and client-oriented approach.

Lastly, Bermudez and Cantos (2012), Irene (2011), Sutanto (2019) as well as Tapado and Delluza (2016) develop their inventory management systems by adopting prototyping methodology. They chose the prototyping methodology because they can frequently communicate with the client to understand the system requirements. Besides, the creation of the prototypes can be a component of obtaining feedback and specifications of the system to improve the software system.

From my standpoint, the waterfall methodology is not suitable for this project. As the waterfall methodology requires a clear understanding of the requirements and cannot adapt to the changing requirements, it is risky to be applied in this project. Besides, I think the agile methodology is also not an appropriate methodology for this project. Since the agile methodology is more team-oriented, it does not suit this individual project. The V-model methodology, which provides testing activities during the software development process, seems reasonable to be employed. However, it might become less flexible when changes occur unexpectedly, where the requiring modifying the test and requirement documents is a pain. After serious consideration, I prefer the RAD prototyping methodology for my project. On the grounds that the prototyping methodology involves the building and reviewing of the working prototype, I can quickly acknowledge and rectify any mistakes or misunderstandings with the stakeholders. Besides, having different iterations allows me to have a clear picture of the objectives that should be achieved in each iteration, thus increasing the project’s efficiency.

2.6 Summary

To sum up, this chapter has gone through the concept of the inventory management system. It was found that there are two types of inventory management systems: manual inventory management system and computerised inventory management system. Besides, four existing applications related to the project, Lazada, senangKIRA, Shopify and WeNiaga, were reviewed, and the common features of the e-commerce and inventory management system were found. Next, there were four inventory analysis techniques studied. The four inventory analysis techniques include ABC analysis, HML analysis, SS analysis and EOQ analysis.

Furthermore, database transaction management and concurrency control was discussed. Different types of concurrency control such as lock-based concurrency control, time-based concurrency control and optimistic concurrency control provide a deep insight into the concurrency problem in the database. Lastly, four types of software development methodologies, including waterfall, V-model, prototyping and agile methodologies, are evaluated and compared to select the most suitable methodology for this project.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This chapter will discuss the software development methodology chosen with detailed phases. Besides, the project plan, including the work breakdown structure and Gantt chart, will be produced. Lastly, this chapter will also discuss the software development tools.

3.2 System Development Methodology

The software development methodology used in this project is the RAD prototyping methodology. The main motive for choosing the prototyping methodology for the project development is the adoption of the iterative system prototype, which allows the client to provide constructive feedback based on the working prototype perpetually until the prototype is accepted as the final system. In this methodology, the system prototype evolves and encompasses more functionalities progressively through each iteration.

The RAD prototyping methodology consists of four major phases: planning, analysis, design, and implementation. Additionally, the analysis, design, and implementation phases will be carried out concurrently to produce one workable system prototype for review purposes. Once the system prototype is accepted as the system to be deployed, the project will enter the implementation phase for testing and finally produce the final system, as illustrated in Figure 3.1.

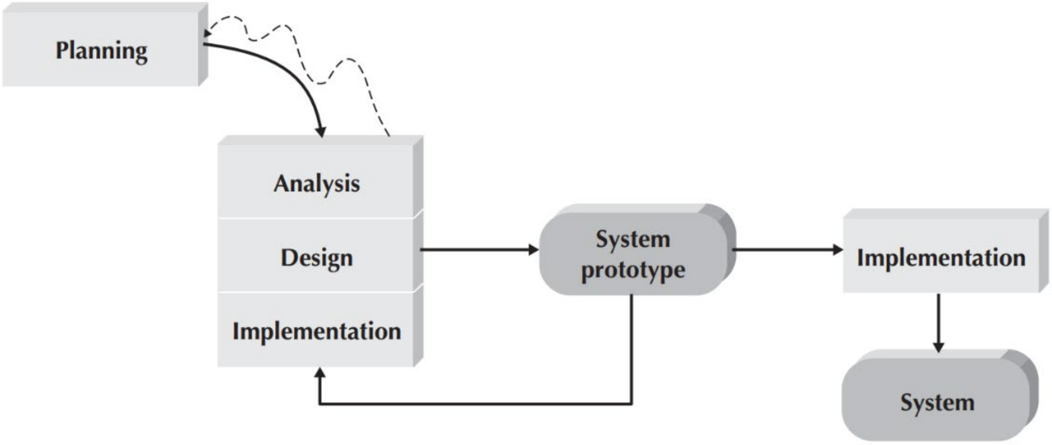


Figure 3.1: RAD Prototyping Methodology (Dennis, et al., 2015)

Since the RAD prototyping methodology proposed by Dennis, et al. (2015) is used for general software development that suits most of the software projects, the prototyping methodology is further modified to complement the project specifically. Through modification, the project further divided the prototyping methodology into four main phases to show the detailed workflow of the project. The four primary phases include planning, iterative prototype development, system testing, and system deployment. Moreover, a brief analysis is performed during the planning phase. Each iteration in the iterative prototyping development phase is added with a testing phase for some early testing activities and an evaluation phase for evaluation and review purposes. In addition, the analysis, design, implementation and testing phases will undergo concurrently in the iterative prototype development phase to produce the system prototype for evaluation. Figure 3.2 shows the modified prototyping methodology for this project.

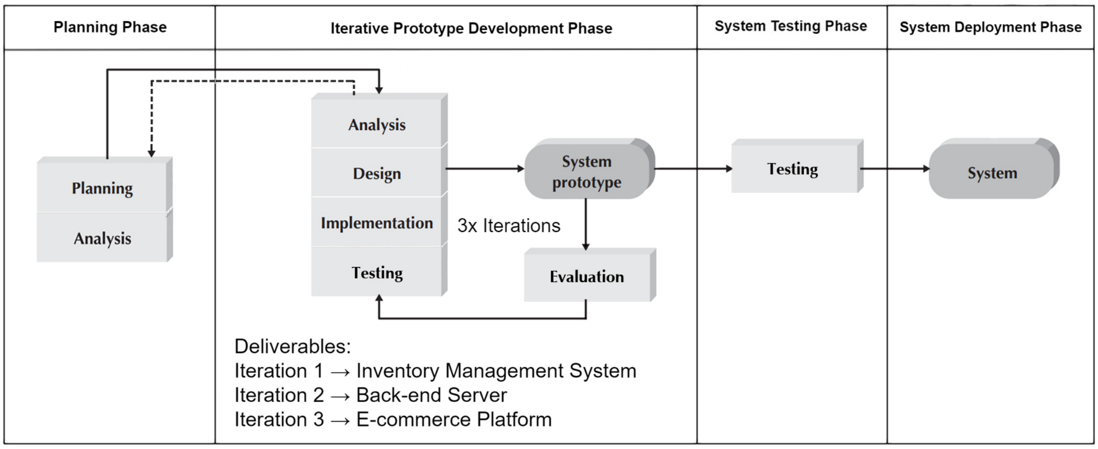


Figure 3.2: Modified Prototyping Methodology

3.2.1 Planning Phase

The first phase of this project is the planning phase. It is the most crucial phase in project development. It helps to identify the project goals, avoid missing project deadlines, reduce the project risks, and eventually produce a satisfying system. This phase focuses on requirements gathering to understand the business requirements from the users, the expectation of the users, and the functionalities required to develop the system.

During the planning phase, the problem encountered by the user was identified. After that, the feasibility study was carried out to determine the project background, goal, objectives, proposed solution, proposed approach, scope, and problem statement. With the deliverables stated, a preliminary report of the project was formulated.

Extensive literature studies and reviews were conducted on the project domain, such as inventory management systems, inventory analysis techniques, database transaction management, and concurrency control. Several existing applications related to the project were reviewed and compared to recognize the features in common so that the common features could be integrated into the system to produce a high-quality and reliable system.

The project plans such as Work Breakdown Structure (WBS) and Gantt chart were developed to create milestones for the project deliverables to be completed in time. The development and testing tools used in this project were also determined. Next, an analysis of the project requirements was performed. The project specifications, such as functional and non-functional requirements, were specified in the analysis phase, with the requirements elicited from the previous activities. Along with the requirements specified, the use case modelling was produced. The use case diagrams and descriptions were developed for each function that will be developed in the project. Lastly, a simple screen prototype was created to provide a basic understanding of the system.

3.2.2 Prototype Development Phase

The project will undergo analysis, design, and implementation phases to produce one workable prototype for evaluation in this prototype development phase. Once the review is completed, the project continues to proceed to the next iteration and perform the same process until the prototype is fully satisfied with all the required functionalities. Throughout this phase, the static analysis tools are used to perform static analysis on the code. This project would have three iterations of the prototype development phase.

a) Iteration 1: Inventory Management System Development

The scope of the first iteration is to produce the entire front-end web application of the inventory management system that can demonstrate the UI and the use case flows for the system functionalities. At the beginning of the development phase, developing a front-end web application helps better understand the design of the back-end architecture. In this iteration, the UI of the web application is designed. The primary purpose of iteration 1 is to check if the functionalities are following the requirements specified. Since the back-end server has not yet been developed, the front-end web application developed in this iteration would work without connecting to the back-end server. After the web application is completed, several testings, such as UI testing, will be performed to ensure that the prototype is working without problems. Then, it will then be shown to the project supervisor for evaluation to check if the prototype satisfies the requirements. The comments and feedback retrieved from the project supervisor will then be used for refinement. The project supervisor will recheck the refined prototype, and this process repeats until the prototype is approved.

b) Iteration 2: Back-end Server Development

The scope of the second iteration is to produce the back-end server with REST API that can connect to the front-end web application developed in iteration 1. This iteration will take the longest to complete since the back-end development involves plenty of logic and might have many bugs to resolve. The system design, such as the software architecture design, entity-relationship diagram, activity diagram, etc., will be produced during this iteration. The database design and API connections will be implemented.

Moreover, all the functionalities and modules will also be implemented in the prototype. Advanced features such as inventory analysis techniques as well as database transaction management and concurrency control will be included in the prototype as well.

After each module is developed, the unit testing and integration testing will be performed to verify the bugs or defects found in the modules as earlier as possible. In addition to this, the test cases of the testing will be developed before the testing is performed. Since the modules developed in the prototype will be reused on the final system, early testing could ensure that the modules are error-free.

By the end of the iteration, a fully functional full-stack web application will be produced. Once the back-end server is completed, it will then be presented to the project supervisor for evaluation to check if the prototype satisfies the requirements. The comments and feedback retrieved from the project supervisor will then be used for refinement. The project supervisor will recheck the refined prototype, and this process repeats until the prototype is approved.

c) Iteration 3: E-commerce Platform Development

The scope of this iteration is to produce a e-commerce application that provides a mobile platform for the users to place their orders with Sharifah Food. During the iteration, the UI design of the web and mobile application will be developed by using cross-platform mobile development tools. After the UI is designed and implemented, the platform will be connected with the back-end server to access and retrieve the data from the server. After the application is completed, several testings, such as UI testing, will be performed to ensure that the mobile application is working. Then, it will then be sent to the project supervisor for evaluation to check if the prototype satisfies the requirements. The comments and feedback retrieved from the project supervisor will then be used for refinement. The project supervisor will recheck the refined prototype, and this process repeats until the prototype is approved.

3.2.3 System Testing Phase

In this phase, the system should be fully functional and finalised. Different types of testing will be performed on the final system, the web, and mobile applications. Before performing the testing, the test plan will be produced as documentation for the testing activities. In addition, the application will undergo four types of testing: unit testing, integration testing, system usability testing, and user acceptance testing. Unit testing and integration testinare used to verify the system works properly. System usability testing evalualates the usability of the system. On the other hand, user acceptance testing verifies if the user requirements are correctly implemented. If there is a 95 percent pass rate of all the tests performed, the project will enter the system deployment phase.

3.2.4 System Deployment Phase

The final review will be carried out with the stakeholders to confirm the final system before delivery in the deployment phase. Besides, the final system and report will be produced and deployed to the internet, marking the completion of the project.

3.3 Project Plan

The project plan is created to provide a detailed schedule of the project. It helps to ensure that the project can be completed within the time-bound adhered to the planned schedule. This project uses the Work Breakdown Structure (WBS) and Gantt Chart for project schedule planning.

3.3.1 Work Breakdown Structure (WBS)

The WBS is a deliverable-oriented deconstruction of the project. It involves a decomposition process, where the project is subdivided into smaller deliverables to become more manageable and measurable. The WBS approach used in this project is the top-down approach, in which the largest portion of the project is broken and subdivided into smaller pieces.

3.3.1.1 WBS List

0.0 Sharifah Food Inventory Management System

1.0 Planning

- 1.1 Preliminary Planning
  - 1.1.1 Background Study
  - 1.1.2 Define Problem Statement
  - 1.1.3 Define Project Goal and Objectives
  - 1.1.4 Define Project Scope
  - 1.1.5 Define Proposed Solution
  - 1.1.6 Define Proposed Approach
- 1.2 Topic Research and Literature Review
  - 1.2.1 Research on Inventory Management System
  - 1.2.2 Research on Inventory Analysis Techniques
  - 1.2.3 Research on Database Transaction Management
  - 1.2.4 Research on Software Development Methodologies
- 1.3 Methodology and Work Planning
  - 1.3.1 Select Methodology
  - 1.3.2 Develop Work Breakdown Structure
  - 1.3.3 Develop Gantt Chart
  - 1.3.4 Select Development Tools
  - 1.3.5 Select Testing Tools
- 1.4 Project Specification
  - 1.4.1 Create Requirement Specifications
    - 1.4.1.1 Functional Requirements
    - 1.4.1.2 Non-functional Requirements
  - 1.4.2 Develop Use Case Model
    - 1.4.2.1 Develop Use Case Diagram
    - 1.4.2.2 Develop Use Case Descriptions
  - 1.4.3 Develop Simple Screen Prototype

2.0 Prototype Development Phase

- 2.1 Iteration 1
  - 2.1.1 Analyse Web Application Requirements
  - 2.1.2 Design Web Application UI
  - 2.1.3 Implement Front-end Web Application

2.1.4 Test Front-end Web Application

2.1.5 Evaluation and Review on Front-end Web Application

2.2 Iteration 2

2.2.1 Analyse Back-end Server Requirements

2.2.2 Design

2.2.2.1 Design System Architecture

2.2.2.2 Design Activity Diagrams

2.2.2.3 Design Data Dictionary

2.2.2.4 Design Entity-Relationship Diagram

2.2.3 Implementation

2.2.3.1 Implement Back-end Server

2.2.3.2 Connect Back-end Server to Web Application

2.2.4 Testing

2.2.4.1 Unit Testing

2.2.4.2 Integration Testing

2.2.5 Evaluation and Review on Back-end Server

2.3 Iteration 3

2.3.1 Analyse Mobile Application Requirements

2.3.2 Design Mobile Application UI

2.3.3 Implementation

2.3.3.1 Implement Mobile Application

2.3.3.2 Connect Back-end Server to Mobile Application

2.3.4 Test Mobile Application

2.3.5 Evaluation and Review on Mobile Application

3.0 System Testing Phase

3.1 Develop Test Plan

3.2 Perform Unit Testing

3.3 Perform Integration Testing

3.4 Perform System Usability Testing

3.5 Perform User Acceptance Testing

4.0 System Deployment Phase

4.1 Final Review

4.2 Prepare Final Report

4.3 Completion of Project

3.3.1.2 WBS Diagram

WBS diagram of the project is attached as **Appendix A**.

3.3.2 Gantt Chart

Gantt Chart is a bar chart used to illustrate the project schedule plan. The Gantt Chart further supports the WBS by graphically showing each project activity's start and end times. The Gantt Chart of the project is attached as **Appendix B**.

3.4 Development Tools

The system to be developed will cover both web and mobile applications. Therefore, both web and mobile development tools will be used in this project. This section will discuss different development tools used in this project.

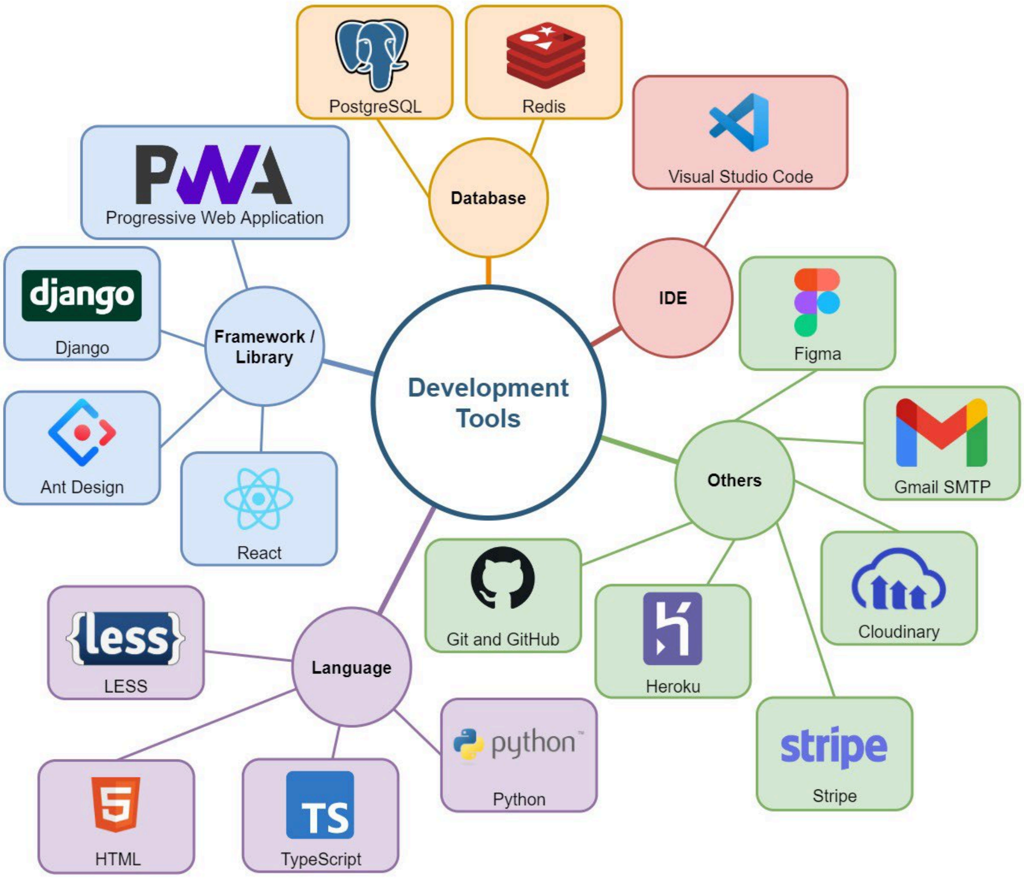


Figure 3.3: Development Tools

3.4.1 Language

a. LESS

LESS, also known as the Leaner Style Sheet, is a stylesheet language extending from CSS that is used specifically for web application development. It is used to style the HTML elements so that the HTML elements can be displayed in different ways. In this project, LESS is used with React and Ant Design to style the web application.

b. HTML

HTML, also known as HyperText Markup Language. It is the standard markup language used in web application development. Nowadays, almost every web application uses HTML to develop web pages. There are many elements, like heading, body, table, link, paragraph, list, image, etc. All of the elements in HTML are enclosed by the start tag (<) and the end tag (>). These elements help to describe the structure of a web page. In this project, HTML is rendered using React to develop the web application.

c. TypeScript

TypeScript, also abbreviated as TS, is a strongly typed programming language extending JavaScript. It is known as the scripting language for web application development. Besides, it is mainly used to develop an interactive web applications. In this project, TypeScript is used by React to add interactive behaviour to the web application.

d. Python

Python is an interpreted high-level programming language. Due to its simplicity, it is one of the most widely used programming languages in the world today. Python is also well known for data science, as it provides many helpful libraries for data science that allow developers to perform data analysis efficiently. In this project, version 3.9 of Python is used to support the back-end server of the system.

3.4.2 Framework / Library

a. Ant Design

Ant Design is an open-source React UI framework. It is one of the most popular frameworks for developing responsive and mobile-friendly web applications. Ant Design provides pre-built LESS components that developers can implement without wasting much time defining the components themselves. It provides various components like buttons, dropdowns, navigation bars, thumbnails, alerts, etc. In this project, Ant Design is used along with React to design the web application using the components available in Ant Design to speed up the web application development process.

b. Django

Django is a Python web framework that allows developers to quickly and efficiently develop a high-quality web application. It follows the Model-Template-Views (MTV) architectural pattern. By using the MTV architectural pattern, it supports fast and parallel development. Django includes a powerful toolkit for building web APIs, the REST framework. The REST framework, also known as Representational State Transfer, is an integration framework in Django that serves as a web API for handling HTTP requests. In this project, the Django framework is used on the back-end server. The web and mobile applications are connected to the back-end server using the Django REST API.

c. React

React is a declarative JavaScript framework used to create interactive UI for web and mobile applications. It is a component-based framework used to manage the application's view layer in the Model-View-Controller (MVC) architecture. The paradigm of React is "Learn Once, Write Anywhere" which emphasizes the development of a single codebase when developing cross-platform applications. In this project, React is used for building the web application.

d. Progressive Web Application (PWA)

PWA is an application built using the web technologies but can deliver a native-like application. The popularity of PWA has risen over time due to its speedy experiences. By using PWA, the users can add the website to their mobile application with just one click. Besides, PWA also provides an offline experience if it is configured to have proper caching. In this project, PWA is used to develop the e-commerce platform so that the platform is available to Sharifah Food customers on mobile and the web.

3.4.3 Database

a. PostgreSQL

PostgreSQL is an open-source Relational Database Management System (RDBMS). As the name implies, it uses an SQL database model to store data. This project will use PostgreSQL as the database to keep all the data information of the system. The main reason for using the SQL database model instead of the NoSQL database model is that the SQL database model is much more reliable in transaction processing and concurrency control to ensure data consistency. Data consistency is the primary concern in this system because the inventory management system should store the inventory accurately. Besides, the Django framework also indicates that it has the best integration with PostgreSQL, and Heroku provides free 1 GB with 10000 rows limitation software deployment with its PostgreSQL. Therefore, PostgreSQL has been selected for this project which uses the SQL database model. Moreover, the studied transaction management and concurrency control techniques are applied to the database.

b. Redis

Redis is an open-source in-memory data structure store. It is mainly used as in-memory storage. In this project, Redis will be used as the cache storage for both the inventory management system and e-commerce platform, as to improve the performance of the server.

3.4.4 Integrated Development Environment

a. Visual Studio Code

Visual Studio Code is a lightweight, open-source source code editor developed by Microsoft. With its various available extensions, it is highly customizable and capable of covering a wide range of programming languages such as Java, C++, C#, Python, PHP, JavaScript, Go, Django, etc. Visual Studio Code has many features like IntelliSense, code refactoring, debugging, version control, working with GitHub, settings synchronization, integrated terminal, code snippets, Emmet support, integrated command-line interface (CLI), etc. In this project, Visual Studio Code is used to write the code needed to develop the system.

3.4.5 Other Development Tools

a. Figma

Figma is both a vector graphics editor and a prototyping tool. It helps developers to develop low and high fidelity prototypes efficiently. By using Figma, applications can have better UI and UX designs. In this project, Figma is used to create low and high fidelity prototypes.

b. Git and GitHub

GitHub is a remote repository hosting service for version control in software development. It is a web server for software version control based on Git, a distributed version control system. In this project, GitHub is used to control the system. Since the project uses a prototyping methodology, different versions of the prototype can be stored on GitHub. If the latest version of the prototype contains errors, it can be reverted to the previous version.

c. Gmail SMTP

Gmail SMTP is a free Simple Mail Transfer Protocol (SMTP) server provided by Google. It can be used to send emails from own Gmail account to another. This project uses Gmail SMTP to send email notifications to the customers, such as the order and customer position updates notifications.

d. Cloudinary

Cloudinary is a company that provides cloud-based image and video management services. The system contains a lot of features that facilitate image management. It allows the user to modify the image without image processing software, the user can pass in the query parameters in the URL, and the system will do its job. This project uses Cloudinary to manage the images of the Sharifah Food goods.

e. Stripe

Stripe is a financial service system that provides an online payment processing service. It allows the users to accept payments from plenty of payment methods. This project uses Stripe as the payment gateway on the e-commerce platform to accept payment from the customers who place orders. Please note that the service provided by Stripe is commercial. Therefore, the payment can only be performed in test mode.

f. Heroku

Heroku is a Platform as a Service (PaaS) that offers developers the to build, run and operate their applications in the cloud. It supports several programming languages, including Python. Besides, it also provides add-ons so that the developers can include different third-party integrations to their applications. This project uses Heroku to deploy the inventory management system and e-commerce platform, as Heroku offers its free integration with PostgreSQL and Redis.

3.5 Testing Tools

The system to be developed will include both web and mobile applications. Therefore, both web and mobile testing tools will be used in this project. This section discusses the various testing tools used in this project.

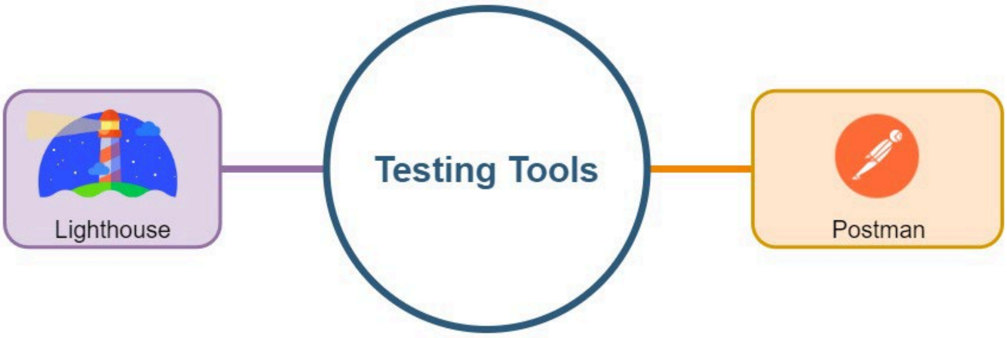


Figure 3.4: Testing Tools

a. Lighthouse

Lighthouse is an automated tool for improving the quality of web pages. Lighthouse helps to test the performance of the web application, the ability to be loaded in offline mode, validation of the aspects of the PWA, etc. In this project, Lighthouse is used to analyse the PWA being developed.

b. Postman

Postman is an API platform that is mainly used for API Testing. It allows the users to send the request to their API and retrieve the response with its graphical user interface. In this project, Postman is used to test the REST API developed in the backend server.

3.6 Summary

In summary, this chapter discussed the software methodology adopted in the project, namely the prototyping method. Furthermore, the prototyping methodology has been modified to fit the project well. Each phase of the revised prototyping methodology has been discussed to give a deep insight into the project. WBS and Gantt Chart were also created to plan the project activities. This will ensure that the project is in line with the planned schedule so that the project can be completed before the given deadline.

In addition, various development tools such as programming language, framework or library, database, IDE and other development tools used in system development have been listed and discussed. Each development tool has its benefits for the development of the system. Finally, the testing tools used for testing the applications were listed and discussed.

CHAPTER 4

PROJECT SPECIFICATION

4.1 Introduction

This chapter will discuss the requirement discovery to show how the requirements are derived. Besides, the functional and non-functional requirements that build up the system will be listed. Lastly, a use case model will be produced with the use case diagram and respective use case descriptions.

4.2 Requirement Discovery

There are two methods performed in the project to elicit the requirements: identification and understanding of the problem stated by the user as well as the review on the existing similar inventory management system. Before the initiation of the project, the problem encounter by Sharifah Food was discussed through the meeting. From the clarification, the problems faced by the company are well understood. Moreover, a total of four existing similar applications, namely Lazada, senangKIRA, Shopify, and WeNiaga, were reviewed and evaluated in terms of their functionalities on both e-commerce platforms and inventory management systems.

After reviewing the four existing computerised inventory management systems, it can be summarized that the common features in the e-commerce platform are browsing the products and promotions, viewing the product, adding products to the shopping cart, checkout, payment gateway, and notification on the order status. On the other hand, the common features in the inventory management systems reviewed are registration and login, product management, stock management, order management, business insights, invoice, sales, and inventory reports generator, as well as notification such as the new order placement and low stock warning notifications. Moreover, it was found that Sharifah Food employs a MLM system to promote its products, in which it recruits agents and dropshippers to stimulate its product sales. In addition, it was discovered that the agents and dropshippers have different discounts on the product price, which gives rise to a new requirement on the system. Therefore, additional features such as registration as agents or dropshippers, discounts management as well as agent and dropshipper management will be included in the system.

4.3 Requirement Specification

The requirement specification is a collection of the requirements to be imposed to implement the system. The requirements refer to the descriptions of what the system should do. It is vital to produce precise requirements as the requirements reflect the need of the users for a system. In this section, the functional requirements and non-functional requirements of the system will be listed.

4.3.1 Functional Requirements

Functional requirements refer to the statement of the services that a system should offer and how a system reacts to particular inputs. The functional requirements are essential in defining the basic system behaviour. Moreover, the system features can be easily derived from the functional requirements. The functional requirements of this system will be divided into two parts: the user side and the admin side.

4.3.1.1 User (Direct Customer, Agent and Dropshipper)

- **Register as Customer**
    a) The system shall allow the user to sign up as a customer by email.
    b) The system shall require the user to enter information such as email and password to register as a customer by email.

- **Register as Agent or Dropshipper**
    a) The system shall allow the user to register as an agent or dropshipper by filling the form containing information such as name, email, phone number, gender, address, occupation, etc.
- **Login**
    a) The system shall allow the user to log in to the e-commerce applications using a registered account.
    b) The system shall send a reset password email to the registered email when the user forgets the password.

- **Profile Management**
  a) The system shall allow the user to modify the user profile by changing the name, password, birthdate, phone number, email, etc.
  b) The system shall allow the user to add multiple shipping addresses.
  c) The system shall prompt the user to enter the password when the user requests to change the password.
  d) The system shall send a verification email to the registered email if the user requests to change the email.

- **Browse Products and Promotions**
  a) The system shall be able to display all the products and promotions.
  b) The system shall allow the user to filter the products and promotions based on different attributes such as the categories, price, popularity, rating, etc.
  c) The system shall allow the user to click on a particular item and redirect to the item page.

- **View Item**
  a) The system shall display the item details such as name, category, description, price, available stocks, etc., to the user when the user selects the item.

- **Add to Shopping Cart**
  a) The system shall allow the registered user to add and store items to the registered user's shopping cart in the database.
  b) The system shall allow the unregistered user to add items to the shopping cart and store them with the in the user's local website storage.
  c) The system shall not allow the users to checkout their shopping cart containing out of stock items.
  d) The system shall display out of stock message to there are out of stock items in the shopping cart.

- **Search Items**
  a) The system shall allow the user to search for specific products or promotions by entering the keywords.
  b) The system shall display a “No Record Found” message when no products or promotions match the search keywords.

- **Order History**
  a) The system shall allow the unregistered users to search their order by entering their email and order number.
  b) The system shall display a list of order history to the registered users.
  c) The system shall allow the user to view the order details containing the order information, order status, order summary, and payment method.
  d) The system shall allow the user to pay for unpaid orders.

- **Checkout**
  a) The system shall allow the user to check out available items on the item detail page or shopping cart.
  b) The system shall display an order summary, including the total price and shipping fee of the order.
  c) The system shall allow the user to choose between pickup or shipment for their order.
  d) The system shall allow the user to select a delivery address saved in the user account or enter a new delivery address.
  e) The system shall allow the user to choose one payment method.
  f) The system shall allow the logged in user to input a discount code for a discount.
  g) The system shall be able to apply a discount code for the order of agent or dropshipper based on their position.

- **Payment**
  a) The system shall be able to redirect the user to the respective payment gateway based on the payment method selected by the user.
  b) The system shall display order status messages such as “Payment Success” or “Payment Failed” once the user is redirected from the payment gateway.
  c) The system shall send an order confirmation email to the user’s email address with the order details once the user has paid the order.

- **Notification**
  a) The system shall notify the user of the order updates through email.

4.3.1.2 Admin (Stockist)

- **Login**
  a) The system shall allow the admin to log in to the inventory management system using a registered account.
  b) The system shall send a verification email to the registered email when the admin forgets the password.

- **Profile Management**
  a) The system shall allow the admin to modify their profile, such as changing the name, password, phone number, email, etc.
  b) The system shall prompt the admin to enter the password when the admin requests to change the password.
  c) The system shall verify the admin password when the admin changes profile information.

- **Product Stock Management**
  a) The system shall allow the admin to add new products by entering the product details such as name, category, price, stock quantity, images, etc.
  b) The system shall display a list containing all products to the admin.
  c) The system shall allow the admin to edit or delete the existing products.
  d) The system shall allow the admin to filter and sort the products based on the different attributes such as stock quantity, price, etc.

- **Promotional Packages Management**
  a) The system shall allow the admin to add new promotional packages by combining different products into one item with a price.
  b) The system shall display a list containing all promotional packages to the admin.
  c) The system shall allow the user to edit or delete existing promotional packages.
  d) The system shall allow the admin to filter and sort the promotional packages based on the different attributes such as stock quantity, price, etc.

- **Order Management**
  a) The system shall record all the orders placed by the users on the e-commerce platform.
  b) The system shall display a list of all orders to the admin.
  c) The system shall allow the admin to add a new order manually, in which the process is similar to the ordering in the e-commerce platform.
  d) The system shall allow the admin to filter and sort the orders based on the different attributes such as order date, price, status, customer type, etc.
  e) The system shall allow the admin to generate the invoices for paid orders.

- **Agent and Dropshipper Management**
  - a) The system shall allow the admin to add new agents and dropshippers by entering their information such as name, email, phone number, gender, address, occupation, etc.
  - b) The system shall display a list of all customers to the admin.
  - c) The system shall display the total order value of each agent and dropshipper per month to the admin.
  - d) The system shall allow the admin to view the agent and dropshipper registrations.
  - e) The system shall allow the admin to accept or reject the agent and dropshipper registrations.
  - f) The system shall inform the applicant about the agent and dropshipper registration status through email after the admin accepts or rejects the registration.

- **Shipping Fee Management**
  - a) The system shall allow the admin to add, edit and delete the shipping fees.
  - b) The system shall display a list of all shipping fees to the admin.
  - c) The system shall allow the admin to customize the weight range and the respective shipping fee for each state.

- **Pickup Location Management**
  - **a)** The system shall allow the admin to add, edit and delete the pickup location.
  - **b)** The system shall display a list of all pickup locations to the admin.

- **Business Insights**
  a) The system shall display the business insights on the dashboard of the inventory management system.
  b) The system shall compute and display the sales and inventory analyses, such as the total sales over time, total orders over time, and sales by-products, in a graphical representation.
  c) The system shall perform inventory analysis techniques such as ABC, HML, SS, and EOQ analyses and display the analysis results to the admin.

- **Invoice, Sales, and Inventory Reports Generator**
  a) The system shall be able to generate an invoice for each order in PDF format.
  b) The system shall be able to generate sales and inventory reports in Excel format.

- **Discount Management**
  a) The system shall allow the admin to add new discount codes applied at checkout on the e-commerce platform.
  b) The system shall allow the admin to edit or delete existing discount codes.
  c) The system shall allow the admin to specify the criteria for applying the discount codes, such as minimum purchase amount, a minimum quantity of items, customer eligibility, etc.

- **Notification**
  b) The system shall notify the admin of the new order placement and low-stock products in the inventory management system.

4.3.2 Non-functional Requirements

Non-functional requirements refer to the restrictions or constraints imposed on the system. It is a requirement that describes the criteria for the system operation. The non-functional requirements to be discussed include availability, development, performance, security, and usability requirements.

4.3.2.1 Availability Requirements

a) The system shall be available for access every day for 24 hours with a maximum downtime of 15 minutes.

4.3.2.2 Development Requirements

a) The system must follow all phases of the revised prototyping methodology.
b) The e-commerce application must be built-in web and mobile platforms, while the inventory management system must be built in the web application.

4.3.2.3 Performance Requirements

a) The system must be able to respond within 20 seconds.
b) The system must be able to handle concurrent access without failure.

4.3.2.4 Security Requirements

a) The system must be able to encrypt the user's data using the SSL encryption standard.
b) The system shall implement the PBKDF2 algorithm with a SHA256 hash to store the password of the users and admins.
c) The system shall suspend the admin account after five failed login attempts.

4.3.2.5 Usability Requirements

a) The system should display the interfaces according to good design principles.
b) The admin should thoroughly understand the inventory management system after 3 to 5 hours of training.

4.4 Use Case

The use case is a model that describes how the users will carry out the tasks on the applications. This project uses use case modelling to understand how the user and admin interact with the system. In this project, the use case divides the system into two subsystems, which are the e-commerce applications and inventory management application. The primary actor for the e-commerce applications is the user, including direct customer, agent, and dropshipper, while the primary actor for the inventory management application is the admin. This section will cover the use case diagram and respective use case descriptions.

4.4.1 Use Case Diagram

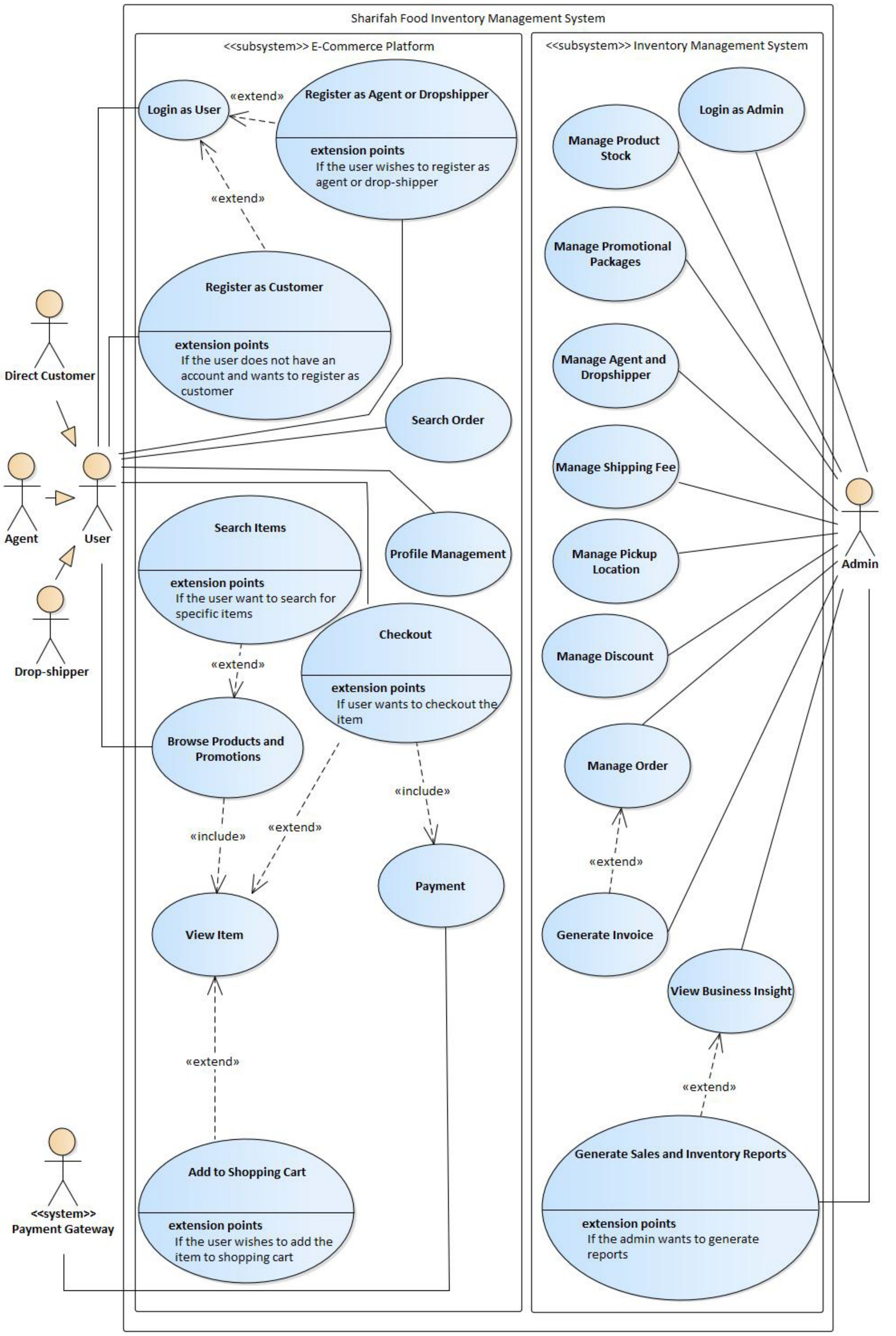

Sharifah Food Inventory Management System
<<subsystem>> E-Commerce Platform
<<subsystem>> Inventory Management System
Login as User
«extend»
Register as Agent or Dropshipper
extension points
If the user wishes to register as agent or drop-shipper
«extend»
Register as Customer
extension points
If the user does not have an account and wants to register as customer
Search Order
Profile Management
Search Items
extension points
If the user want to search for specific items
«extend»
Checkout
extension points
If user wants to checkout the item
Browse Products and Promotions
«include»
«extend»
«include»
Payment
View Item
«extend»
Add to Shopping Cart
extension points
If the user wishes to add the item to shopping cart
Direct Customer
Agent
User
Drop-shipper
<<system>>
Payment Gateway
Login as Admin
Manage Product Stock
Manage Promotional Packages
Manage Agent and Dropshipper
Manage Shipping Fee
Manage Pickup Location
Manage Discount
Manage Order
«extend»
Generate Invoice
View Business Insight
«extend»
Generate Sales and Inventory Reports
extension points
If the admin wants to generate reports
Admin

Figure 4.1: Use Case Diagram

4.4.2 Use Case Descriptions

4.4.2.1 Login as User

Table 4.1: Use Case Description - Login User

Name of the Use Case: Login as User	ID: A1	Importance Level: Moderate
Actor of the Use Case: User	Type of the Use Case: Detail, Essential	
Stakeholders of the Use Case: User: The user who wants to log in to the e-commerce applications		
Brief Description of the Use Case: This use case describes how the user logs in to the e-commerce applications.		
Trigger of the Use Case: The user wishes to log in to their accounts in the e-commerce application.		
Relationships of the Use Case: Association: User Extend: Register as Customer, Register as Agent or Dropshipper Generalization: Direct Customer, Agent, Dropshipper		
Standard Event Flows: 1. The user enters the e-commerce login page. 2. The user enters the registered email and password. If the user forgot the password, perform S1. Forgot Password. 3. The system validates the email and password. 4. The user successfully logged in to the account.		

Table 4.1 (Continued)

SubFlows:

S1. Forgot Password

1. The user enters a registered email.
2. The system sends a validation link to the email.
3. The user clicks on the validation link and redirected to the reset password page.
4. The user enters a new password.
5. The password is successfully changed to the new password.

Alternate/Exceptional Flows:

1a. Do not have an account and wants to register as a customer

1. Perform **Use Case A2: Register as Customer**.

1b. Register as agent and dropshipper

1. The user enters the e-commerce agent and dropshipper registration page.
2. The user fills and submits the form with the user information.
3. The system adds the request to the pending list and displays a message informing the user to wait for confirmation.

3a. Invalid email or password

1. The system displays an “Invalid email or password” message.

1b-2a. Incomplete information

1. The system displays an error message and asks the user to input the incomplete information.
2. The user fills up the incomplete information and submits it again.

4.4.2.2 Register as Customer

Table 4.2: Use Case Description - Register as Customer

<table>
<tr><td>Name of the Use Case:
Register as Customer</td><td colspan="2">ID:
A2</td><td>Importance Level:
Moderate</td></tr>
<tr><td colspan="2">Actor of the Use Case:
User</td><td colspan="2">Type of the Use Case:
Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders of the Use Case:
User: The user who wants to register as a customer for the e-commerce applications</td></tr>
<tr><td colspan="4">Brief Description of the Use Case:
This use case describes how the user registers as a customer for the e-commerce applications.</td></tr>
<tr><td colspan="4">Trigger of the Use Case:
The user wishes to register an account as a customer for the e-commerce applications.</td></tr>
<tr><td colspan="4">Relationships of the Use Case:
Association: User
Generalization: Direct Customer, Agent, Dropshipper</td></tr>
<tr><td colspan="4">Standard Event Flows:
1. The user enters the e-commerce customer registration page.
2. The user enters information such as name, email, password.
3. The system validates the user information.
4. The user is successfully registered as a customer.</td></tr>
<tr><td colspan="4">Alternate/Exceptional Flows:
3a. The email is found registered
The system displays an “Email has been registered” message.</td></tr>
</table>

4.4.2.3 Browse Products and Promotions

Table 4.3: Use Case Description - Browse Products and Promotions

<table>
<tr><td colspan="2">Name of the Use Case:
Browse Products and Promotions</td><td colspan="2">ID:
A3</td><td>Importance Level:
High</td></tr>
<tr><td colspan="3">Actor of the Use Case:
User</td><td colspan="2">Type of the Use Case:
Detail, Essential</td></tr>
<tr><td colspan="5">Stakeholders of the Use Case:
User: The user who wants to browse the catalogue of the products and promotions</td></tr>
<tr><td colspan="5">Brief Description of the Use Case:
This use case describes how the user browses the catalogue of the products and promotions.</td></tr>
<tr><td colspan="5">Trigger of the Use Case:
The user wishes to browse the products and promotions in the e-commerce applications.</td></tr>
<tr><td colspan="5">Relationships of the Use Case:
Association: User
Include: View Item
Extend: Search Items, Add to Shopping Cart, Add to Wishlist, Checkout
Generalization: Direct Customer, Agent, Dropshipper</td></tr>
<tr><td colspan="5">Standard Event Flows:
1. The user enters the products page.
2. The system displays a list of all products and promotions.
3. The user can now browse the products and promotions. If the user wants to search for a specific item, perform 3a. Search Items.
4. If the user wants to view the shopping cart, peform 4a. View Shopping Cart,
5. The user clicks on the product or promotion to look for the detailed information. Perform S1. View Item.</td></tr>
</table>

Table 4.3 (Continued)

SubFlows: S1. View Item 1. The user enters the specific product details page. 2. All the information about the item will be shown. If the user wishes to add the item to the shopping cart, perform S1-3a. Add to Shopping Cart.
Alternate/Exceptional Flows: 3a. Search Items 1. The user enters the keywords of the item in the search bar. 2. The items with the keywords in their names will be displayed to the user. 3a-2a. No items were found for the keywords 1. The system displays a “No Record Found” message. 4a. View Shopping Cart 1. The user clicks the shopping cart icon. 2. The system redirects the user to the shopping cart page and display a list of user’s shopping cart items. 3. If the user wants to check out the items in the shopping cart, perform **Use Case A4: Checkout**. S1-3a. Add to Shopping Cart 1. The user clicks the “Add to Cart” button. 2. The system displays a successful message, and the item is added to the shopping cart.

4.4.2.4 Profile Management

Table 4.4: Use Case Description - Profile Management

Name of the Use Case: Profile Management	ID: A4	Importance Level: Medium
Actor of the Use Case: User	Type of the Use Case: Detail, Essential	
Stakeholders of the Use Case: User: The registered user who wants to view or edit profile, address book, and order history.		
Brief Description of the Use Case: This use case describes how the registered user can view and edit profiles, address books, and order history.		
Trigger of the Use Case: The registered user wishes to view or edit their profile, address book, and order history.		
Relationships of the Use Case: Association: User Generalization: Direct Customer, Agent, Dropshipper		
Standard Event Flows: 1. The registered user clicks the ‘Profile’ menu or user icon. 2. The registered user can either choose to view the account information, address book, or order history. a. If the registered user wants to view or edit the account information, perform S1. Manage account information b. If the registered user wants to view or edit the address book, perform S2. Manage address book. c. If the registered user wants to view the order history, perform S3. View order history.		

Table 4.4 (Continued)

SubFlows:

S1. Manage account information

1. The user clicks on the 'Account Information' tab in the menu, and the system redirects the user to the account information page.
2. The system displays the user's account information.
3. The user clicks on the 'Edit' button.
4. The system displays a modal form to the user.
5. The user edits and submits the form along with the user's password.
6. The system validates the user's password and updates the user's account information.

S2. Manage address book

1. The user clicks on the 'Address Book' tab in the menu, and the system redirects the user to the address book page.
2. The system displays a list of addresses of the users.
3. The user can choose to add, modify or delete the address.
   a. If the user wants to add a new address, perform S2-3a Add new address.
   b. If the user wants to edit an existing address, perform S2-3b Edit existing address.
   c. If the user wants to view an existing address, perform S2-3c View existing address.

S3. View order history

1. The user clicks on the 'Order History' tab in the menu, and the system redirects the user to the order history page.
2. The system displays a list of orders placed by the user.
3. The user clicks on order.
4. The system displays detailed information about the order.

Table 4.4 (Continued)

Alternate/Exceptional Flows:

S2-3a. Add new address

1. The user clicks on the ‘Add address’ button.
2. The system displays a modal form to the user.
3. The user fills and submits the modal form with all the address information.
4. The system validates the input from the form and saves the user's new address.

S2-3b. Edit existing address

1. The user clicks on the ‘Edit’ button on the address card.
2. The system displays a modal form with the existing address information.
3. The user edits the information and submits the form.
4. The system validates the input from the form and edits the selected address.

S2-3c. View existing address

1. The user clicks on the address card in the list.
2. The system redirects the user to the specific order details page.
3. The system displays all the order information to the user.

4.4.2.5 Search Order

Table 4.5: Use Case Description - Search Order

<table>
<tr><td>Name of the Use Case:
Search Order</td><td colspan="2">ID:
A5</td><td>Importance Level:
Medium</td></tr>
<tr><td colspan="2">Actor of the Use Case:
User</td><td colspan="2">Type of the Use Case:
Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders of the Use Case:
User: The user who wants to search a placed order.</td></tr>
<tr><td colspan="4">Brief Description of the Use Case:
This use case describes how the user can search for a placed order.</td></tr>
<tr><td colspan="4">Trigger of the Use Case:
The user wishes to search for a placed order.</td></tr>
<tr><td colspan="4">Relationships of the Use Case:
Association: User
Generalization: Direct Customer, Agent, Dropshipper</td></tr>
<tr><td colspan="4">Standard Event Flows:
1. The user clicks the “Search Order” tab in the menu.
2. The system redirects the user to the search order page.
3. The user enters the email entered when placing the order and the order number received from the order confirmation email.
4. The system redirects the user to the specific order details page.
5. The system displays all the order information to the user.</td></tr>
<tr><td colspan="4">Alternate/Exceptional Flows:
3a. Order not found
1. The system displays an ‘Order not found’ message to the user.</td></tr>
</table>

4.4.2.6 Checkout

Table 4.6: Use Case Description - Checkout

<table>
<tr><td>Name of the Use Case:
Checkout</td><td colspan="2">ID:
A6</td><td>Importance Level:
High</td></tr>
<tr><td colspan="2">Actor of the Use Case:
User</td><td colspan="2">Type of the Use Case:
Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders of the Use Case:
User: The user who wants to check out the items.</td></tr>
<tr><td colspan="4">Brief Description of the Use Case:
This use case describes how the user can place an order by checking out the items.</td></tr>
<tr><td colspan="4">Trigger of the Use Case:
The user wishes to place the order.</td></tr>
<tr><td colspan="4">Relationships of the Use Case:
Association: User, Payment Gateway
Include: Payment
Exclude: View Item
Generalization: Direct Customer, Agent, Dropshipper</td></tr>
<tr><td colspan="4">Standard Event Flows:
1. The user clicks the “Checkout” button in the shopping cart.
2. The system shows the order summary with all the checked item(s).
3. If the user wish to view the checked item, perform 3a. View Item.
4. The user may input a discount code, if the user has logged in. Perform 4a. Input a discount code.
5. The user selects the delivery address or pickup location.
6. The user chooses the payment method.
7. Once the user confirms the order, the user clicks the “Place Order” button. Perform **S1. Payment**.</td></tr>
</table>

Table 4.6 (Continued)

SubFlows: S1. Payment 1. The user is redirected to the payment gateway with respect to the payment method. 2. The user pays the order through the payment gateway. 3. Once the user has paid, the user is redirected back to the application. 4. The system shows a "Payment Successful" message, and an order confirmation email will be sent to the user.
Alternate/Exceptional Flows: 3a. View Item 1. The user clicks on the item name listed in the order summary card. 2. The system redirects the user to the item details page. 3. The information about the item is shown to the user. 4a. Input a discount code 1. The user enters the discount code in the textbox and clicks the "Apply" button. 2. The system checks if the discount code is valid. 3. The discount code is applied to the order. 3a-3a. Invalid discount code 1. The system displays an "Invalid voucher code" message. S1-4a Failed payment 1. The system displays a "Payment Failed" message. 2. The user can pay for the order via order history or search for the order on the order searching page.

4.4.2.7 Login as Admin

Table 4.7: Use Case Description - Login Admin

<table>
<tr><td>Name of the Use Case:
Login as Admin</td><td>ID:
B1</td><td>Importance Level:
High</td></tr>
<tr><td colspan="2">Actor of the Use Case:
Admin</td><td>Type of the Use Case:
Detail, Essential</td></tr>
<tr><td colspan="3">Stakeholders of the Use Case:
Admin: The admin who wants to log in to the inventory management system.</td></tr>
<tr><td colspan="3">Brief Description of the Use Case:
This use case describes how the admin login to the inventory management system.</td></tr>
<tr><td colspan="3">Trigger of the Use Case:
The admin wishes to access the inventory management system.</td></tr>
<tr><td colspan="3">Relationships of the Use Case:
Association: Admin</td></tr>
<tr><td colspan="3">Standard Event Flows:
1. The admin enters the inventory management system login page.
2. The admin enters the registered username and password. If the admin forgot the password. Perform S1. Forgot Password.
3. The system validates the username and password.
4. The admin successfully logged in to the account.</td></tr>
<tr><td colspan="3">SubFlows:
S1. Forgot Password
1. The admin enters a registered username.
2. The system sends a verification link to the admin email.
3. The admin enters a new password.
4. The password is successfully changed to the new password.</td></tr>
</table>

Table 4.7 (Continued)

Alternate/Exceptional Flows: 3a. Invalid username or password 1. The system displays an error message. 2. If admin enters invalid username and password for five times, perform 3a-2a. Maximum 5 password attempts reached. 3a-2a. Maximum 5 password attempts reached 1. The system locks the user session for 1 hour. 2. The system displayed a message informing the user that the session is locked.

4.4.2.8 Manage Product Stock

Table 4.8: Use Case Description - Manage Product Stock

Name of the Use Case: Manage Product Stock	ID: B2		Importance Level: High
Actor of the Use Case: Admin		Type of the Use Case: Detail, Essential	
Stakeholders of the Use Case: Admin: The admin who wants to manage the product and inventory stock.			
Brief Description of the Use Case: This use case describes how the admin manages the product and inventory stock.			
Trigger of the Use Case: The admin wishes to add, edit or delete product and inventory stock.			
Relationships of the Use Case: Association: Admin			

Table 4.8 (Continued)

Standard Event Flows:

1. The admin enters the product management page in the inventory management system.
2. The system displays a list of all products to the admin.
3. The admin can choose to add, modify or delete the product and its inventory stock.
    a. If the admin wants to add a new product, perform S1. Add a new product.
    b. If the admin wants to modify an existing product, perform S2. Modify an existing product.
    c. If the admin wants to delete an existing product, perform S3. Delete an existing product.

SubFlows:

S1. Add a new product

1. The admin clicks the “Add Product” button.
2. The admin enters the product information.
3. The admin clicks the “Save” button.
4. The system adds the new product to the database.

S2. Modify an existing product

1. The admin clicks the “Edit” button.
2. The admin modifies the product information.
3. The admin clicks the “Save” button.
4. The system updates the product in the database.

S3. Delete an existing product

1. The admin clicks the “Delete” button.
2. The system prompts a confirmation message.
3. The admin confirms the message.
4. The system deletes the product in the database.

Table 4.8 (Continued)

Alternate/Exceptional Flows: S1-4a Incomplete product information 1. The system displays an “Incomplete information” message. S3-3a Admin rejects confirmation message 1. The system aborts the delete operation.

4.4.2.9 Manage Promotional Packages

Table 4.9: Use Case Description - Manage Promotional Packages

Name of the Use Case: Manage Promotional Packages	ID: B3	Importance Level: High
Actor of the Use Case: Admin		Type of the Use Case: Detail, Essential
Stakeholders of the Use Case: Admin: The admin who wants to manage the promotional packages.		
Brief Description of the Use Case: This use case describes how the admin manages the promotional packages.		
Trigger of the Use Case: The admin wishes to add, edit or delete the promotional packages.		
Relationships of the Use Case: Association: Admin		
Standard Event Flows: 1. The admin enters the promotional package management page in the inventory management system. 2. The system displays a list of all promotional packages to the admin. 3. The admin can choose to add, modify or delete the promotional packages.		

Table 4.9 (Continued)

a. If the admin wants to add a new promotional package, perform S1. Add a new promotional package. b. If the admin wants to modify an existing promotional package, perform S2. Modify an existing promotional package. c. If the admin wants to delete an existing promotional package, perform S3. Delete an existing promotional package.
SubFlows: S1. Add a new promotional package 1. The admin clicks the “Add Package” button. 2. The admin enters the promotional package information. 3. The admin adds multiple products into the package. 4. The admin clicks the “Save” button. 5. The system adds the new promotional packages by combining different products to the database. S2. Modify an existing promotional package 1. The admin clicks the “Edit” button. 2. The admin modifies the package information. 3. The admin clicks the “Save” button. 4. The system updates the package in the database. S3. Delete an existing promotional package 1. The admin clicks the “Delete” button. 2. The system prompts a confirmation message. 3. The admin confirms the message. 4. The system deletes the package in the database.
Alternate/Exceptional Flows: S1-5a Incomplete promotional package information 1. The system displays an “Incomplete information” message. S3-3a Admin rejects confirmation message 1. The system aborts the delete operation.

4.4.2.10 Manage Agent and Dropshipper

Table 4.10: Use Case Description - Manage Agent and Dropshipper

<table>
<tr><td>Name of the Use Case:
Manage Agent and Dropshipper</td><td colspan="2">ID:
B4</td><td>Importance Level:
High</td></tr>
<tr><td colspan="2">Actor of the Use Case:
Admin</td><td colspan="2">Type of the Use Case:
Detail, Essential</td></tr>
<tr><td colspan="4">Stakeholders of the Use Case:
Admin: The admin who wants to manage the agents and dropshippers.</td></tr>
<tr><td colspan="4">Brief Description of the Use Case:
This use case describes how the admin manages the agents and dropshippers.</td></tr>
<tr><td colspan="4">Trigger of the Use Case:
The admin wishes to view and manage the agents and dropshippers.</td></tr>
<tr><td colspan="4">Relationships of the Use Case:
Association: Admin</td></tr>
<tr><td colspan="4">Standard Event Flows:
1. The admin enters the customer management page in the inventory management system.
2. The system displays a list of all the customers to the admin.
3. The admin can choose to add, modify, or delete the agents and dropshippers.
a. If the admin wants to add a new agent or dropshipper, perform S1. Add a new agent or dropshipper.
b. If the admin wants to suspend or activate an agent or dropshipper, perform S2. Suspend or activate an existing agent or dropshipper.
c. If the admin wants to accept or reject an agent or dropshipper registration, perform S3. Accept or reject an agent or dropshipper registration.</td></tr>
</table>

Table 4.10 (Continued)

SubFlows: S1. Add a new agent or dropshipper 1. The admin clicks the “Add” button. 2. The admin enters the agent or dropshipper information and selects the applicant's role (agent or dropshipper). 3. The admin clicks the “Save” button. 4. The system adds the new agent or dropshipper to the database. S2. Suspend or accept an existing agent or dropshipper 1. The admin clicks the “Suspend” or “Activate” button on an agent or a dropshipper. 2. The system prompts a confirmation message. 3. The admin confirms the message. 4. The system suspends or activates the agent or dropshipper account based on admin’s input. 5. The system sends an email notification to the admin or dropshipper. S3. Accept or reject an agent or dropshipper registration 1. The admin enters the registration management page. 2. The admin clicks the “Accept” or “Reject” button on an agent or a dropshipper registration. 3. The system prompts a confirmation message. 4. The admin confirms the message. 5. The system accepts or rejects the agent or dropshipper account based on admin’s input. 6. The system sends an email notification to the applicant.
Alternate/Exceptional Flows: S1-4a Incomplete agent or dropshipper information 1. The system displays an “Incomplete information” message.

4.4.2.11 Manage Discount

Table 4.11: Use Case Description - Manage Discount

Name of the Use Case: Manage Discount	ID: B5	Importance Level: High
Actor of the Use Case: Admin	Type of the Use Case: Detail, Essential	
Stakeholders of the Use Case: Admin: The admin who wants to manage the discount codes for e-commerce platforms.		
Brief Description of the Use Case: This use case describes how the admin manages the discount codes.		
Trigger of the Use Case: The admin wishes to add, edit or delete the discount codes for e-commerce platofmrs.		
Relationships of the Use Case: Association: Admin		
Standard Event Flows: 1. The admin enters the promotional package management page in the inventory management system. 2. The system displays a list of all promotional packages to the admin. 3. The admin can choose to add, modify or delete the promotional packages. a. If the admin wants to add a new discount code, perform S1. Add a new discount code. b. If the admin wants to modify an existing discount code, perform S2. Modify an existing discount code. c. If the admin wants to delete an existing discount code, perform S3. Delete an existing discount code.		

Table 4.11 (Continued)

S1. Add a new discount code 1. The admin clicks the “Create Discount Code” button. 2. The admin enters the discount code information. 3. The admin clicks the “Save” button. 4. The system adds the new discount code information to the database. S2. Modify an existing discount code 1. The admin clicks the “Edit” button. 2. The admin modifies the discount code information. 3. The admin clicks the “Save” button. 4. The system updates the discount code information in the database. S3. Delete an existing discount code 1. The admin clicks the “Delete” button. 2. The system prompts a confirmation message. 3. The admin confirms the message. 4. The system deletes the discount code information in the database.
Alternate/Exceptional Flows: S1-5a Incomplete discount code information 1. The system displays an “Incomplete information” message. S3-3a Admin rejects confirmation message The system aborts the delete operation.

4.4.2.12 Manage Order

Table 4.12: Use Case Description - Manage Order

<table>
<tr><td colspan="2">Name of the Use Case:
Manage Order</td><td>ID:
B6</td><td colspan="2">Importance Level:
High</td></tr>
<tr><td colspan="3">Actor of the Use Case:
Admin</td><td colspan="2">Type of the Use Case:
Detail, Essential</td></tr>
<tr><td colspan="5">Stakeholders of the Use Case:
Admin: The admin who wants to manage the sales and orders.</td></tr>
<tr><td colspan="5">Brief Description of the Use Case:
This use case describes how the admin manages the sales and orders.</td></tr>
<tr><td colspan="5">Trigger of the Use Case:
The admin wishes to view the details of the order placed by the customer.</td></tr>
<tr><td colspan="5">Relationships of the Use Case:
Association: Admin
Extend: Generate Invoice</td></tr>
<tr><td colspan="5">Standard Event Flows:
1. The admin enters the order management page in the inventory management system.
2. The system displays a list of all the orders to the admin.
3. The admin selects different criteria to filter.
4. The system filters the result required by the admin.
5. The admin can add new orders manually. Perform S1. Add New Order.
6. The admin can view the specific order details. Perform S2. View Order Details.</td></tr>
</table>

Table 4.12 (Continued)

SubFlows: S1. Add New Order 1. The admin clicks the “Add Order” button. 2. The system redirects the admin to the e-commerce platform. S2. View Order Details 1. The admin clicks an order in the list of all orders. 2. The system shows all the details of the order selected. 3. The admin can generate an invoice for the order. Perform S2-3a. Generate Invoice.
Alternate/Exceptional Flows: S2-3a Generate Invoice 1. The system generates an invoice in PDF format containing all the information of the order.

4.4.2.13 View Business Insights

Table 4.13: Use Case Description - View Business Insights

<table>
<tr><td>Name of the Use Case:
View Business Insights</td><td>ID:
B7</td><td>Importance Level:
Medium</td></tr>
<tr><td colspan="2">Actor of the Use Case:
Admin</td><td>Type of the Use Case:
Detail, Essential</td></tr>
<tr><td colspan="3">Stakeholders of the Use Case:
Admin: The admin who wants to view the analysis of the company's business.</td></tr>
<tr><td colspan="3">Brief Description of the Use Case:
This use case describes how the admin views different types of business insight.</td></tr>
<tr><td colspan="3">Trigger of the Use Case:
The admin wishes to gain insight into the business development.</td></tr>
<tr><td colspan="3">Relationships of the Use Case:
Association: Admin
Extend: Generate Sales and Inventory Reports</td></tr>
<tr><td colspan="3">Standard Event Flows:
1. The admin enters the business insights page of the inventory management system.
2. The system displays a list of different analyses, such as statistics, sales key metrics analysis and item ranking list to the admin.
3. The admin clicks on the particular sales key metrics. Perform S1. Show analysis details.
4. If the admin wishes to generate a report. Perform 4a. Generate Sales and Inventory Reports.
5. If the admin wishes to view the inventory analysis. Perform 5a. View Inventory Analysis.</td></tr>
</table>

Table 4.13 (Continued)

SubFlows:

S1. Show analysis details

1. The user clicks on a particular sales key metrics.
2. The systems show more detailed analysis in a line chart.
3. The admin generates a report button. Perform S1-3a. Generate Sales and Inventory Reports.

Alternate/Exceptional Flows:

4a. Generate Sales and Inventory Reports

1. The admin clicks the “Generate Reports” buttons.
2. The system generates the sales order report in Excel format to the admin.

5a. View Inventory Analysis

1. The admin enters the inventory analysis pages.
2. The system displays the ABC analysis of the products to the admin.
3. The admin select one out of four inventory analyses, which are ABC, HML, EOQ, and SS analyses.
4. The system displays respective inventory analysis to the admin.

S1-3a. Generate Sales and Inventory Reports

1. The admin clicks the “Generate Reports” buttons.
2. The system generates the sales and Inventory analysis report in Excel format.

4.4.2.14 Manage Shipping Fee

Table 4.14: Use Case Description - Manage Shipping Fee

Name of the Use Case: Manage Shipping Fee	ID: B8	Importance Level: Medium
Actor of the Use Case: Admin	Type of the Use Case: Detail, Essential	
Stakeholders of the Use Case: Admin: The admin who wants to manage the shipping fees for e-commerce platforms.		
Brief Description of the Use Case: This use case describes how the admin manages the shipping fees.		
Trigger of the Use Case: The admin wishes to add, edit or delete the shipping fees for e-commerce platforms.		
Relationships of the Use Case: Association: Admin		
Standard Event Flows: 1. The admin enters the shipping fee management page in the inventory management system. 2. The system displays a list of all shipping fees to the admin. 3. The admin can choose to add, modify or delete the shipping fee. a. If the admin wants to add a new shipping fee, perform S1. Add a new shipping fee. b. If the admin wants to modify an existing shipping fee, perform S2. Modify an existing shipping fee. c. If the admin wants to delete an existing shipping fee, perform S3. Delete an existing shipping fee.		

Table 4.14 (Continued)

S1. Add a new shipping fee 1. The admin clicks the “Add Shipping Fee” button. 2. The admin enters the shipping fee information. 3. The admin clicks the “Save” button. 4. The system adds the new shipping fee information to the database. S2. Modify an existing shipping fee 1. The admin clicks the “Edit” button. 2. The admin modifies the shipping fee information. 3. The admin clicks the “Save” button. 4. The system updates the shipping fee information in the database. S3. Delete an existing shipping fee 1. The admin clicks the “Delete” button. 2. The system prompts a confirmation message. 3. The admin confirms the message. 4. The system deletes the shipping fee information in the database.
Alternate/Exceptional Flows: S1-3a Incomplete discount code information 1. The system displays an “Incomplete information” message. S3-3a Admin rejects confirmation message 1. The system aborts the delete operation.

4.4.2.15 Manage Pickup Location

Table 4.15: Use Case Description - Manage Pickup Location

Name of the Use Case: Manage Shipping Fee	ID: B9	Importance Level: Medium
Actor of the Use Case: Admin	Type of the Use Case: Detail, Essential	
Stakeholders of the Use Case: Admin: The admin who wants to manage the pickup locations for orders on e-commerce platforms.		
Brief Description of the Use Case: This use case describes how the admin manages the pickup locations.		
Trigger of the Use Case: The admin wishes to add, edit or delete the pickup location for e-commerce platforms.		
Relationships of the Use Case: Association: Admin		
Standard Event Flows: 4. The admin enters the pickup location management page in the inventory management system. 5. The system displays a list of all pickup locations to the admin. 6. The admin can choose to add, modify or delete the pickup locations. a. If the admin wants to add a new pickup location, perform S1. Add a new pickup location. b. If the admin wants to modify an existing pickup location, perform S2. Modify an existing pickup location. c. If the admin wants to delete an existing pickup location, perform S3. Delete an existing pickup location.		

Table 4.15 (Continued)

S1. Add a new pickup location 5. The admin clicks the “Add Pickup Location” button. 6. The admin enters the pickup location information. 7. The admin clicks the “Save” button. 8. The system adds the new pickup location information to the database. S2. Modify an existing pickup location 5. The admin clicks the “Edit” button. 6. The admin modifies the pickup location information. 7. The admin clicks the “Save” button. 8. The system updates the pickup location information in the database. S3. Delete an existing pickup location 5. The admin clicks the “Delete” button. 6. The system prompts a confirmation message. 7. The admin confirms the message. 8. The system deletes the pickup location information in the database.
Alternate/Exceptional Flows: S1-3a Incomplete pickup location information 2. The system displays an “Incomplete information” message. S3-3a Admin rejects confirmation message 2. The system aborts the delete operation.

4.5 Summary

All in all, this chapter has discussed how the requirements are elicited. A list of the functional and non-functional requirements of the Sharifah Food inventory management system was produced from the requirements discovery. Lastly, use case modelling was developed with the list of functional requirements defined. By developing the use case modelling, the flow of the applications' activities is described to understand the applications better.

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

This chapter will discuss the design of the system, which includes the system architecture design, entity-relationship design, data dictionary, activity diagram design, and screen prototype design.

5.2 System Architecture Design

The system architecture design for the system is a three-tier client-server architecture design.

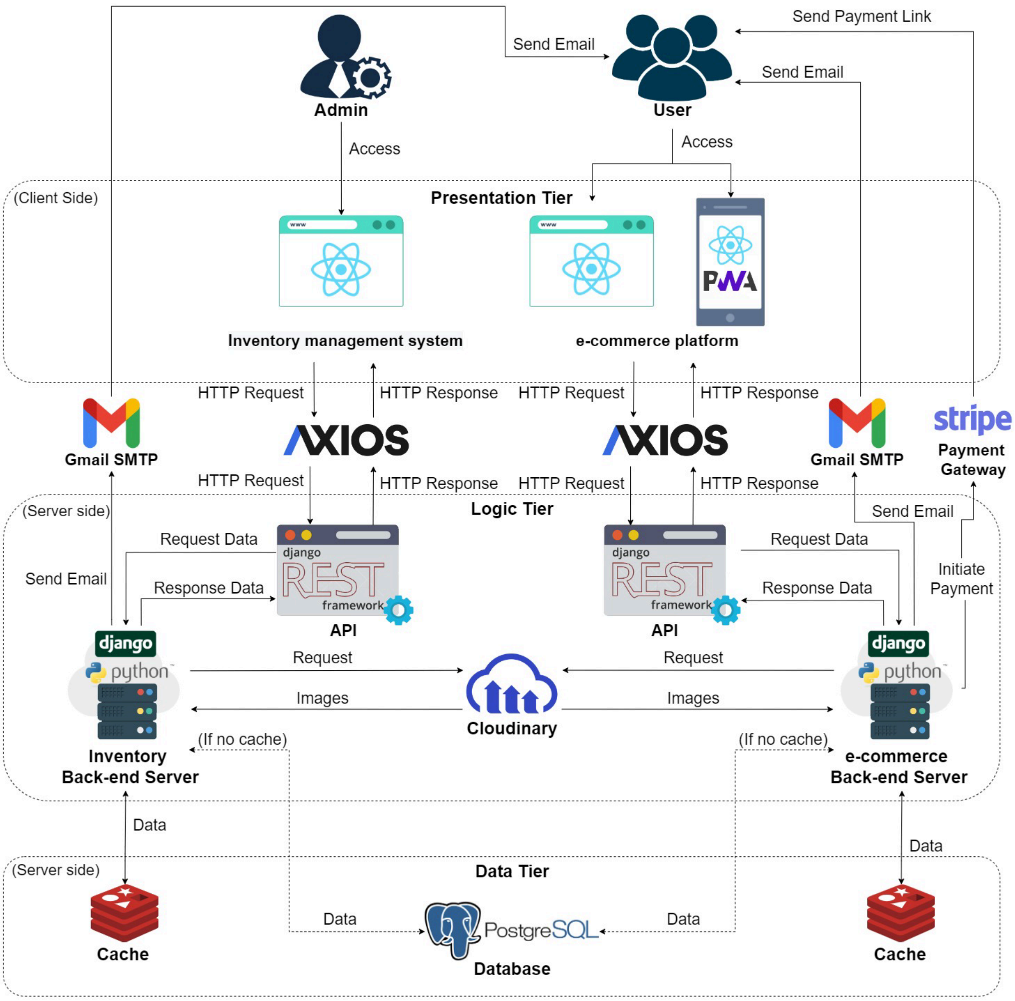


Figure 5.1: System Architecture Design

Figure 5.1: System Architecture shows the system architecture design for the system. There are three tiers in the systems: presentation tier, logic tier, and data tier.

The presentation tier is the topmost tier of the system. It includes the UI for the admins and users to interface and communicate with the inventory management system and the e-commerce platform.

In terms of the presentation tier, only the admin can access the inventory management system via the web application in this system. In contrast, the users can access the e-commerce platform via the web application and PWA mobile platform. The inventory management systems and e-commerce platforms connect to the backend server via Axios, a promise-based HTTP client library, which helps transmit the asynchronous requests from the front-end to the API endpoints in the backend server. Besides, the front-end applications of both systems are built with React, a Javascript framework for building user interfaces.

On the other hand, the data tier is the bottommost tier that stores all the data and information. The main objective of the data tier is to reduce the coupling of the data stored with the system server processing the logic. This data tier includes the database and cache server of the system.

In terms of the data tier, PostgreSQL was used as the database that provides the data to the backend servers in terms of the data tier. Besides, Redis was also used to cache the constantly retrieved data so that the backend server does not require to retrieve the same data often, allowing the database to have a better performance.

The logic tier, also known as the application tier, is designed to bridge the communication between the presentation tier and data tier. It acts as the middle layer to receive the request from the presentation tier and process the data retrieved from the data tier, allowing the data in the data tier to be independent of the server. The logic tier can also be called the application's heart because all the logic controls are included in this tier. This logic tier includes the Django backend servers and its REST framework for the inventory management system and e-commerce platform.

In terms of the logic tier, the backend system of the inventory management system and the e-commerce platforms are separated into two systems to reduce the coupling between the two systems, allowing better maintainability between the two systems. The backend system is built with Django, a python-based web framework that adopts the MTV architecture. Besides, the Django REST framework was used to develop RESTful API to communicate with the front-end applications.

Moreover, some additional third-party applications were used to facilitate the features of the inventory management system and the e-commerce platforms. Cloudinary was used to store the images, such as the thumbnails and the images of the products and packages of Sharifah Food. Besides, the Gmail SMTP server was used to send the notifications to the customer for order updates and updates on the agent and dropshipper status.

5.3 Entity-Relationship Diagram

The entity-relationship diagram, also known as ERD, is a diagram that contains the entities and the relationships between the entities. It helps to provide a deep insight into how the entities are related. This section will divide the ERD into three models: conceptual ERD, logical ERD, and physical ERD.

5.3.1 Conceptual ERD

The conceptual ERD is the simplest ERD to describe the entity and its relationship. It is used to provide the overall idea of the entities' relationship. Figure 5.2: Conceptual ERD shows the conceptual ERD of the system.

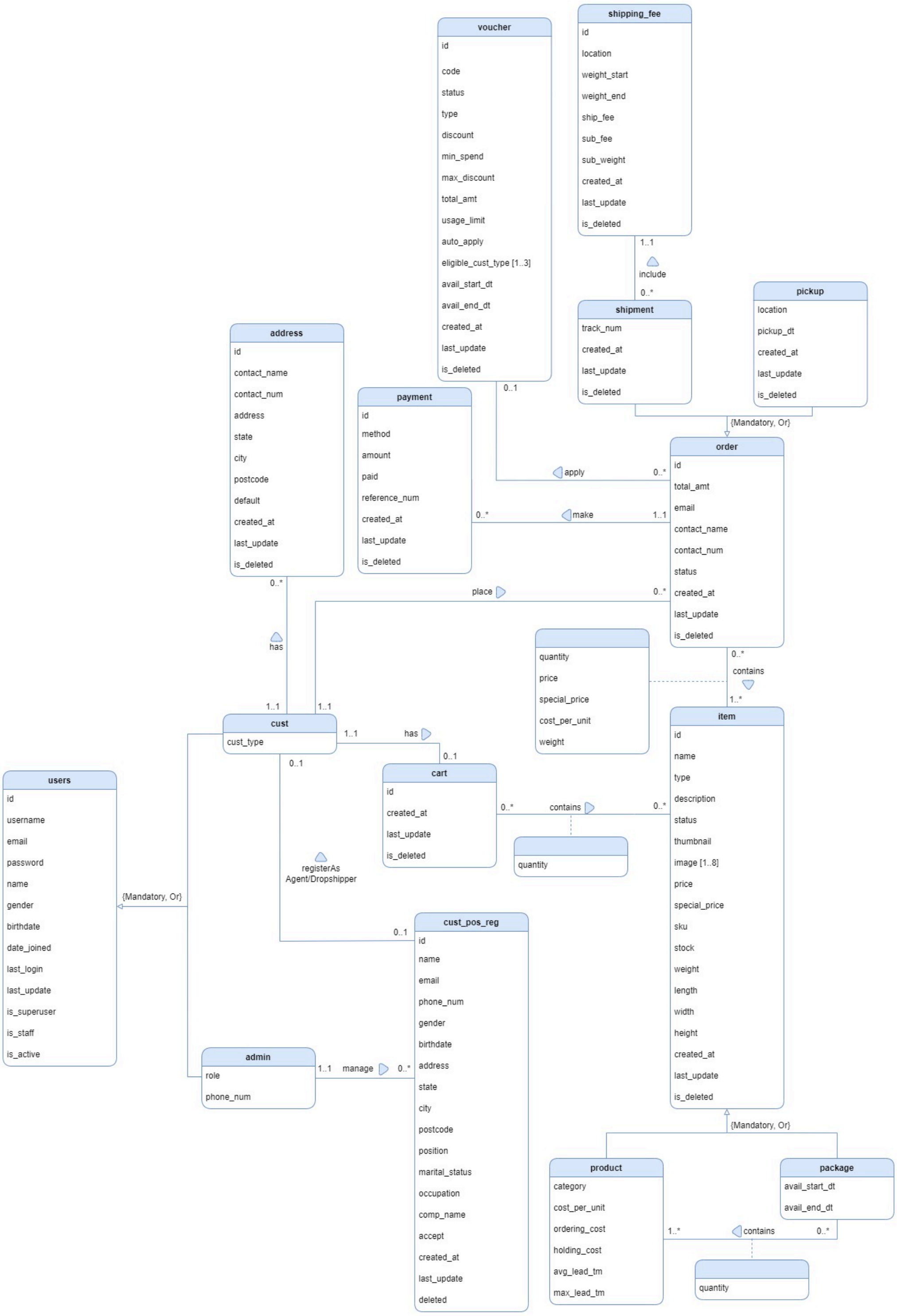


Figure 5.2: Conceptual ERD

5.3.2 Logical ERD

The logical ERD provides more detailed information than the conceptual ERD. It includes the primary and foreign keys of each table. Besides, it also provides the normalized tables from the conceptual ERD to reduce the data redundancy in such a manner that the database would not suffer from the update anomalies, such as insertion, deletion, and modification anomalies. Figure 5.3 shows the logical ERD of the system.

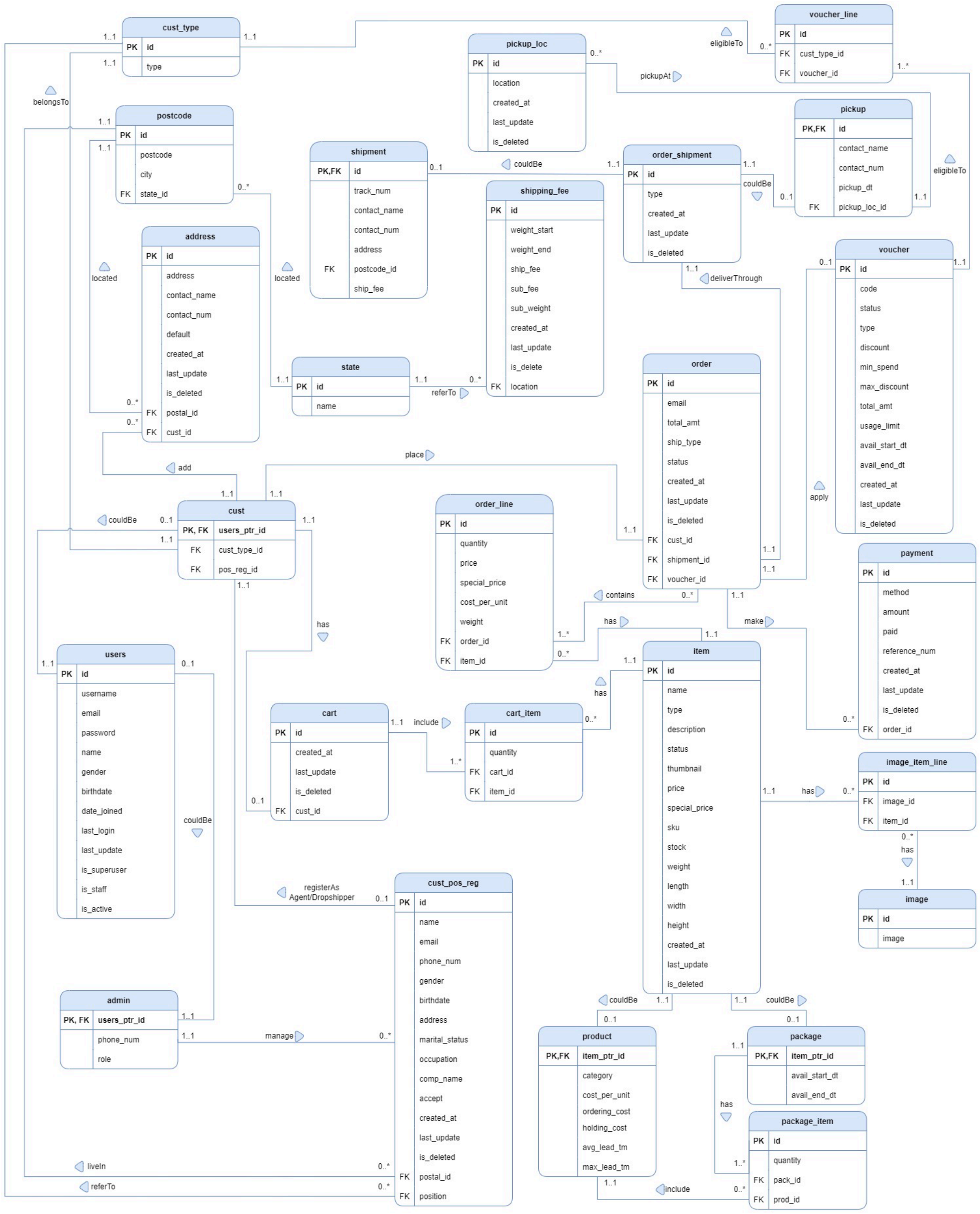


Figure 5.3: Logical ERD

5.3.3 Physical ERD

The physical ERD represents the actual database design. It is essential to consider the restrictions and conventions of the database used. Besides, it should not contain the reserved word held by the database. It also includes the column type of each data field. Figure 5.4 shows the logical ERD of the system.

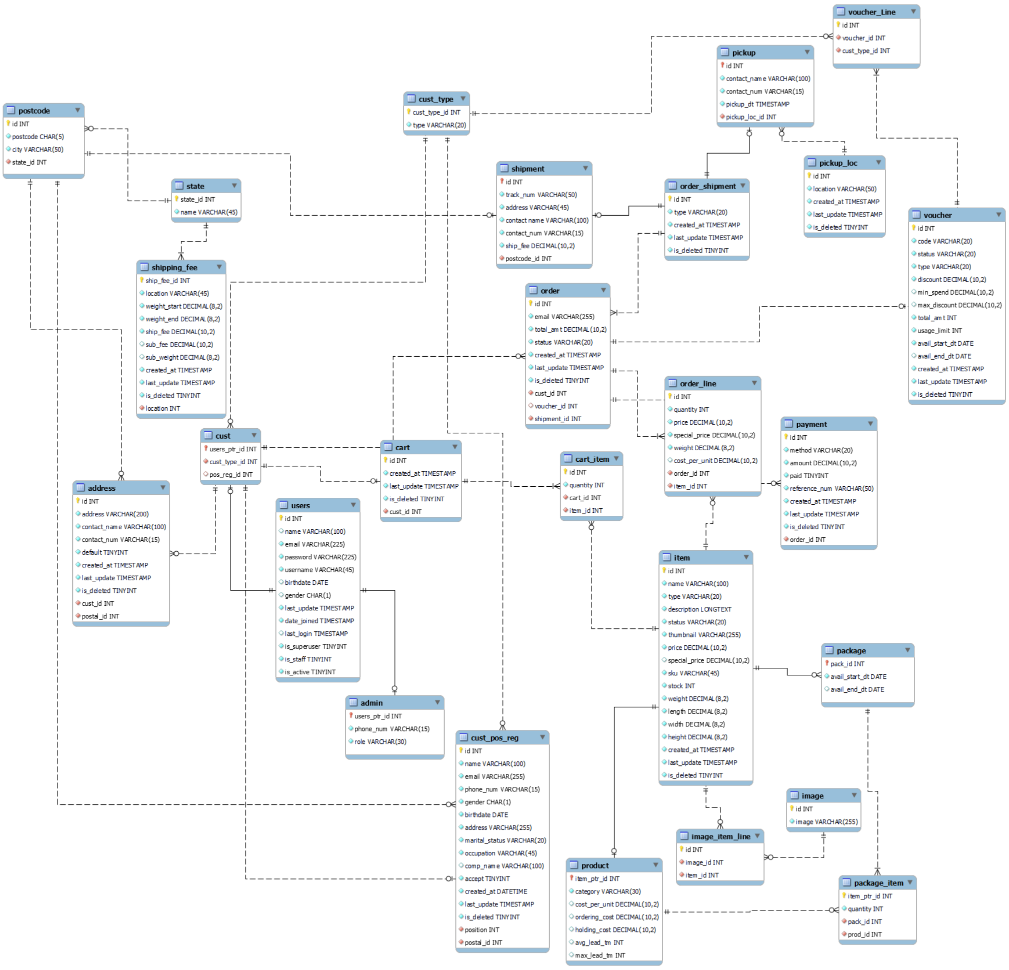


Figure 5.4: Physical ERD

5.4 Data Dictionary

The data dictionary is a collection of the metadata about the data fields found in the database. It provides the structure and content of the tables in the database and describes the meaning of each data field. The data dictionary of the system is shown below. The data types used in the data dictionary are based on the data types in PostgreSQL. The ‘varying’ data type is identical to the ‘varchar’ data type in MySQL.

Table Name: users

Table 5.1: Users Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a user	integer	✓	
name	User’s name	varying (100)		
email	User’s email	varying (255)		
password	User’s password stored in SHA256 hash	varying (255)		
username	User’s username	varying (45)		
birthdate	User’s birthdate	date		
gender	User’s gender	varying (1)		
last_update	Last modification made to the record	timestamp		
date_joined	The first date when a user is registered	timestamp		
last_login	Last login time of a user	timestamp		
is_superuser	Indicator to check if the user is a superuser	boolean		
is_staff	Indicator to check if the user is a staff	boolean		
is_active	Indicator to check if the user’s account is active	boolean		

Table Name: admin

Table 5.2: Admin Table Data Dictionary

Field Name	Description	Data Type	PK	FK
user_ptr_id	Unique identifier of a admin	integer	✓	users
role	Admin's role (superadmin, admin)	varying (20)		
phone_num	Admin's phone number	varying (15)		

Table Name: cust

Table 5.3: Cust Table Data Dictionary

Field Name	Description	Data Type	PK	FK
user_ptr_id	Unique identifier of a customer	integer	✓	users
cust_type_id	Customer's type reference id	integer		cust_type
pos_reg_id	Agent or Dropshipper registration information reference id	integer		cust_pos_reg

Table Name: cust_type

Table 5.4: Cust Type Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a customer type	integer	✓	
type	Name of the customer type (Direct customer, Agent, Dropshipper)	varying (20)		

Table Name: state

Table 5.5: State Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a state	integer	✓	
name	Name of the state	varying (45)		

Table Name: postcode

Table 5.6: Postcode Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a postcode	integer	✓	
postcode	Postal Code	varying (5)		
city	City of the postal code	varying (50)		
state_id	State reference id of the postal code	integer		state

Table Name: address

Table 5.7: Address Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of an address	integer	✓	
address	Information of the address	varying (200)		
contact_name	Contact Name of the order receiver	varying (100)		
contact_num	Contact Number of the order receiver	varying (15)		
default	Indicator to check if the address is the customer's default address	boolean		
created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		
cust_id	Customer reference id	integer		cust
postcode_id	Postcode reference id for the address	integer		postcode

Table Name: item

Table 5.8: Item Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of an item	integer	✓	
name	Item's name	varying (100)		
type	Item's type (product, package)	varying (20)		
description	Item's description	text		
status	Item's status	varying (20)		
thumbnail	Item's thumbnail location path in the Cloudinary	varying (255)		
price	Item's original price	numeric (10,2)		
special_price	Item's discounted price that is displayed to the customers	numeric (10,2)		
sku	Item's stock keeping unit	varying(45)		
stock	Item's current stock	integer		
weight	Item's weight in grams	numeric (8,2)		
length	Item's length in cm	numeric (8,2)		
width	Item's width in cm	numeric (8,2)		
height	Item's height in cm	numeric (8,2)		
created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		

Table Name: product

Table 5.9: Product Table Data Dictionary

Field Name	Description	Data Type	PK	FK
item_ptr_id	Unique identifier of a product	integer	✓	item
category	Product's category (ready to eat, ready to cook, paste, others)	varying (30)		
cost_per_unit	Product's unit price	numeric (10,2)		
ordering_cost	Product's fixed costs per purchase order	numeric (10,2)		
holding_cost	Cost of holding the product in stock per unit, per month	numeric (10,2)		
avg_lead_tm	Product's average lead time when restocking	integer		
max_lead_tm	Product's maximum lead time when restocking	integer		

Table Name: package

Table 5.10: Package Table Data Dictionary

Field Name	Description	Data Type	PK	FK
item_ptr_id	Unique identifier of a package	integer	✓	item
avail_start_dt	Package's launching date	date		
avail_end_dt	Package's ending date	date		

Table Name: image

Table 5.11: Image Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of an image	integer	✓	
image	Image location in the Cloudinary	varying (255)		

Table Name: image_item_line

Table 5.12: Image Item Line Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of an image item line	integer	✓	
image_id	Reference id of an image	integer		image
item_id	Reference id of an item	integer		item

Table Name: shipment

Table 5.13: Shipment Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a shipment	integer	✓	
track_num	Tracking Number of the shipment	varying (5)		
address	Destination address to be delivered	varying (200)		
contact_name	Contact Name of the order receiver	varying (100)		
contact_num	Contact Number of the order receiver	varying (15)		
ship_fee	Total Shipping Fee of the order	numeric (10,2)		
postcode_id	Reference id of the address's postcode			postcode

Table Name: pickup_loc

Table 5.14: Pickup Location Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a pickup location	integer	✓	
location	Name of the pickup location	varying (80)		
created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		

Table Name: pickup

Table 5.15: Pickup Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a pickup	integer	✓	
contact_name	Contact Name of the order receiver	varying (100)		
contact_num	Contact Number of the order receiver	varying (15)		
pickup_dt	Pickup Time of the order	timestamp		
pickup_loc_id	Reference id of the pickup location	integer		pickup_loc

Table Name: order_shipment

Table 5.16: Order Shipment Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of an order shipment	integer	✓	
type	Type of the shipment (pickup, shipping)	varying (20)		

Table 5.16 (Continued)

created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		

Table Name: voucher

Table 5.17: Voucher Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a voucher	integer	✓	
code	Voucher code to be entered when placing the orders	varying (20)		
status	Voucher's status	varying (20)		
type	Voucher's type (amount, percentage)	varying (20)		
discount	Voucher's discount amount	numeric (10,2)		
min_spend	Minimum order spend required to apply the voucher	numeric (10,2)		
max_discount	Maximum discount amount when applying the voucher	numeric (10,2)		
total_amt	Total number of a voucher available to redeem	integer		
usage_limit	The number of times a customer can apply the voucher	integer		

Table 5.17 (Continued)

auto_apply	Whether to auto-apply the voucher when a customer checkout the order	boolean		
avail_start_dt	Voucher's launching date	date		
avail_end_dt	Voucher's ending date	date		
created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		

Table Name: voucher_line

Table 5.18: Voucher Line Table Data Dictionary

Field Name	**Description**	**Data Type**	**PK**	**FK**
id	Unique identifier of a voucher line	integer	✓	
voucher_id	Reference id of the voucher	integer		voucher
cust_type_id	Reference id of the customer type that the voucher applies to	integer		cust_type

Table Name: payment

Table 5.19: Payment Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a payment	integer	✓	
method	Payment method (card, fpx)	varying (20)		
amount	Voucher's discount amount	numeric (10,2)		
paid	Indicator to check if the payment is successful	boolean		
reference_num	Reference	numeric (10,2)		
created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		
order_id	Reference id for order	integer		order

Table Name: order

Table 5.20: Order Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of an order	integer	✓	
total_amt	Order's total amount	numeric (10,2)		
status	Order status	varying (20)		
email	Customer email (to update status to the unregistered customer)	varying (255)		

Table 5.20 (Continued)

created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		
cust_id	Reference id of the customer who places the order	integer		cust
shipment_id	Reference id for the order shipment	integer		shipment
voucher_id	Reference id for the voucher applied to the order	integer		voucher

Table Name: order_line

Table 5.21: Order Line Table Data Dictionary

Field Name	**Description**	**Data Type**	**PK**	**FK**
id	Unique identifier of an order line	integer	✓	
quantity	The total quantity of an item placed in the order	integer		
price	Price of an item placed in the order	numeric (10,2)		
special_price	Special price of an item placed in the order	numeric (10,2)		
cost_per_unit	The unit price of an item placed in the order	numeric (10,2)		
weight	Weight of an item placed in the order	numeric (8,2)		
order_id	Reference id of the order	integer		order
item_id	Reference id of the item	integer		item

Table Name: cart

Table 5.22: Cart Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a cart	integer	✓	
created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		
cust_id	Reference id of the customer	integer		cust

Table Name: cart_item

Table 5.23: Cart Item Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a cart item	integer	✓	
quantity	The total quantity of an item placed in the cart	integer		
cart_id	Reference id of the cart	integer		cart
item_id	Reference id of the item	integer		item

Table Name: shipping_fee

Table 5.24: Shipping Fee Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a shipping_fee	integer	✓	
weight_start	The matching starting weight for the shipping_fee with the order’s total weight	numeric (8,2)		

Table 5.24 (Continued)

weight_end	The matching ending weight for the shipping_fee with the order's total weight	numeric (8,2)		
ship_fee	Total shipping fee	numeric (10,2)		
sub_fee	The subsequent fee to be applied for sub_weight if the order's total weight exceeds the weight_end	numeric (10,2)		
sub_weight	The subsequent weight to be charged after the order's total weight exceeds the weight_end	numeric (8,2)		
created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		
location_id	Reference id for the state of the shipping fee	integer		state

Table Name: cust_pos_reg

Table 5.25: Customer Position Registration Table Data Dictionary

Field Name	**Description**	**Data Type**	**PK**	**FK**
id	Unique identifier of a customer position registration	integer	✓	
name	Applicant's name	varying (100)		
marital status	Applicant's marital status	varying (20)		
email	Applicant's email	varying (255)		
phone_num	Applicant's phone number	varying (15)		
gender	Applicant's gender	varying (1)		
birthdate	Applicant's birthdate	date		
address	Applicant's address	varying (255)		
occupation	Applicant's current job	varying (45)		
comp_name	Applicant's company name (optional)	varying (100)		
accept	Indicator to check if the applicant is accepted	boolean		
created_at	Creation time of the record	timestamp		
last_update	Last modification made to the record	timestamp		
is_deleted	Indicator to check if the record has been deleted	boolean		
position_id	Reference id of the position to be registered by the applicant	integer		cust_type
postcode_id	Reference id of the postcode of the applicant	integer		postcode

Table Name: package_item

Table 5.26: Package Item Table Data Dictionary

Field Name	Description	Data Type	PK	FK
id	Unique identifier of a package item	integer	✓	
quantity	The total quantity of an item placed in the package	integer		
pack_id	Reference id of the package	integer		package
item_id	Reference id of the item	integer		item

5.5 Activity Diagrams

Activity diagrams are behavioural diagrams that illustrate the behaviour of a system, and it helps to model the business processes by showing the overall flow of execution. In this section, the activity diagrams will be the extension of the use case diagram developed in Chapter 4, which visualizes the use case at an in-depth level.

5.5.1 Inventory Management System

This section shows the activity diagrams related to the inventory management system.

5.5.1.1 Login As Admin

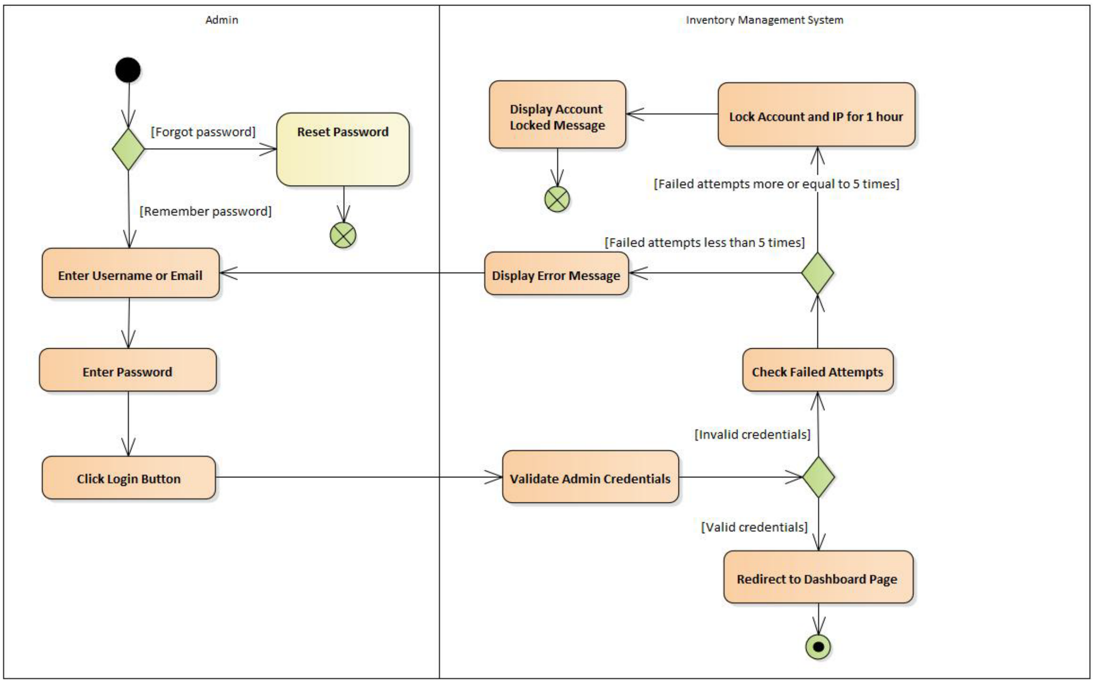

Admin
Inventory Management System
[Forgot password]
Reset Password
[Remember password]
Enter Username or Email
Enter Password
Click Login Button
Display Account Locked Message
Lock Account and IP for 1 hour
[Failed attempts more or equal to 5 times]
[Failed attempts less than 5 times]
Display Error Message
Check Failed Attempts
[Invalid credentials]
Validate Admin Credentials
[Valid credentials]
Redirect to Dashboard Page

Figure 5.5: Activity Diagram - Login As Admin

5.5.1.2 Reset Password

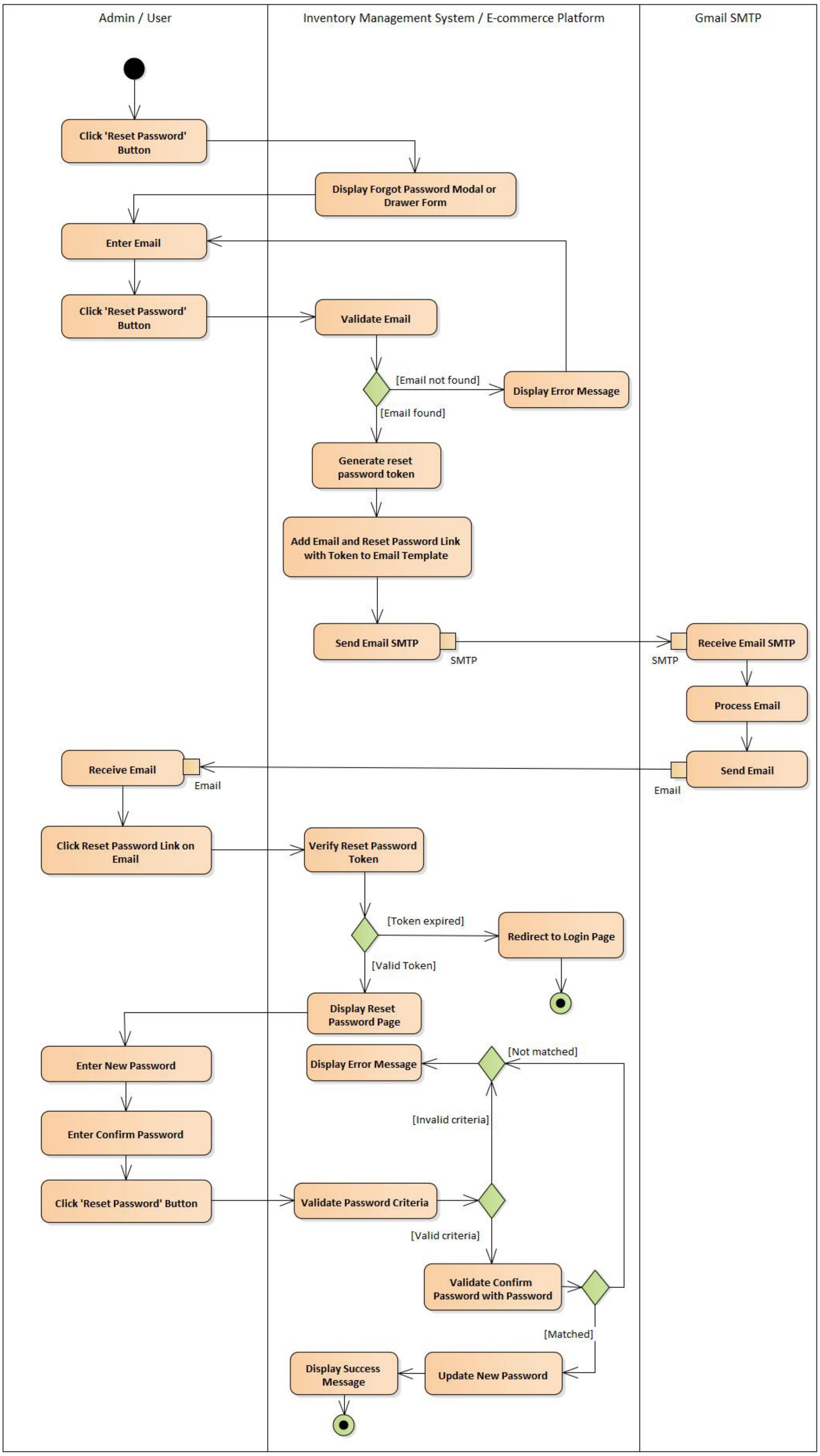

Admin / User
Inventory Management System / E-commerce Platform
Gmail SMTP
Click 'Reset Password' Button
Display Forgot Password Modal or Drawer Form
Enter Email
Click 'Reset Password' Button
Validate Email
[Email not found]
Display Error Message
[Email found]
Generate reset password token
Add Email and Reset Password Link with Token to Email Template
Send Email SMTP
SMTP
SMTP
Receive Email SMTP
Process Email
Receive Email
Email
Send Email
Email
Click Reset Password Link on Email
Verify Reset Password Token
[Token expired]
Redirect to Login Page
[Valid Token]
Display Reset Password Page
Enter New Password
Display Error Message
[Not matched]
[Invalid criteria]
Enter Confirm Password
Click 'Reset Password' Button
Validate Password Criteria
[Valid criteria]
Validate Confirm Password with Password
[Matched]
Display Success Message
Update New Password

Figure 5.6: Activity Diagram - Reset Password

5.5.1.3 Manage Product Stock

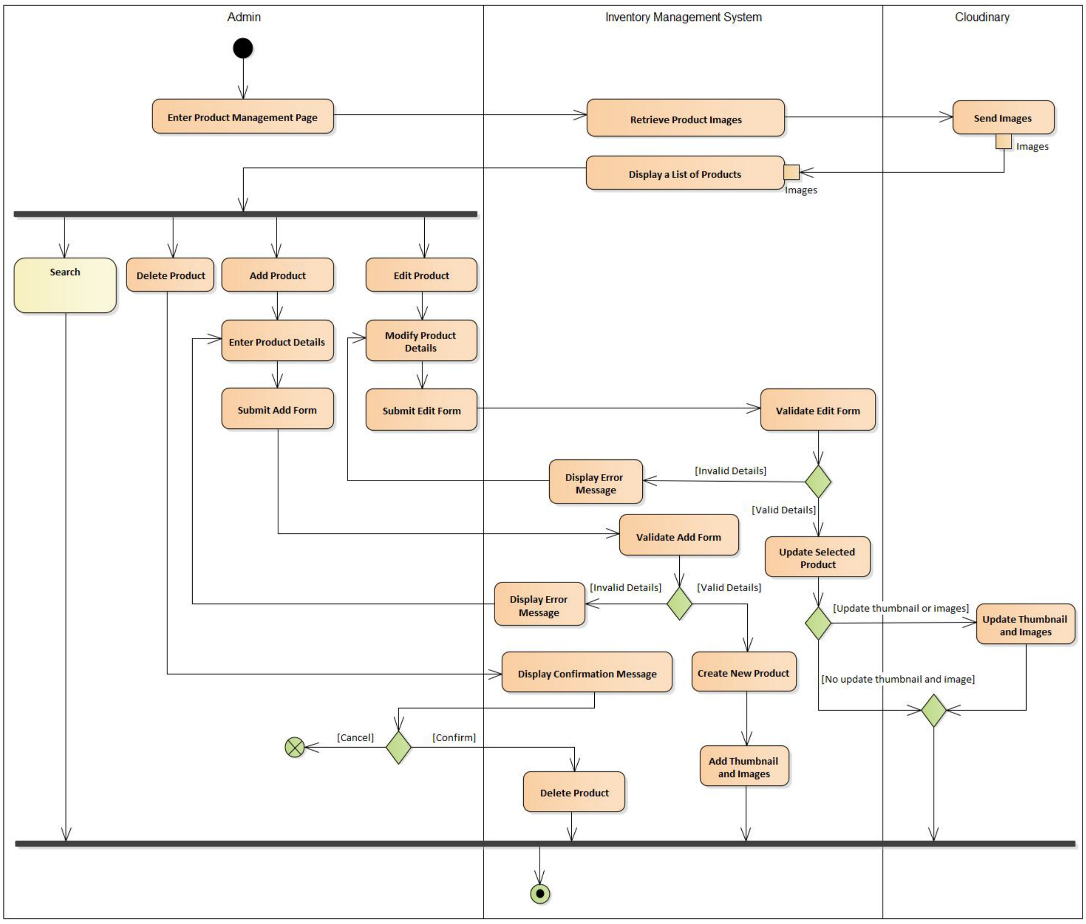


Figure 5.7: Activity Diagram - Manage Product Stock

5.5.1.4 Manage Promotional Packages

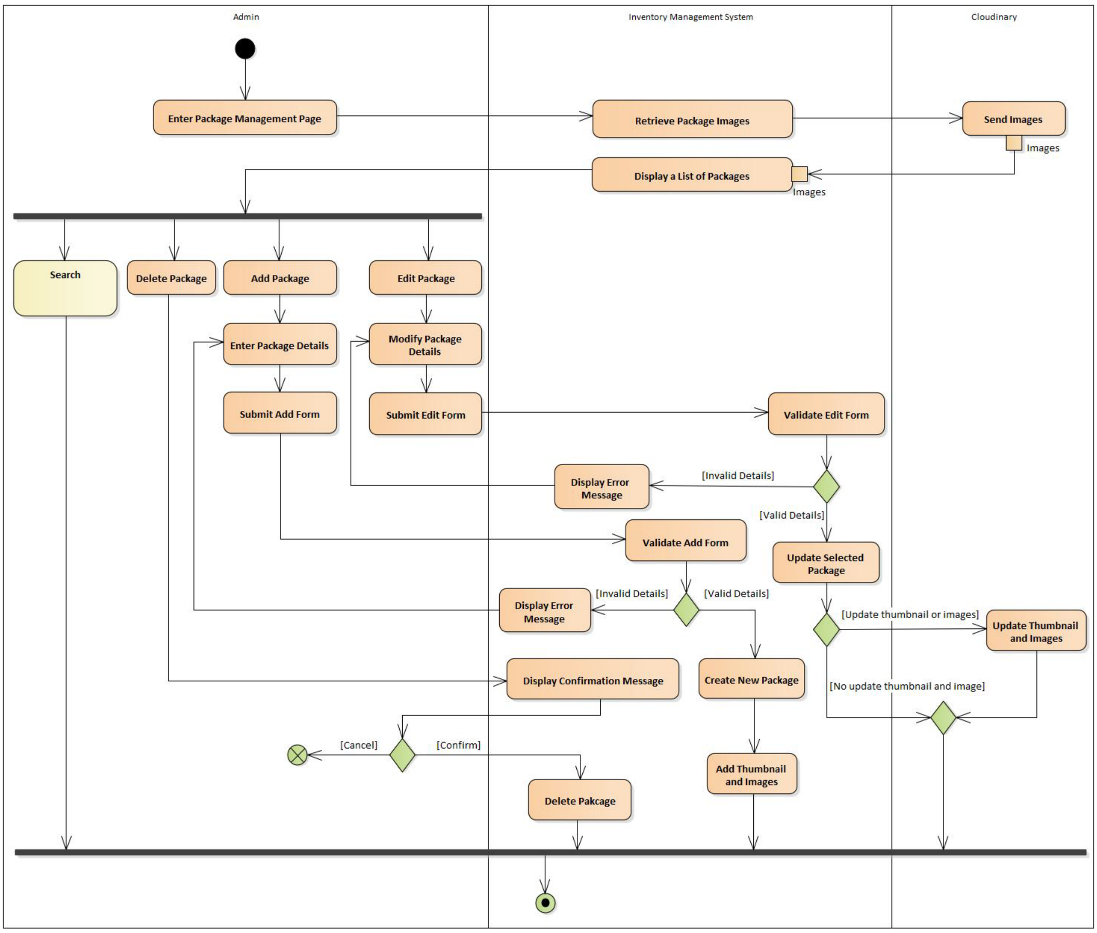


Figure 5.8: Activity Diagram - Manage Promotional Packages

5.5.1.5 Manage Agent and Dropshipper

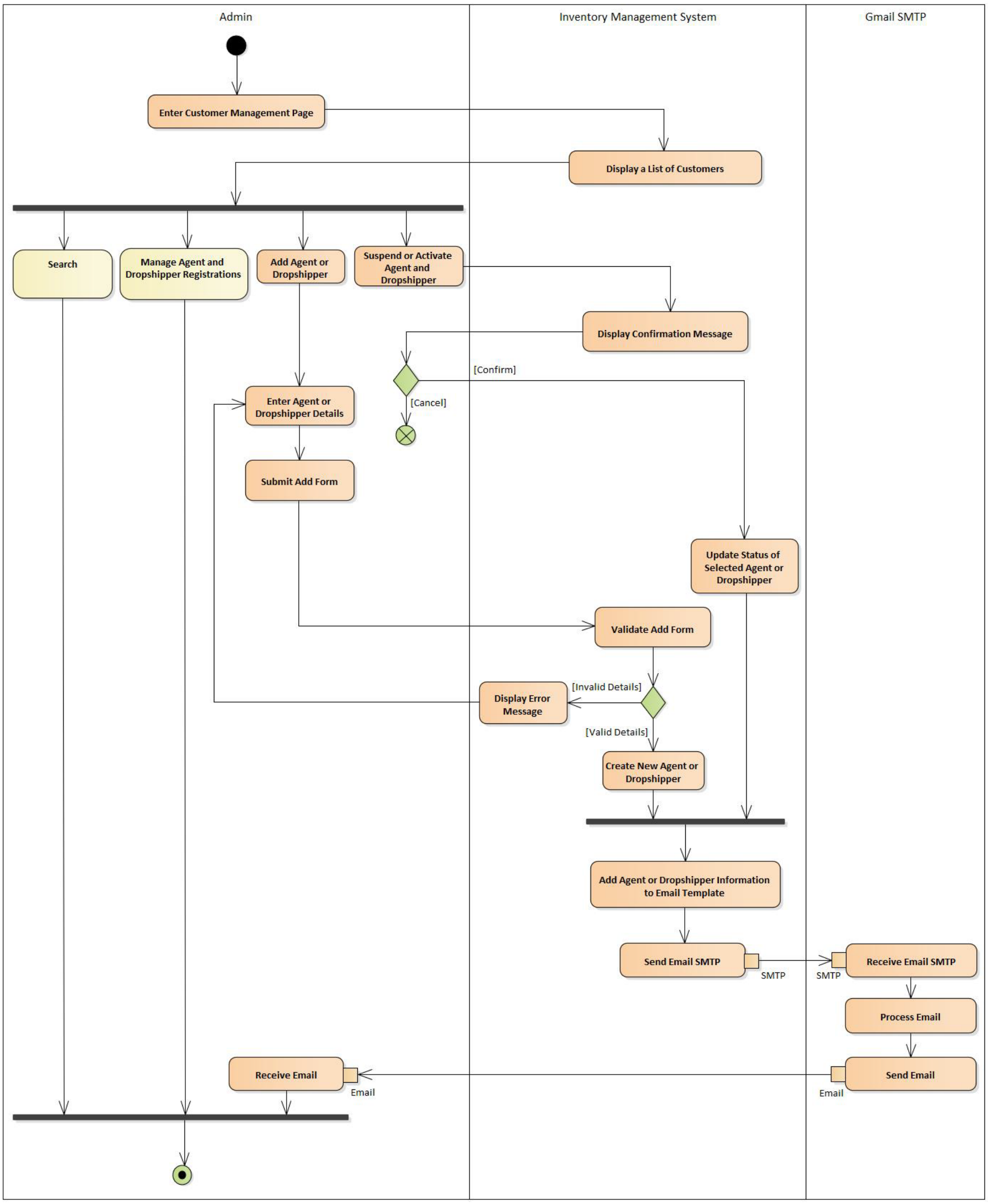

Admin
Inventory Management System
Gmail SMTP
Enter Customer Management Page
Display a List of Customers
Search
Manage Agent and Dropshipper Registrations
Add Agent or Dropshipper
Suspend or Activate Agent and Dropshipper
Display Confirmation Message
[Confirm]
[Cancel]
Enter Agent or Dropshipper Details
Submit Add Form
Update Status of Selected Agent or Dropshipper
Validate Add Form
Display Error Message
[Invalid Details]
[Valid Details]
Create New Agent or Dropshipper
Add Agent or Dropshipper Information to Email Template
Send Email SMTP
SMTP
SMTP
Receive Email SMTP
Process Email
Send Email
Email
Receive Email
Email

Figure 5.9: Activity Diagram - Manage Agent or Dropshipper

5.5.1.6 Manage Agent and Dropshipper Registrations

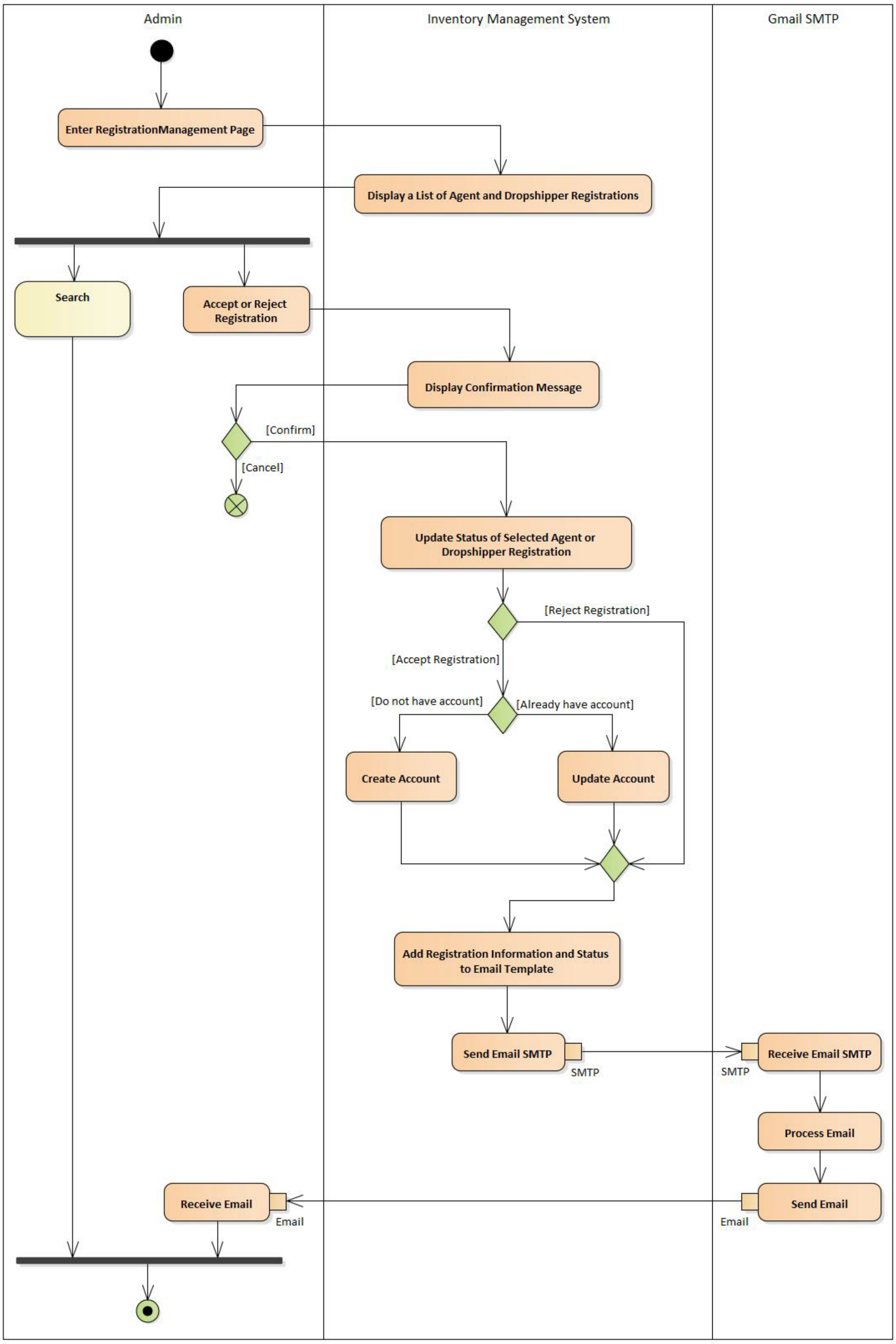

Admin
Inventory Management System
Gmail SMTP
Enter RegistrationManagement Page
Display a List of Agent and Dropshipper Registrations
Search
Accept or Reject Registration
Display Confirmation Message
[Confirm]
[Cancel]
Update Status of Selected Agent or Dropshipper Registration
[Reject Registration]
[Accept Registration]
[Do not have account]
[Already have account]
Create Account
Update Account
Add Registration Information and Status to Email Template
Send Email SMTP
SMTP
SMTP
Receive Email SMTP
Process Email
Send Email
Email
Receive Email
Email

Figure 5.10: Activity Diagram - Manage Agent and Dropshipper Registrations

5.5.1.7 Manage Shipping Fee

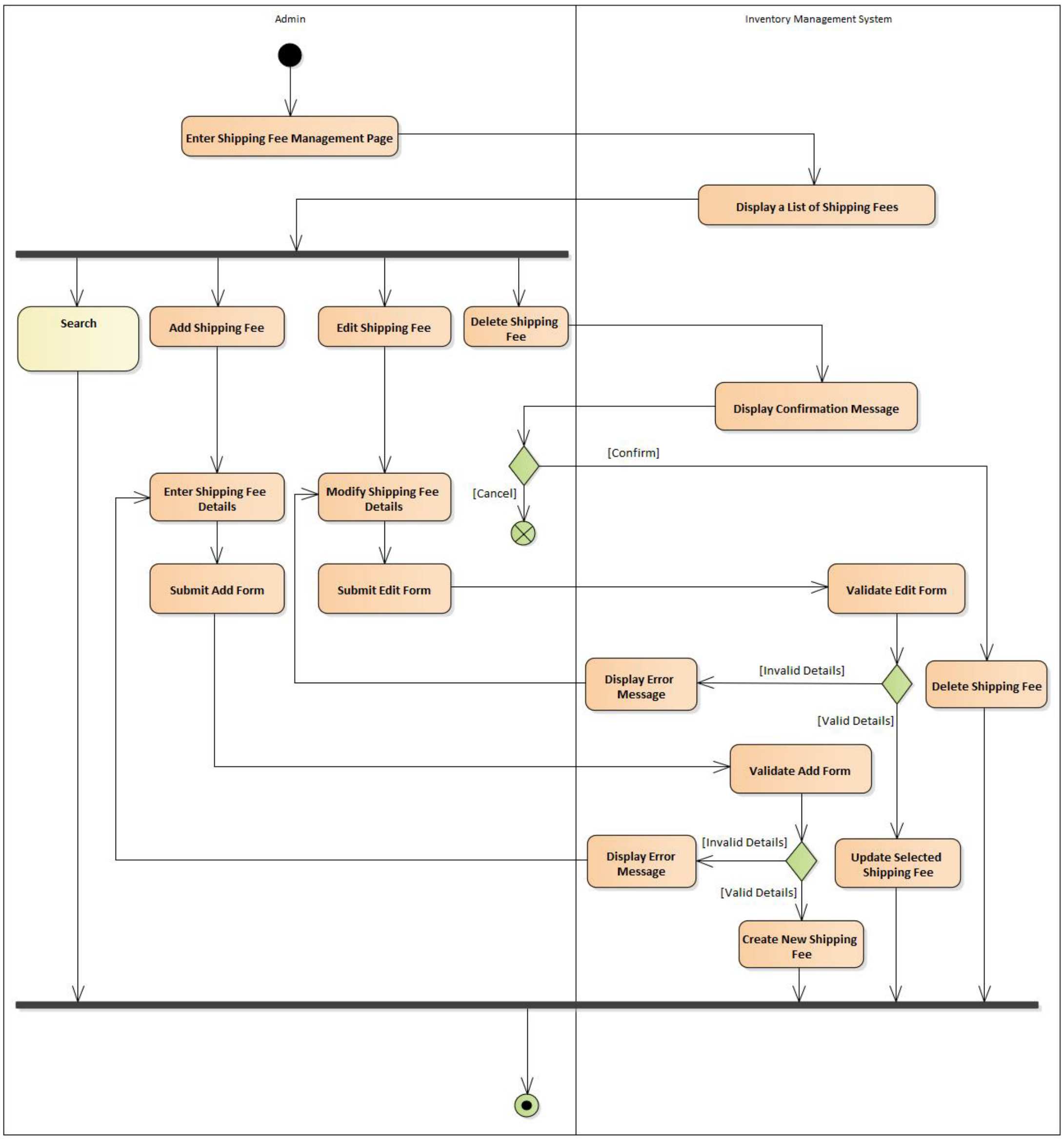


Figure 5.11: Activity Diagram – Manage Shipping Fee

5.5.1.8 Manage Pickup Location

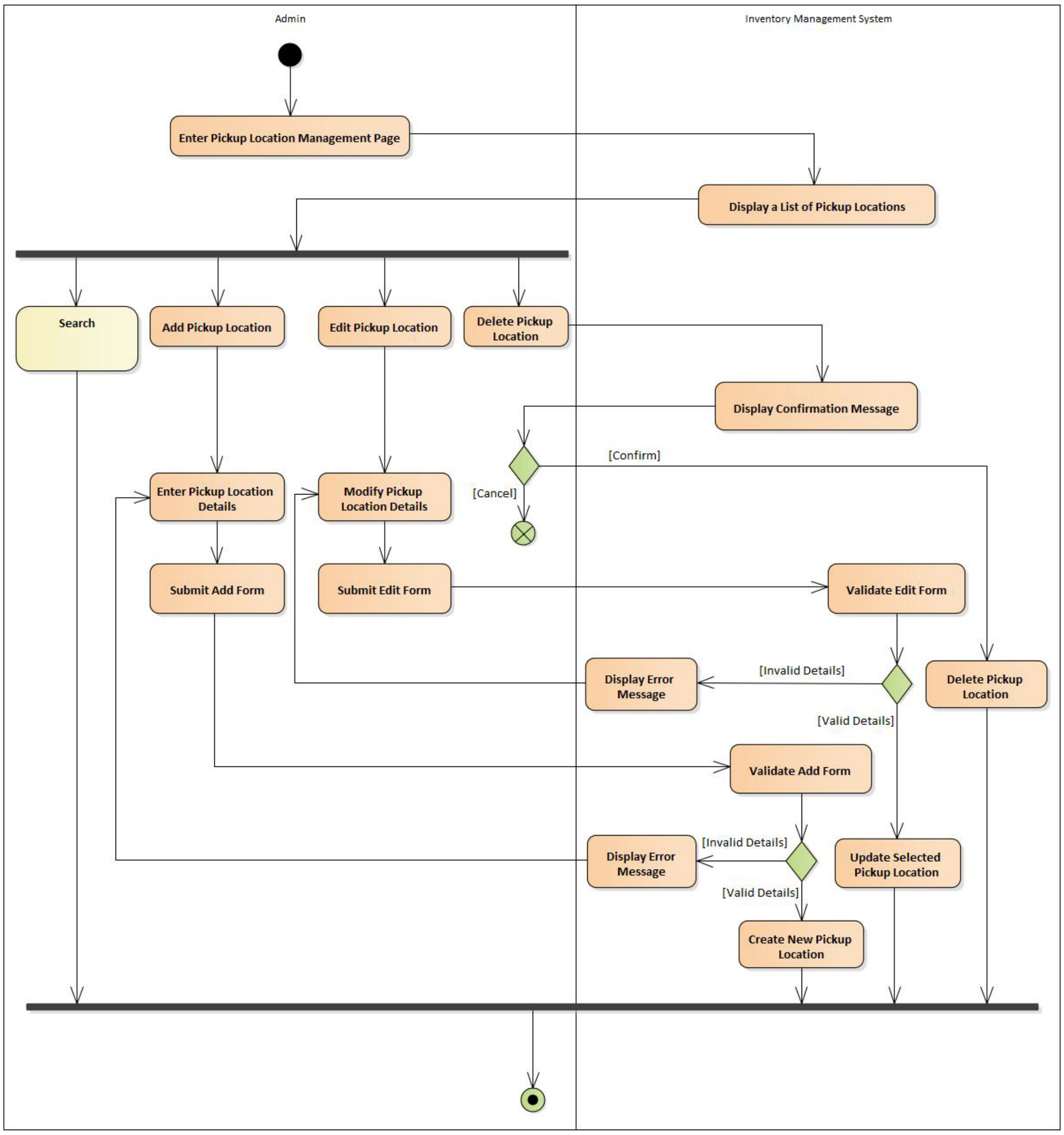


Figure 5.12: Activity Diagram - Manage Pickup Location

5.5.1.9 Manage Discount

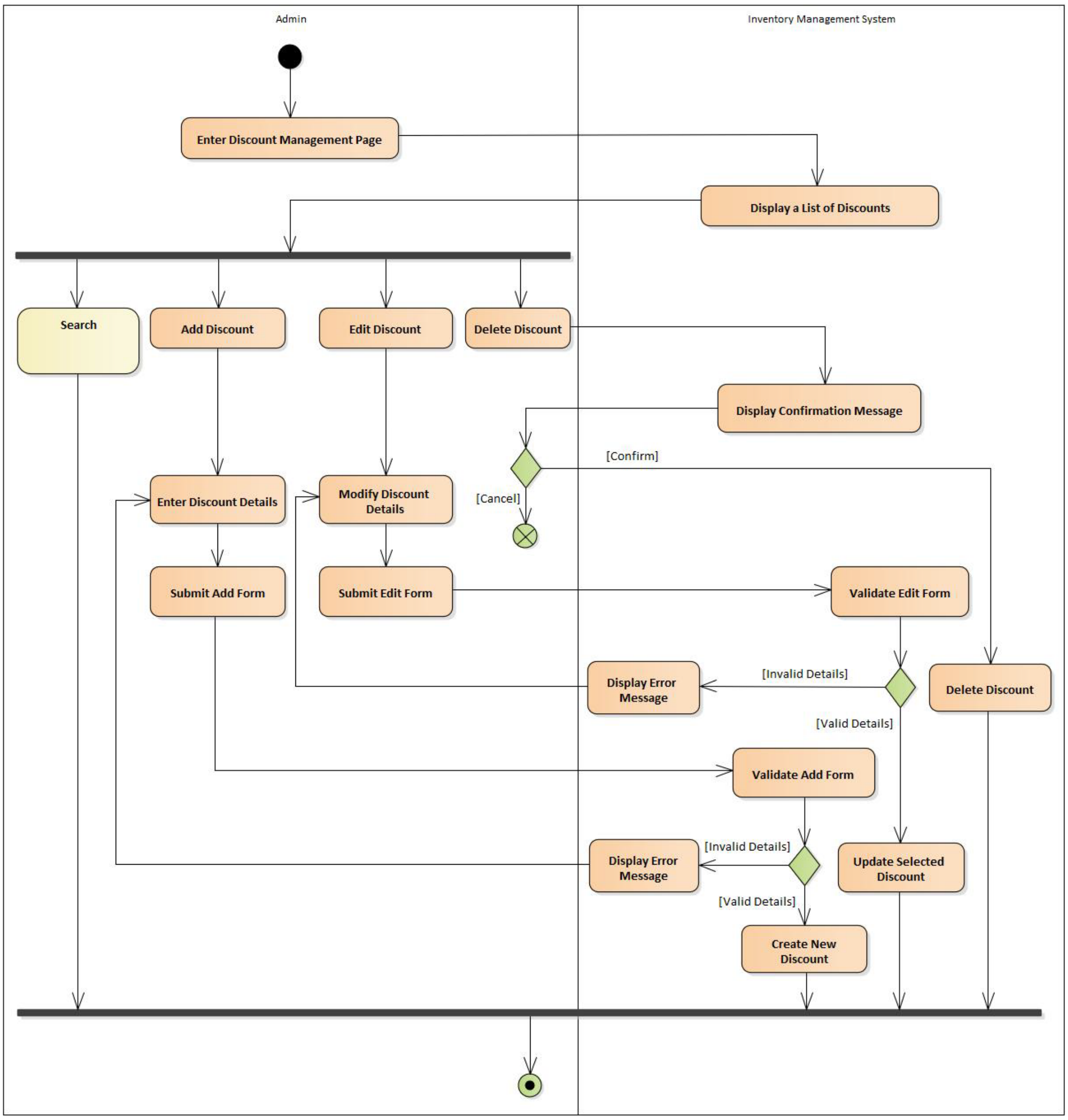


Figure 5.13: Activity Diagram - Manage Discount

5.5.1.10 Manage Order

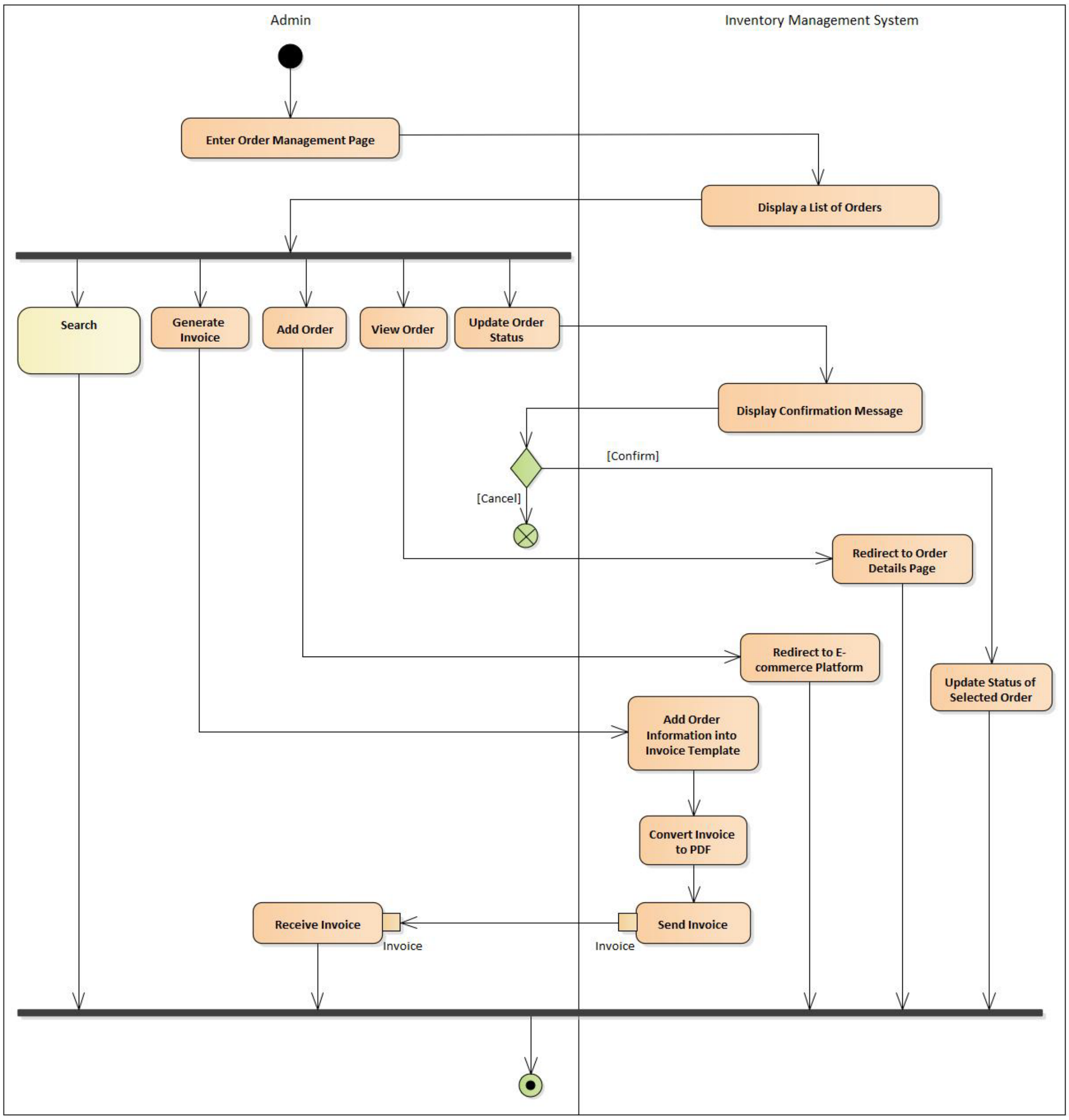

Admin
Inventory Management System
Enter Order Management Page
Display a List of Orders
Search
Generate Invoice
Add Order
View Order
Update Order Status
Display Confirmation Message
[Confirm]
[Cancel]
Redirect to Order Details Page
Redirect to E-commerce Platform
Update Status of Selected Order
Add Order Information into Invoice Template
Convert Invoice to PDF
Receive Invoice
Invoice
Send Invoice
Invoice

Figure 5.14: Activity Diagram - Manage Order

5.5.1.11 View Business Insights

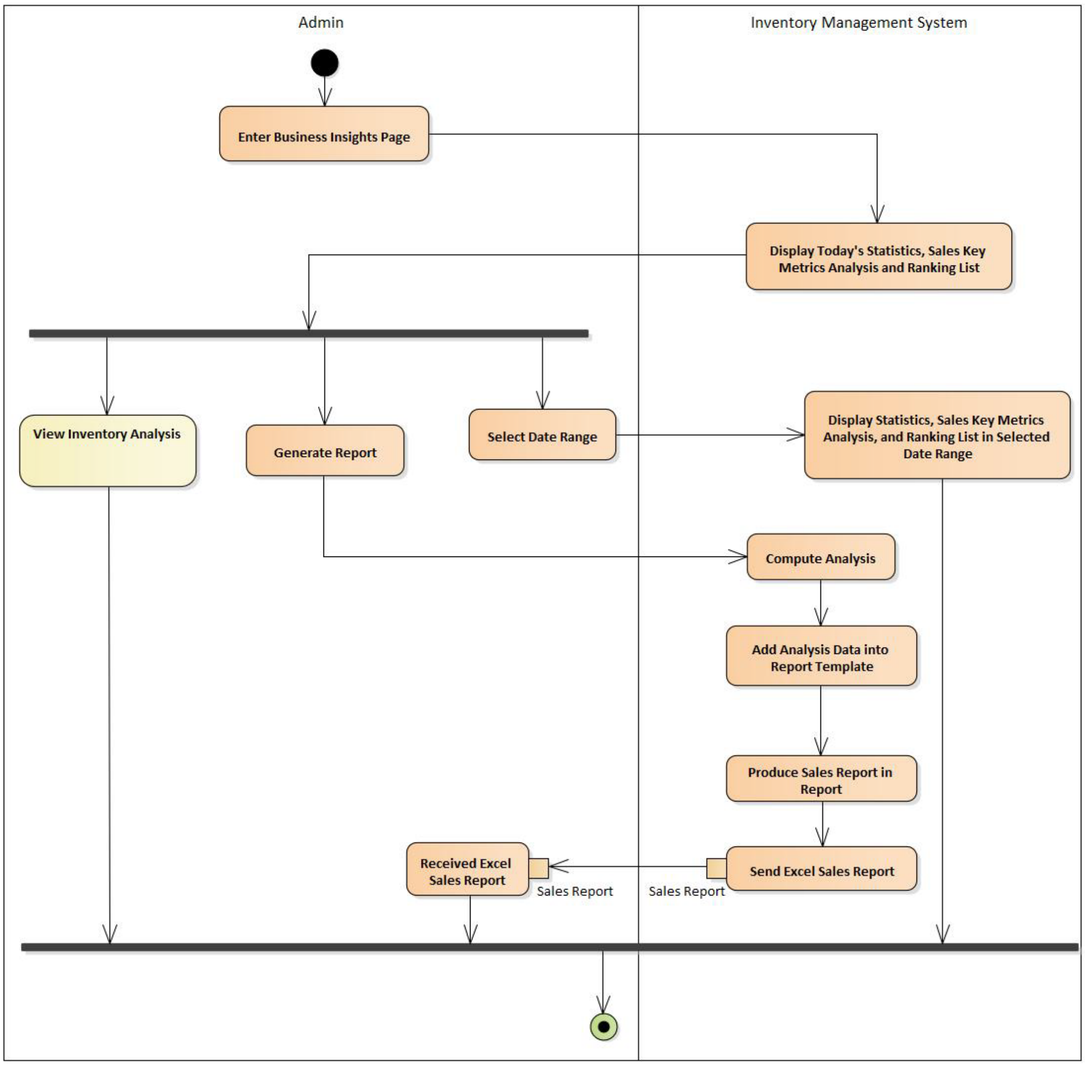

Admin
Inventory Management System
Enter Business Insights Page
Display Today's Statistics, Sales Key Metrics Analysis and Ranking List
View Inventory Analysis
Generate Report
Select Date Range
Display Statistics, Sales Key Metrics Analysis, and Ranking List in Selected Date Range
Compute Analysis
Add Analysis Data into Report Template
Produce Sales Report in Report
Received Excel Sales Report
Sales Report
Sales Report
Send Excel Sales Report

Figure 5.15: Activity Diagram - View Business Insights

5.5.1.12 View Inventory Analysis

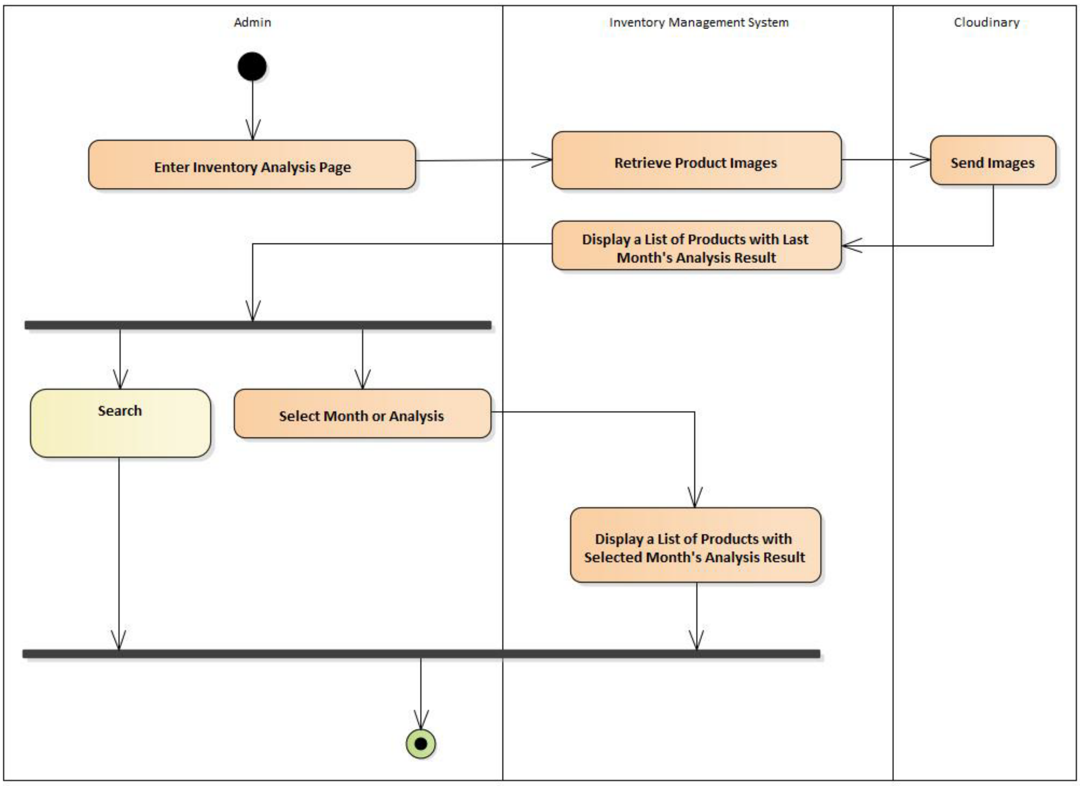

Admin
Inventory Management System
Cloudinary
Enter Inventory Analysis Page
Retrieve Product Images
Send Images
Display a List of Products with Last Month's Analysis Result
Search
Select Month or Analysis
Display a List of Products with Selected Month's Analysis Result

Figure 5.16: Activity Diagram - View Inventory Analysis

5.5.1.13 Search

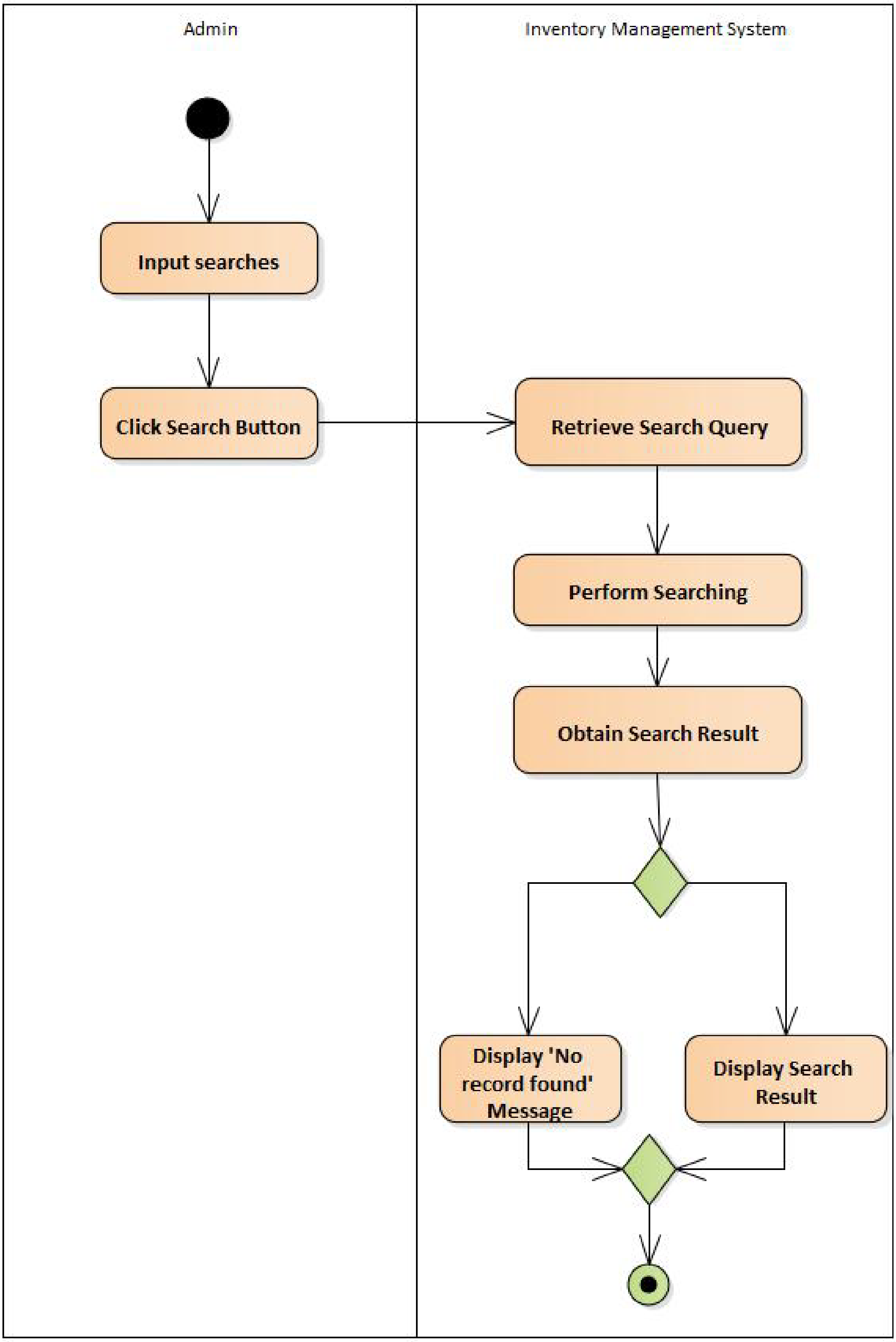


Figure 5.17: Activity Diagram - Search

5.5.2 E-commerce Platform

This section shows the activity diagrams related to the e-commerce platform.

5.5.2.1 Login as User

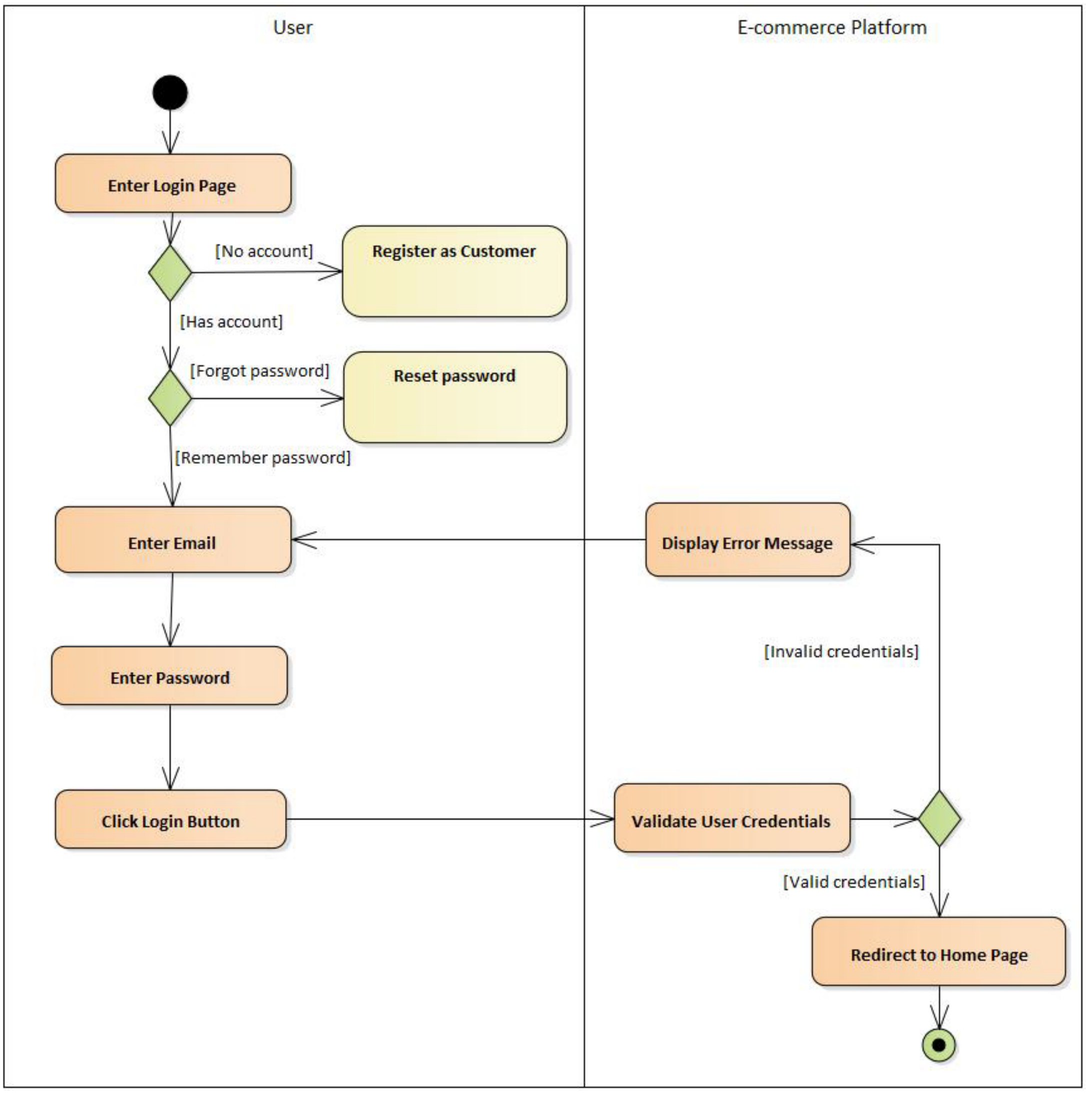


Figure 5.18: Activity Diagram - Login as User

5.5.2.2 Register as Customer

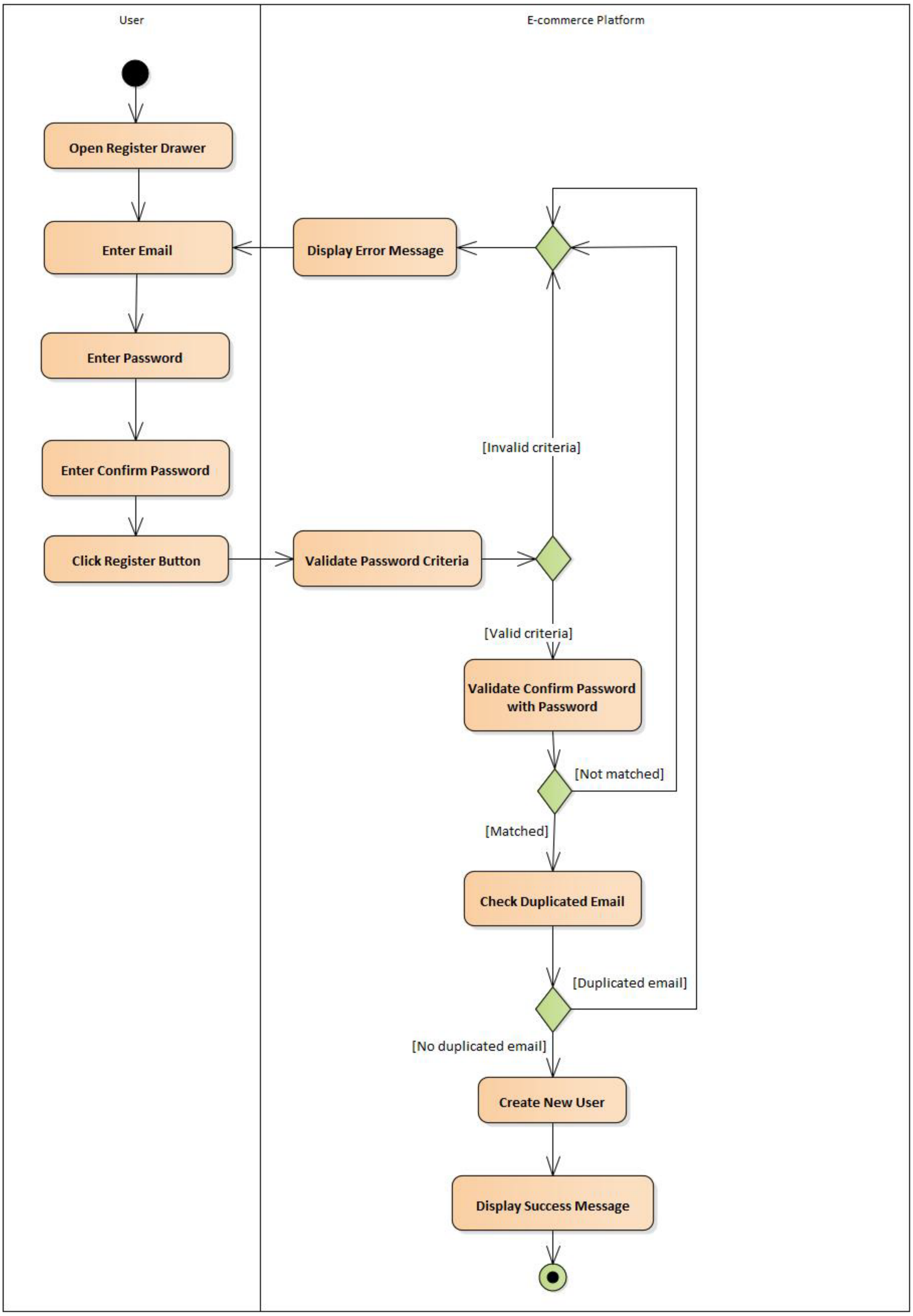


Figure 5.19: Activity Diagram - Register as Customer

5.5.2.3 Register as Agent or Dropshipper

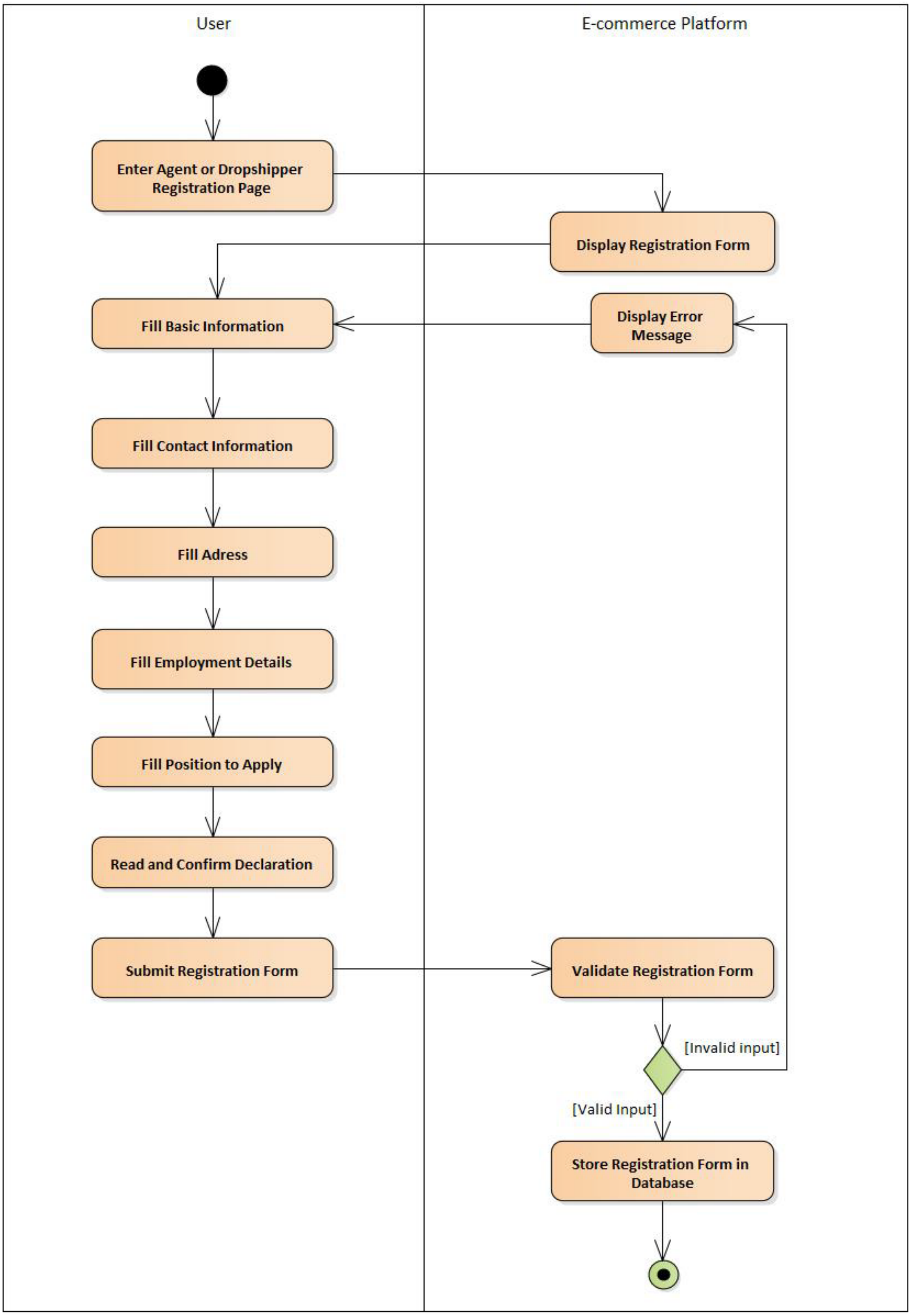


Figure 5.20: Activity Diagram - Register as Agent or Dropshipper

5.5.2.4 Browse Products and Promotions

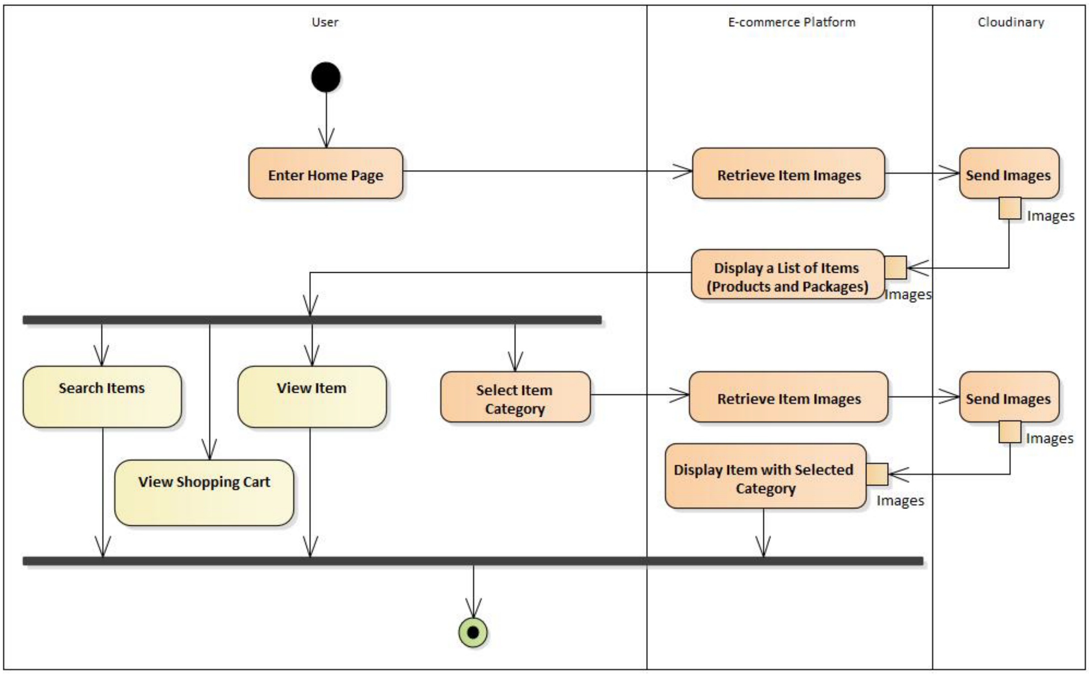


Figure 5.21: Activity Diagram - Browse Products and Promotions

5.5.2.5 Search Items

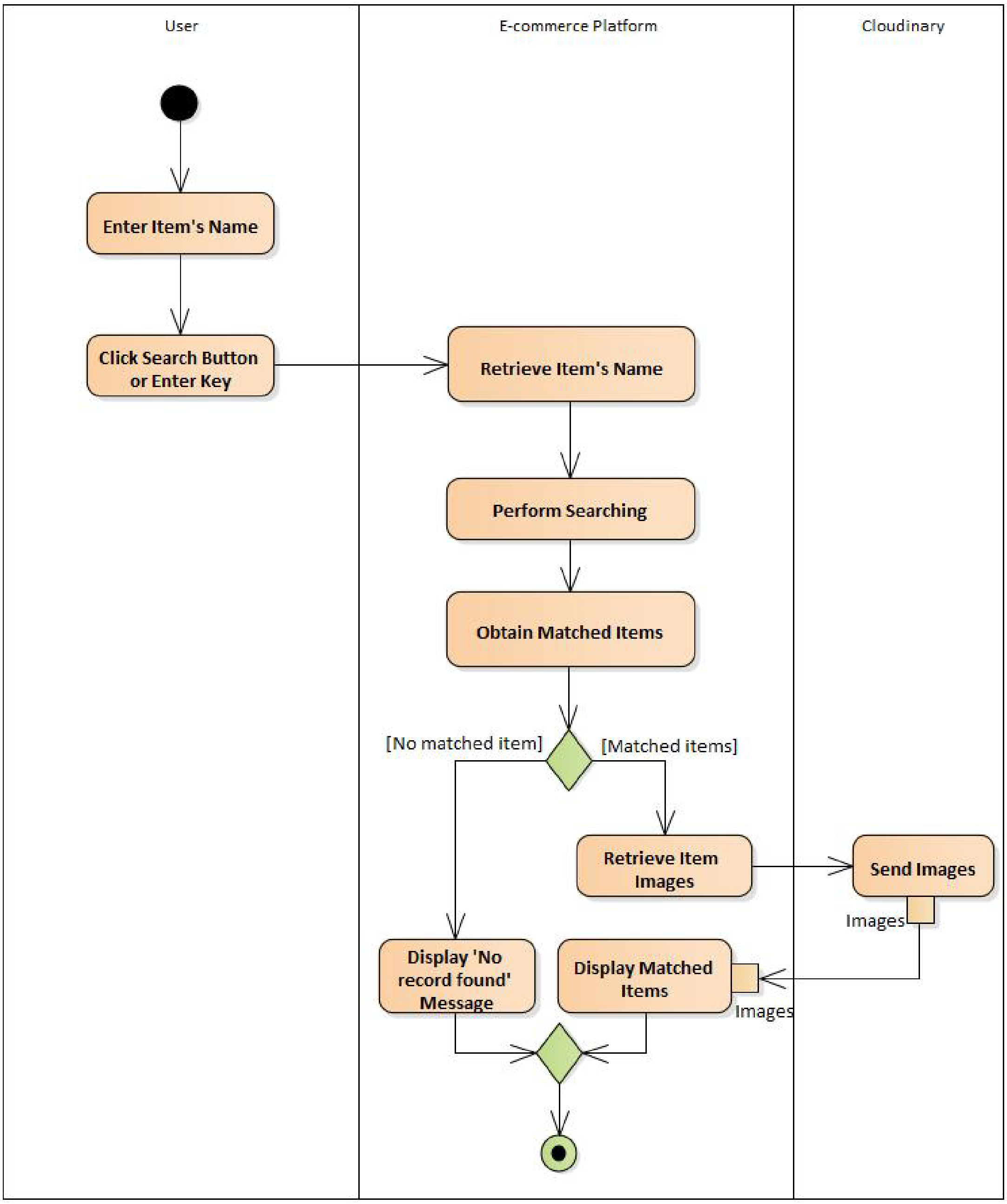


Figure 5.22: Activity Diagram - Search Items

5.5.2.6 View Shopping Cart

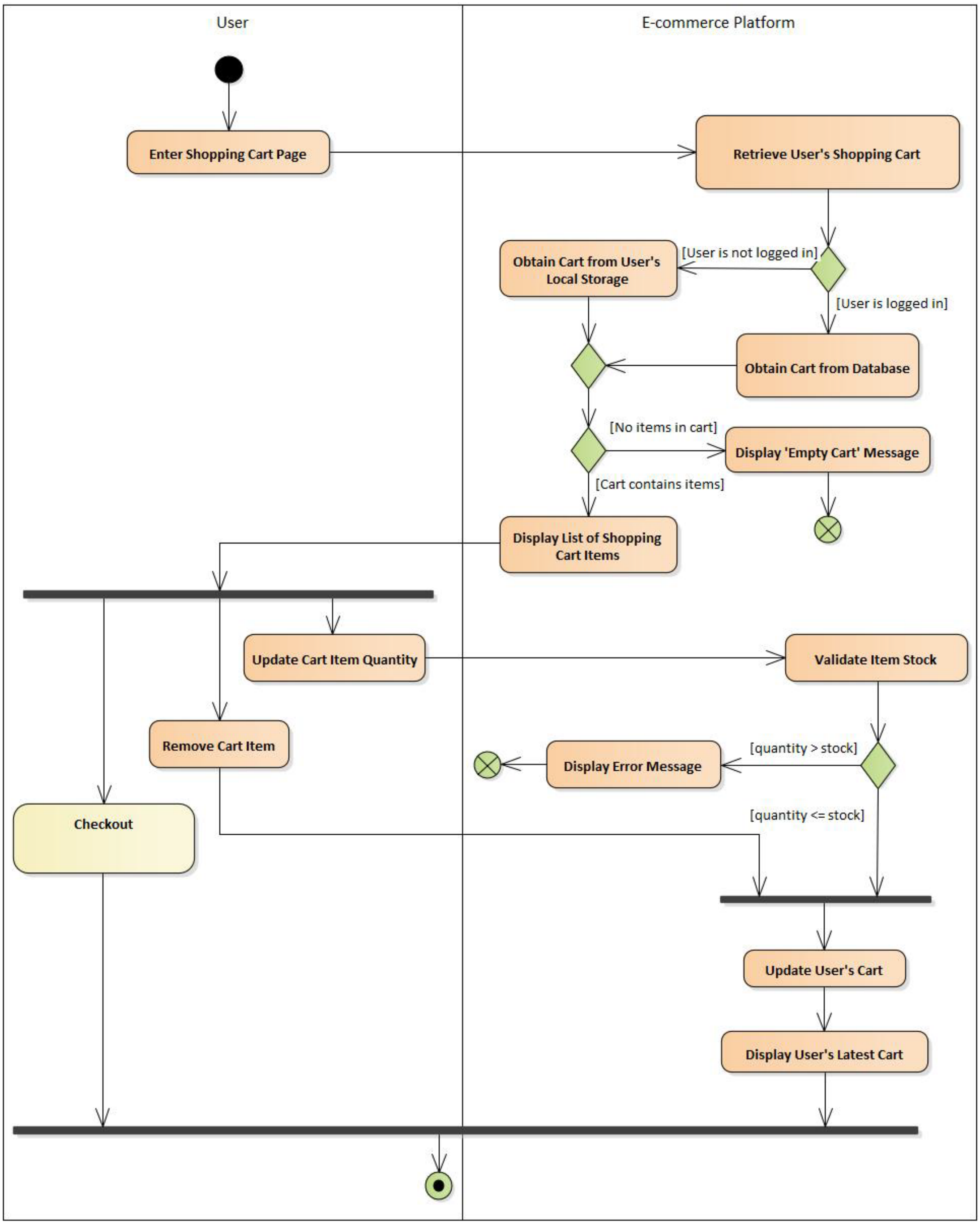


Figure 5.23: Activity Diagram - View Shopping Cart

5.5.2.7 View Item

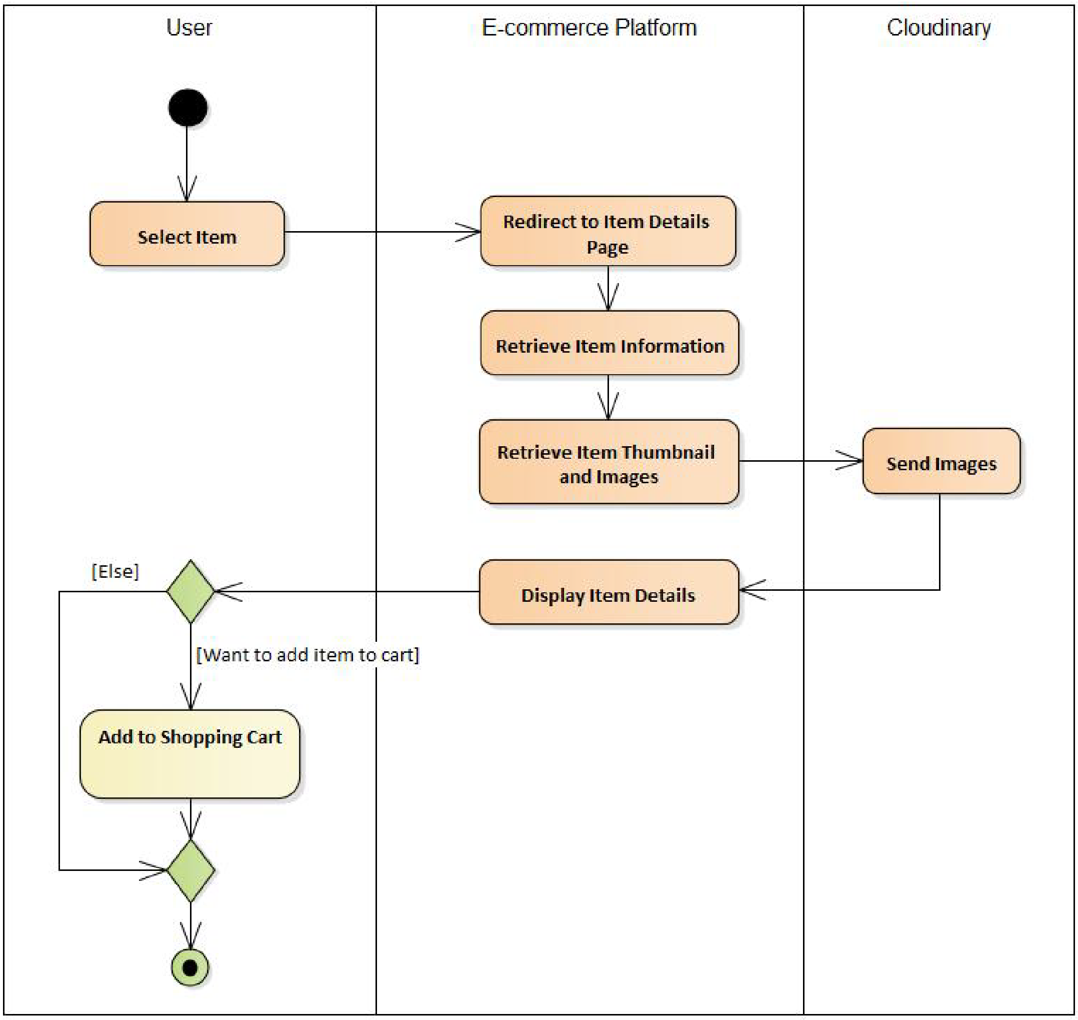

User
E-commerce Platform
Cloudinary
Select Item
Redirect to Item Details Page
Retrieve Item Information
Retrieve Item Thumbnail and Images
Send Images
Display Item Details
[Else]
[Want to add item to cart]
Add to Shopping Cart

Figure 5.24: Activity Diagram - View Item

5.5.2.8 Add to Shopping Cart

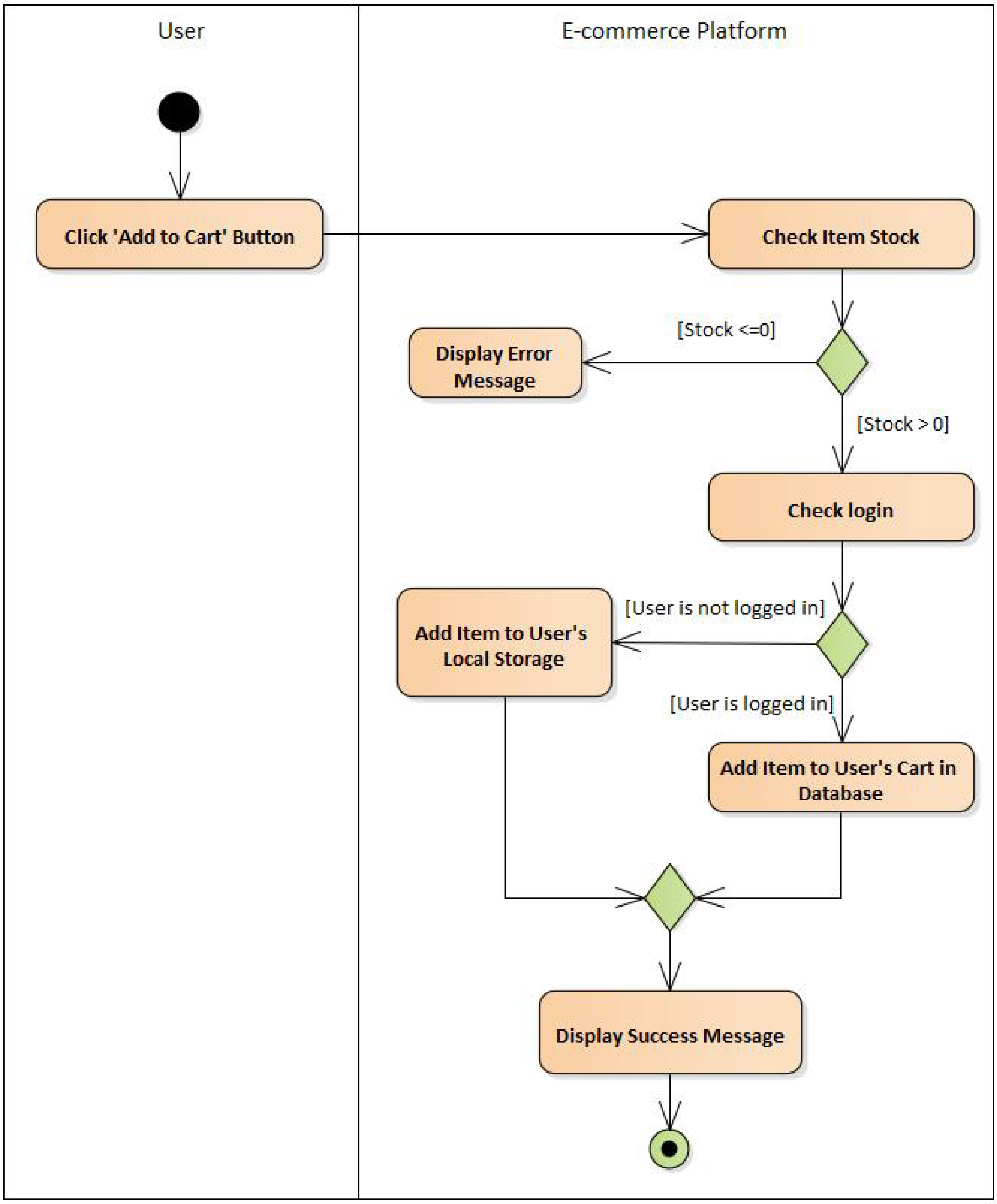


Figure 5.25: Activity Diagram - Add to Shopping Cart

5.5.2.9 Checkout

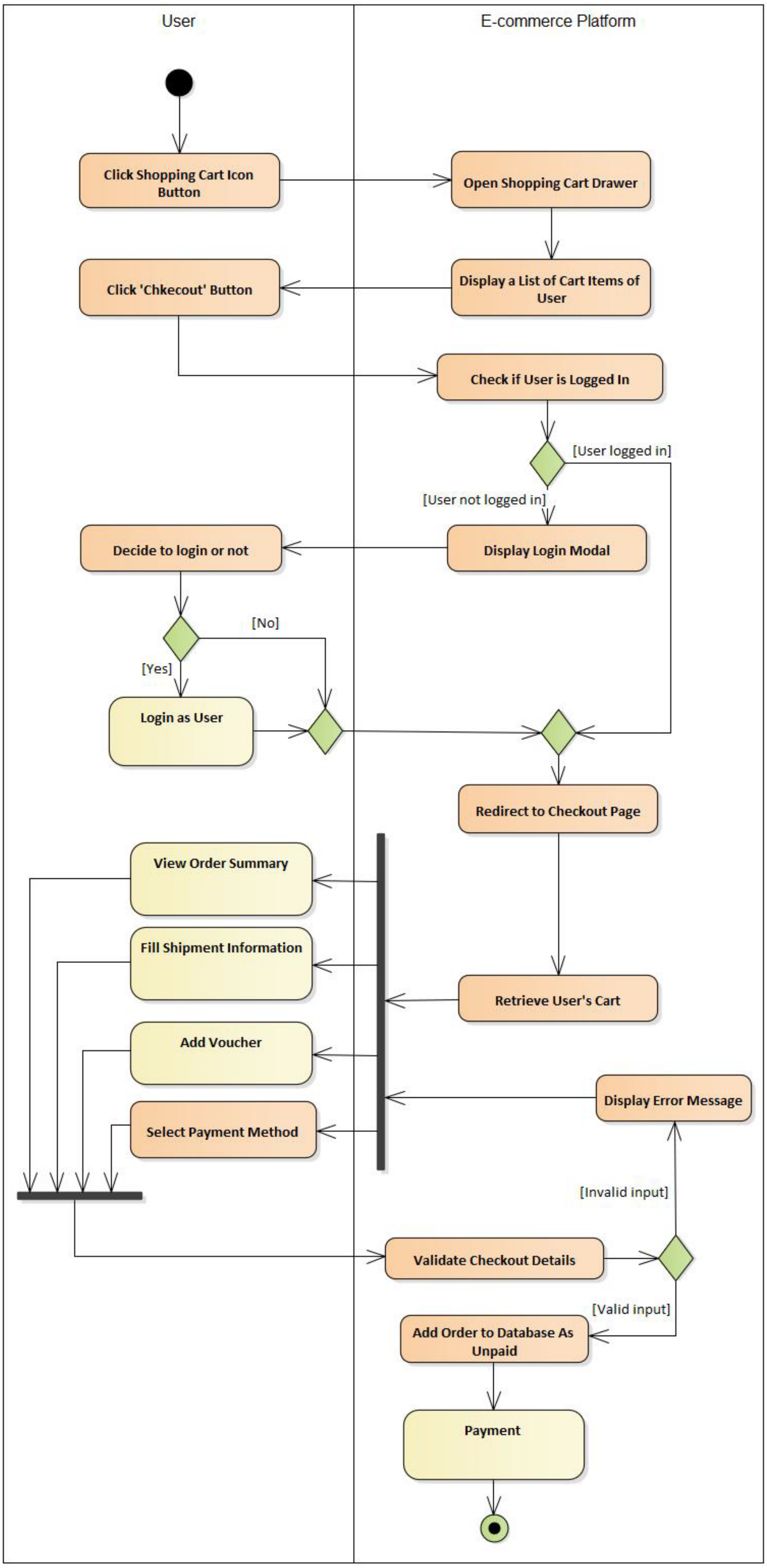


Figure 5.26: Activity Diagram - Checkout

5.5.2.10 View Order Summary

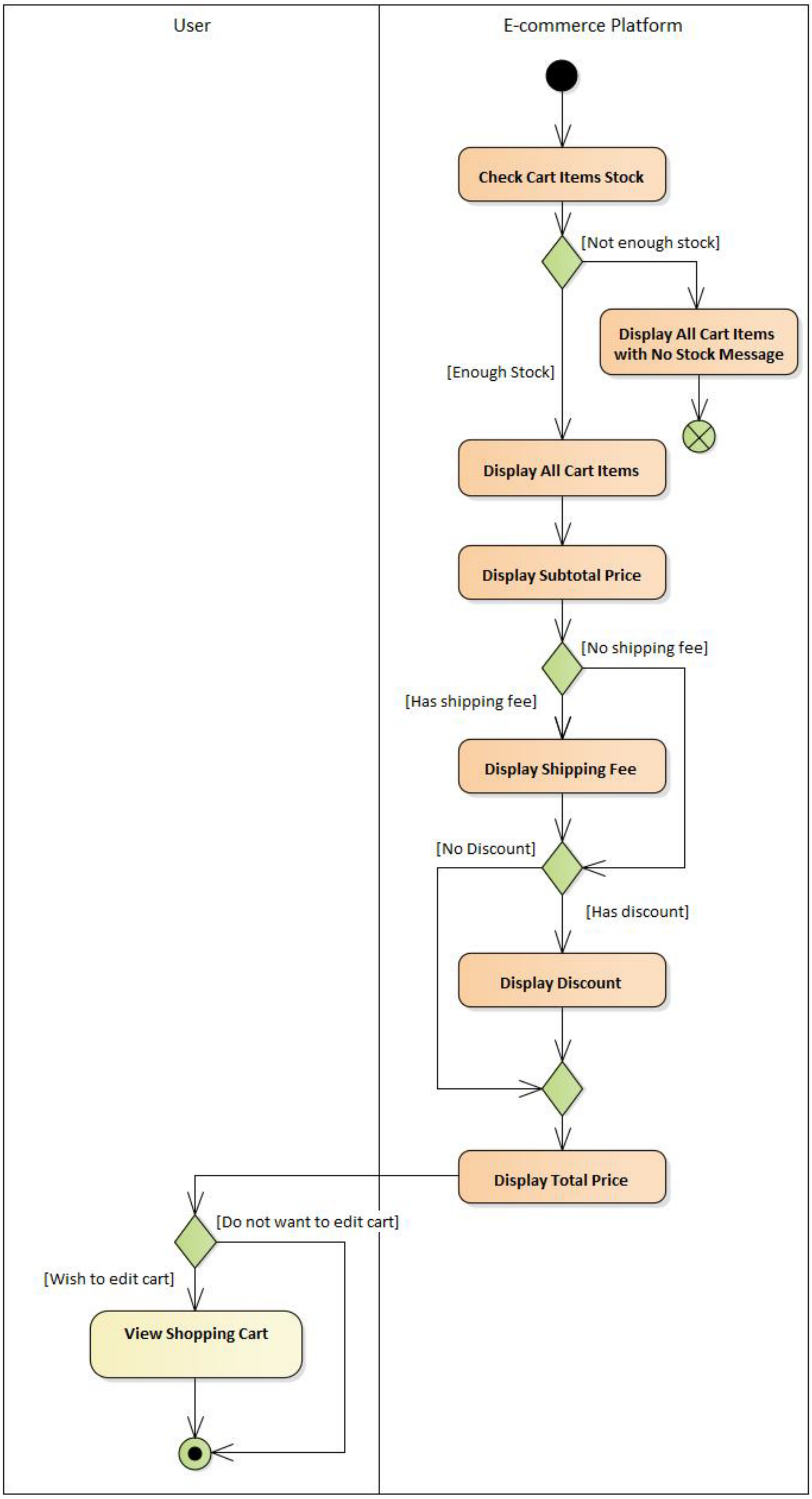


Figure 5.27: Activity Diagram - View Order Summary

5.5.2.11 Fill Shipment Information

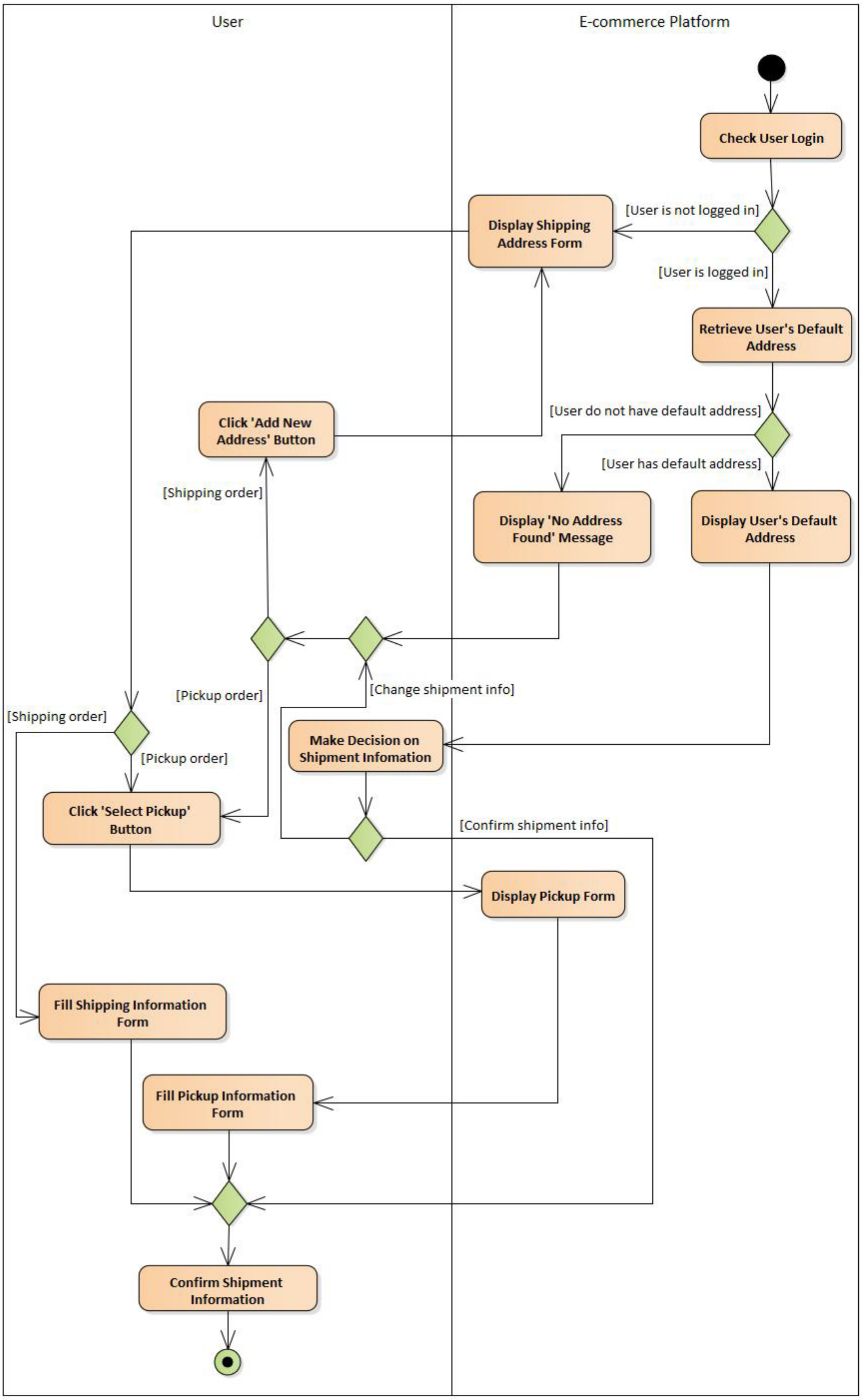

User
E-commerce Platform
Check User Login
[User is not logged in]
Display Shipping Address Form
[User is logged in]
Retrieve User's Default Address
[User do not have default address]
Click 'Add New Address' Button
[User has default address]
Display 'No Address Found' Message
Display User's Default Address
[Shipping order]
[Pickup order]
[Change shipment info]
[Shipping order]
[Pickup order]
Make Decision on Shipment Infomation
Click 'Select Pickup' Button
[Confirm shipment info]
Display Pickup Form
Fill Shipping Information Form
Fill Pickup Information Form
Confirm Shipment Information

Figure 5.28: Activity Diagram - Fill Shipment Information

5.5.2.12 Add Voucher

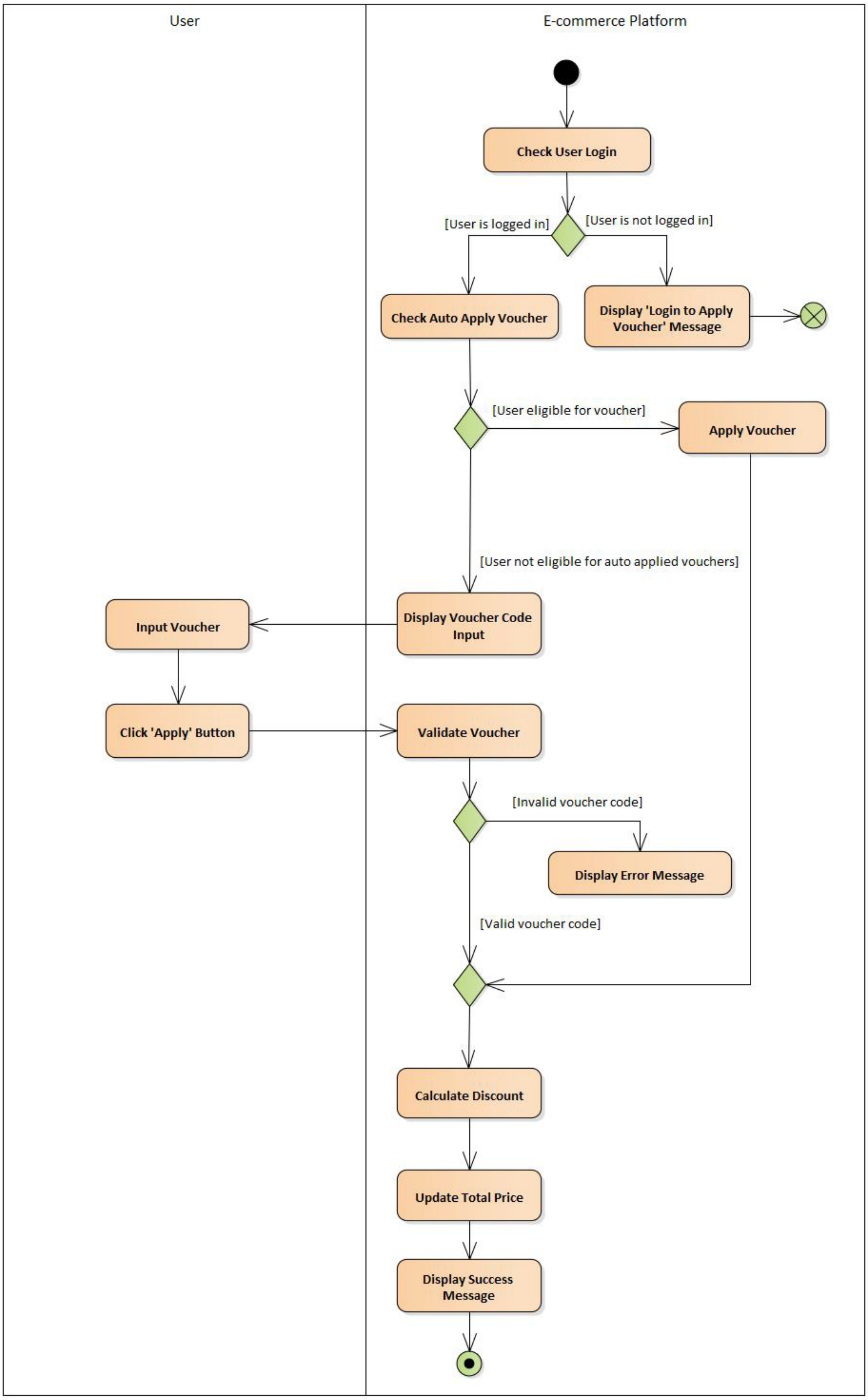


Figure 5.29: Activity Diagram - Add Voucher

5.5.2.13 Payment

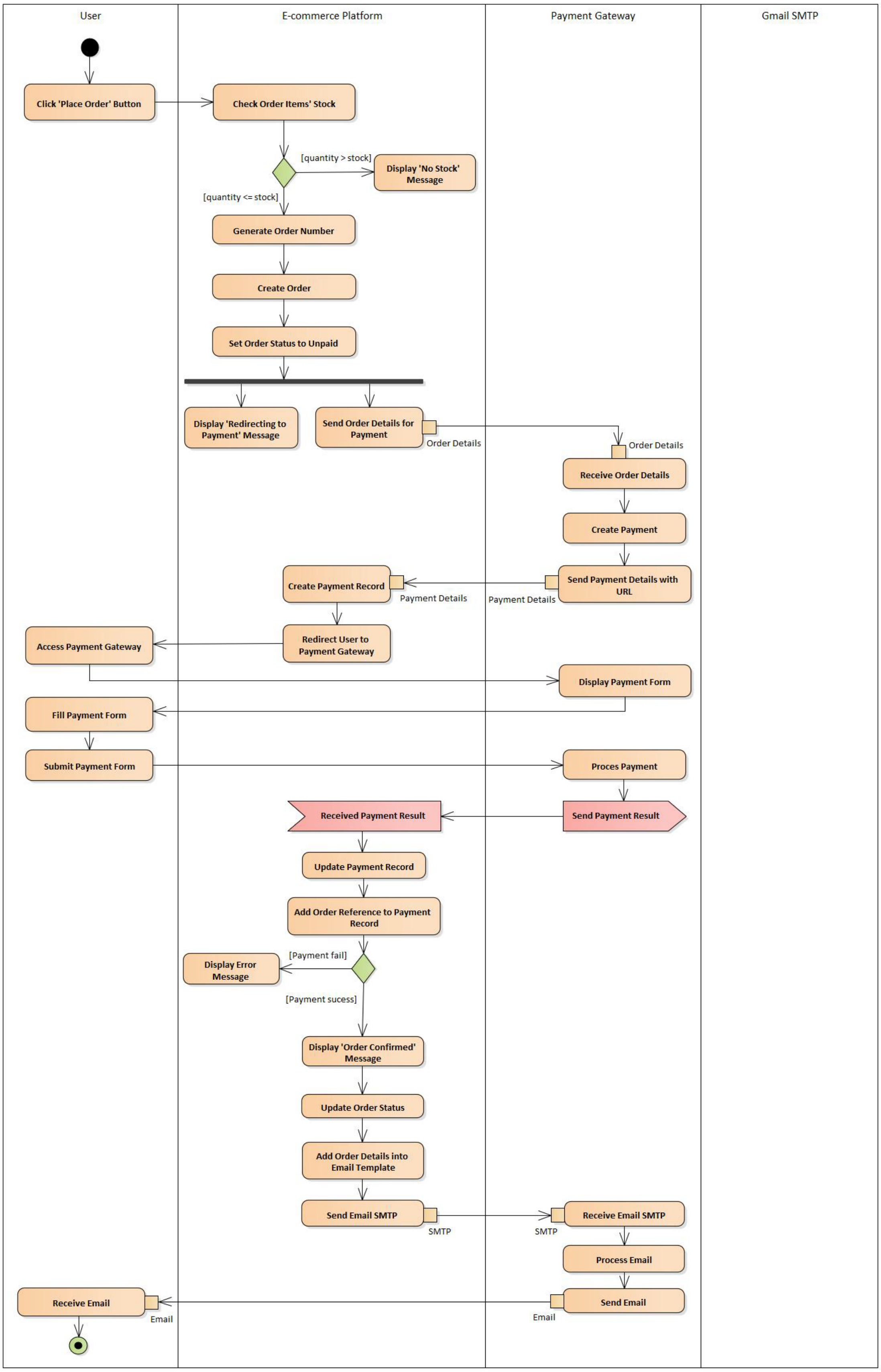


Figure 5.30: Activity Diagram - Payment

5.5.2.14 Search Order

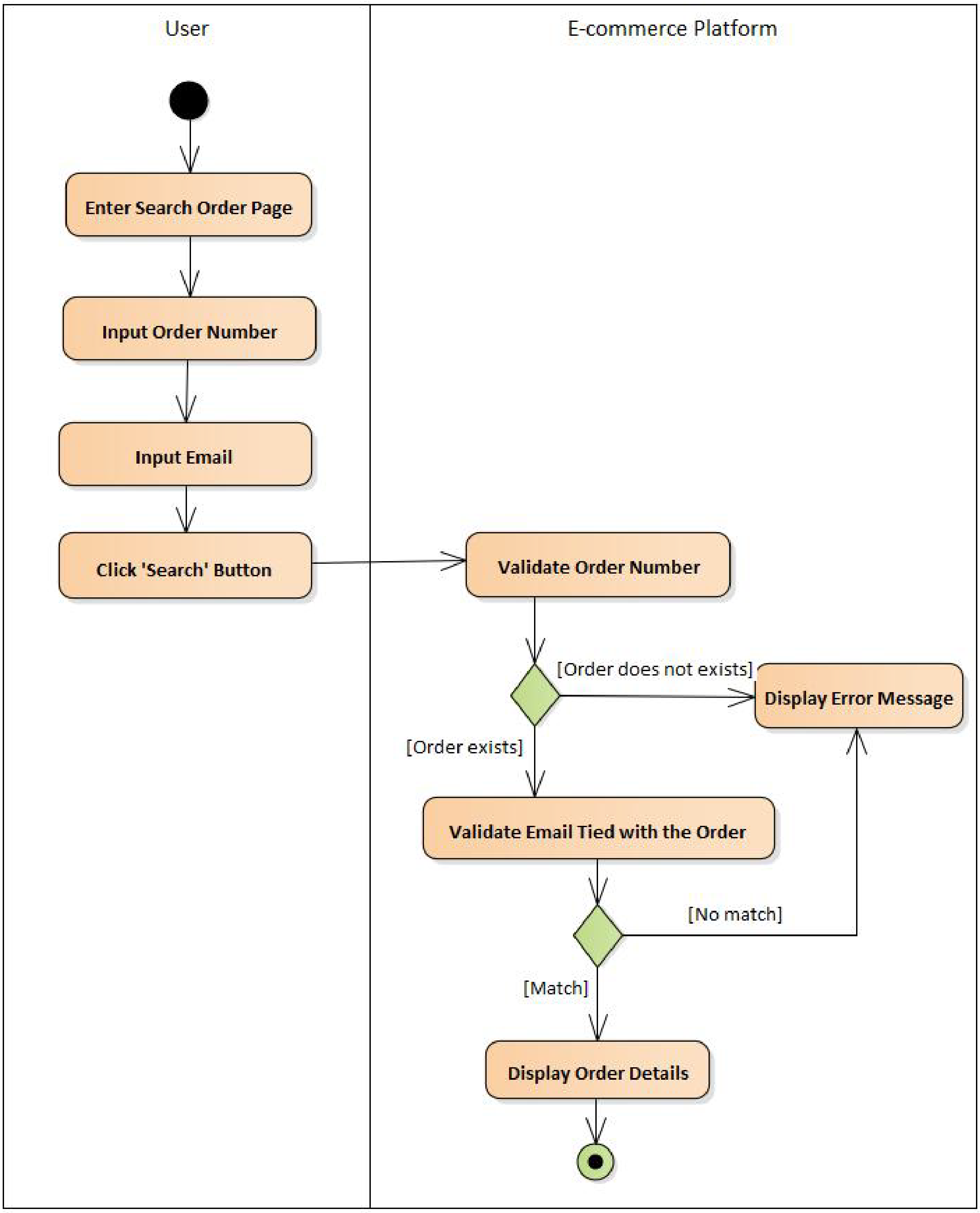


Figure 5.31: Activity Diagram - Search Order

5.5.2.15 Profile Management

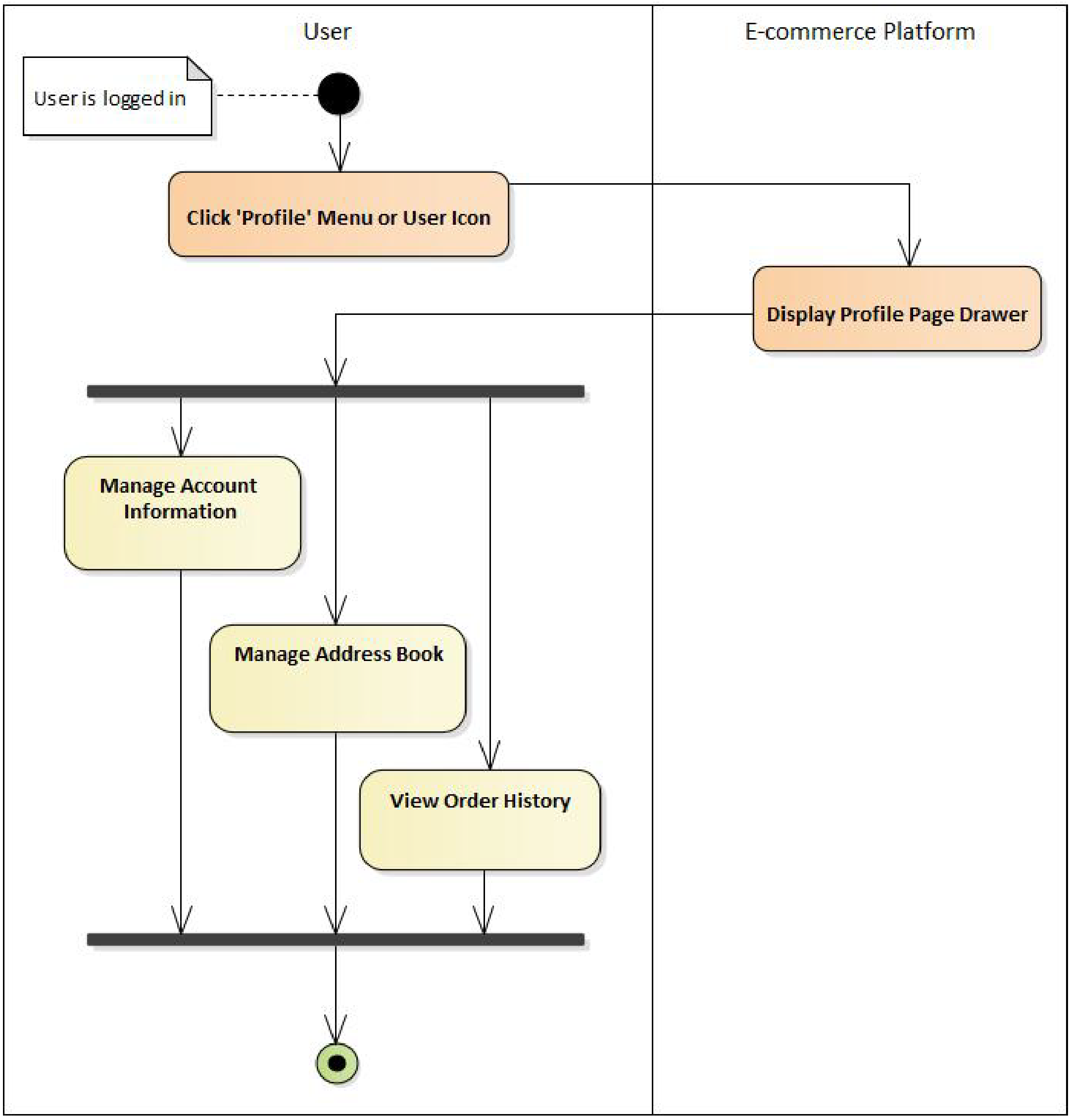

User
E-commerce Platform
User is logged in
Click 'Profile' Menu or User Icon
Display Profile Page Drawer
Manage Account Information
Manage Address Book
View Order History

Figure 5.32: Activity Diagram - Profile Management

5.5.2.16 Manage Account Information

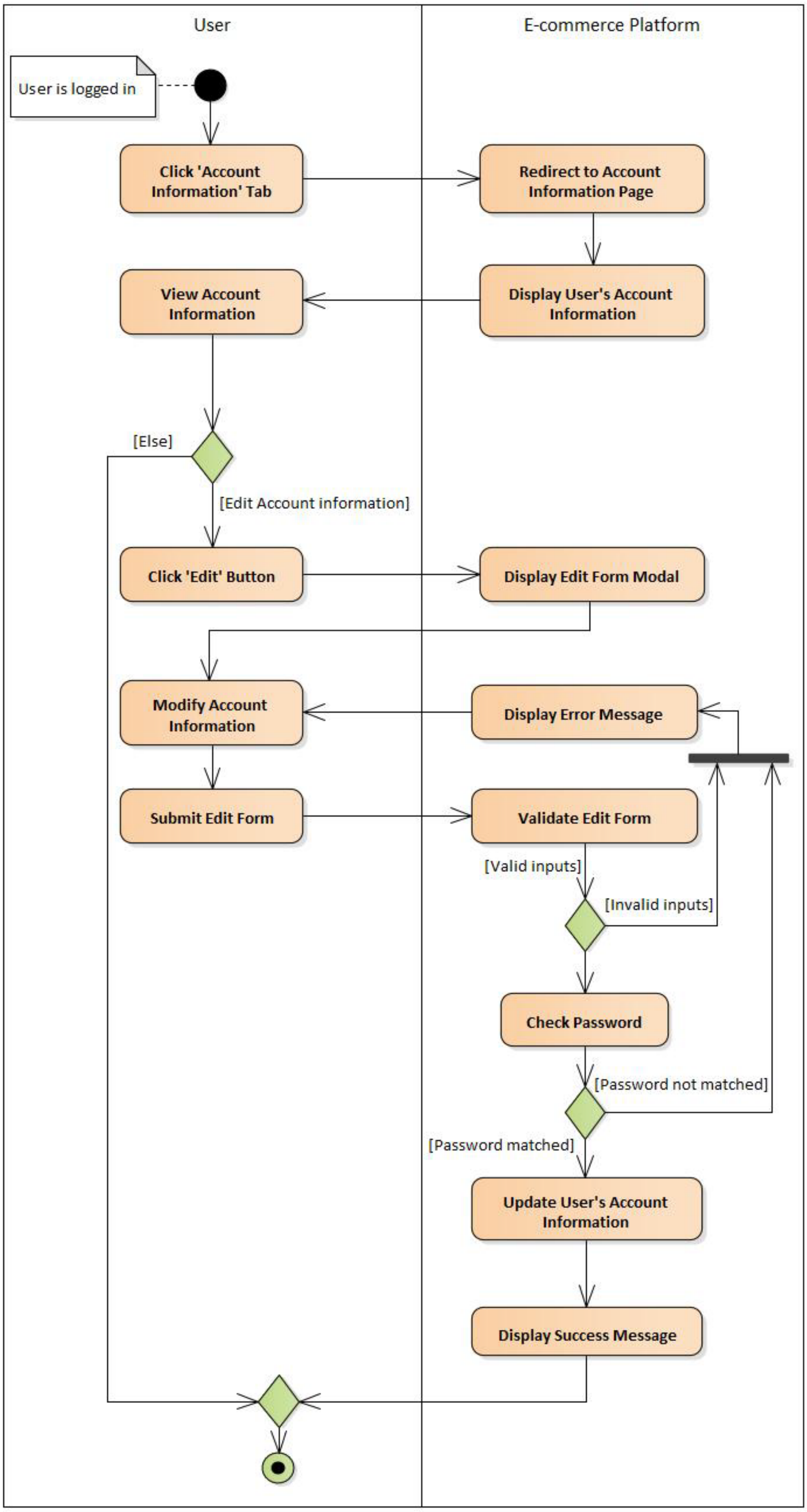


Figure 5.33: Activity Diagram - View or Edit Account Information

5.5.2.17 Manage Address Book

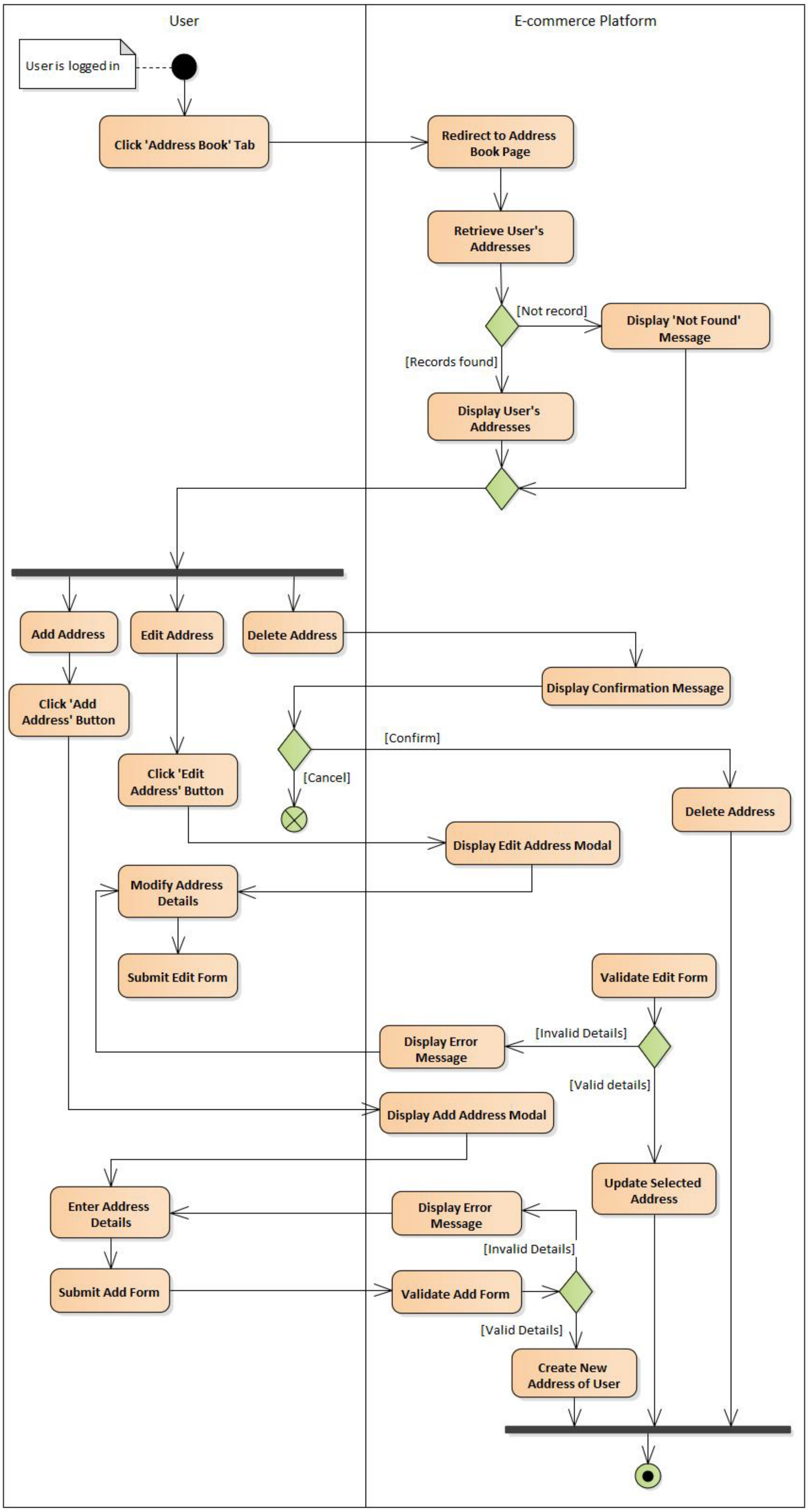


Figure 5.34: Activity Diagram - Manage Address

5.5.2.18 View Order History

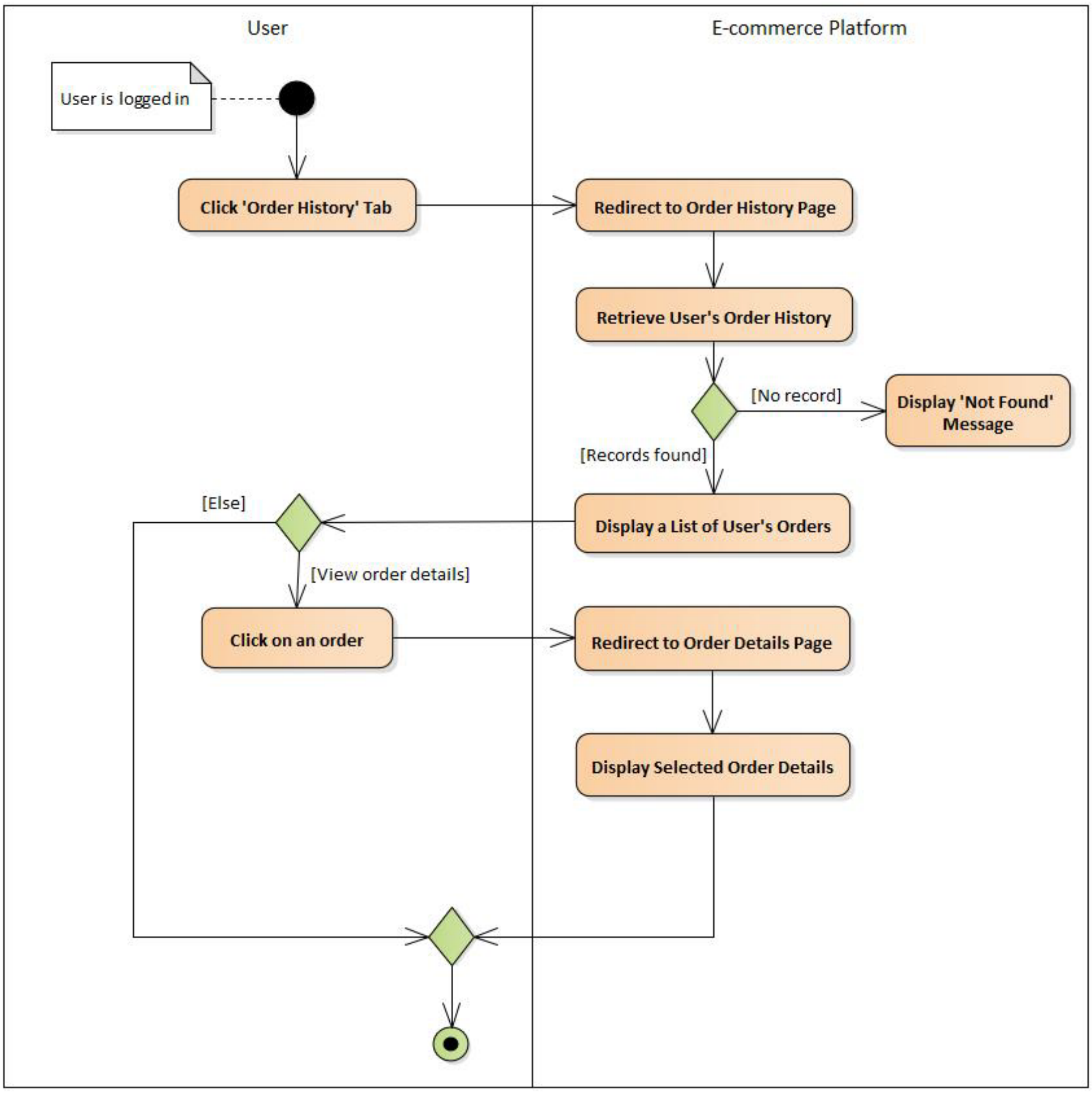

User
E-commerce Platform
User is logged in
Click 'Order History' Tab
Redirect to Order History Page
Retrieve User's Order History
[No record]
Display 'Not Found' Message
[Records found]
Display a List of User's Orders
[Else]
[View order details]
Click on an order
Redirect to Order Details Page
Display Selected Order Details

Figure 5.35: Activity Diagram – View Order History

5.6 Screen Prototype Design

The prototype is designed to simulate the real system, providing additional insights into how the system will like. This section will discuss the prototype developed for the inventory management system. As the prototype is to simulate the real system, the real design of the system after implementation might vary from the prototype design.

5.6.1 Inventory Management System Prototype

5.6.1.1 Login Page

The admins will log into the inventory management system on the login page. To log into the system, the admins must enter their username and password and click the 'Login' button. Besides, the admin can enter the e-commerce platform by clicking the 'Go Back to Sharifah E-commerce' button. Figure 5.36 shows the login page UI.

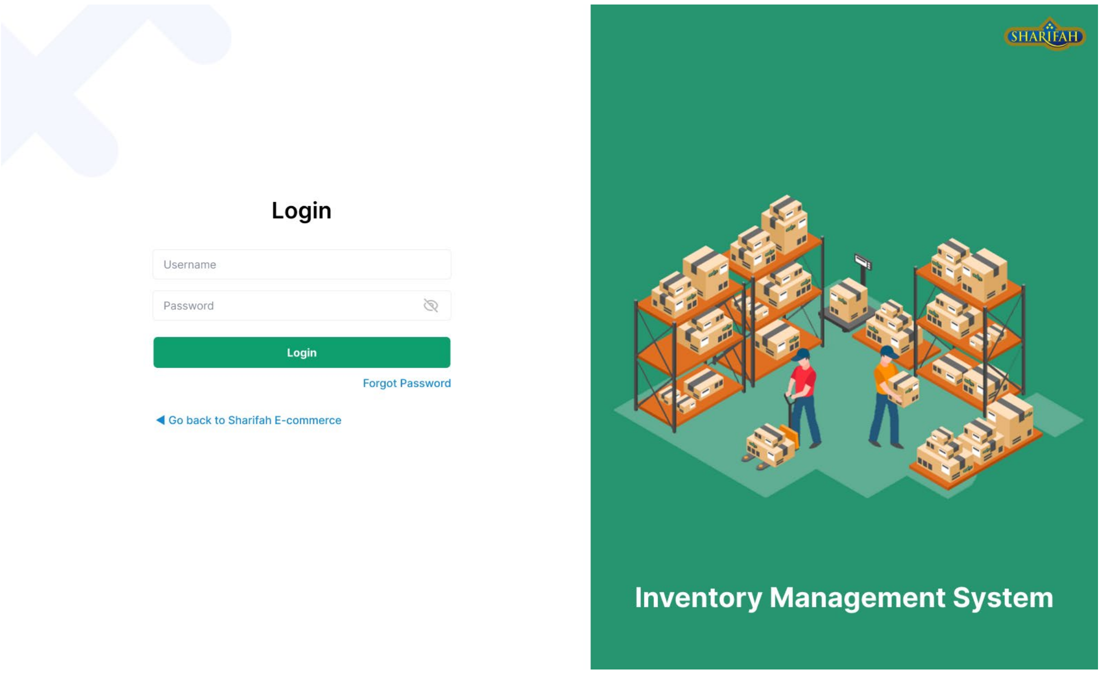
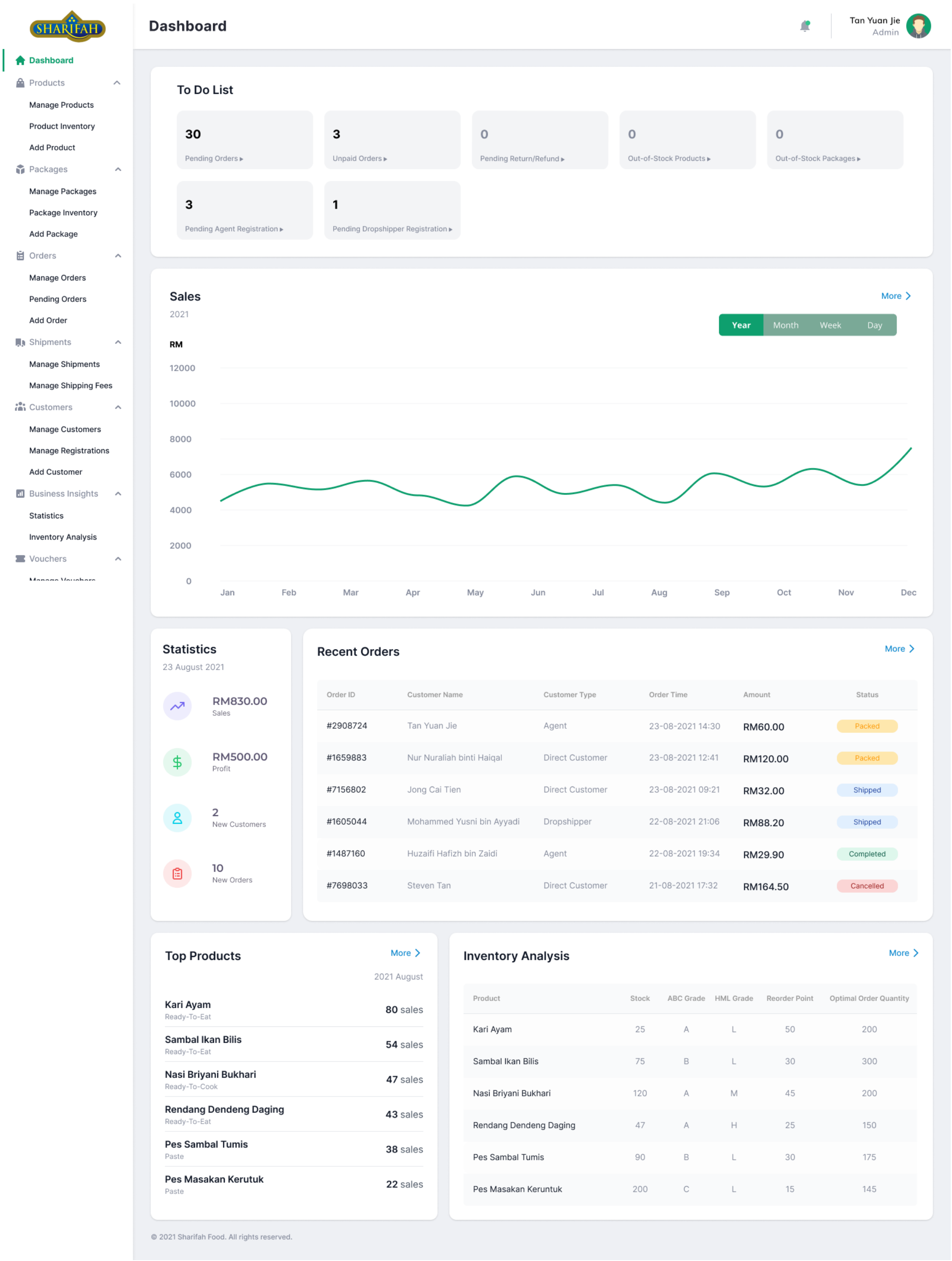


Figure 5.36: Login Page Prototype

5.6.1.2 Dashboard Page

Once the admin has logged into the system, the dashboard page will be displayed. The admin can view the to-do list on the dashboard page, which tells the admin about the pending item to be reviewed. Besides, the sales analysis will also be displayed to the admin. The admin can choose to view the sales analysis in the current year, month, week, or day. There are also today's statistics on the sales, profit, number of customers, and number of new orders. Moreover, the admin can also have some insights into recent orders, top products, and inventory analysis. Figure 5.37 shows the dashboard page.



Dashboard

To Do List

30 Pending Orders ▸

3 Unpaid Orders ▸

0 Pending Return/Refund ▸

0 Out-of-Stock Products ▸

0 Out-of-Stock Packages ▸

3 Pending Agent Registration ▸

1 Pending Dropshipper Registration ▸

Sales

2021

Statistics

23 August 2021

RM830.00 Sales

RM500.00 Profit

2 New Customers

10 New Orders

Recent Orders

Order ID	Customer Name	Customer Type	Order Time	Amount	Status
#2908724	Tan Yuan Jie	Agent	23-08-2021 14:30	RM60.00	Packed
#1659883	Nur Nuraliah binti Haiqal	Direct Customer	23-08-2021 12:41	RM120.00	Packed
#7156802	Jong Cai Tien	Direct Customer	23-08-2021 09:21	RM32.00	Shipped
#1605044	Mohammed Yusni bin Ayyadi	Dropshipper	22-08-2021 21:06	RM88.20	Shipped
#1487160	Huzaifi Hafizh bin Zaidi	Agent	22-08-2021 19:34	RM29.90	Completed
#7698033	Steven Tan	Direct Customer	21-08-2021 17:32	RM164.50	Cancelled

Top Products

2021 August

Kari Ayam Ready-To-Eat	80 sales
Sambal Ikan Bilis Ready-To-Eat	54 sales
Nasi Briyani Bukhari Ready-To-Cook	47 sales
Rendang Dendeng Daging Ready-To-Eat	43 sales
Pes Sambal Tumis Paste	38 sales
Pes Masakan Kerutuk Paste	22 sales

Inventory Analysis

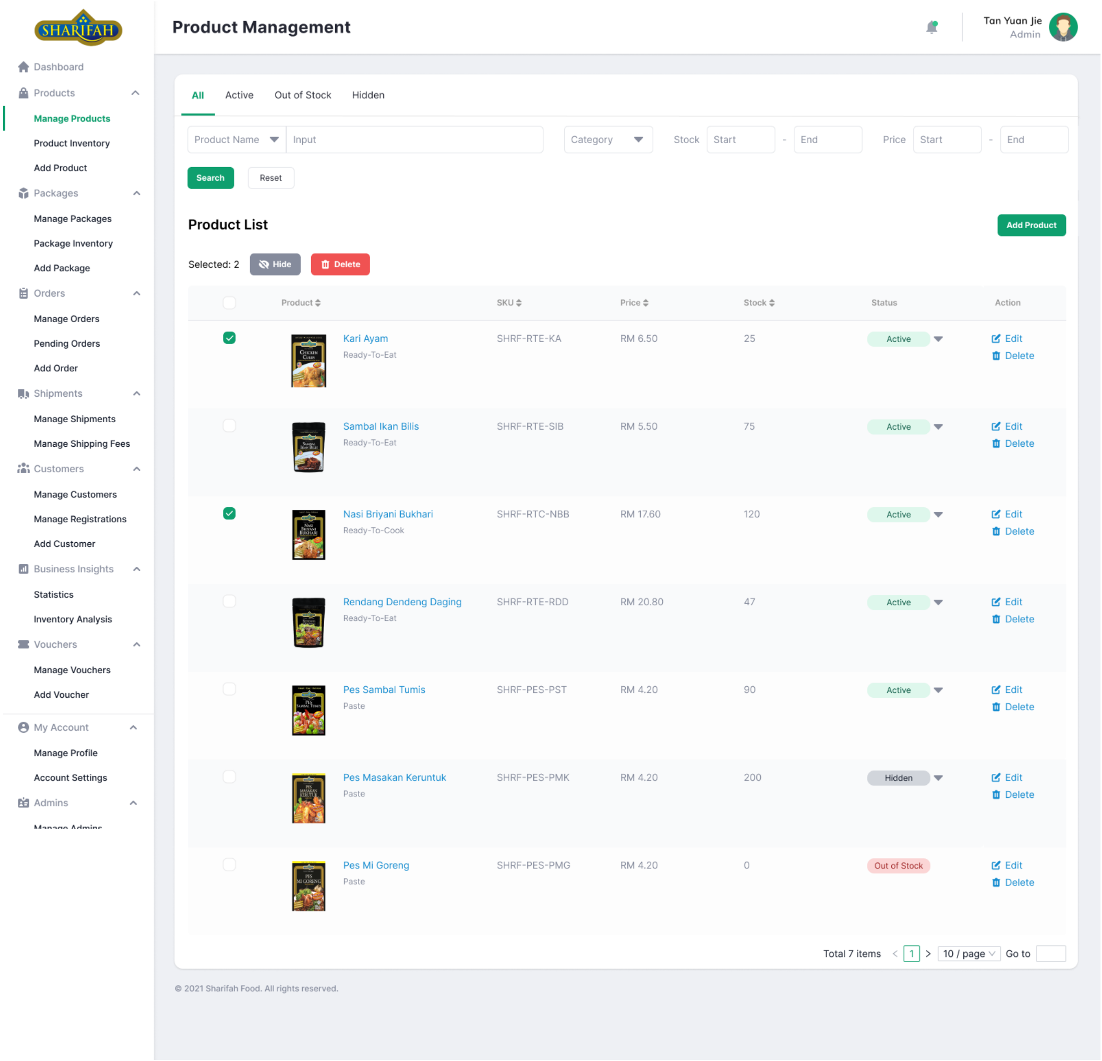
Product	Stock	ABC Grade	HML Grade	Reorder Point	Optimal Order Quantity
Kari Ayam	25	A	L	50	200
Sambal Ikan Bilis	75	B	L	30	300
Nasi Briyani Bukhari	120	A	M	45	200
Rendang Dendeng Daging	47	A	H	25	150
Pes Sambal Tumis	90	B	L	30	175
Pes Masakan Keruntuk	200	C	L	15	145

© 2021 Sharifah Food. All rights reserved.

Figure 5.37: Dashboard Page Prototype

5.6.1.3 Management Pages

There are different module management pages that allow the admin to manage the modules. The admin can manage the modules such as products, packages, orders, customers, and vouchers by clicking the 'Manage' menu on the sidebar. Once the admin clicks on the menu, the system will redirect to the respective management page. On the respective management page, the admin can view a list of records, filters the records, sort the products and perform some CRUD actions on the records. The figures below show all the management pages available in the inventory management system.



Product Management

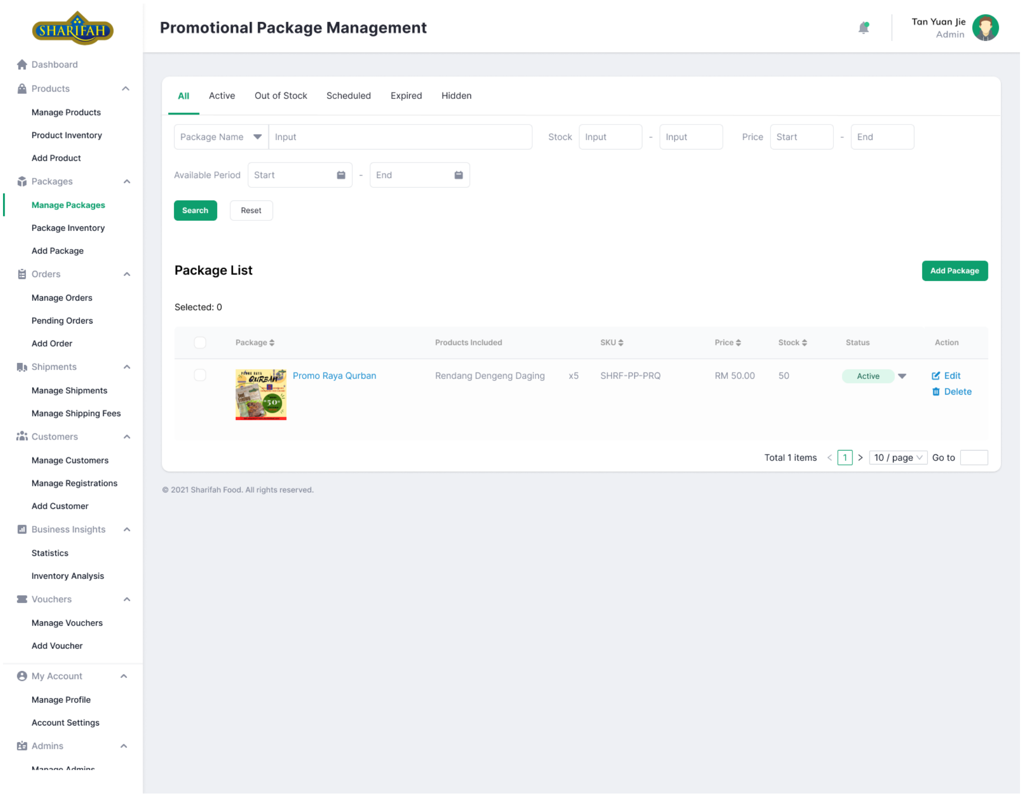
Product List

Selected: 2

□	Product	SKU	Price	Stock	Status	Action
☑	Kari Ayam Ready-To-Eat	SHRF-RTE-KA	RM 6.50	25	Active	Edit Delete
□	Sambal Ikan Bilis Ready-To-Eat	SHRF-RTE-SIB	RM 5.50	75	Active	Edit Delete
☑	Nasi Briyani Bukhari Ready-To-Cook	SHRF-RTC-NBB	RM 17.60	120	Active	Edit Delete
□	Rendang Dendeng Daging Ready-To-Eat	SHRF-RTE-RDD	RM 20.80	47	Active	Edit Delete
□	Pes Sambal Tumis Paste	SHRF-PES-PST	RM 4.20	90	Active	Edit Delete
□	Pes Masakan Keruntuk Paste	SHRF-PES-PMK	RM 4.20	200	Hidden	Edit Delete
□	Pes Mi Goreng Paste	SHRF-PES-PMG	RM 4.20	0	Out of Stock	Edit Delete

© 2021 Sharifah Food. All rights reserved.

Figure 5.38: Product Management Page Prototype



Promotional Package Management

Package Name: Input | Stock: Input - Input | Price: Start - End

Available Period: Start - End

Package List

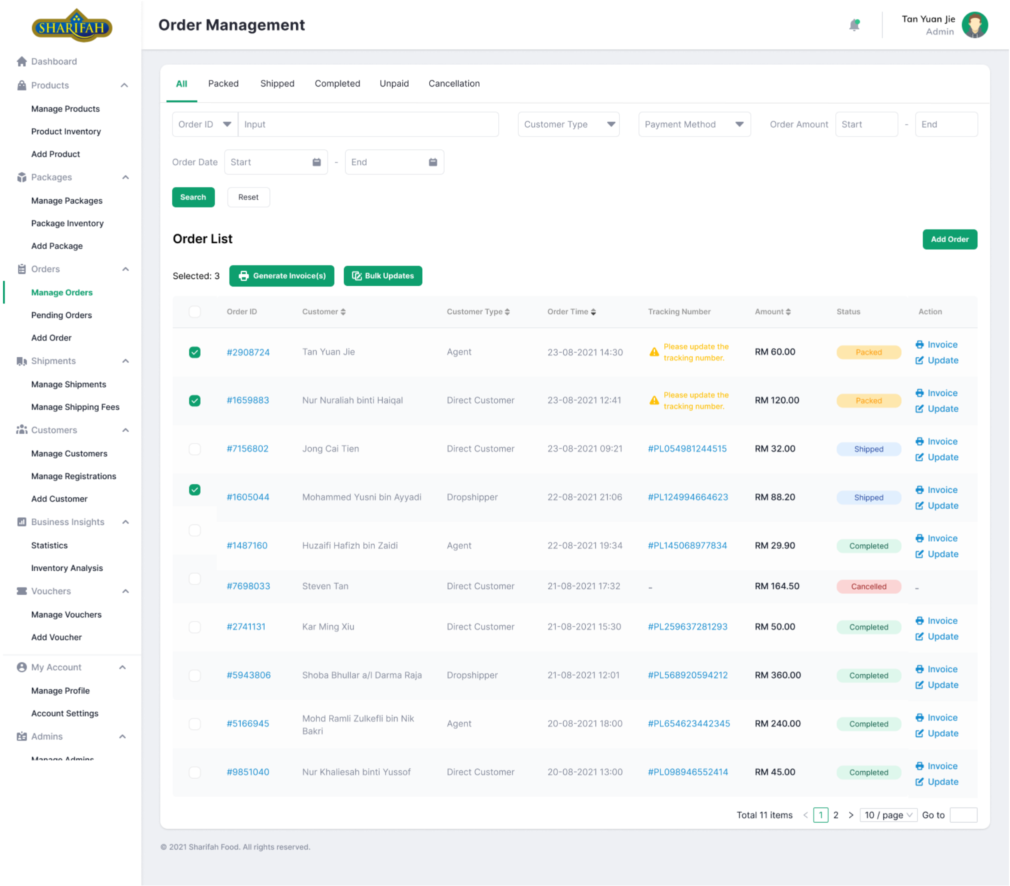
Selected: 0

□	Package	Products Included		SKU	Price	Stock	Status	Action
□	Promo Raya Qurban	Rendang Dengeng Daging	x5	SHRF-PP-PRQ	RM 50.00	50	Active	Edit Delete

Total 1 items < 1 > 10 / page Go to

© 2021 Sharifah Food. All rights reserved.

Figure 5.39: Package Management Page Prototype



Order Management

Order ID: Input | Customer Type | Payment Method | Order Amount: Start - End

Order Date: Start - End

Order List

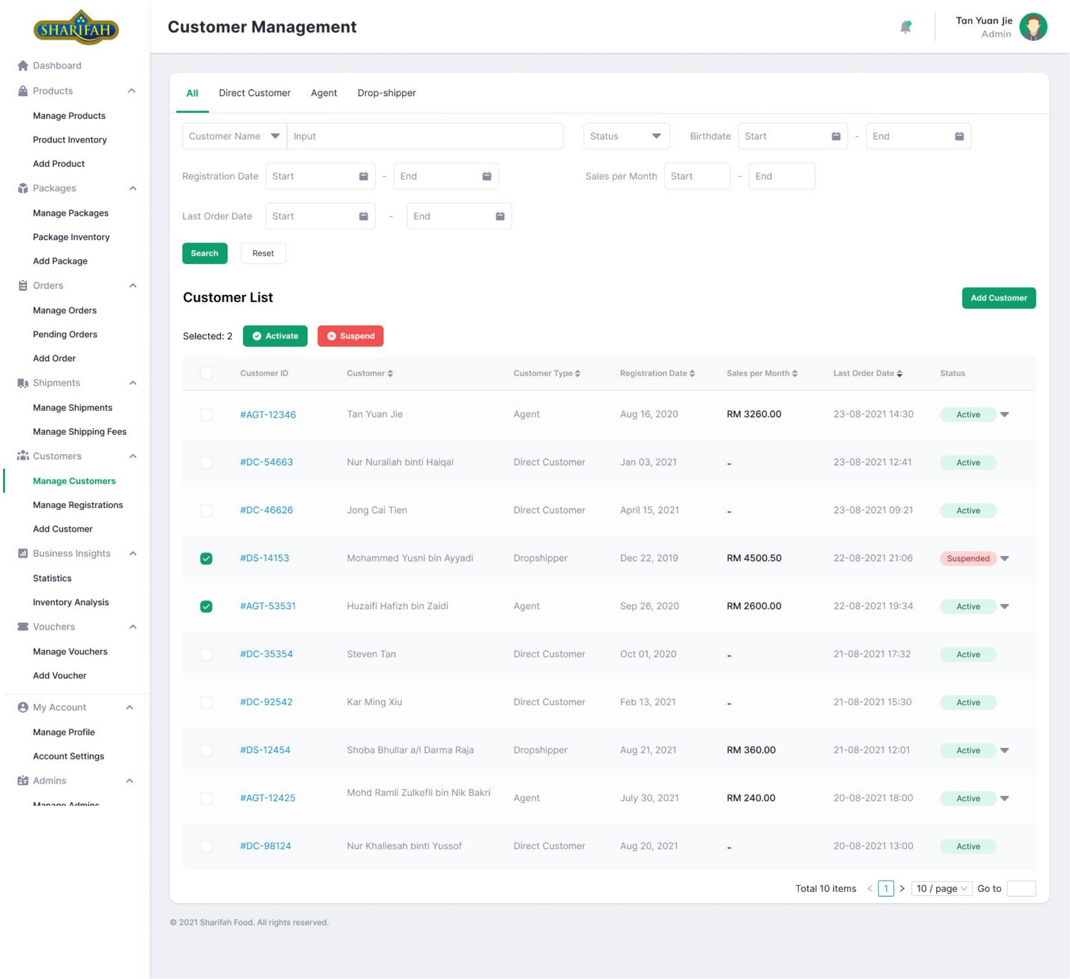
Selected: 3

□	Order ID	Customer	Customer Type	Order Time	Tracking Number	Amount	Status	Action
☑	#2908724	Tan Yuan Jie	Agent	23-08-2021 14:30	⚠ Please update the tracking number.	RM 60.00	Packed	Invoice Update
☑	#1659883	Nur Nuraliah binti Haiqal	Direct Customer	23-08-2021 12:41	⚠ Please update the tracking number.	RM 120.00	Packed	Invoice Update
□	#7156802	Jong Cai Tien	Direct Customer	23-08-2021 09:21	#PL054981244515	RM 32.00	Shipped	Invoice Update
☑	#1605044	Mohammed Yusni bin Ayyadi	Dropshipper	22-08-2021 21:06	#PL124994664623	RM 88.20	Shipped	Invoice Update
□	#1487160	Huzaifi Hafizh bin Zaidi	Agent	22-08-2021 19:34	#PL145068977834	RM 29.90	Completed	Invoice Update
□	#7698033	Steven Tan	Direct Customer	21-08-2021 17:32	-	RM 164.50	Cancelled	-
□	#2741131	Kar Ming Xiu	Direct Customer	21-08-2021 15:30	#PL259637281293	RM 50.00	Completed	Invoice Update
□	#5943806	Shoba Bhullar a/l Darma Raja	Dropshipper	21-08-2021 12:01	#PL568920594212	RM 360.00	Completed	Invoice Update
□	#5166945	Mohd Ramli Zulkefli bin Nik Bakri	Agent	20-08-2021 18:00	#PL654623442345	RM 240.00	Completed	Invoice Update
□	#9851040	Nur Khaliesah binti Yussof	Direct Customer	20-08-2021 13:00	#PL098946552414	RM 45.00	Completed	Invoice Update

Total 11 items < 1 2 > 10 / page Go to

© 2021 Sharifah Food. All rights reserved.

Figure 5.40: Order Management Page Prototype



Customer Management

Customer Name: Input | Status | Birthdate: Start - End

Registration Date: Start - End | Sales per Month: Start - End

Last Order Date: Start - End

Customer List

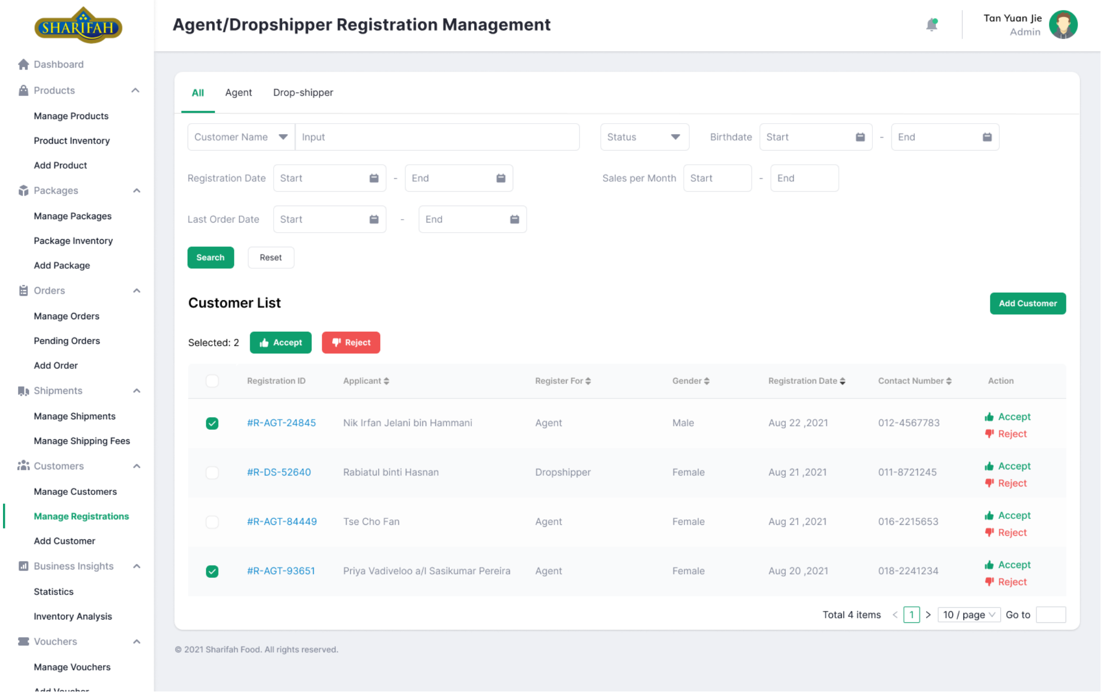
Selected: 2

	Customer ID	Customer	Customer Type	Registration Date	Sales per Month	Last Order Date	Status
□	#AGT-12346	Tan Yuan Jie	Agent	Aug 16, 2020	RM 3260.00	23-08-2021 14:30	Active
□	#DC-54663	Nur Nuraliah binti Haiqal	Direct Customer	Jan 03, 2021	-	23-08-2021 12:41	Active
□	#DC-46626	Jong Cai Tien	Direct Customer	April 15, 2021	-	23-08-2021 09:21	Active
☑	#DS-14153	Mohammed Yusni bin Ayyadi	Dropshipper	Dec 22, 2019	RM 4500.50	22-08-2021 21:06	Suspended
☑	#AGT-53531	Huzaifi Hafizh bin Zaidi	Agent	Sep 26, 2020	RM 2600.00	22-08-2021 19:34	Active
□	#DC-35354	Steven Tan	Direct Customer	Oct 01, 2020	-	21-08-2021 17:32	Active
□	#DC-92542	Kar Ming Xiu	Direct Customer	Feb 13, 2021	-	21-08-2021 15:30	Active
□	#DS-12454	Shoba Bhullar a/l Darma Raja	Dropshipper	Aug 21, 2021	RM 360.00	21-08-2021 12:01	Active
□	#AGT-12425	Mohd Ramli Zulkefli bin Nik Bakri	Agent	July 30, 2021	RM 240.00	20-08-2021 18:00	Active
□	#DC-98124	Nur Khaliesah binti Yussof	Direct Customer	Aug 20, 2021	-	20-08-2021 13:00	Active

Total 10 items < 1 > 10 / page Go to

© 2021 Sharifah Food. All rights reserved.

Figure 5.41: Customer Management Page Prototype



Agent/Dropshipper Registration Management

Customer Name: Input | Status | Birthdate: Start - End

Registration Date: Start - End | Sales per Month: Start - End

Last Order Date: Start - End

Customer List

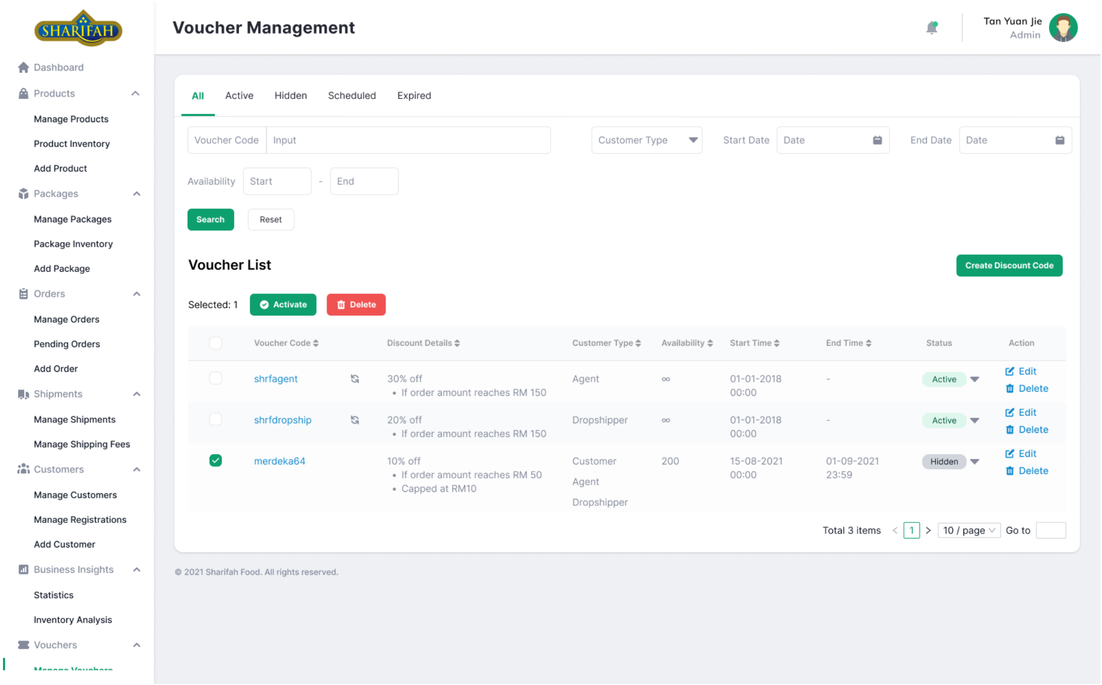
Selected: 2

	Registration ID	Applicant	Register For	Gender	Registration Date	Contact Number	Action
☑	#R-AGT-24845	Nik Irfan Jelani bin Hammani	Agent	Male	Aug 22 ,2021	012-4567783	Accept Reject
□	#R-DS-52640	Rabiatul binti Hasnan	Dropshipper	Female	Aug 21 ,2021	011-8721245	Accept Reject
□	#R-AGT-84449	Tse Cho Fan	Agent	Female	Aug 21 ,2021	016-2215653	Accept Reject
☑	#R-AGT-93651	Priya Vadiveloo a/l Sasikumar Pereira	Agent	Female	Aug 20 ,2021	018-2241234	Accept Reject

Total 4 items < 1 > 10 / page Go to

© 2021 Sharifah Food. All rights reserved.

Figure 5.42: Customer Registration Management Page Prototype



Voucher Management

Voucher Code: Input

Customer Type

Start Date: Date

End Date: Date

Availability: Start - End

Voucher List

Selected: 1

□	Voucher Code	Discount Details	Customer Type	Availability	Start Time	End Time	Status	Action
□	shrfagent	30% off • If order amount reaches RM 150	Agent	∞	01-01-2018 00:00	-	Active	Edit Delete
□	shrfdropship	20% off • If order amount reaches RM 150	Dropshipper	∞	01-01-2018 00:00	-	Active	Edit Delete
☑	merdeka64	10% off • If order amount reaches RM 50 • Capped at RM10	Customer Agent Dropshipper	200	15-08-2021 00:00	01-09-2021 23:59	Hidden	Edit Delete

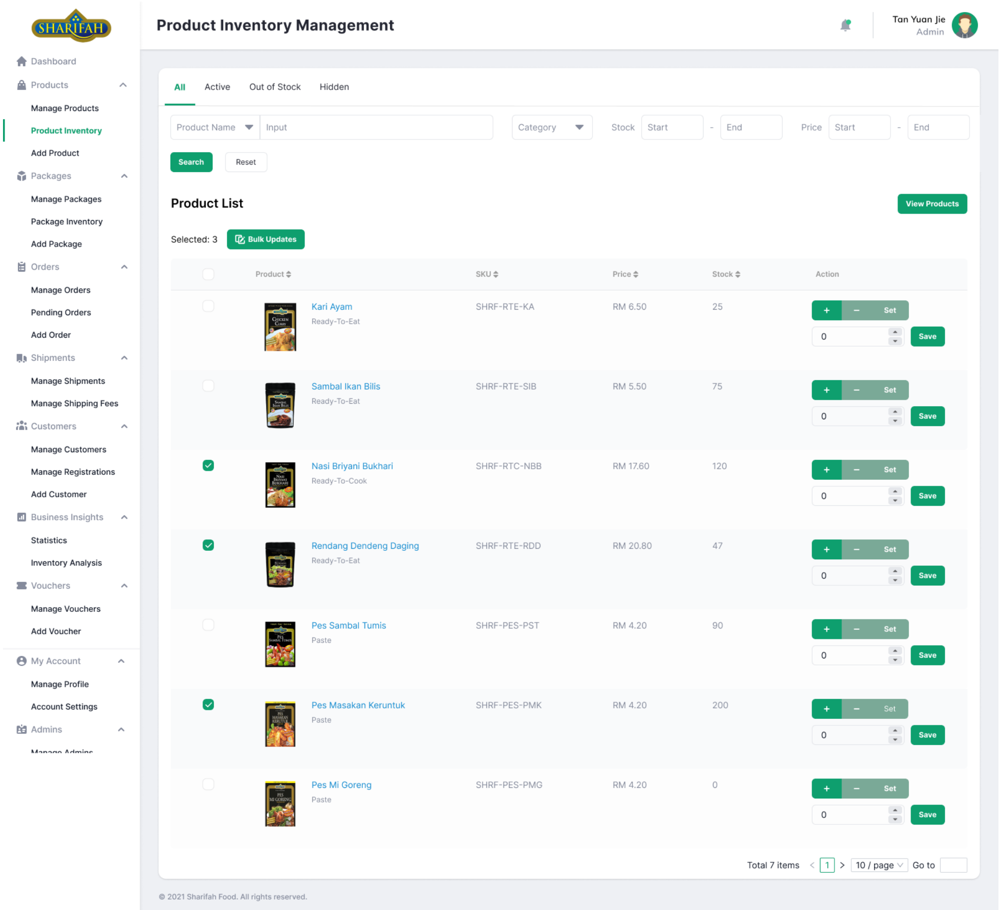
Total 3 items < 1 > 10 / page Go to

© 2021 Sharifah Food. All rights reserved.

Figure 5.43: Voucher Management Page Prototype

5.6.1.4 Inventory Management Pages

Like the management pages, the system allows the admin to manage the inventory for the products and packages specifically. On the inventory management pages, the admin can directly modify the items' stock without entering the item details pages one by one. The figures below show all the inventory management pages available in the inventory management system.



Product Inventory Management

Product List

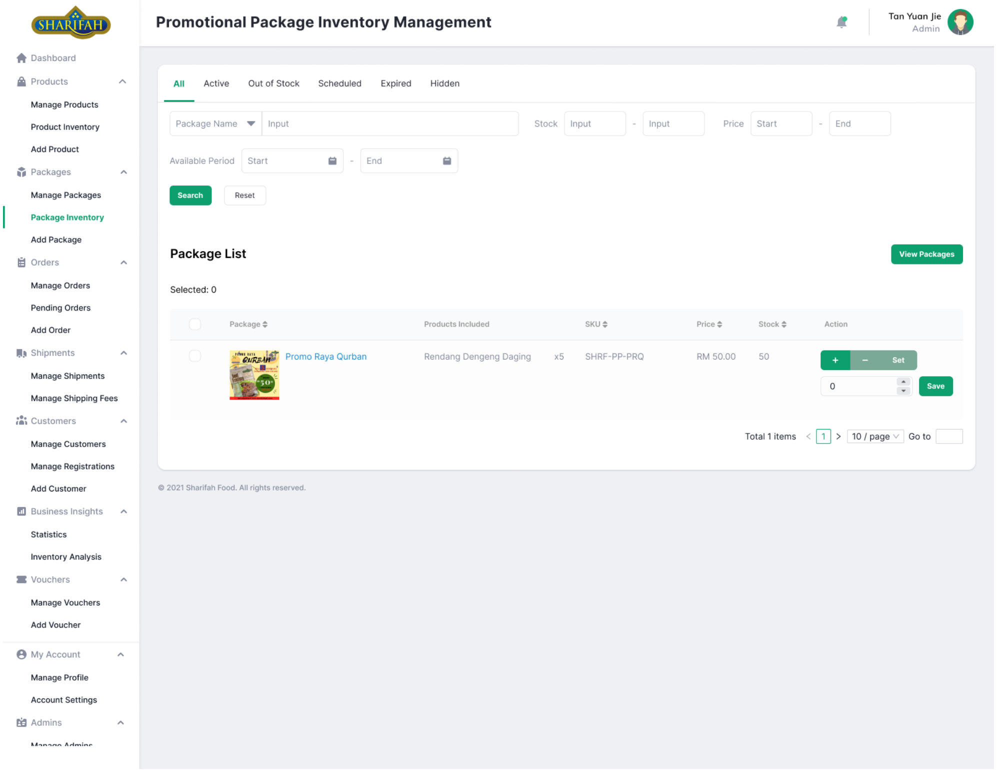
Selected: 3

	Product		SKU	Price	Stock	Action
□		Kari Ayam Ready-To-Eat	SHRF-RTE-KA	RM 6.50	25	0
□		Sambal Ikan Bilis Ready-To-Eat	SHRF-RTE-SIB	RM 5.50	75	0
☑		Nasi Briyani Bukhari Ready-To-Cook	SHRF-RTC-NBB	RM 17.60	120	0
☑		Rendang Dendeng Daging Ready-To-Eat	SHRF-RTE-RDD	RM 20.80	47	0
□		Pes Sambal Tumis Paste	SHRF-PES-PST	RM 4.20	90	0
☑		Pes Masakan Keruntuk Paste	SHRF-PES-PMK	RM 4.20	200	0
□		Pes Mi Goreng Paste	SHRF-PES-PMG	RM 4.20	0	0

Total 7 items

© 2021 Sharifah Food. All rights reserved.

Figure 5.44: Product Inventory Management Page Prototype



Promotional Package Inventory Management

Package List

Selected: 0

	Package		Products Included		SKU	Price	Stock	Action
□		Promo Raya Qurban	Rendang Dengeng Daging	x5	SHRF-PP-PRQ	RM 50.00	50	0

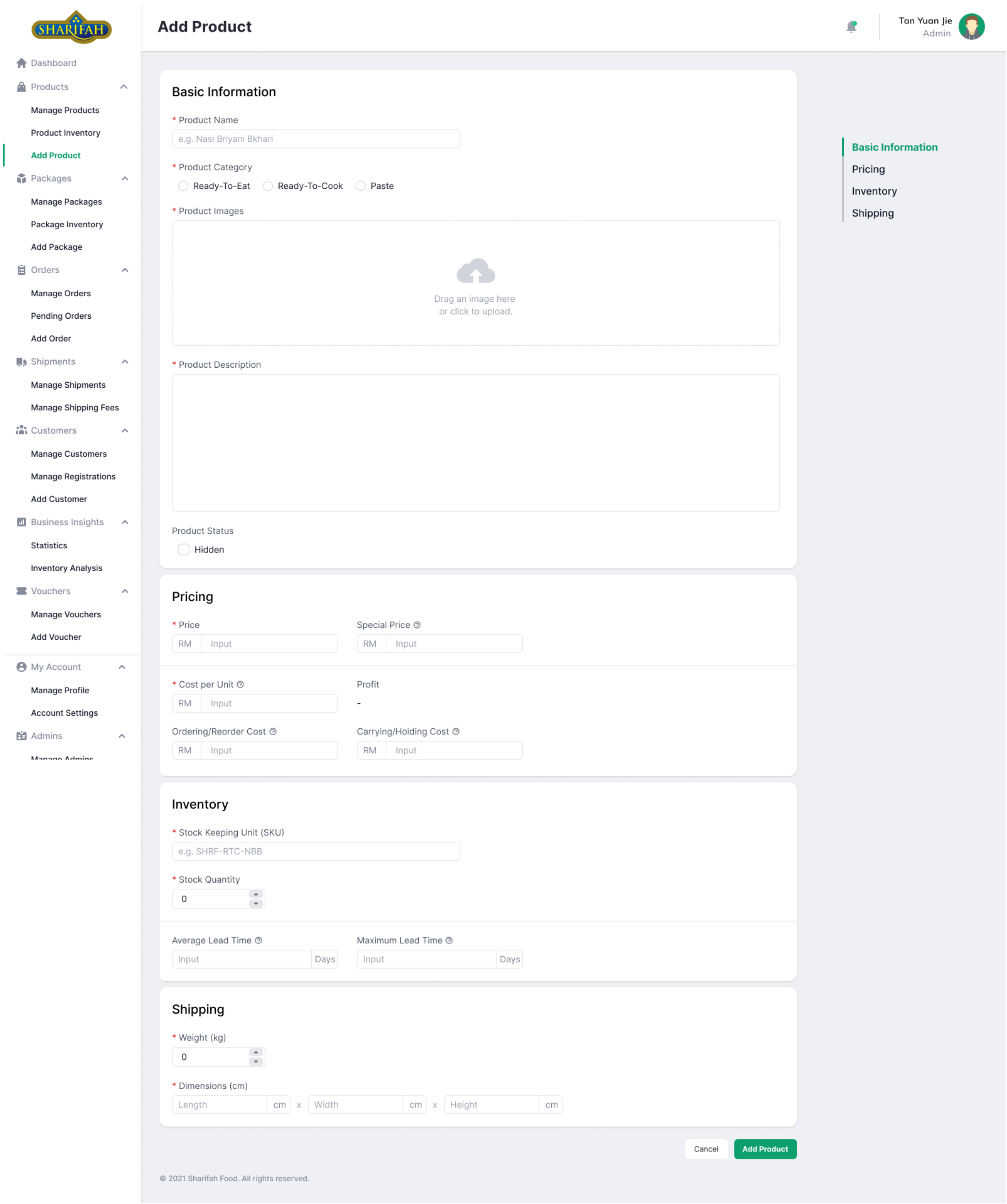
Total 1 items

© 2021 Sharifah Food. All rights reserved.

Figure 5.45: Package Inventory Management Page Prototype

5.6.1.5 Add and Edit Record Page

Besides viewing the list of records, the inventory management system also allows the admin to add and edit the specific record. The admin can add the record by clicking the ‘Add’ button and edit the record by clicking the record title or the ‘Edit’ button. The figures below show all the add and edit pages available in the inventory management system.



Add Product

Basic Information

* Product Name

e.g. Nasi Briyani Bkhari

* Product Category

○ Ready-To-Eat ○ Ready-To-Cook ○ Paste

* Product Images

Drag an image here
or click to upload.

* Product Description

Product Status

□ Hidden

Pricing

* Price	Special Price ⓘ
RM Input	RM Input
* Cost per Unit ⓘ	Profit
RM Input	-
Ordering/Reorder Cost ⓘ	Carrying/Holding Cost ⓘ
RM Input	RM Input

Inventory

* Stock Keeping Unit (SKU)

e.g. SHRF-RTC-NBB

* Stock Quantity

0

Average Lead Time ⓘ	Maximum Lead Time ⓘ
Input Days	Input Days

Shipping

* Weight (kg)

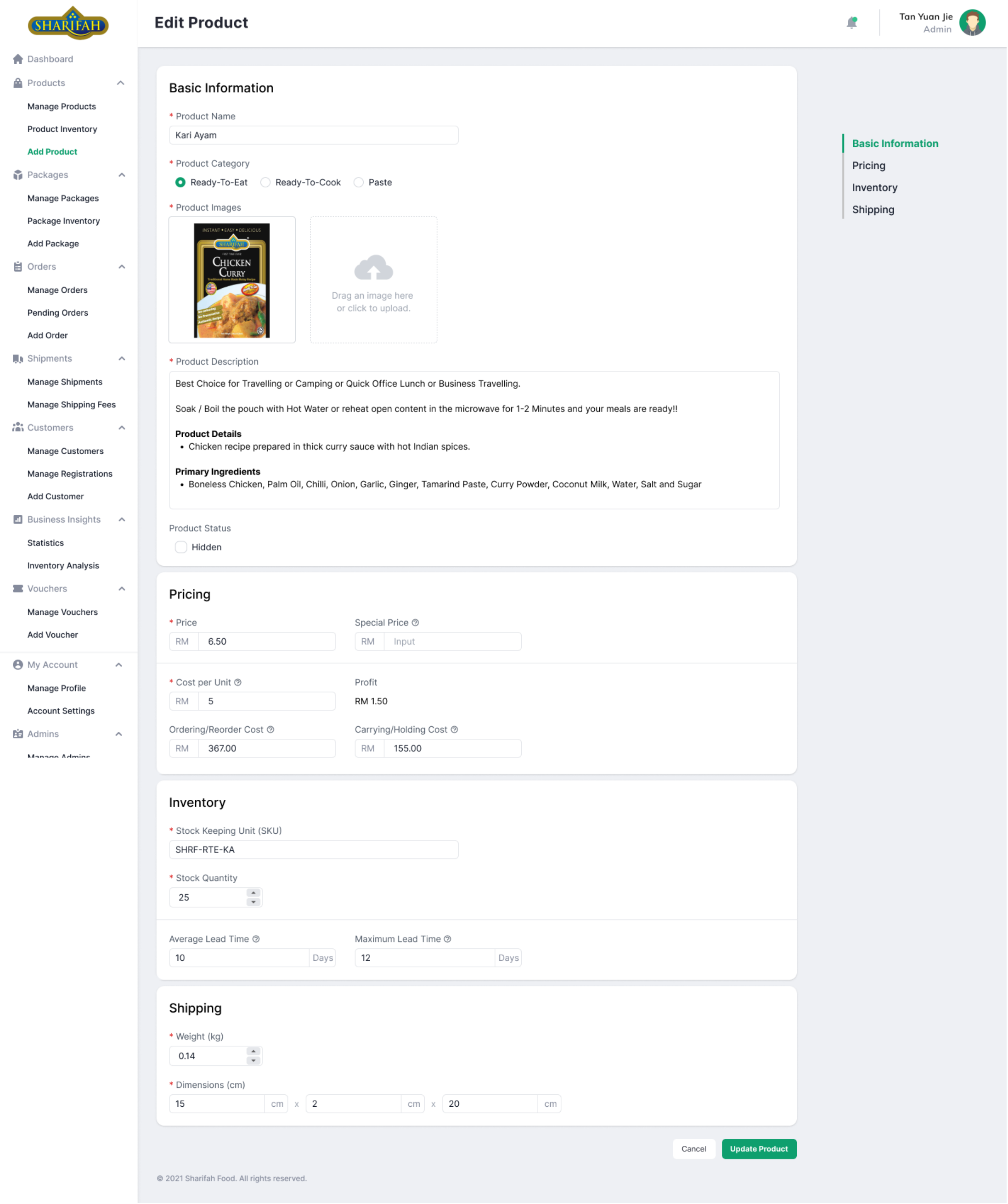
0

* Dimensions (cm)

Length cm x Width cm x Height cm

© 2021 Sharifah Food. All rights reserved.

Figure 5.46: Add Product Page Prototype



Edit Product

Basic Information

* Product Name
Kari Ayam

* Product Category
● Ready-To-Eat ○ Ready-To-Cook ○ Paste

* Product Images

* Product Description

Best Choice for Travelling or Camping or Quick Office Lunch or Business Travelling.

Soak / Boil the pouch with Hot Water or reheat open content in the microwave for 1-2 Minutes and your meals are ready!!

Product Details
- Chicken recipe prepared in thick curry sauce with hot Indian spices.

Primary Ingredients
- Boneless Chicken, Palm Oil, Chilli, Onion, Garlic, Ginger, Tamarind Paste, Curry Powder, Coconut Milk, Water, Salt and Sugar

Product Status
□ Hidden

Pricing

* Price: RM 6.50
Special Price: RM Input

* Cost per Unit: RM 5
Profit: RM 1.50

Ordering/Reorder Cost: RM 367.00
Carrying/Holding Cost: RM 155.00

Inventory

* Stock Keeping Unit (SKU)
SHRF-RTE-KA

* Stock Quantity
25

Average Lead Time: 10 Days
Maximum Lead Time: 12 Days

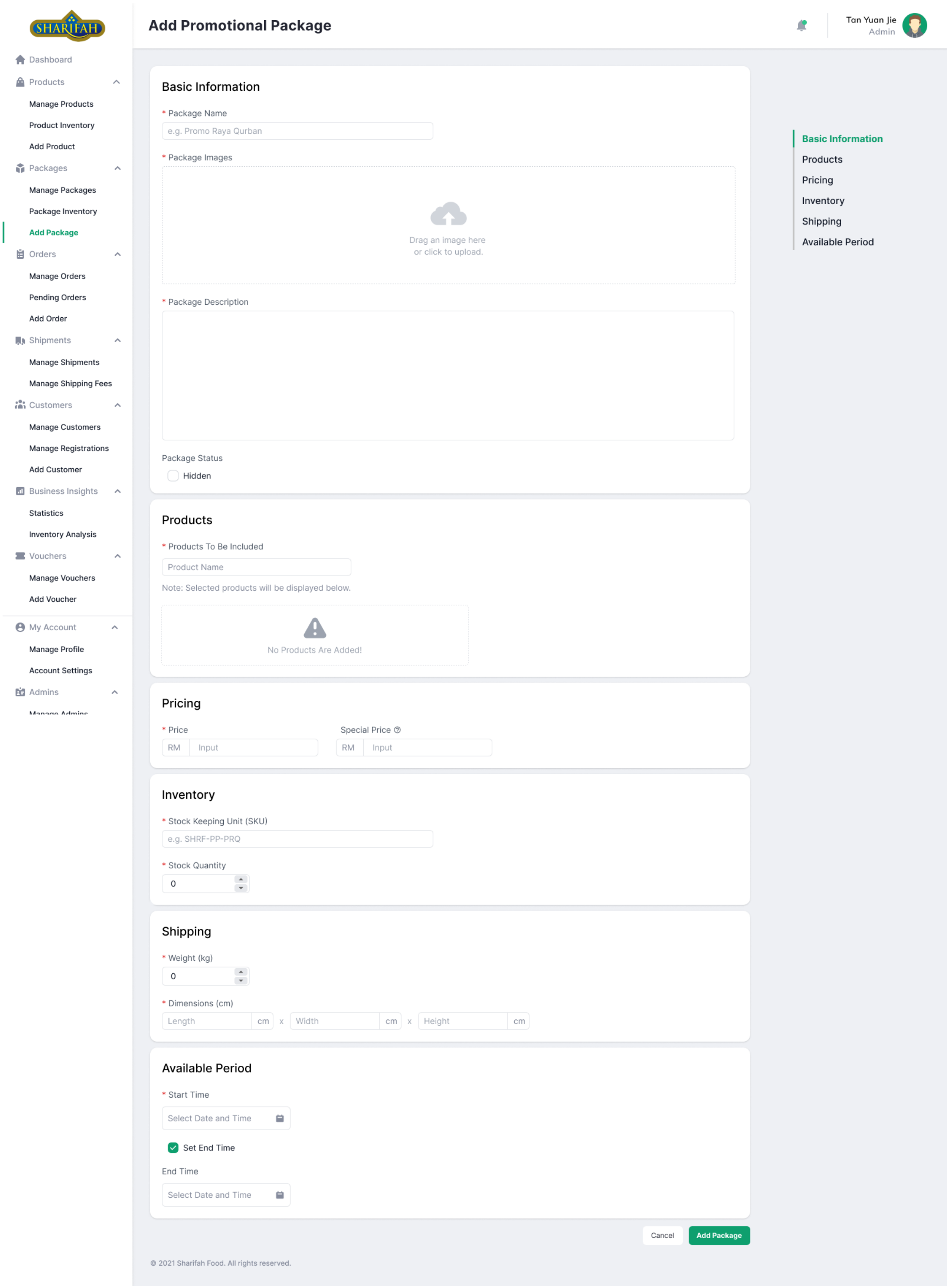
Shipping

* Weight (kg)
0.14

* Dimensions (cm)
15 cm x 2 cm x 20 cm

© 2021 Sharifah Food. All rights reserved.

Figure 5.47: Edit Product Page Prototype



Add Promotional Package

Basic Information

* Package Name

e.g. Promo Raya Qurban

* Package Images

Drag an image here
or click to upload.

* Package Description

Package Status

□ Hidden

Products

* Products To Be Included

Product Name

Note: Selected products will be displayed below.

No Products Are Added!

Pricing

* Price RM Input

Special Price RM Input

Inventory

* Stock Keeping Unit (SKU)

e.g. SHRF-PP-PRQ

* Stock Quantity

0

Shipping

* Weight (kg)

0

* Dimensions (cm)

Length cm x Width cm x Height cm

Available Period

* Start Time

Select Date and Time

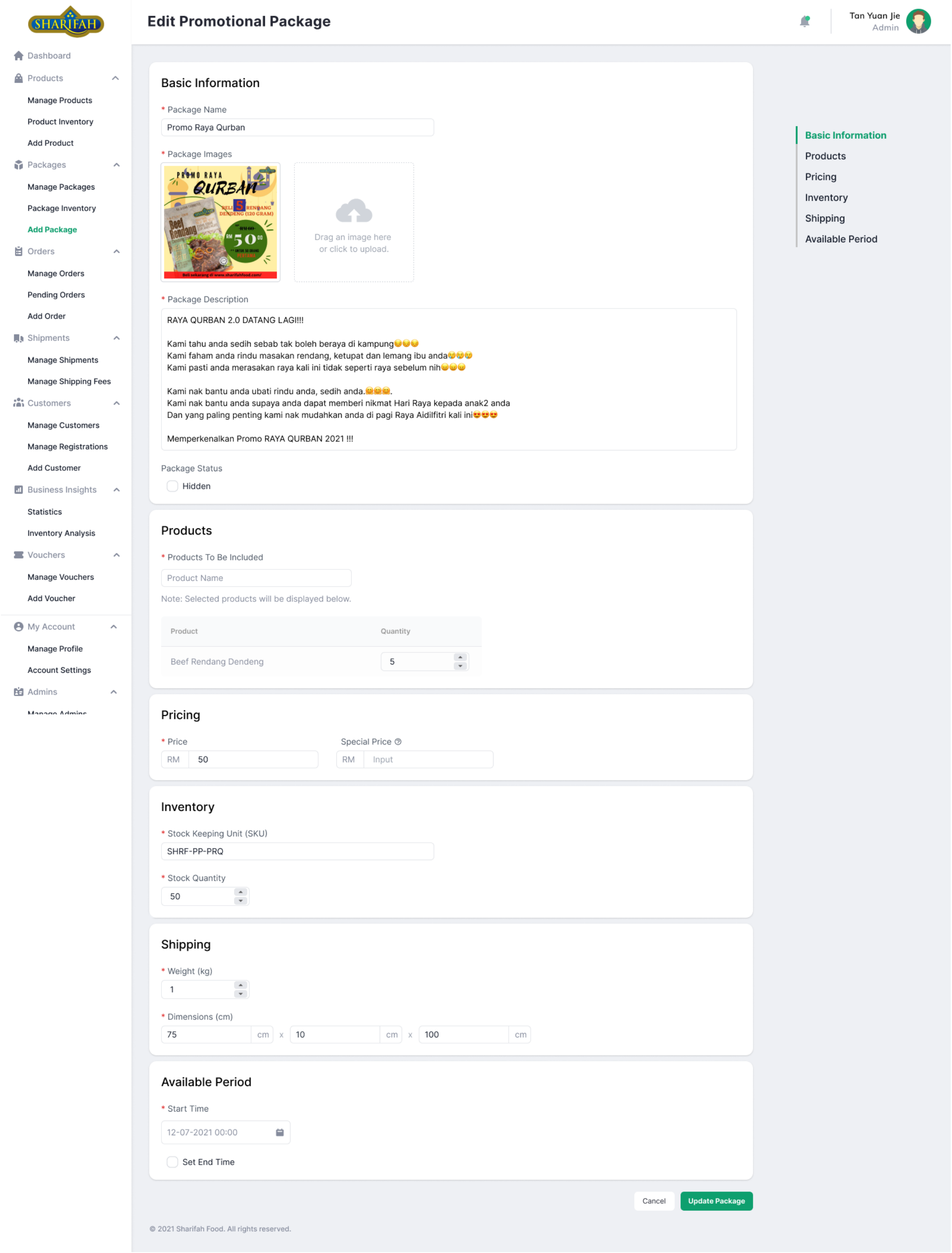
☑ Set End Time

End Time

Select Date and Time

© 2021 Sharifah Food. All rights reserved.

Figure 5.48: Add Package Page Prototype



Edit Promotional Package

Tan Yuan Jie
Admin

Basic Information

* Package Name: Promo Raya Qurban

* Package Images

PROMO RAYA
QURBAN
Beef Rendang
RENDANG DENDENG (120 GRAM)
RM 60
RM 50 00
UNTUK 50 ORANG PERTAMA
Beli sekarang di www.sharifahfood.com/
Drag an image here or click to upload.

* Package Description

RAYA QURBAN 2.0 DATANG LAGI!!!

Kami tahu anda sedih sebab tak boleh beraya di kampung😔😔😔
Kami faham anda rindu masakan rendang, ketupat dan lemang ibu anda😢😢😢
Kami pasti anda merasakan raya kali ini tidak seperti raya sebelum nih😖😖😖

Kami nak bantu anda ubati rindu anda, sedih anda.🤗🤗🤗.
Kami nak bantu anda supaya anda dapat memberi nikmat Hari Raya kepada anak2 anda
Dan yang paling penting kami nak mudahkan anda di pagi Raya Aidilfitri kali ini😍😍😍

Memperkenalkan Promo RAYA QURBAN 2021 !!!

Package Status
☐ Hidden

Products

* Products To Be Included: Product Name

Note: Selected products will be displayed below.

Product	Quantity
Beef Rendang Dendeng	5

Pricing

* Price: RM 50

Special Price: RM Input

Inventory

* Stock Keeping Unit (SKU): SHRF-PP-PRQ

* Stock Quantity: 50

Shipping

* Weight (kg): 1

* Dimensions (cm): 75 cm x 10 cm x 100 cm

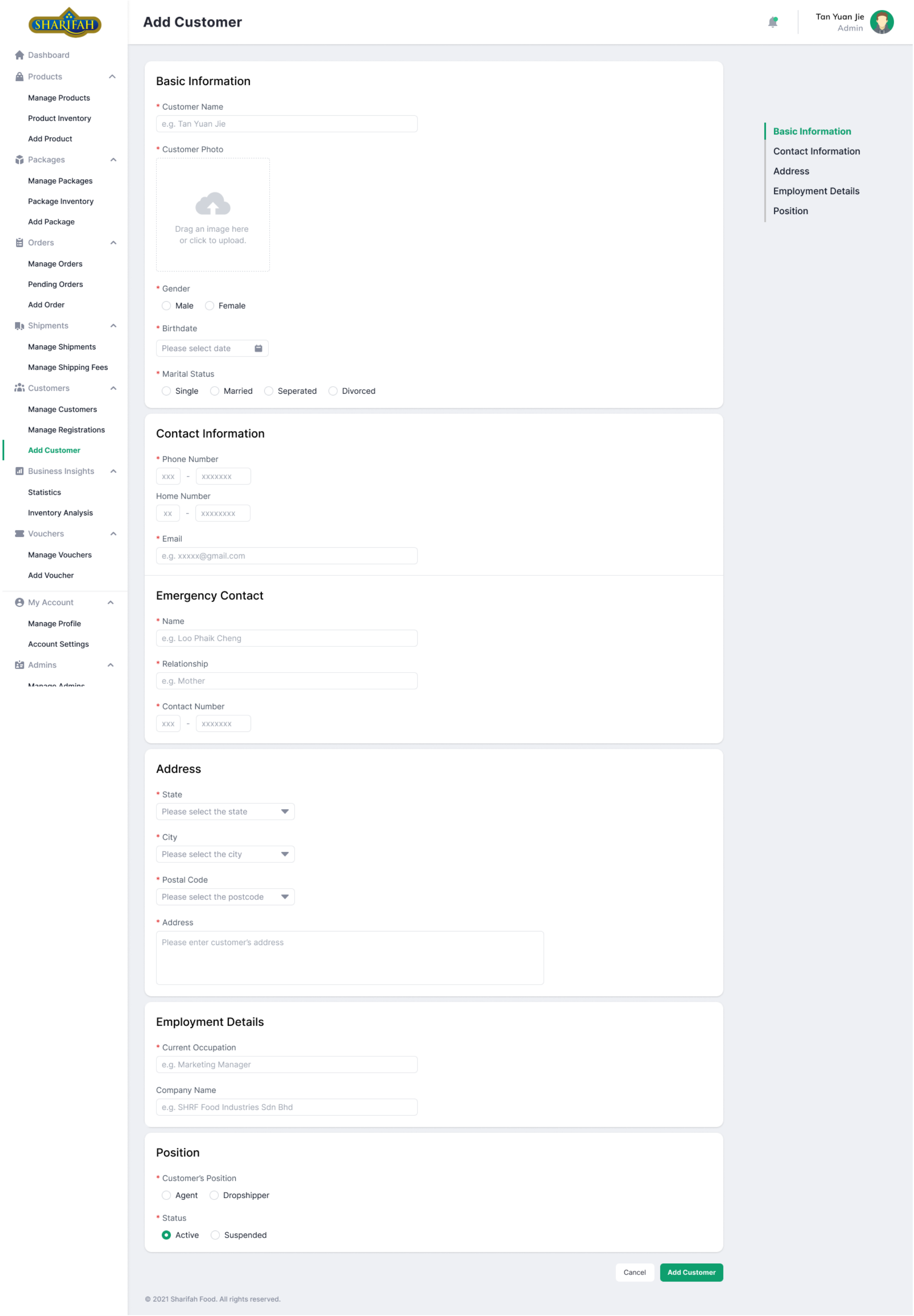
Available Period

* Start Time: 12-07-2021 00:00

☐ Set End Time

© 2021 Sharifah Food. All rights reserved.

Figure 5.49: Edit Package Page Prototype



Add Customer

Basic Information

* Customer Name
e.g. Tan Yuan Jie

* Customer Photo
Drag an image here or click to upload.

* Gender
○ Male ○ Female

* Birthdate
Please select date

* Marital Status
○ Single ○ Married ○ Seperated ○ Divorced

Contact Information

* Phone Number
xxx - xxxxxxx

Home Number
xx - xxxxxxxx

* Email
e.g. xxxxx@gmail.com

Emergency Contact

* Name
e.g. Loo Phaik Cheng

* Relationship
e.g. Mother

* Contact Number
xxx - xxxxxxx

Address

* State
Please select the state

* City
Please select the city

* Postal Code
Please select the postcode

* Address
Please enter customer's address

Employment Details

* Current Occupation
e.g. Marketing Manager

Company Name
e.g. SHRF Food Industries Sdn Bhd

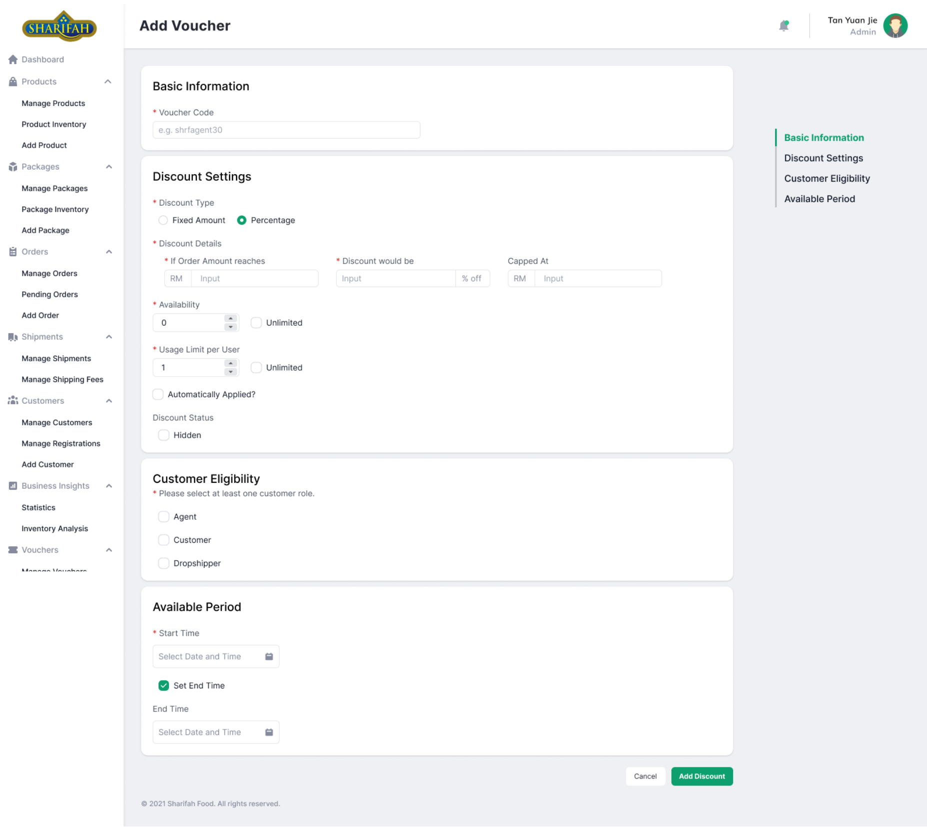
Position

* Customer's Position
○ Agent ○ Dropshipper

* Status
● Active ○ Suspended

© 2021 Sharifah Food. All rights reserved.

Figure 5.50: Add Customer Page Prototype



Add Voucher

Basic Information

* Voucher Code: e.g. shrfagent30

Discount Settings

* Discount Type: ○ Fixed Amount ● Percentage

* Discount Details

* If Order Amount reaches: RM Input

* Discount would be: Input % off

Capped At: RM Input

* Availability: 0 □ Unlimited

* Usage Limit per User: 1 □ Unlimited

□ Automatically Applied?

Discount Status: □ Hidden

Customer Eligibility

* Please select at least one customer role.

□ Agent

□ Customer

□ Dropshipper

Available Period

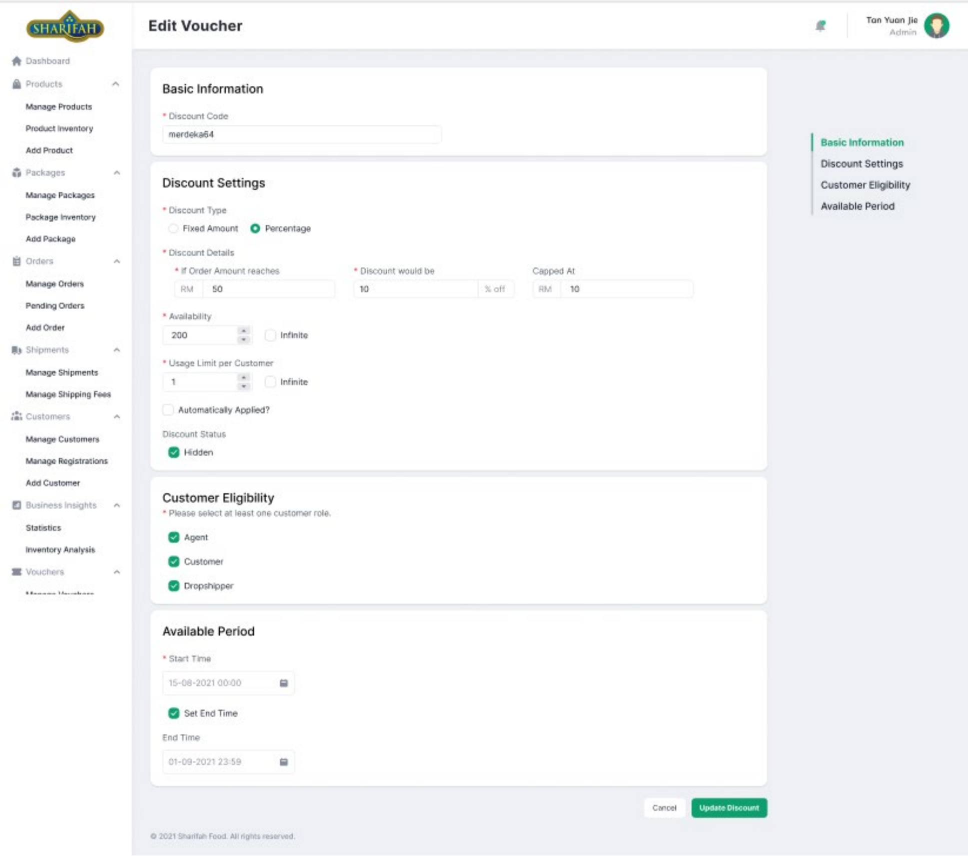
* Start Time: Select Date and Time

☑ Set End Time

End Time: Select Date and Time

© 2021 Sharifah Food. All rights reserved.

Figure 5.51: Add Voucher Page Prototype



Edit Voucher

Basic Information

* Discount Code: merdeka64

Discount Settings

* Discount Type: ○ Fixed Amount ● Percentage

* Discount Details

* If Order Amount reaches: RM 50

* Discount would be: 10 % off

Capped At: RM 10

* Availability: 200 □ Infinite

* Usage Limit per Customer: 1 □ Infinite

□ Automatically Applied?

Discount Status: ☑ Hidden

Customer Eligibility

* Please select at least one customer role.

☑ Agent

☑ Customer

☑ Dropshipper

Available Period

* Start Time: 15-08-2021 00:00

☑ Set End Time

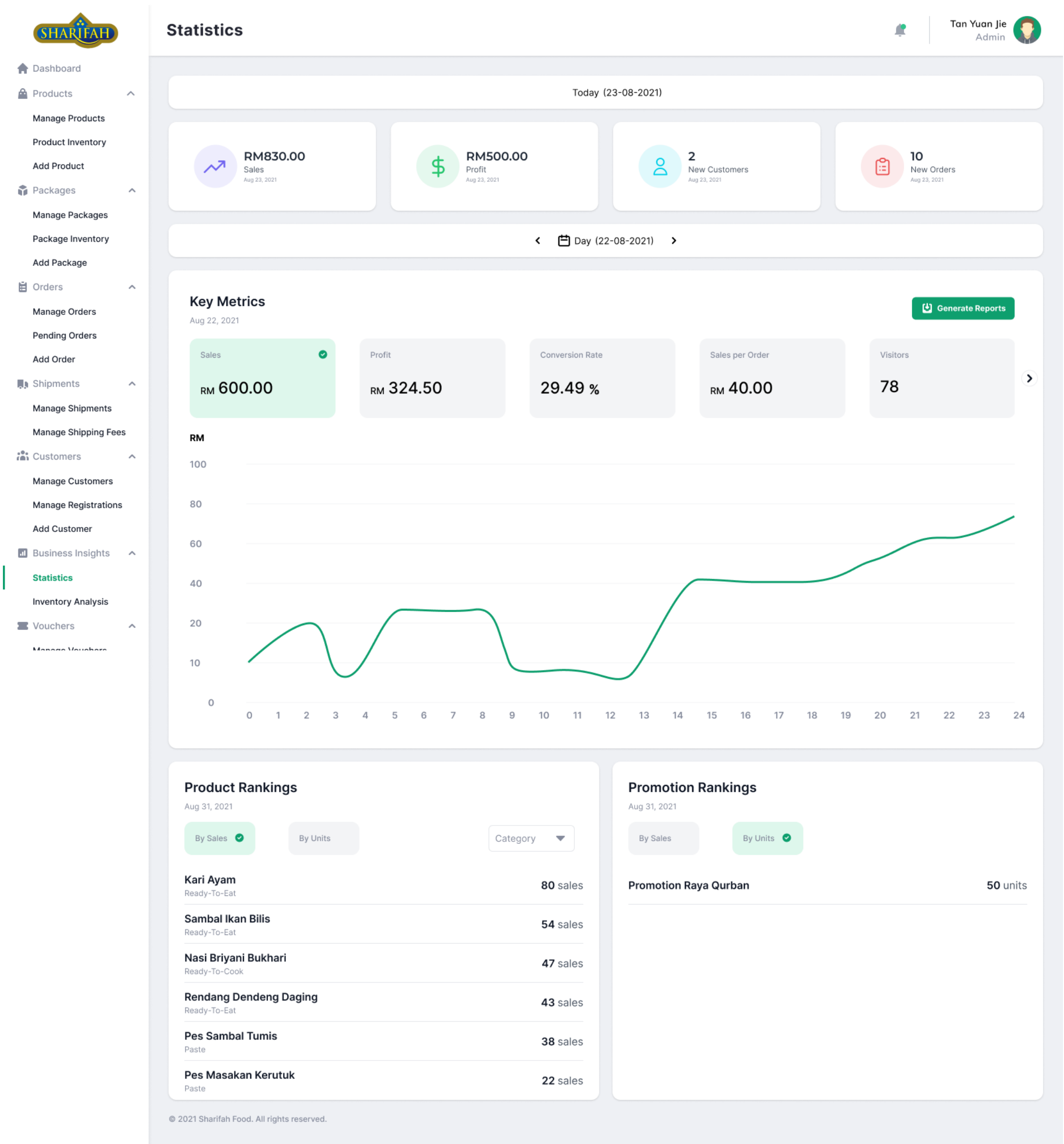
End Time: 01-09-2021 23:59

© 2021 Sharifah Food. All rights reserved.

Figure 5.52: Edit Voucher Page Prototype

5.6.1.6 Statistics Page

The statistics page is one of the pages under the business insights. It focuses on sales analysis. Besides, the attractive point of the statistics page is its eight key sales metrics analysis. The key metrics analysis displays two results, the sum of the value of the key metrics over the selected period and the time-series line graph of the key metrics. Moreover, the statistics page displays the product rankings and package rankings over the chosen period. Figure 5.53 shows the statistics page.



Statistics

Today (23-08-2021)

RM830.00
Sales
Aug 23, 2021

RM500.00
Profit
Aug 23, 2021

2
New Customers
Aug 23, 2021

10
New Orders
Aug 23, 2021

Day (22-08-2021)

Key Metrics
Aug 22, 2021

Sales	Profit	Conversion Rate	Sales per Order	Visitors
RM 600.00	RM 324.50	29.49 %	RM 40.00	78

Product Rankings
Aug 31, 2021

Kari Ayam Ready-To-Eat	80 sales
Sambal Ikan Bilis Ready-To-Eat	54 sales
Nasi Briyani Bukhari Ready-To-Cook	47 sales
Rendang Dendeng Daging Ready-To-Eat	43 sales
Pes Sambal Tumis Paste	38 sales
Pes Masakan Kerutuk Paste	22 sales

Promotion Rankings
Aug 31, 2021

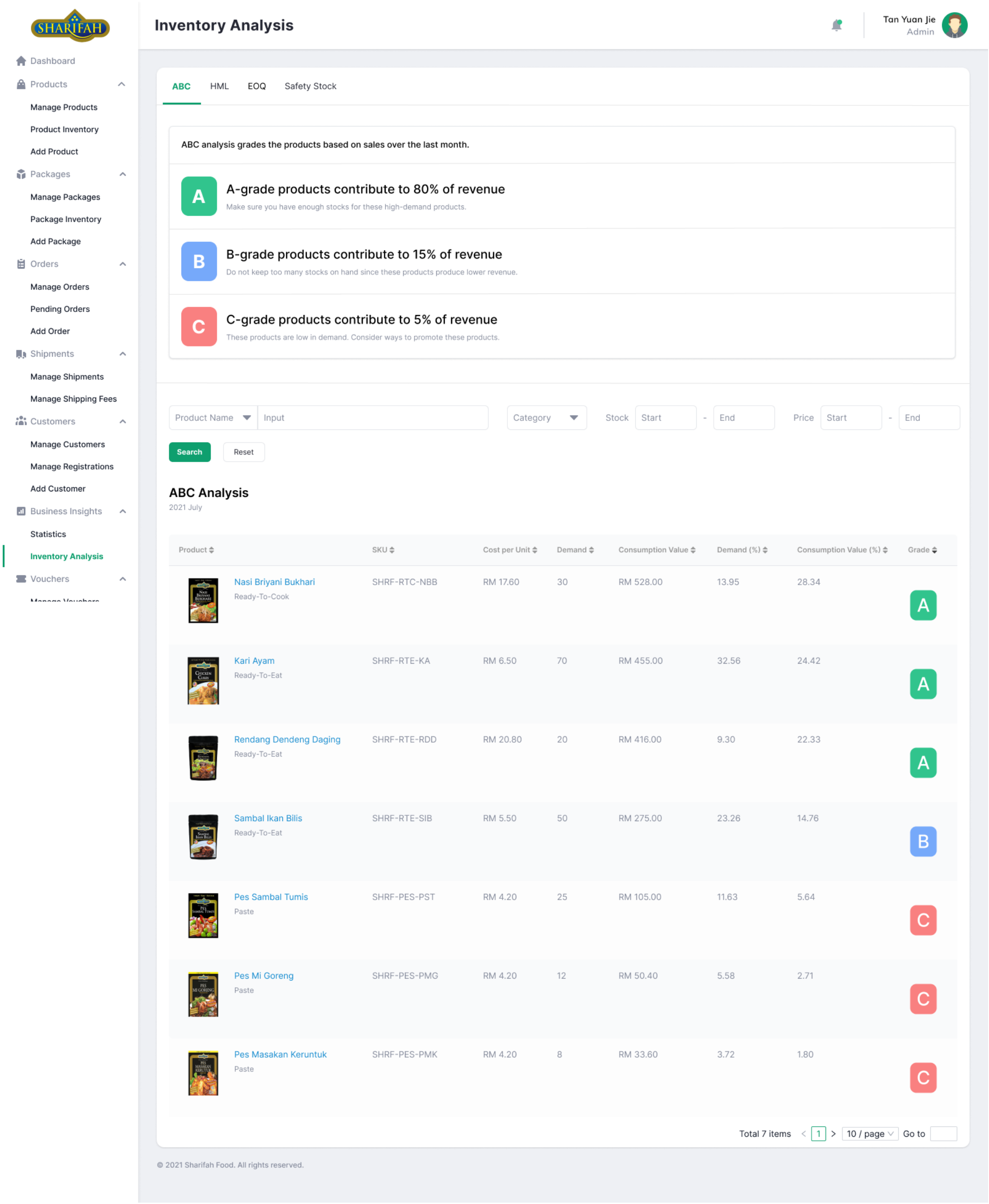
Promotion Raya Qurban	50 units

© 2021 Sharifah Food. All rights reserved.

Figure 5.53: Statistics Page Prototype

5.6.1.7 Inventory Analyses Page

Besides the sales analysis, the business insights in the inventory management system also provide inventory analyses. There are four inventory analyses available, including ABC, HML, Optimal Order Quantity, and Safety Stock. The figures below show all the inventory analyses pages available in the inventory management system.



ABC analysis grades the products based on sales over the last month.

A A-grade products contribute to 80% of revenue
Make sure you have enough stocks for these high-demand products.

B B-grade products contribute to 15% of revenue
Do not keep too many stocks on hand since these products produce lower revenue.

C C-grade products contribute to 5% of revenue
These products are low in demand. Consider ways to promote these products.

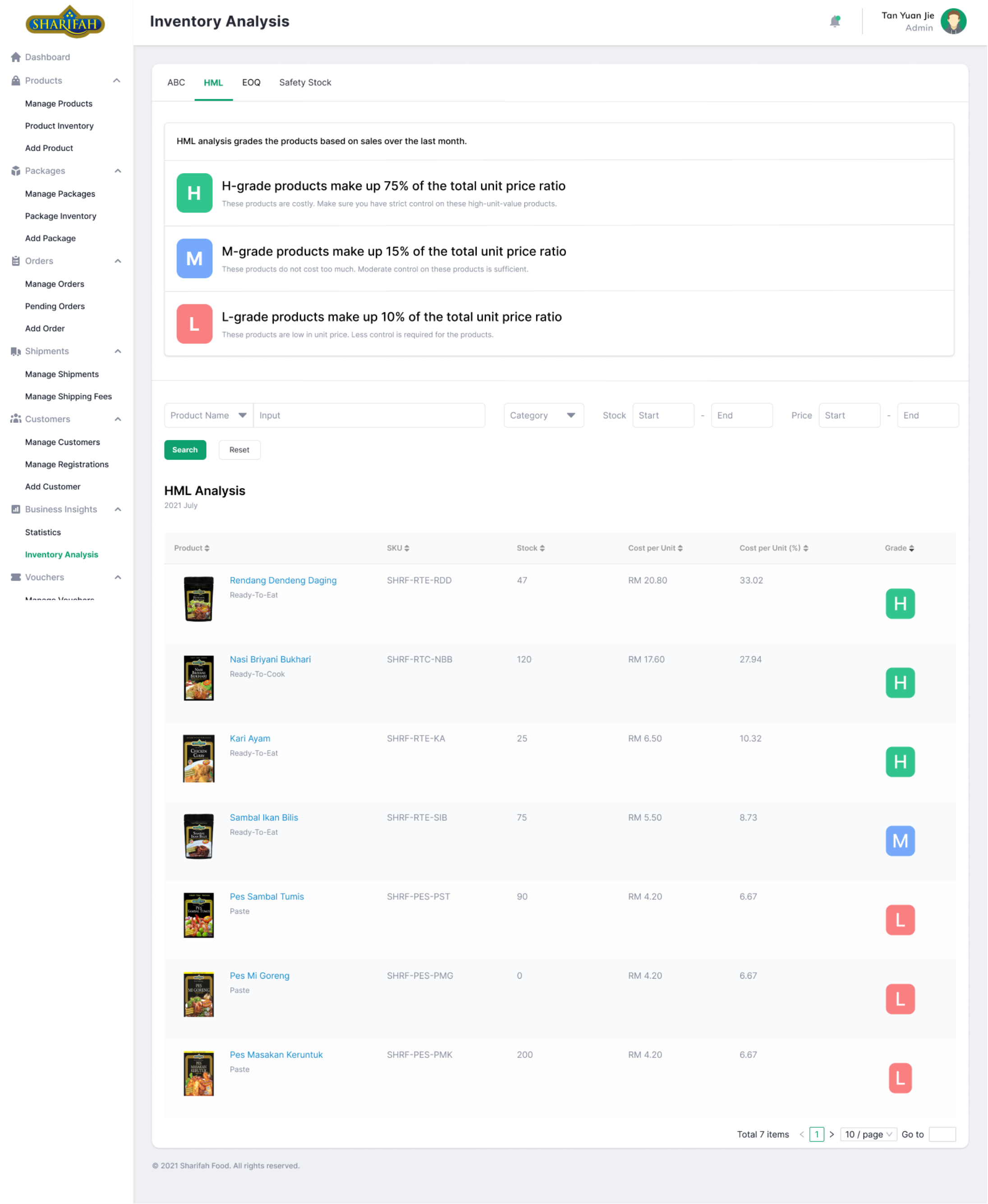
ABC Analysis
2021 July

Product	SKU	Cost per Unit	Demand	Consumption Value	Demand (%)	Consumption Value (%)	Grade
Nasi Briyani Bukhari Ready-To-Cook	SHRF-RTC-NBB	RM 17.60	30	RM 528.00	13.95	28.34	A
Kari Ayam Ready-To-Eat	SHRF-RTE-KA	RM 6.50	70	RM 455.00	32.56	24.42	A
Rendang Dendeng Daging Ready-To-Eat	SHRF-RTE-RDD	RM 20.80	20	RM 416.00	9.30	22.33	A
Sambal Ikan Bilis Ready-To-Eat	SHRF-RTE-SIB	RM 5.50	50	RM 275.00	23.26	14.76	B
Pes Sambal Tumis Paste	SHRF-PES-PST	RM 4.20	25	RM 105.00	11.63	5.64	C
Pes Mi Goreng Paste	SHRF-PES-PMG	RM 4.20	12	RM 50.40	5.58	2.71	C
Pes Masakan Keruntuk Paste	SHRF-PES-PMK	RM 4.20	8	RM 33.60	3.72	1.80	C

Total 7 items < 1 > 10 / page Go to

© 2021 Sharifah Food. All rights reserved.

Figure 5.54: ABC Analysis Page Prototype



Inventory Analysis

HML analysis grades the products based on sales over the last month.

H H-grade products make up 75% of the total unit price ratio

These products are costly. Make sure you have strict control on these high-unit-value products.

M M-grade products make up 15% of the total unit price ratio

These products do not cost too much. Moderate control on these products is sufficient.

L L-grade products make up 10% of the total unit price ratio

These products are low in unit price. Less control is required for the products.

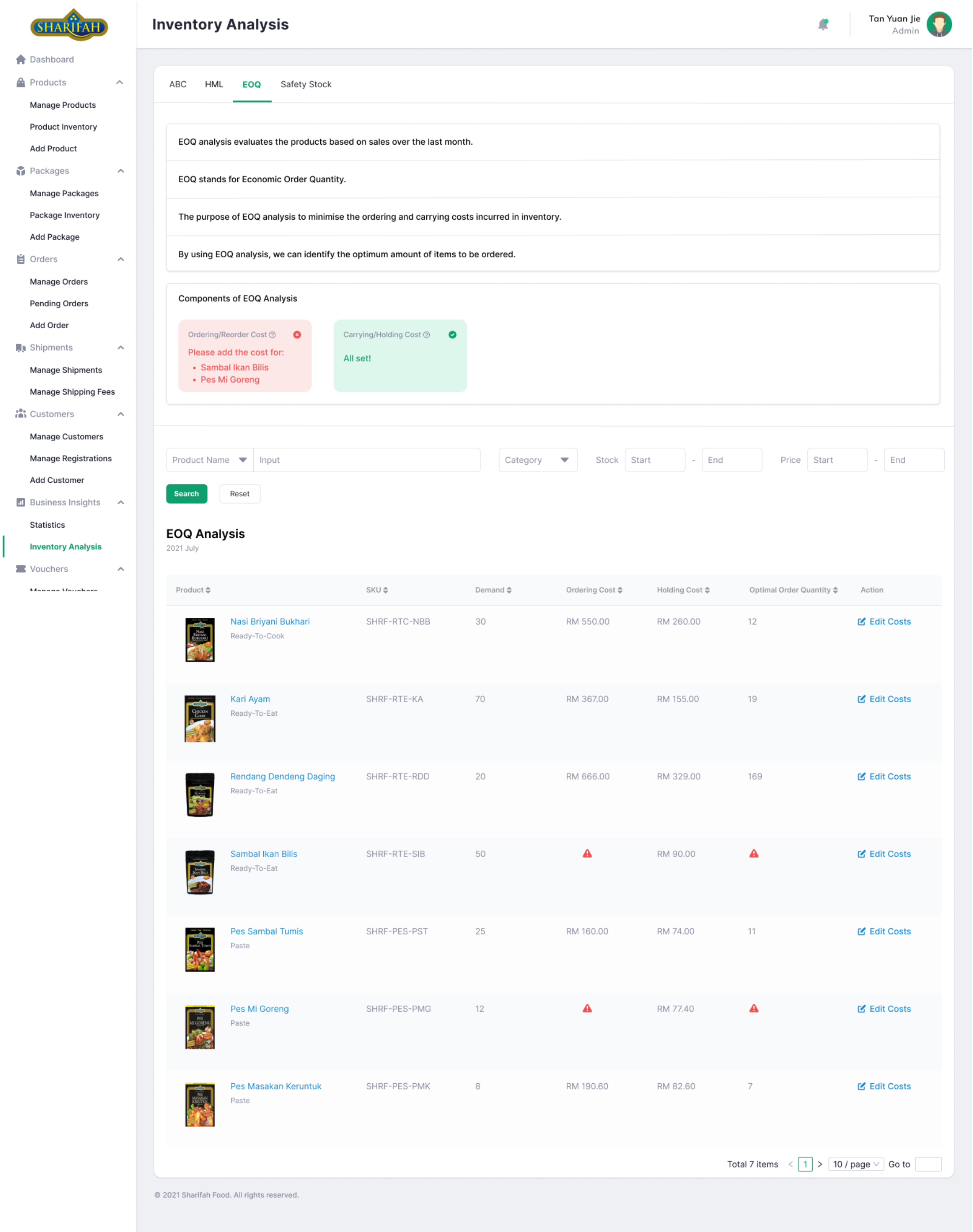
HML Analysis

2021 July

Product	SKU	Stock	Cost per Unit	Cost per Unit (%)	Grade
Rendang Dendeng Daging Ready-To-Eat	SHRF-RTE-RDD	47	RM 20.80	33.02	H
Nasi Briyani Bukhari Ready-To-Cook	SHRF-RTC-NBB	120	RM 17.60	27.94	H
Kari Ayam Ready-To-Eat	SHRF-RTE-KA	25	RM 6.50	10.32	H
Sambal Ikan Bilis Ready-To-Eat	SHRF-RTE-SIB	75	RM 5.50	8.73	M
Pes Sambal Tumis Paste	SHRF-PES-PST	90	RM 4.20	6.67	L
Pes Mi Goreng Paste	SHRF-PES-PMG	0	RM 4.20	6.67	L
Pes Masakan Keruntuk Paste	SHRF-PES-PMK	200	RM 4.20	6.67	L

© 2021 Sharifah Food. All rights reserved.

Figure 5.55: HML Analysis Prototype



Inventory Analysis

EOQ analysis evaluates the products based on sales over the last month.

EOQ stands for Economic Order Quantity.

The purpose of EOQ analysis to minimise the ordering and carrying costs incurred in inventory.

By using EOQ analysis, we can identify the optimum amount of items to be ordered.

Components of EOQ Analysis

Ordering/Reorder Cost

Please add the cost for:
- Sambal Ikan Bilis
- Pes Mi Goreng

Carrying/Holding Cost

All set!

EOQ Analysis

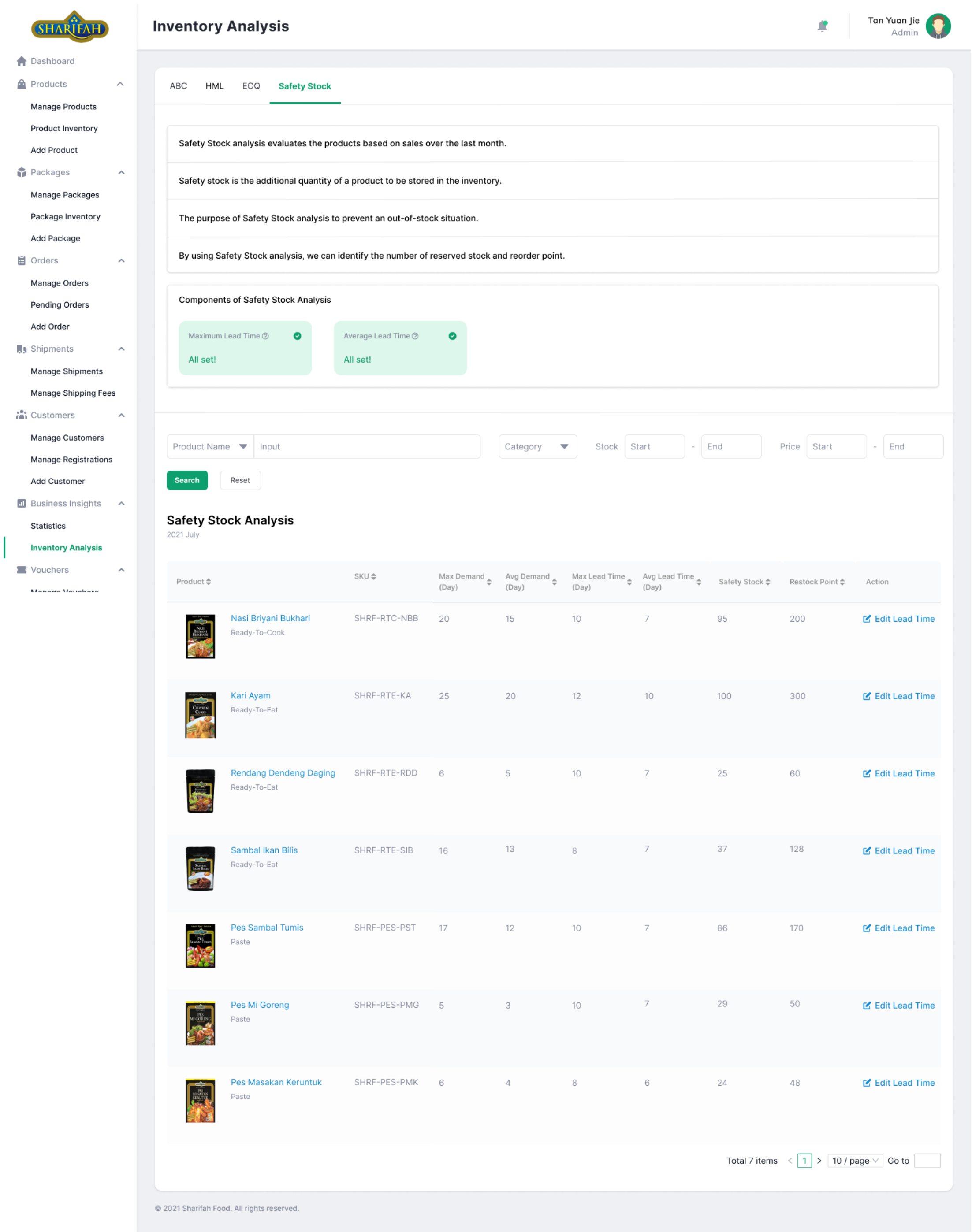
2021 July

Product	SKU	Demand	Ordering Cost	Holding Cost	Optimal Order Quantity	Action
Nasi Briyani Bukhari Ready-To-Cook	SHRF-RTC-NBB	30	RM 550.00	RM 260.00	12	Edit Costs
Kari Ayam Ready-To-Eat	SHRF-RTE-KA	70	RM 367.00	RM 155.00	19	Edit Costs
Rendang Dendeng Daging Ready-To-Eat	SHRF-RTE-RDD	20	RM 666.00	RM 329.00	169	Edit Costs
Sambal Ikan Bilis Ready-To-Eat	SHRF-RTE-SIB	50	⚠	RM 90.00	⚠	Edit Costs
Pes Sambal Tumis Paste	SHRF-PES-PST	25	RM 160.00	RM 74.00	11	Edit Costs
Pes Mi Goreng Paste	SHRF-PES-PMG	12	⚠	RM 77.40	⚠	Edit Costs
Pes Masakan Keruntuk Paste	SHRF-PES-PMK	8	RM 190.60	RM 82.60	7	Edit Costs

Total 7 items < 1 > 10 / page Go to

© 2021 Sharifah Food. All rights reserved.

Figure 5.56: EOQ Analysis Page Prototype



Inventory Analysis

Safety Stock analysis evaluates the products based on sales over the last month.

Safety stock is the additional quantity of a product to be stored in the inventory.

The purpose of Safety Stock analysis to prevent an out-of-stock situation.

By using Safety Stock analysis, we can identify the number of reserved stock and reorder point.

Components of Safety Stock Analysis

Maximum Lead Time ✓ All set!

Average Lead Time ✓ All set!

Safety Stock Analysis

2021 July

Product	SKU	Max Demand (Day)	Avg Demand (Day)	Max Lead Time (Day)	Avg Lead Time (Day)	Safety Stock	Restock Point	Action
Nasi Briyani Bukhari Ready-To-Cook	SHRF-RTC-NBB	20	15	10	7	95	200	Edit Lead Time
Kari Ayam Ready-To-Eat	SHRF-RTE-KA	25	20	12	10	100	300	Edit Lead Time
Rendang Dendeng Daging Ready-To-Eat	SHRF-RTE-RDD	6	5	10	7	25	60	Edit Lead Time
Sambal Ikan Bilis Ready-To-Eat	SHRF-RTE-SIB	16	13	8	7	37	128	Edit Lead Time
Pes Sambal Tumis Paste	SHRF-PES-PST	17	12	10	7	86	170	Edit Lead Time
Pes Mi Goreng Paste	SHRF-PES-PMG	5	3	10	7	29	50	Edit Lead Time
Pes Masakan Keruntuk Paste	SHRF-PES-PMK	6	4	8	6	24	48	Edit Lead Time

Total 7 items < 1 > 10 / page Go to

© 2021 Sharifah Food. All rights reserved.

Figure 5.57: Safety Stock Analysis Page Prototype

5.7 Summary

All in all, this chapter has discussed the detailed system design. The project moved to a more sophisticated system design based on the requirements elicited from the use case modelling in Chapter 3. A detailed three-tier client-server system architecture design has been illustrated to provide a deep comprehension of the implemented system. Besides, entity-relationship diagrams from conceptual to physical have been introduced to show the relationship between the entities in the system. In addition, the data dictionary further empowers the entity-relationship diagrams by explaining the descriptions of each data field in the database. Moreover, a prototype design of the inventory management system has been developed. The prototype design allows the users and the developers to visualize the actual systems.

CHAPTER 6

SYSTEM IMPLEMENTATION

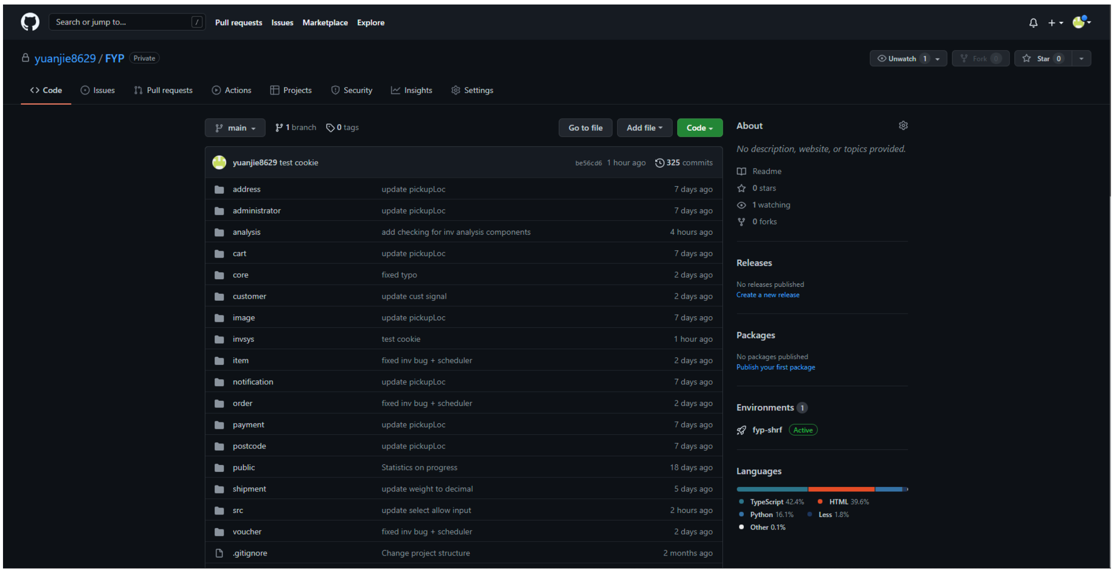
6.1 Introduction

This chapter describes the in-depth system implementation of the Sharifah Food inventory management system and the Sharifah Food E-commerce platform. The implementations to be discussed will cover the project setup, API endpoint design, implementation of the inventory management system and e-commerce platform, and system deployment.

6.2 Project Setup

In order to reduce the coupling between the inventory management system and the e-commerce platform, a decision was made to separate the inventory management system and e-commerce system into two projects. Apart from reducing the coupling between the systems, separating both systems also help increase the security of the systems, preventing malicious users from accessing the inventory management system through the e-commerce platform.

Both projects are set up with the Django framework for the back-end servers and React for the front-end applications. Moreover, Github was used as a version control system to store the versions of the developed systems. The practice of constantly committing the source code to Github has also been applied to make the projects more maintainable and allow quick reverting to the previous commits if any issues are found on the latest version. Figure 6.1 and Figure 6.2 show the Github Repository for the developed inventory management system and e-commerce platform.



yuanjie8629 / FYP Private

yuanjie8629 test cookie be56cd6 1 hour ago 325 commits

address	update pickupLoc	7 days ago
administrator	update pickupLoc	7 days ago
analysis	add checking for inv analysis components	4 hours ago
cart	update pickupLoc	7 days ago
core	fixed typo	2 days ago
customer	update cust signal	2 days ago
image	update pickupLoc	7 days ago
invsys	test cookie	1 hour ago
item	fixed inv bug + scheduler	2 days ago
notification	update pickupLoc	7 days ago
order	fixed inv bug + scheduler	2 days ago
payment	update pickupLoc	7 days ago
postcode	update pickupLoc	7 days ago
public	Statistics on progress	18 days ago
shipment	update weight to decimal	5 days ago
src	update select allow input	2 hours ago
voucher	fixed inv bug + scheduler	2 days ago
.gitignore	Change project structure	2 months ago

About

No description, website, or topics provided.

Releases

No releases published

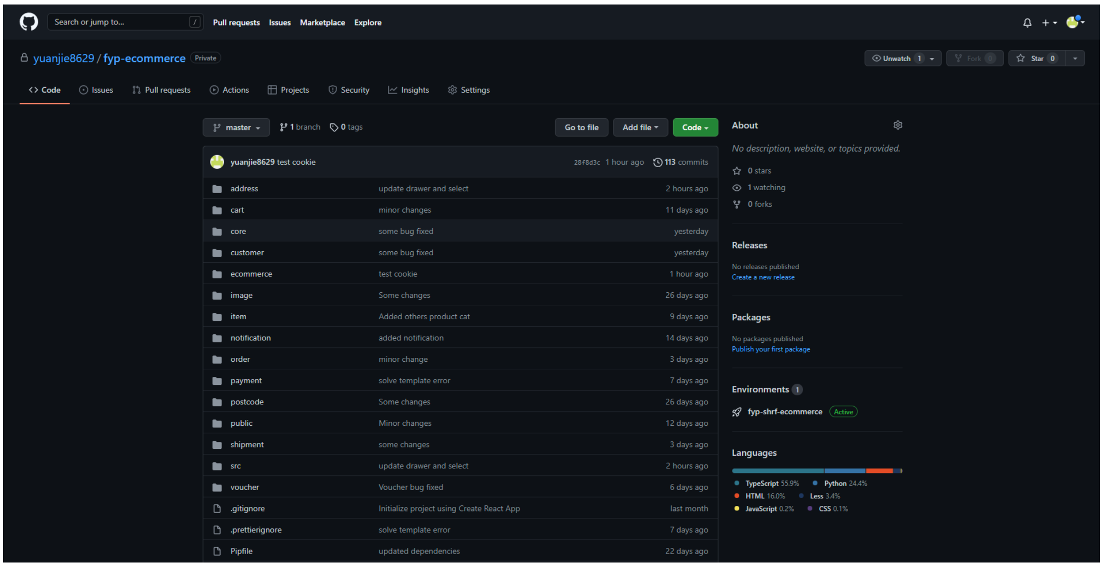
Packages

No packages published

Environments 1

fyp-shrf Active

Figure 6.1: Github Repository of Inventory Management System



yuanjie8629 / fyp-ecommerce Private

yuanjie8629 test cookie 28f8d3c 1 hour ago 113 commits

address	update drawer and select	2 hours ago
cart	minor changes	11 days ago
core	some bug fixed	yesterday
customer	some bug fixed	yesterday
ecommerce	test cookie	1 hour ago
image	Some changes	26 days ago
item	Added others product cat	9 days ago
notification	added notification	14 days ago
order	minor change	3 days ago
payment	solve template error	7 days ago
postcode	Some changes	26 days ago
public	Minor changes	12 days ago
shipment	some changes	3 days ago
src	update drawer and select	2 hours ago
voucher	Voucher bug fixed	6 days ago
.gitignore	Initialize project using Create React App	last month
.prettierignore	solve template error	7 days ago
Pipfile	updated dependencies	22 days ago

About

No description, website, or topics provided.

Releases

No releases published

Packages

No packages published

Environments 1

fyp-shrf-ecommerce Active

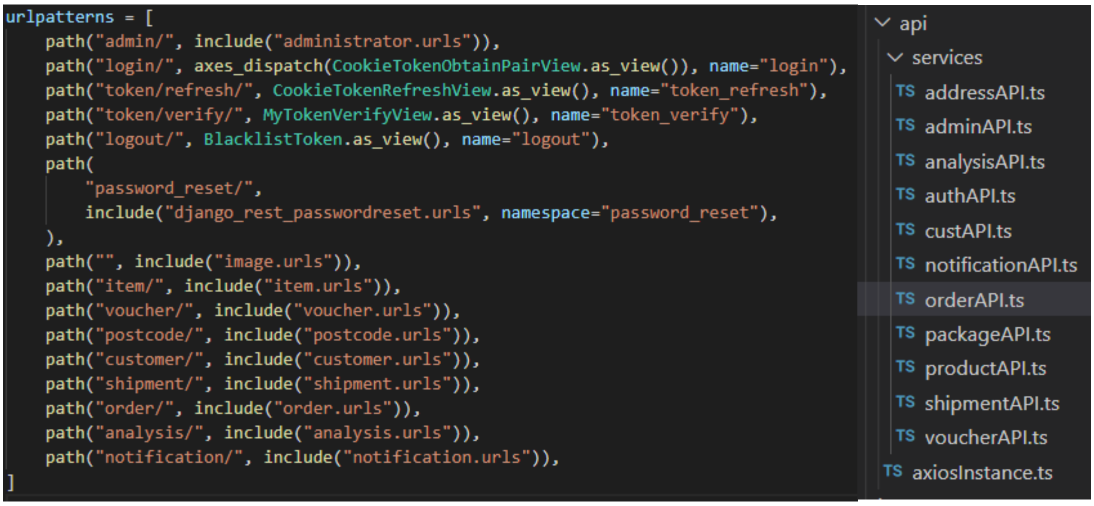
Figure 6.2: Github Repository of E-commerce Platform

6.3 API Endpoint Design

In this project, API was used along with the HTTP protocol to connect the back-end server to the front-end application. An API endpoint is a URL that allows the API protocol to receive and send the request to the systems it interacts with. In order to perform CRUD operation on a data model, the front-end application would require the API endpoints of the back-end server.

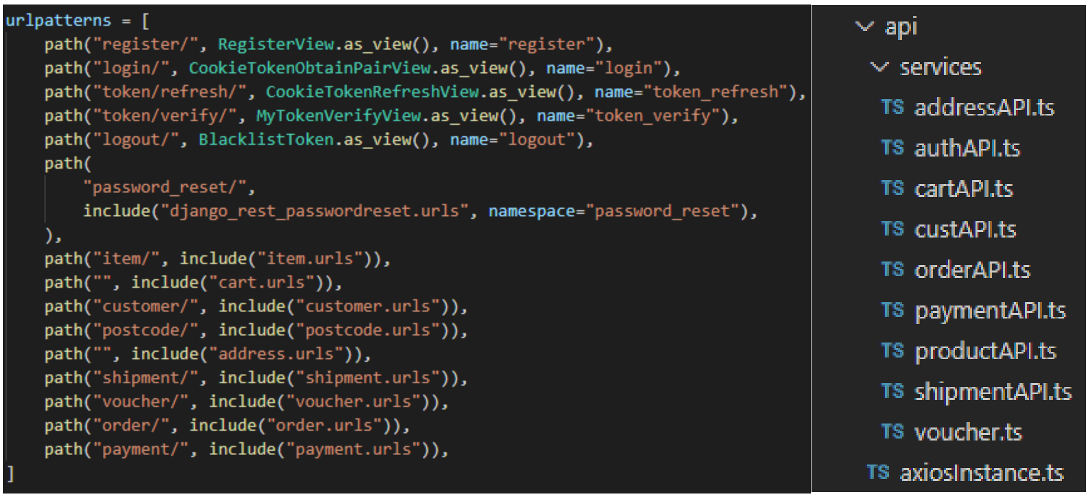
There are five commonly used HTTP methods for the API endpoints developed in the inventory management system and the e-commerce platform: GET, POST, PUT, PATCH, and DELETE. The GET method was used to retrieve the specified resource data from the server. The POST method was mainly used to create a new resource for the server. The PUT and PATCH methods were both used to update the resources. Nonetheless, the PUT method is only used to update the entire data, while the PATCH method partially updates the data.

The Django REST framework was used to develop the API endpoints for both the inventory management system and e-commerce platform. Besides, since Django supports the MTV software design pattern, the API endpoints were developed for the models in the systems, which can also be referred to as tables in the database. Figure 6.3 and Figure 6.4 show the overview of API routes available in the inventory management system and e-commerce platform.



```
urlpatterns = [
    path("admin/", include("administrator.urls")),
    path("login/", axes_dispatch(CookieTokenObtainPairView.as_view()), name="login"),
    path("token/refresh/", CookieTokenRefreshView.as_view(), name="token_refresh"),
    path("token/verify/", MyTokenVerifyView.as_view(), name="token_verify"),
    path("logout/", BlacklistToken.as_view(), name="logout"),
    path(
        "password_reset/",
        include("django_rest_passwordreset.urls", namespace="password_reset"),
    ),
    path("", include("image.urls")),
    path("item/", include("item.urls")),
    path("voucher/", include("voucher.urls")),
    path("postcode/", include("postcode.urls")),
    path("customer/", include("customer.urls")),
    path("shipment/", include("shipment.urls")),
    path("order/", include("order.urls")),
    path("analysis/", include("analysis.urls")),
    path("notification/", include("notification.urls")),
]
```

Figure 6.3: Overview of API Routes in Inventory Management System



```
urlpatterns = [
    path("register/", RegisterView.as_view(), name="register"),
    path("login/", CookieTokenObtainPairView.as_view(), name="login"),
    path("token/refresh/", CookieTokenRefreshView.as_view(), name="token_refresh"),
    path("token/verify/", MyTokenVerifyView.as_view(), name="token_verify"),
    path("logout/", BlacklistToken.as_view(), name="logout"),
    path(
        "password_reset/",
        include("django_rest_passwordreset.urls", namespace="password_reset"),
    ),
    path("item/", include("item.urls")),
    path("", include("cart.urls")),
    path("customer/", include("customer.urls")),
    path("postcode/", include("postcode.urls")),
    path("", include("address.urls")),
    path("shipment/", include("shipment.urls")),
    path("voucher/", include("voucher.urls")),
    path("order/", include("order.urls")),
    path("payment/", include("payment.urls")),
]
```

```
∨ api
  ∨ services
    TS addressAPI.ts
    TS authAPI.ts
    TS cartAPI.ts
    TS custAPI.ts
    TS orderAPI.ts
    TS paymentAPI.ts
    TS productAPI.ts
    TS shipmentAPI.ts
    TS voucher.ts
  TS axiosInstance.ts
```

Figure 6.4: Overview of API Routes in E-commerce Platform

6.3.1 API Endpoints for Inventory Management System

This section lists all the API routes available in the inventory management system.

Table 6.1: API Endpoints for Inventory Management System - Admin

Model: admin		
Method	**Route**	**Desription**
POST	/login/	Login admin (JWT token)
POST	/logout/	Logout admin (JWT token)
POST	/token/verify/	Verify JWT token
POST	/token/refresh/	Refresh JWT token (issue new refresh token)
GET	/admin/<int: id>/	Retrieve admin details by admin id
PUT	/admin/<int: id>/	Update admin details by admin id
PUT	/admin/<int: id>/change_pass/	Change admin’s password by admin id
POST	/password_reset/	Request a reset password token with email
POST	/password_reset/validate_token/	Validate a reset password token (return HTTP status 200 for the valid token)
POST	/password_reset/confirm/	Set the password of the admin using a valid token

Table 6.2: API Endpoints for Inventory Management System - Customer

Model: cust		
Method	**Route**	**Desription**
GET	/customer/{query}	Retrieve an overview of all customer details
GET	/customer/<int: id>/	Retrieve a customer details by customer id
POST	/customer/	Create a new customer
PUT	/customer/<int: id>/	Update an existing customer by customer id
PATCH	/customer/<int: id>/	Partially update an existing customer by customer id
POST	/customer/status/update/	Bulk update the customers' status (activate/suspend)

Table 6.3: API Endpoints for Inventory Management System - Customer Registration

Model: cust_pos_reg		
Method	**Route**	**Desription**
GET	/customer/registration/{query}	Retrieve an overview of all customer registration details
GET	/customer/registration/<int: id>/	Retrieve a customer registration details by registration id
PATCH	/customer/registration/update/	Bulk update the agent and dropshipper registration status (accept/reject)

Table 6.4: API Endpoints for Inventory Management System - Notification

Model: notification		
Method	**Route**	**Desription**
GET	/notification/	Retrieve all notification information
PATCH	/notification/read/	Bulk read the notification (update read fields of the notifications to indicate that notifications have been read)

Table 6.5: API Endpoints for Inventory Management System - Order

Model: order		
Method	**Route**	**Desription**
GET	/order/{query}	Retrieve overview of all order details
GET	/order/<int: id>/	Retrieve a customer registration details by order number (id)
PATCH	/order/track_num/update/	Bulk update the orders' tracking number
PATCH	/order/pickup/update/	Bulk update the orders' pickup date
PATCH	/order/cancel/	Bulk update the orders' status to 'cancel'
GET	/order/invoice/<int: id>/	Retrieve an order invoice by order number (id)
POST	/order/invoices/bulk/	Bulk retrieve the order invoices by order numbers (ids)

Table 6.6: API Endpoints for Inventory Management System - Item

Model: item, product		
Method	**Route**	**Desription**
GET	/item/product/prev/{query}	Retrieve previews of all products
GET	/item/product/prev/all/	Retrieve previews of all products without pagination
GET	/item/product/<int: id>/	Retrieve product details by product id
POST	/item/product/	Create a new product (multipart/form-data)
PATCH	/item/product/<int: id>/	Partially update an existing product by product id
DELETE	/item/product/<int: id>/	Delete an existing product by product id
PATCH	/item/product/bulk/update/	Bulk update the products (status)
Model: item, package		
Method	**Route**	**Desription**
GET	/item/package/prev/{query}	Retrieve previews of all packages
GET	/item/package/<int: id>/	Retrieve package details by package id
POST	/item/package/	Create a new package (multipart/form-data)
PATCH	/item/package/<int: id>/	Partially update an existing package by package id
DELETE	/item/package/<int: id>/	Delete an existing package by package id
PATCH	/item/package/bulk/update/	Bulk update the package (status)
Model: item		
PATCH	/item/bulk/delete/	Bulk delete the items (products/packages)

Table 6.7: API Endpoints for Inventory Management System - Postcode

Model: postcode		
Method	Route	Desription
GET	/state/	Retrieve all states in Malaysia
GET	/postcode/	Retrieve all postcodes and cities in Malaysia
GET	/postcode/<int: id>/	Retrieve a postcode details by postcode id

Table 6.8: API Endpoints for Inventory Management System - Shipping Fee

Model: shipping_fee		
Method	Route	Desription
GET	/shipment/shipping_fee/{query}	Retrieve an overview of all shipping fee records
GET	/shipment/shipping_fee/<int: id>/	Retrieve a shipping fee details by shipping fee id
POST	/shipment/shipping_fee/	Create a new shipping fee
PUT	/shipment/shipping_fee/<int: id>/	Update an existing shipping fee by shipping fee id
PATCH	/shipment/shipping_fee/<int: id>/	Partially update an existing shipping fee by shipping fee id
POST	/shipment/shipping_fee/bulk/delete/	Bulk delete the shipping fees
GET	/shipment/shipping_fee/state/	Retrieve the states that do not contain any shipping fee record

Table 6.9: API Endpoints for Inventory Management System - Pickup Location

Model: pickup_loc		
Method	Route	Desription
GET	/shipment/pickup_loc/{query}	Retrieve all pickup location
GET	/shipment/pickup_loc/<int: id>/	Retrieve a pickup location details by pickup location id
POST	/shipment/pickup_loc/	Create a new pickup location

Table 6.9 (Continued)

PUT	/shipment/pickup_loc/<int: id>/	Update an existing pickup location by pickup location id
PATCH	/shipment/ pickup_loc/<int: id>/	Partially update an existing pickup location by pickup location id
POST	/shipment/ pickup_loc/bulk/delete/	Bulk delete the pickup locations

Table 6.10: API Endpoints for Inventory Management System - Voucher

Model: voucher		
Method	**Route**	**Desription**
GET	/voucher/{query}	Retrieve an overview of all voucher
GET	/voucher/<int: id>/	Retrieve a voucher details by voucher id
POST	/voucher/	Create a new voucher
PUT	/voucher/<int: id>/	Update an existing voucher by voucher id
PATCH	/voucher/<int: id>/	Partially update an existing voucher by voucher id
DELETE	/voucher/<int: id>/	Delete an existing voucher by voucher id
POST	/voucher/bulk/update/	Bulk update the vouchers
POST	/voucher/bulk/delete/	Bulk delete the vouchers

Table 6.11: API Endpoints for Inventory Management System - Analysis

Analysis		
Method	**Route**	**Desription**
GET	/analysis/to_do_list/	Retrieve a list of to-do list summary (pending order shipment, pending order pickup, unpaid orders, out-of-stock products, out-of-stock packages, pending agent registration and pending dropshipper registration)
GET	/analysis/statistics/	Retrieve a list of statistics over a specified period (sales, profit, new customer and new order)
GET	/analysis/key_metrics/{query}	Retrieve a list of key metrics data over a specified period based on the query (from_date, to_date, key, date_type)
GET	/analysis/key_metrics/summary/{query}	Retrieve a list of total key metrics data values over a specified period based on the query (from_date, to_date)
GET	/analysis/key_metrics/report/{query}	Retrieve an Excel report of key metrics analysis over a specified period based on the query (from_date, to_date, date type)
GET	/analysis/abc/{query}	Retrieve ABC analysis of all products

Table 6.11 (Continued)

GET	/analysis/hml/{query}	Retrieve HML analysis of all products
GET	/analysis/eoq/{query}	Retrieve EOQ analysis of all products
GET	/analysis/ss/{query}	Retrieve SS analysis of all products
GET	/item/product/ranking/sales/{query}	Retrieve product ranking by sales based on the query (from_date, to_date, category, limit, offset)
GET	/item/product/ranking/units/{query}	Retrieve product ranking by units sold based on the query (from_date, to_date, category, limit, offset)
GET	/item/package/ranking/sales/{query}	Retrieve package ranking by sales based on the query (from_date, to_date, limit, offset)
GET	/item/package/ranking/units/{query}	Retrieve package ranking by units sold based on the query (from_date, to_date, limit, offset)

6.3.2 API Endpoints for E-commerce Platform

This section lists all the API routes available in the e-commerce platform.

Table 6.12: API Endpoints for E-commerce Platform - Address

Model: address		
Method	**Route**	**Desription**
GET	/address/	Retrieve a list of addresses that belong to the requesting user
GET	/address/<int: id>/	Retrieve an address details by address id
POST	/address/	Create a new address for a user
PUT	/address/<int: id>/	Update an existing address by address id
PATCH	/address/<int: id>/	Partially update an existing address by address id
DELETE	/address/<int: id>/	Delete an existing address by address id
GET	/address/default/	Retrieve the default address of the request user

Table 6.13: API Endpoints for E-commerce Platform – Cart

Model: cart		
Method	**Route**	**Desription**
GET	/cart/<int: id>/	Retrieve a cart details by user id
POST	/cart/<int: id>/add/	Add a cart item with quantity to the cart by user id. Return HTTP 406 status if the item stock is less than the quantity
POST	/cart/<int: id>/remove/	Reduce a cart item quantity from the cart by user id. If the item stock is less than the quantity or the specified quantity is 0, remove the item from the cart

Table 6.13 (Continued)

POST	/cart/<int: id>/set/	Set the quantity of an existing cart item to the quantity specified. Return HTTP 406 status if the item stock is less than the quantity specified
POST	/cart/details/	Check the quantity of cart items with their stock and calculate the subtotal price, shipping fee, and total price for the unregistered user's cart

Table 6.14: API Endpoints for E-commerce Platform - Customer

Model: cust		
Method	**Route**	**Desription**
POST	/login/	Login customer (JWT token)
POST	/logout/	Logout customer (JWT token)
POST	/token/verify/	Verify JWT token
POST	/token/refresh/	Refresh JWT token (issue new refresh token)
GET	/customer/<int: id>/	Retrieve customer details by customer id
PUT	/customer/<int: id>/	Update customer details by customer id
PATCH	/customer/<int: id>/	Partially update customer details by customer id
PUT	/customer/<int: id>/change_pass/	Change customer's password by customer id
POST	/customer/position/registration/	Create a new agent/dropshipper position registration record
POST	/password_reset/	Request a reset password token with email

Table 6.14 (Continued)

POST	/password_reset/validate_token/	Validate a reset password token (return HTTP status 200 for the valid token)
POST	/password_reset/confirm/	Set the password of the admin using a valid token

Table 6.15: API Endpoints for E-commerce Platform - Item

Model: item, product, package		
Method	**Route**	**Desription**
GET	/item/{query}	Retrieve a list of items (products and packages)
GET	/item/active/{query}	Retrieve a list of items (products and packages) with active status
GET	/item/product/{query}	Retrieve a list of products with active status
GET	/item/package/{query}	Retrieve a list of packages with active status
GET	/item/<int: id>/	Retrieve item details by item id

Table 6.16: API Endpoints for E-commerce Platform - Order

Model: order		
Method	**Route**	**Desription**
GET	/order/{query}	Retrieve an overview of all orders that belong to the requesting user
GET	/order/<int: id>/	Retrieve an order details by order number (id)
POST	/order/search/	Check if the order exists with the given order number and email. It is used for the order searching feature for unregistered users.
POST	/order/	Create a new order

Table 6.17: API Endpoints for E-commerce Platform - Payment

Model: payment		
Method	**Route**	**Desription**
POST	/payment/create_checkout_session/	Create a checkout session with the Stripe payment gateway to obtain the payment URL for the order
POST	/payment/stripe_webhooks/	Retrieve the real-time payment information notification of an order from the payment gateway.

Table 6.18: API Endpoints for E-commerce Platform - Postcode

Model: postcode		
Method	**Route**	**Desription**
GET	/state/	Retrieve all states in Malaysia
GET	/postcode/	Retrieve all postcodes and cities in Malaysia
GET	/postcode/<int: id>/	Retrieve a postcode details by postcode id

Table 6.19: API Endpoints for E-commerce Platform - Pickup Location

Model: pickup_loc		
Method	**Route**	**Desription**
GET	/shipment/pickup_loc/	Retrieve all pickup location

Table 6.20: API Endpoints for E-commerce Platform - Voucher

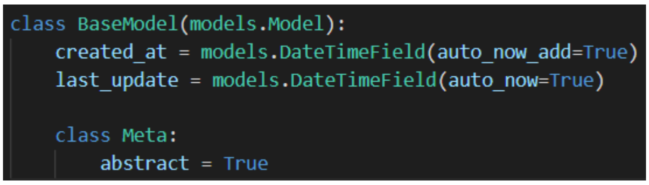
Model: voucher		
Method	**Route**	**Desription**
POST	/voucher/check/	Check if the voucher is valid for the requesting user. Return HTTP status 404 if the voucher is invalid.
POST	/voucher/check/auto/	Check if there is an auto-applied voucher during the order checkout for the requesting user. Return HTTP status 404 if the voucher is invalid.

6.4 System Enhancement

In order to bring the project to the enterprise level, some further implementations were performed to enhance the inventory management system and e-commerce platform. This section covers the implementations of the model inheritance and polymorphic relationships, soft delete design pattern, API pagination, data audit trail, and database transaction management and concurrency control.

6.4.1 Model Inheritance and Polymorphic Relationships

The project uses model inheritance to provide the base attributes to all the models inherited from the base model. The base model contains “created_at” and “last_update” fields. The “created_at” field holds the creation time of an instance, while the “last_update” field holds the latest time when a model instance is updated. Figure 6.5 shows the code snippet of the base model developed.



```
class BaseModel(models.Model):
    created_at = models.DateTimeField(auto_now_add=True)
    last_update = models.DateTimeField(auto_now=True)

    class Meta:
        abstract = True
```

Figure 6.5: Code Snippet of Base Model

Besides, model inheritance was also used to solve the polymorphic relationships. A polymorphic relationship refers to the ability of a model to belong to more than one of the other models on a single association. Simply put, a model can take many forms. In the project, there are several polymorphic relationships between the models. For instance, an item could be a product or a package. Besides, an order could also be a shipment order or a pickup order.

One of the challenges of designing the database schemas is resolving the polymorphic relationships. The main problem with polymorphic relationships in the database design is the referential integrity. There are several common solutions to the polymorphic relationships, such as the single table, class table, and concrete table inheritances.

Single table inheritance combines all the subclasses into a single table, and it is the most straightforward solution to solve the polymorphic relationship. Nonetheless, the issue of the single table inheritance is that it has a high coupling within the inheritance hierarchy and will result in a lot of null values. For example, if the item is a package, then the fields related to the product will be empty.

Class table inheritance represents the superclass and all subclasses with one relation for each class. However, this solution increases the complexity of performing a join operation and thus becomes less performant.

Concrete table inheritance combines the superclass attributes with each subclass, resulting in one relation for each combined superclass and subclass. This solution reduces the null values issues encountered by the single table inheritance. However, it was discovered during database design that it complicates the database schema.

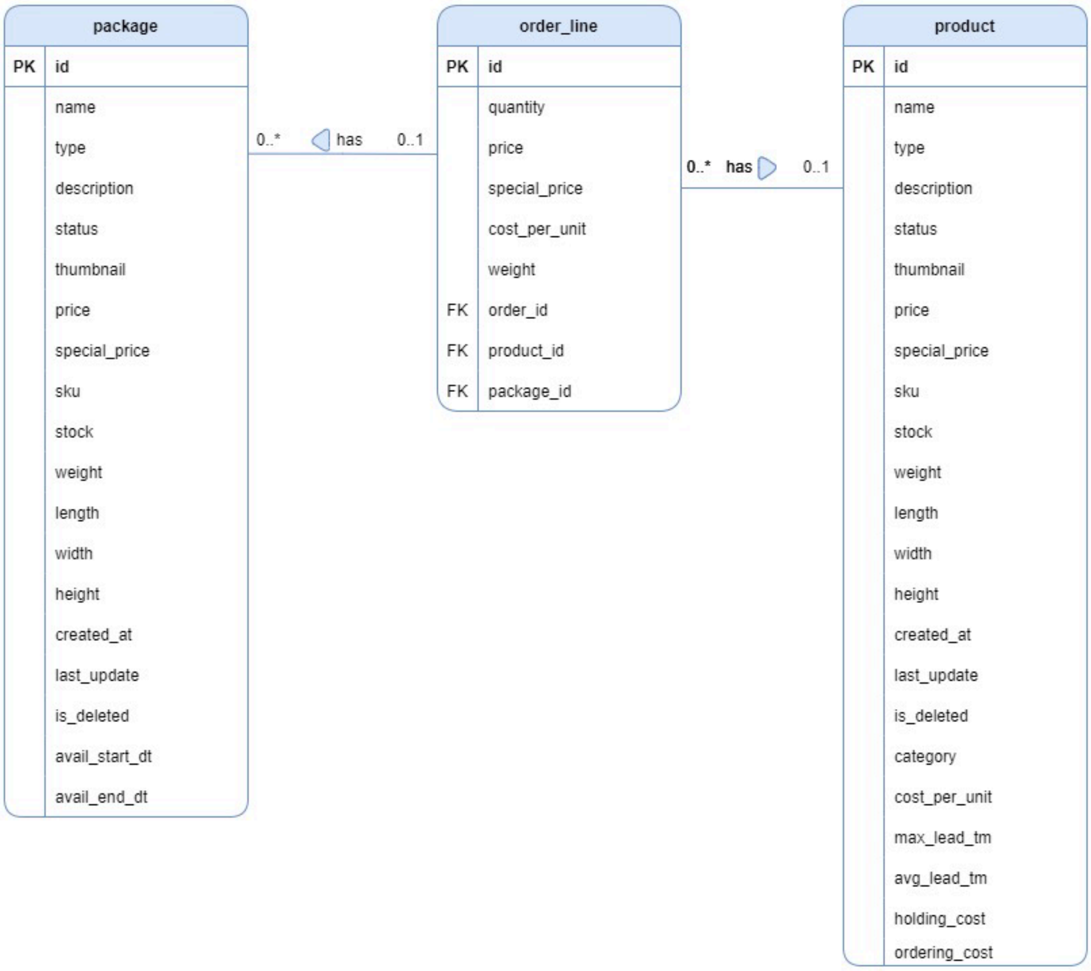


Figure 6.6: Concrete Table Inheritance

Figure 6.6 shows an example segment of the database schema using the concrete table inheritance. As we can see, the order_line relation would require foreign key references to each subclass of the item relation, which are the product and package relations, to guarantee the referential integrity. If there are more subclasses of items in the future implementation, the order_line relation would require more foreign key references to the subclasses, making the relations more difficult to maintain.

On the grounds of this, a mapped superclass table inheritance was used to overcome the problem of polymorphic relationships. The mapped superclass table inheritance contains one relation for the superclass and one relation for each subclass. Besides, the subclass’s primary key is a foreign key reference to the superclass's primary key. Figure 6.7 shows the outcome of the mapped superclass table inheritance that will be used to solve the polymorphic relationships in the project.

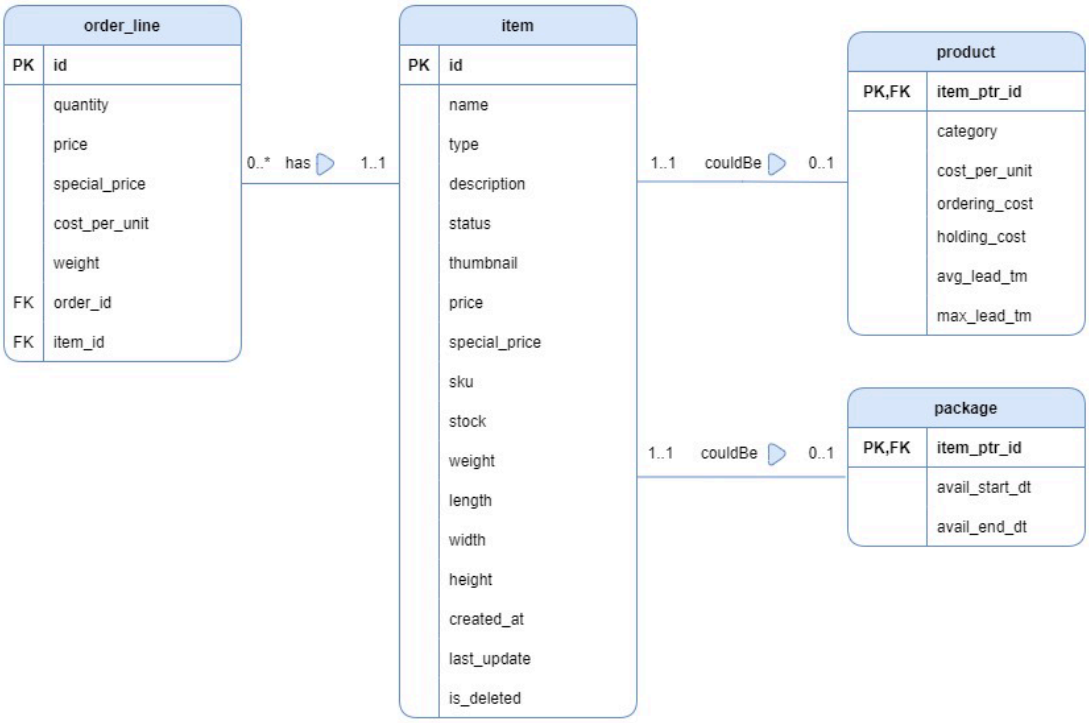
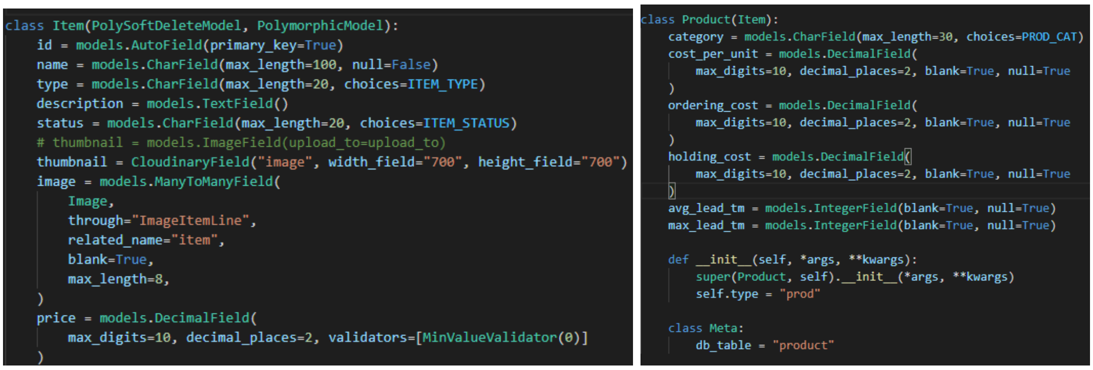


Figure 6.7: Mapped Superclass Table Inheritance

In order to make the model inheritance for the polymorphic relationships easier, a Django library named django-polymorphic was used. Figure 6.8 shows the code snippets of the polymorphic model inheritance of Item and Product models.



```
class Item(PolySoftDeleteModel, PolymorphicModel):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=100, null=False)
    type = models.CharField(max_length=20, choices=ITEM_TYPE)
    description = models.TextField()
    status = models.CharField(max_length=20, choices=ITEM_STATUS)
    # thumbnail = models.ImageField(upload_to=upload_to)
    thumbnail = CloudinaryField("image", width_field="700", height_field="700")
    image = models.ManyToManyField(
        Image,
        through="ImageItemLine",
        related_name="item",
        blank=True,
        max_length=8,
    )
    price = models.DecimalField(
        max_digits=10, decimal_places=2, validators=[MinValueValidator(0)]
    )
```

```
class Product(Item):
    category = models.CharField(max_length=30, choices=PROD_CAT)
    cost_per_unit = models.DecimalField(
        max_digits=10, decimal_places=2, blank=True, null=True
    )
    ordering_cost = models.DecimalField(
        max_digits=10, decimal_places=2, blank=True, null=True
    )
    holding_cost = models.DecimalField(
        max_digits=10, decimal_places=2, blank=True, null=True
    )
    avg_lead_tm = models.IntegerField(blank=True, null=True)
    max_lead_tm = models.IntegerField(blank=True, null=True)

    def __init__(self, *args, **kwargs):
        super(Product, self).__init__(*args, **kwargs)
        self.type = "prod"

    class Meta:
        db_table = "product"
```

Figure 6.8: Code Snippets of Polymorphic Model Inheritance

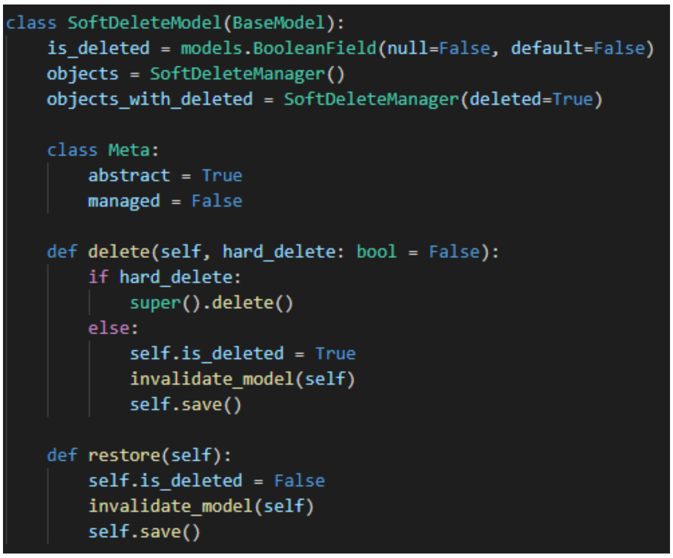
6.4.2 Soft Delete Design Pattern

Soft delete is a system design pattern used widely by business applications, and it allows the system to delete the record without actually erasing the record.

Since many administrators will be operating the inventory management system, it is essential to ensure that the admins do not simply or accidentally delete the records. Thus, this project utilizes the soft delete design pattern to reduce the risk of losing the data.

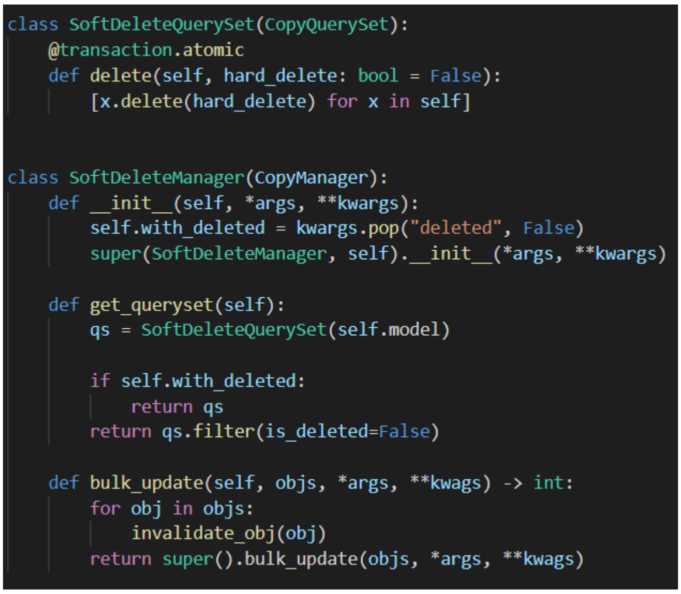
In order to apply the soft delete pattern, a soft-delete base model was developed so that the other models that apply the soft delete pattern could inherit from the base model. The soft-delete base model includes the “is_deleted” field to indicate that the record is deleted. For example, if the admin deleted the product, the system will mark the “is_deleted” field as “True” instead of performing a delete operation on the product.

Besides, a custom Django model manager for soft deletion was also developed to provide the query operations that exclude soft-deleted records. By using the custom soft-delete model manager, the system will automatically filter out the deleted records without needing to explicitly indicate the filter in the query set. In addition, the soft delete model also provides the methods to hard delete the data and restore the soft-deleted records. In an effort to ease the super admin to restore the deleted records, the soft deletion concept has also been integrated with the Django administration site.



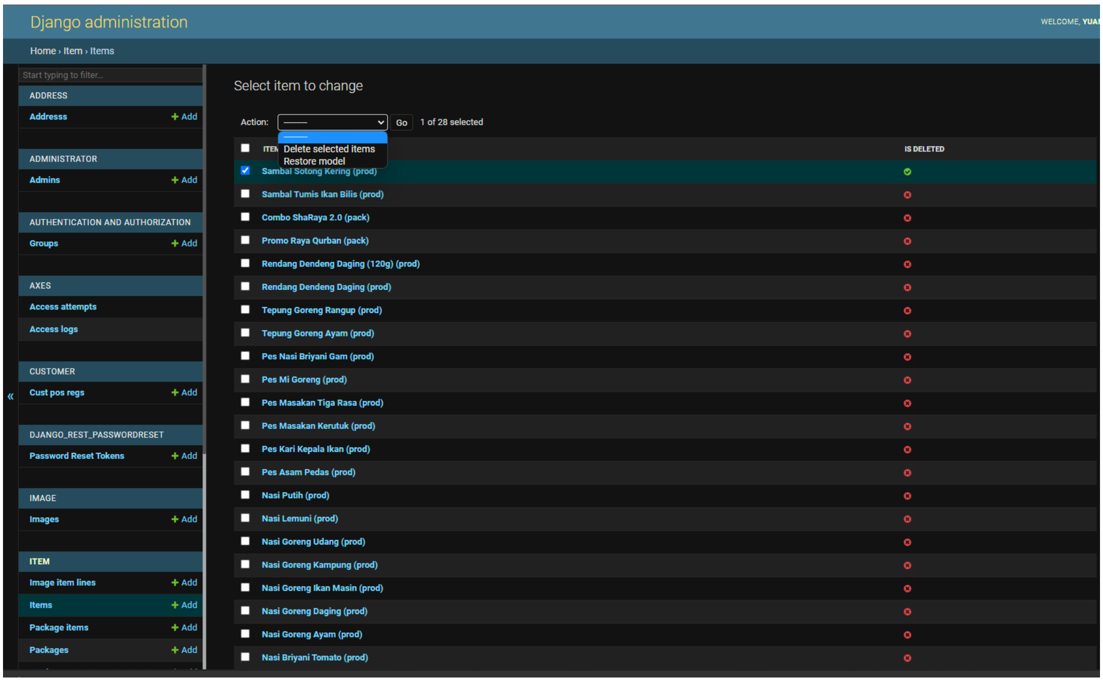
```
class SoftDeleteModel(BaseModel):
    is_deleted = models.BooleanField(null=False, default=False)
    objects = SoftDeleteManager()
    objects_with_deleted = SoftDeleteManager(deleted=True)

    class Meta:
        abstract = True
        managed = False

    def delete(self, hard_delete: bool = False):
        if hard_delete:
            super().delete()
        else:
            self.is_deleted = True
            invalidate_model(self)
            self.save()

    def restore(self):
        self.is_deleted = False
        invalidate_model(self)
        self.save()
```

Figure 6.9: Code Snippet of Soft Delete Model



```python
class SoftDeleteQuerySet(CopyQuerySet):
    @transaction.atomic
    def delete(self, hard_delete: bool = False):
        [x.delete(hard_delete) for x in self]


class SoftDeleteManager(CopyManager):
    def __init__(self, *args, **kwargs):
        self.with_deleted = kwargs.pop("deleted", False)
        super(SoftDeleteManager, self).__init__(*args, **kwargs)

    def get_queryset(self):
        qs = SoftDeleteQuerySet(self.model)

        if self.with_deleted:
            return qs
        return qs.filter(is_deleted=False)

    def bulk_update(self, objs, *args, **kwags) -> int:
        for obj in objs:
            invalidate_obj(obj)
        return super().bulk_update(objs, *args, **kwags)
```

Figure 6.10: Code Snippet of Soft Delete Manager



Select item to change

Action: ——— 1 of 28 selected

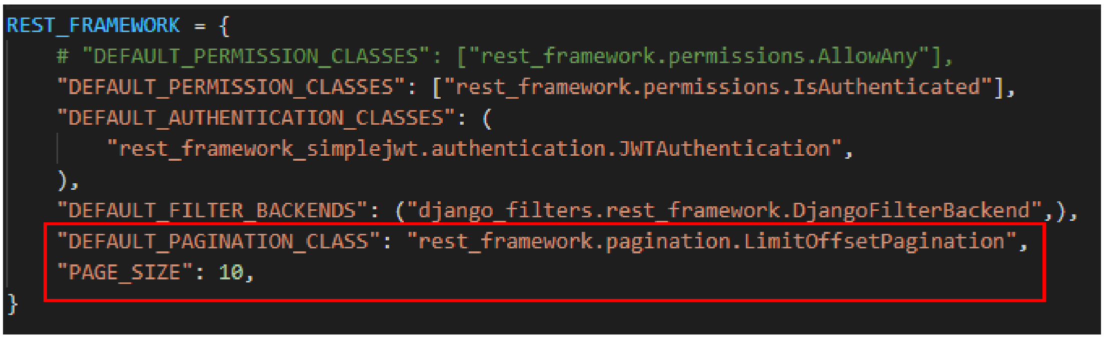
	ITEM	IS DELETED
☑	Sambal Sotong Kering (prod)	✓
□	Sambal Tumis Ikan Bilis (prod)	✗
□	Combo ShaRaya 2.0 (pack)	✗
□	Promo Raya Qurban (pack)	✗
□	Rendang Dendeng Daging (120g) (prod)	✗
□	Rendang Dendeng Daging (prod)	✗
□	Tepung Goreng Rangup (prod)	✗
□	Tepung Goreng Ayam (prod)	✗
□	Pes Nasi Briyani Gam (prod)	✗
□	Pes Mi Goreng (prod)	✗
□	Pes Masakan Tiga Rasa (prod)	✗
□	Pes Masakan Kerutuk (prod)	✗
□	Pes Kari Kepala Ikan (prod)	✗
□	Pes Asam Pedas (prod)	✗
□	Nasi Putih (prod)	✗
□	Nasi Lemuni (prod)	✗
□	Nasi Goreng Udang (prod)	✗
□	Nasi Goreng Kampung (prod)	✗
□	Nasi Goreng Ikan Masin (prod)	✗
□	Nasi Goreng Daging (prod)	✗
□	Nasi Goreng Ayam (prod)	✗
□	Nasi Briyani Tomato (prod)	✗

Figure 6.11: Soft Deletion Concept in Django Administration Site

6.4.3 API Pagination

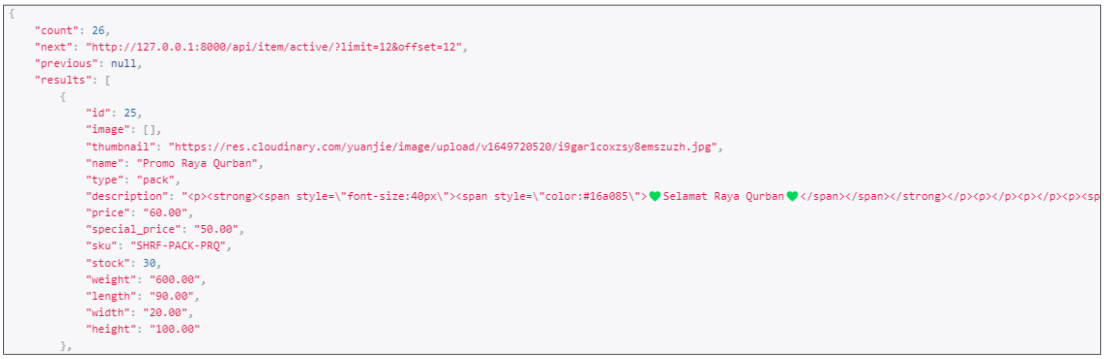
As the system grows, there will be more data. Without proper limits on the amount of the data sent from an API, it could return millions of data, which would result in extraneous network traffic. Therefore, a pagination concept was applied to the project.

The inventory management system and e-commerce platform implemented the offset pagination with the Django REST framework. Additionally, the front-end application also implemented the paging UI components to accommodate the pagination implemented in the back-end server.



```
REST_FRAMEWORK = {
    # "DEFAULT_PERMISSION_CLASSES": ["rest_framework.permissions.AllowAny"],
    "DEFAULT_PERMISSION_CLASSES": ["rest_framework.permissions.IsAuthenticated"],
    "DEFAULT_AUTHENTICATION_CLASSES": (
        "rest_framework_simplejwt.authentication.JWTAuthentication",
    ),
    "DEFAULT_FILTER_BACKENDS": ("django_filters.rest_framework.DjangoFilterBackend",),
    "DEFAULT_PAGINATION_CLASS": "rest_framework.pagination.LimitOffsetPagination",
    "PAGE_SIZE": 10,
}
```

Figure 6.12: API Offset Pagination Settings



```
{
    "count": 26,
    "next": "http://127.0.0.1:8000/api/item/active/?limit=12&offset=12",
    "previous": null,
    "results": [
        {
            "id": 25,
            "image": [],
            "thumbnail": "https://res.cloudinary.com/yuanjie/image/upload/v1649720520/i9gar1coxzsy8emszuzh.jpg",
            "name": "Promo Raya Qurban",
            "type": "pack",
            "description": "<p><strong><span style=\"font-size:40px\"><span style=\"color:#16a085\">💚Selamat Raya Qurban💚</span></span></strong></p><p></p><p></p><p><sp
            "price": "60.00",
            "special_price": "50.00",
            "sku": "SHRF-PACK-PRQ",
            "stock": 30,
            "weight": "600.00",
            "length": "90.00",
            "width": "20.00",
            "height": "100.00"
        },
```

Figure 6.13: Paginated API Response

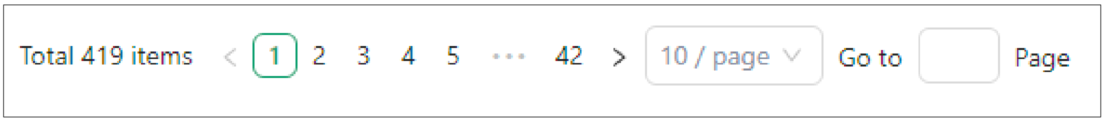


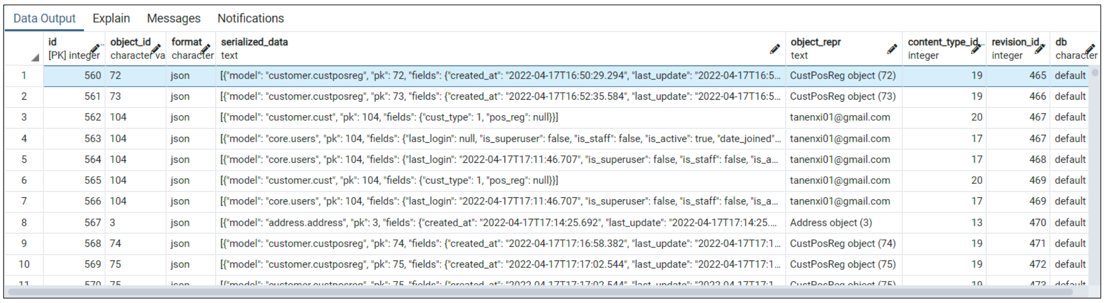
Figure 6.14: Example of Paging UI Component

6.4.4 Data Audit Trail

A data audit trail is a detailed log that stores the data-related operations, and it consists of the changes made to the data and the time when the changes occur. As mentioned, the inventory management system will be used by many administrators, and therefore, any updates on the data should be reflected to make the data reversion painless.

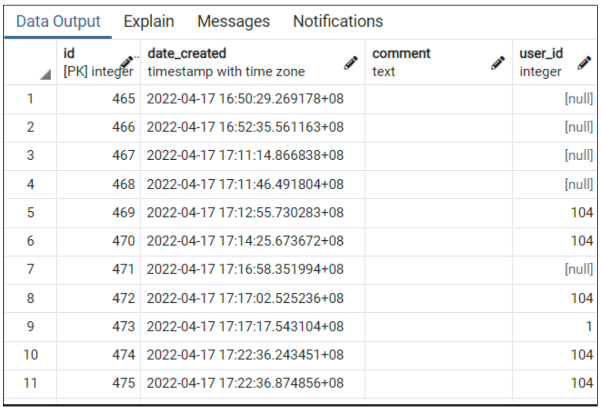
In order to automate the data audit trail process, a Django library named django-reversion was used. By using the django-reversion, the model instance can be rolled back to any point in the history, even for a deleted model instance.

After installing and configuring the django-reversion library, two tables were created in the database: reversion_revision and reversion_version. The reversion_revision table stores the date created and the user who triggered the action. On the other hand, the reversion_version stores the serialized data and the id of the model instance being modified. The django-reversion has also been integrated with the Django administration site to provide a better interface for the super admin to manage the data reversion history.



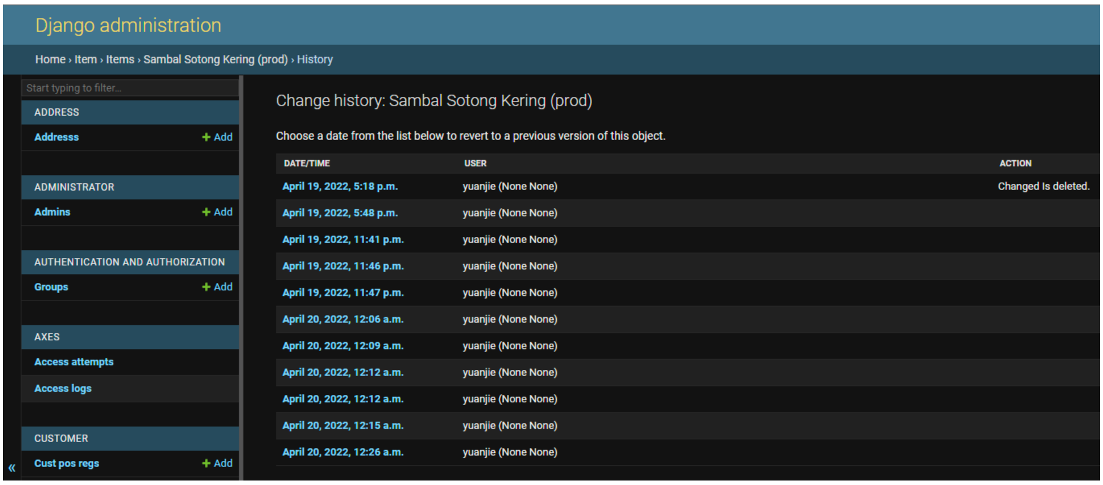
	id [PK] integer	object_id character va	format character	serialized_data text	object_repr text	content_type_id integer	revision_id integer	db character
1	560	72	json	[{"model": "customer.custposreg", "pk": 72, "fields": {"created_at": "2022-04-17T16:50:29.294", "last_update": "2022-04-17T16:5...	CustPosReg object (72)	19	465	default
2	561	73	json	[{"model": "customer.custposreg", "pk": 73, "fields": {"created_at": "2022-04-17T16:52:35.584", "last_update": "2022-04-17T16:5...	CustPosReg object (73)	19	466	default
3	562	104	json	[{"model": "customer.cust", "pk": 104, "fields": {"cust_type": 1, "pos_reg": null}}]	tanenxi01@gmail.com	20	467	default
4	563	104	json	[{"model": "core.users", "pk": 104, "fields": {"last_login": null, "is_superuser": false, "is_staff": false, "is_active": true, "date_joined"...	tanenxi01@gmail.com	17	467	default
5	564	104	json	[{"model": "core.users", "pk": 104, "fields": {"last_login": "2022-04-17T17:11:46.707", "is_superuser": false, "is_staff": false, "is_a...	tanenxi01@gmail.com	17	468	default
6	565	104	json	[{"model": "customer.cust", "pk": 104, "fields": {"cust_type": 1, "pos_reg": null}}]	tanenxi01@gmail.com	20	469	default
7	566	104	json	[{"model": "core.users", "pk": 104, "fields": {"last_login": "2022-04-17T17:11:46.707", "is_superuser": false, "is_staff": false, "is_a...	tanenxi01@gmail.com	17	469	default
8	567	3	json	[{"model": "address.address", "pk": 3, "fields": {"created_at": "2022-04-17T17:14:25.692", "last_update": "2022-04-17T17:14:25....	Address object (3)	13	470	default
9	568	74	json	[{"model": "customer.custposreg", "pk": 74, "fields": {"created_at": "2022-04-17T17:16:58.382", "last_update": "2022-04-17T17:1...	CustPosReg object (74)	19	471	default
10	569	75	json	[{"model": "customer.custposreg", "pk": 75, "fields": {"created_at": "2022-04-17T17:17:02.544", "last_update": "2022-04-17T17:1...	CustPosReg object (75)	19	472	default
11	570	75	json	[{"model": "customer.custposreg", "pk": 75, "fields": {"created_at": "2022-04-17T17:17:02.544", "last_update": "2022-04-17T17:1...	CustPosReg object (75)	19	473	default

Figure 6.15: Sample Data in reversion_version



	id [PK] integer	date_created timestamp with time zone	comment text	user_id integer
1	465	2022-04-17 16:50:29.269178+08		[null]
2	466	2022-04-17 16:52:35.561163+08		[null]
3	467	2022-04-17 17:11:14.866838+08		[null]
4	468	2022-04-17 17:11:46.491804+08		[null]
5	469	2022-04-17 17:12:55.730283+08		104
6	470	2022-04-17 17:14:25.673672+08		104
7	471	2022-04-17 17:16:58.351994+08		[null]
8	472	2022-04-17 17:17:02.525236+08		104
9	473	2022-04-17 17:17:17.543104+08		1
10	474	2022-04-17 17:22:36.243451+08		104
11	475	2022-04-17 17:22:36.874856+08		104

Figure 6.16: Sample Data of reversion_revision



Change history: Sambal Sotong Kering (prod)

Choose a date from the list below to revert to a previous version of this object.

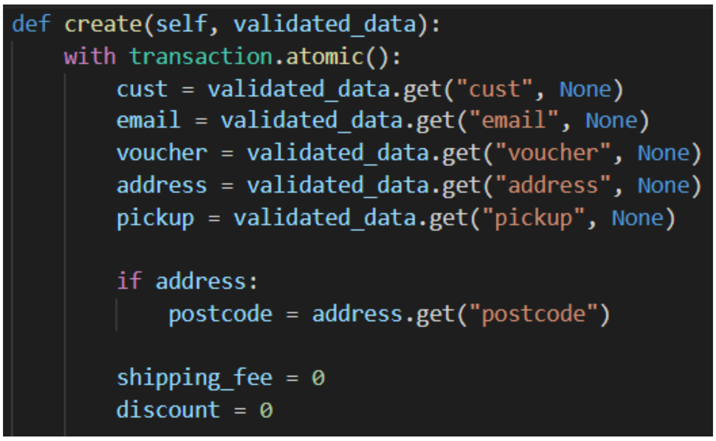
DATE/TIME	USER	ACTION
April 19, 2022, 5:18 p.m.	yuanjie (None None)	Changed Is deleted.
April 19, 2022, 5:48 p.m.	yuanjie (None None)	
April 19, 2022, 11:41 p.m.	yuanjie (None None)	
April 19, 2022, 11:46 p.m.	yuanjie (None None)	
April 19, 2022, 11:47 p.m.	yuanjie (None None)	
April 20, 2022, 12:06 a.m.	yuanjie (None None)	
April 20, 2022, 12:09 a.m.	yuanjie (None None)	
April 20, 2022, 12:12 a.m.	yuanjie (None None)	
April 20, 2022, 12:12 a.m.	yuanjie (None None)	
April 20, 2022, 12:15 a.m.	yuanjie (None None)	
April 20, 2022, 12:26 a.m.	yuanjie (None None)	

Figure 6.17: Data Audit Trail in Django Administration Site

6.4.5 Database Transaction Management and Concurrency Control

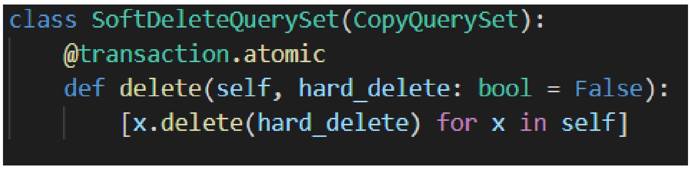
As discussed in Chapter 2, database transaction management and concurrency control help maintain the database's consistency and integrity while allowing the users to access the system simultaneously. Therefore, the project has implemented database transaction management.

In order to ensure that the system only commits to the database after the request is processed and the response is produced without problem, all the requests of the systems developed are wrapped in a transaction. In Django, there are two ways to wrap the request in a transaction. The first way is to use the transaction decorator or context manager provided by the django.db library, as shown in Figure 6.18 and Figure 6.19. The code beneath the context manager will be executed inside a transaction with the transaction context manager. Alternately, the transaction decorator wraps the entire function inside a transaction.



```python
def create(self, validated_data):
    with transaction.atomic():
        cust = validated_data.get("cust", None)
        email = validated_data.get("email", None)
        voucher = validated_data.get("voucher", None)
        address = validated_data.get("address", None)
        pickup = validated_data.get("pickup", None)

        if address:
            postcode = address.get("postcode")

        shipping_fee = 0
        discount = 0
```

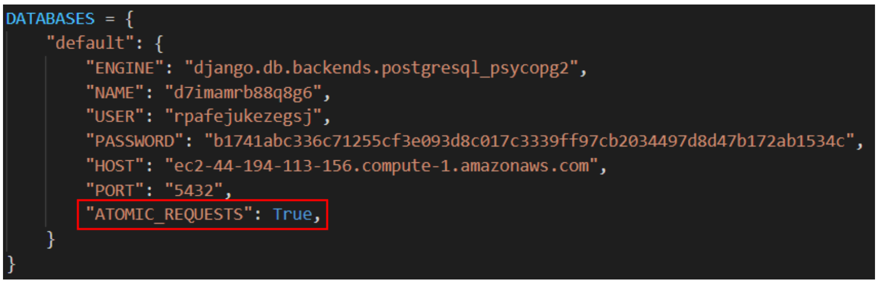
Figure 6.18: Code Snippet of Transaction Conext Manager



```
class SoftDeleteQuerySet(CopyQuerySet):
    @transaction.atomic
    def delete(self, hard_delete: bool = False):
        [x.delete(hard_delete) for x in self]
```

Figure 6.19: Code Snippet of Transaction Decorator

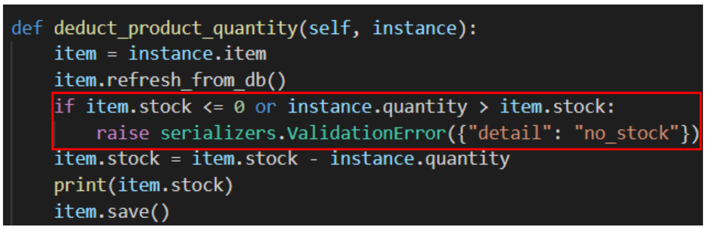
Since the transaction context manager and transaction decorator must be explicitly defined to wrap the function or code in a transaction, it is very tedious to define them in every request explicitly. Hence, the project configures the database settings to enable the "ATOMIC_REQUESTS" to wrap every request in a transaction without defining the decorator or context manager. Figure 6.20 shows the database settings to allow the wrapping of transactions for all requests.



```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql_psycopg2",
        "NAME": "d7imamrb88q8g6",
        "USER": "rpafejukezegsj",
        "PASSWORD": "b1741abc336c71255cf3e093d8c017c3339ff97cb2034497d8d47b172ab1534c",
        "HOST": "ec2-44-194-113-156.compute-1.amazonaws.com",
        "PORT": "5432",
        "ATOMIC_REQUESTS": True,
    }
}
```

Figure 6.20: Code Snippet of Database Transaction Management Settings

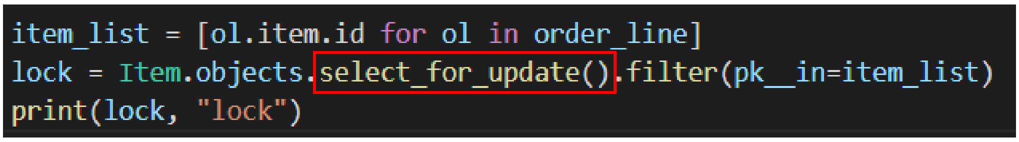
After configuring the database settings, it is critical to raise an exception to roll back the transaction if something goes wrong. Figure 6.21 shows the code snippet of raising an exception to roll back the transaction if the item is not in stock when the user places an order.



```
def deduct_product_quantity(self, instance):
    item = instance.item
    item.refresh_from_db()
    if item.stock <= 0 or instance.quantity > item.stock:
        raise serializers.ValidationError({"detail": "no_stock"})
    item.stock = item.stock - instance.quantity
    print(item.stock)
    item.save()
```

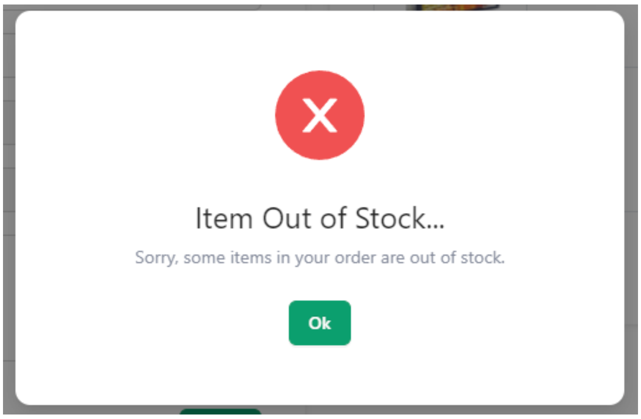
Figure 6.21: Code Snippet of Raising Exception to Trigger Transaction Rollback

Furthermore, the project also applied lock-based concurrency control to lock the table rows until the end of the transaction. The lock-based concurrency control is applied when the user places an order on the e-commerce platform to ensure that no simultaneous users can successfully place orders when the items in the orders are out of stock. Figure 6.22 shows the use of the select_for_update function in Django to lock the rows of the items when a user places an order, preventing other users from accessing the item records before the transaction ends. If the users place orders simultaneously and the item stock can only fulfill the order of the first user, the succeeding users will receive an error message, as shown in Figure 6.23.



```
item_list = [ol.item.id for ol in order_line]
lock = Item.objects.select_for_update().filter(pk__in=item_list)
print(lock, "lock")
```

Figure 6.22: Code Snippet of Lock-based Concurrency Control Implementation



Item Out of Stock...

Sorry, some items in your order are out of stock.

Figure 6.23: Item Out of Stock When Placing Order

6.5 Performance Optimization

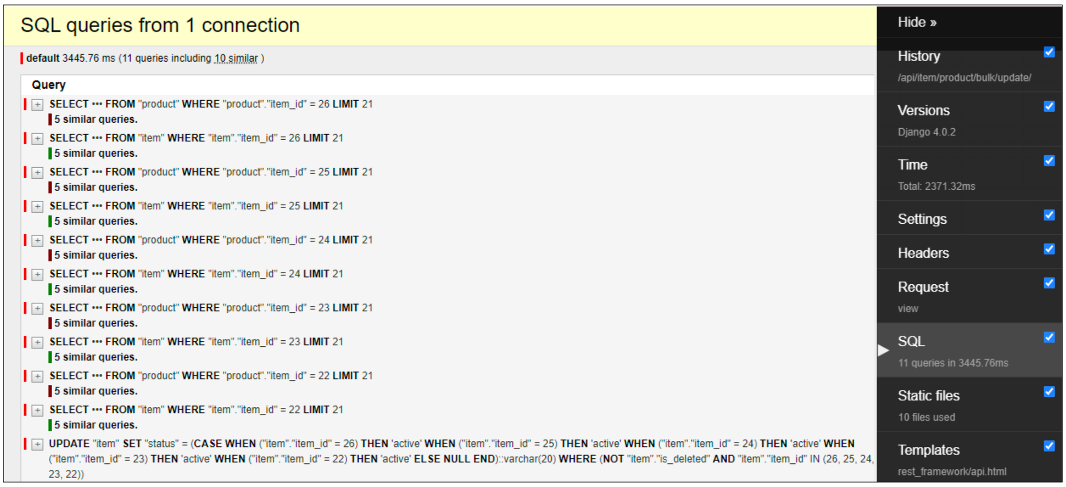
Performance is one of the significant considerations when developing a system, and it can be measured by the time constraints and resources allocation to show the system's effectiveness. A performant system is said to have better responsiveness and stability and be able to withstand a heavy workload. In this section, two methods used in the project to improve the system’s performance will be discussed: database query optimization and cache implementation.

6.5.1 Database Query Optimization

As Django automatically generates the database queries, some query optimizations can be performed, especially for models with one-to-one and one-to-many relationships. The default behaviour of Django when querying one-to-one and one-to-many relationships is that it will perform the select operations on each record one by one.

In order to optimize the database queries, the functions such as select_related() and prefetch_related() were used. When dealing with the one-to-one relationship, the select_related function was used in the queryset. On the other hand, when dealing with one-to-many relationships, the prefetch_related function was used. Both functions perform the join operations to reduce the number of database accesses, thereby boosting the system’s performance.

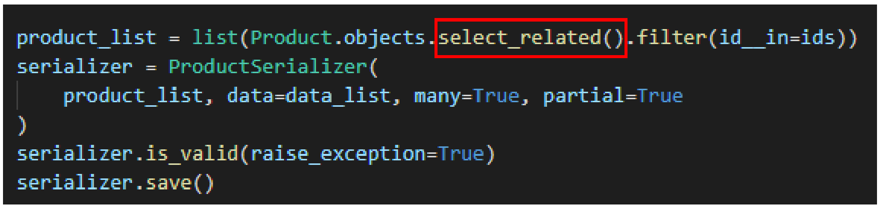
A Django library named Django Debug Toolbar was used to observe the system’s performance. As shown in Figure 6.24, there were 11 queries to obtain and update the product records. However, there were only 2 queries required after applying the select_related function in the query set, as shown in Figure 6.25 and Figure 6.26.



SQL queries from 1 connection

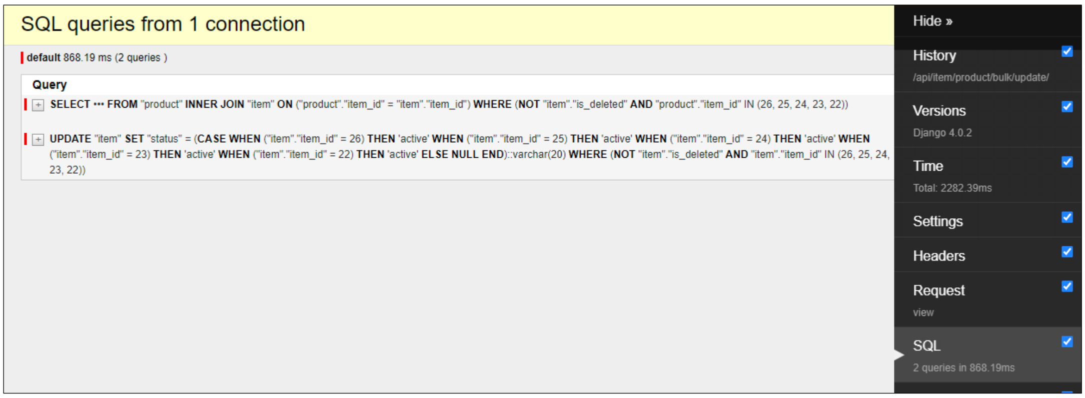
```
default 3445.76 ms (11 queries including 10 similar )

Query
SELECT ••• FROM "product" WHERE "product"."item_id" = 26 LIMIT 21
  5 similar queries.
SELECT ••• FROM "item" WHERE "item"."item_id" = 26 LIMIT 21
  5 similar queries.
SELECT ••• FROM "product" WHERE "product"."item_id" = 25 LIMIT 21
  5 similar queries.
SELECT ••• FROM "item" WHERE "item"."item_id" = 25 LIMIT 21
  5 similar queries.
SELECT ••• FROM "product" WHERE "product"."item_id" = 24 LIMIT 21
  5 similar queries.
SELECT ••• FROM "item" WHERE "item"."item_id" = 24 LIMIT 21
  5 similar queries.
SELECT ••• FROM "product" WHERE "product"."item_id" = 23 LIMIT 21
  5 similar queries.
SELECT ••• FROM "item" WHERE "item"."item_id" = 23 LIMIT 21
  5 similar queries.
SELECT ••• FROM "product" WHERE "product"."item_id" = 22 LIMIT 21
  5 similar queries.
SELECT ••• FROM "item" WHERE "item"."item_id" = 22 LIMIT 21
  5 similar queries.
UPDATE "item" SET "status" = (CASE WHEN ("item"."item_id" = 26) THEN 'active' WHEN ("item"."item_id" = 25) THEN 'active' WHEN ("item"."item_id" = 24) THEN 'active' WHEN
("item"."item_id" = 23) THEN 'active' WHEN ("item"."item_id" = 22) THEN 'active' ELSE NULL END)::varchar(20) WHERE (NOT "item"."is_deleted" AND "item"."item_id" IN (26, 25, 24,
23, 22))
```

Figure 6.24: Query Performance Before Optimization Using select_related



```
product_list = list(Product.objects.select_related().filter(id__in=ids))
serializer = ProductSerializer(
    product_list, data=data_list, many=True, partial=True
)
serializer.is_valid(raise_exception=True)
serializer.save()
```

Figure 6.25: Code Snippet of Applying select_related Function



SQL queries from 1 connection

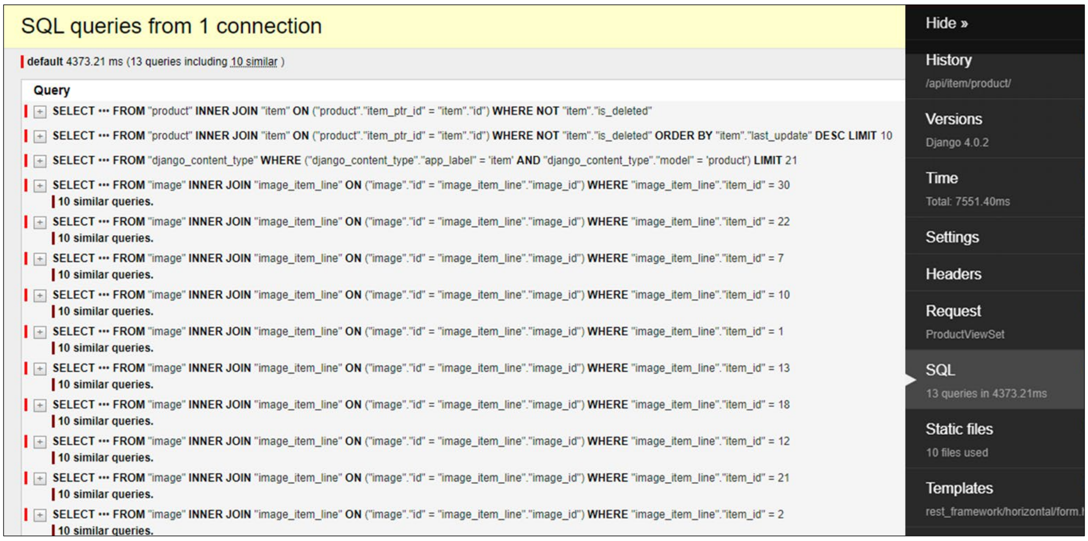
default 868.19 ms (2 queries)

Query

```
SELECT ••• FROM "product" INNER JOIN "item" ON ("product"."item_id" = "item"."item_id") WHERE (NOT "item"."is_deleted" AND "product"."item_id" IN (26, 25, 24, 23, 22))

UPDATE "item" SET "status" = (CASE WHEN ("item"."item_id" = 26) THEN 'active' WHEN ("item"."item_id" = 25) THEN 'active' WHEN ("item"."item_id" = 24) THEN 'active' WHEN ("item"."item_id" = 23) THEN 'active' WHEN ("item"."item_id" = 22) THEN 'active' ELSE NULL END)::varchar(20) WHERE (NOT "item"."is_deleted" AND "item"."item_id" IN (26, 25, 24, 23, 22))
```

Figure 6.26: Query Performance After Optimization Using select_related

In terms of the one-to-many relationships, Figure 6.27, Figure 6.28 and Figure 6.29 show the usage of the prefetch_related function to optimize the query when retrieving the products with images. As we can see from the figures, the number of queries reduced from 12 times to 4 times.



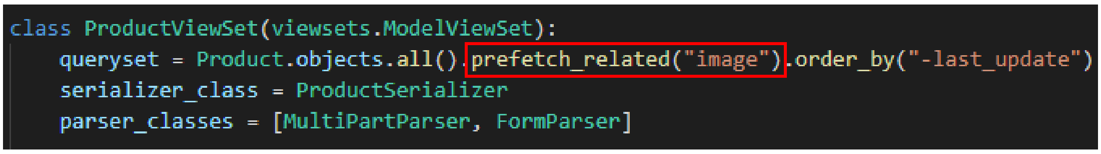
SQL queries from 1 connection

default 4373.21 ms (13 queries including 10 similar)

Query

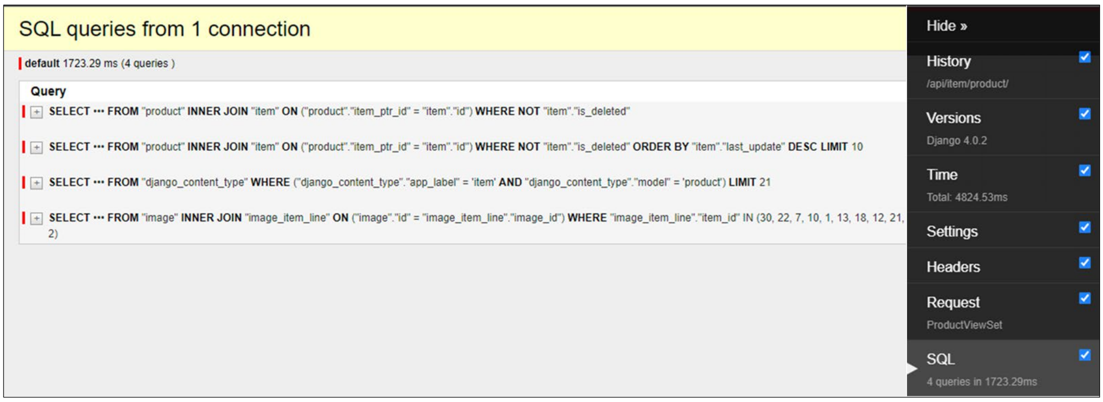
```
SELECT ••• FROM "product" INNER JOIN "item" ON ("product"."item_ptr_id" = "item"."id") WHERE NOT "item"."is_deleted"
SELECT ••• FROM "product" INNER JOIN "item" ON ("product"."item_ptr_id" = "item"."id") WHERE NOT "item"."is_deleted" ORDER BY "item"."last_update" DESC LIMIT 10
SELECT ••• FROM "django_content_type" WHERE ("django_content_type"."app_label" = 'item' AND "django_content_type"."model" = 'product') LIMIT 21
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 30
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 22
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 7
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 10
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 1
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 13
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 18
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 12
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 21
 10 similar queries.
SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" = 2
 10 similar queries.
```

Figure 6.27:Query Performance Before Optimization Using prefetch_related



```
class ProductViewSet(viewsets.ModelViewSet):
    queryset = Product.objects.all().prefetch_related("image").order_by("-last_update")
    serializer_class = ProductSerializer
    parser_classes = [MultiPartParser, FormParser]
```

Figure 6.28: Code Snippet of Applying prefetch_related Function



SQL queries from 1 connection

default 1723.29 ms (4 queries)

Query

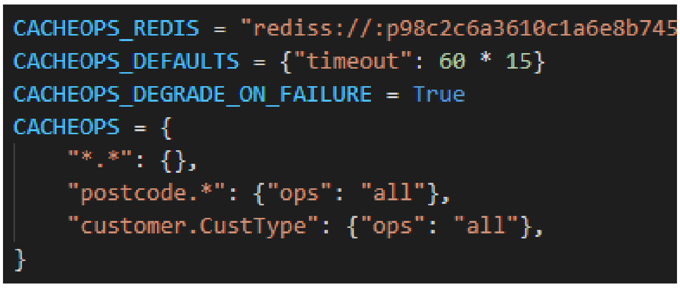
```
SELECT ••• FROM "product" INNER JOIN "item" ON ("product"."item_ptr_id" = "item"."id") WHERE NOT "item"."is_deleted"

SELECT ••• FROM "product" INNER JOIN "item" ON ("product"."item_ptr_id" = "item"."id") WHERE NOT "item"."is_deleted" ORDER BY "item"."last_update" DESC LIMIT 10

SELECT ••• FROM "django_content_type" WHERE ("django_content_type"."app_label" = 'item' AND "django_content_type"."model" = 'product') LIMIT 21

SELECT ••• FROM "image" INNER JOIN "image_item_line" ON ("image"."id" = "image_item_line"."image_id") WHERE "image_item_line"."item_id" IN (30, 22, 7, 10, 1, 13, 18, 12, 21,
2)
```

Figure 6.29: Query Performance After Optimization Using prefetch_realted

6.5.2 Cache Implementation

Proper caching can help to speed up the system's performance and increase efficiency. With cache, the system does not require to keep hitting the database for querying the results. Instead, the system stores the queries in temporary storage and retrieves them whenever needed.

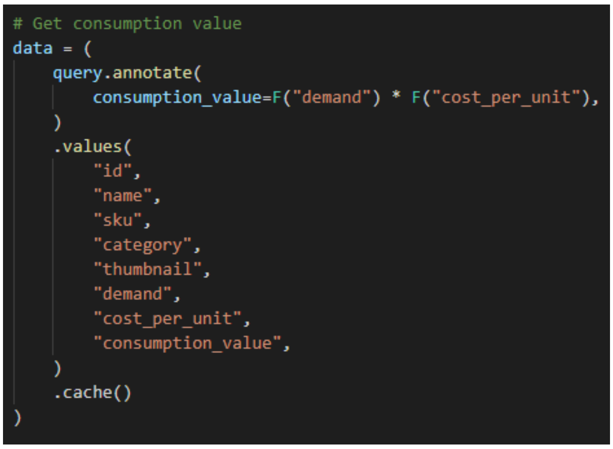
The project uses Redis as cache storage to cache the data that rarely changes. As shown in Figure 6.30, the models in the postcode app, including Postcode and State models and the CustType models, are set up as caches because they rarely change.



```
CACHEOPS_REDIS = "rediss://:p98c2c6a3610c1a6e8b745
CACHEOPS_DEFAULTS = {"timeout": 60 * 15}
CACHEOPS_DEGRADE_ON_FAILURE = True
CACHEOPS = {
    "*.*": {},
    "postcode.*": {"ops": "all"},
    "customer.CustType": {"ops": "all"},
}
```

Figure 6.30: Cache Settings

Besides, the cache is also used for inventory analyses so that the system does not require to compute the inventory analysis results repeatedly. Figure 6.31 shows one of the code snippets of cache implementation for the inventory analysis.



```python
# Get consumption value
data = (
    query.annotate(
        consumption_value=F("demand") * F("cost_per_unit"),
    )
    .values(
        "id",
        "name",
        "sku",
        "category",
        "thumbnail",
        "demand",
        "cost_per_unit",
        "consumption_value",
    )
    .cache()
)
```

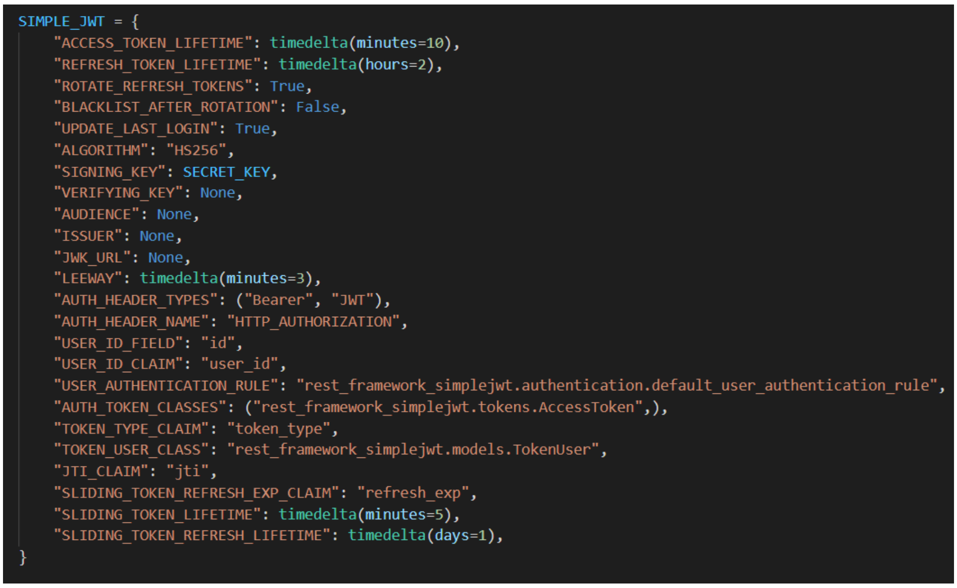
Figure 6.31: Code Snippet of Cache Implementation for ABC analysis

6.6 Implementation of Inventory Management System

This section discusses the features implemented in the inventory management system.

6.6.1 Authentication

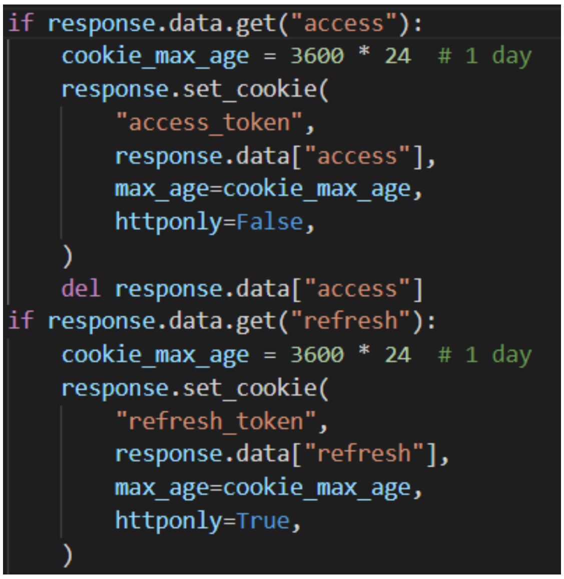
The authentication of the inventory management system mainly relies on JSON Web Token (JWT) authentication. There are two tokens involved in the authentication of the inventory management system: access token and refresh token. When the user login to the system, the system will issue an access token and a refresh token. After that, the access token will be added to the authentication header of each request before it is sent to the API. The refresh token has longer life as it is used to refresh the access token once it expires. Figure 6.32 shows the JWT settings on the inventory system.



```python
SIMPLE_JWT = {
    "ACCESS_TOKEN_LIFETIME": timedelta(minutes=10),
    "REFRESH_TOKEN_LIFETIME": timedelta(hours=2),
    "ROTATE_REFRESH_TOKENS": True,
    "BLACKLIST_AFTER_ROTATION": False,
    "UPDATE_LAST_LOGIN": True,
    "ALGORITHM": "HS256",
    "SIGNING_KEY": SECRET_KEY,
    "VERIFYING_KEY": None,
    "AUDIENCE": None,
    "ISSUER": None,
    "JWK_URL": None,
    "LEEWAY": timedelta(minutes=3),
    "AUTH_HEADER_TYPES": ("Bearer", "JWT"),
    "AUTH_HEADER_NAME": "HTTP_AUTHORIZATION",
    "USER_ID_FIELD": "id",
    "USER_ID_CLAIM": "user_id",
    "USER_AUTHENTICATION_RULE": "rest_framework_simplejwt.authentication.default_user_authentication_rule",
    "AUTH_TOKEN_CLASSES": ("rest_framework_simplejwt.tokens.AccessToken",),
    "TOKEN_TYPE_CLAIM": "token_type",
    "TOKEN_USER_CLASS": "rest_framework_simplejwt.models.TokenUser",
    "JTI_CLAIM": "jti",
    "SLIDING_TOKEN_REFRESH_EXP_CLAIM": "refresh_exp",
    "SLIDING_TOKEN_LIFETIME": timedelta(minutes=5),
    "SLIDING_TOKEN_REFRESH_LIFETIME": timedelta(days=1),
}
```

Figure 6.32: JWT Authentication Settings of Inventory Management System

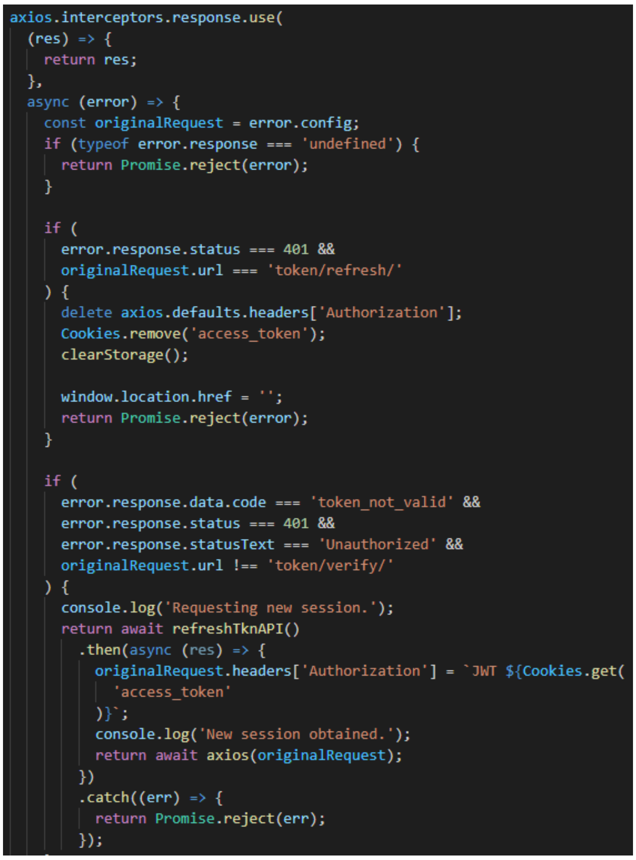
In addition, the access token only lives for 10 minutes, while the refresh token lives for 2 hours to increase the system's security. Besides, the tokens are stored in the cookies instead of the local storage because third-party scripts can easily compromise the system if the tokens are stored in the local storage. Besides, as shown in Figure 6.33, the access token is not set to the HttpOnly cookie as the front-end application requires retrieving and adding the access token to the requests' authentication header. On the other hand, the refresh token is set to HttpOnly cookie to reduce the risk of a Cross-Site Scripting (XSS) exploit.



```
if response.data.get("access"):
    cookie_max_age = 3600 * 24  # 1 day
    response.set_cookie(
        "access_token",
        response.data["access"],
        max_age=cookie_max_age,
        httponly=False,
    )
    del response.data["access"]
if response.data.get("refresh"):
    cookie_max_age = 3600 * 24  # 1 day
    response.set_cookie(
        "refresh_token",
        response.data["refresh"],
        max_age=cookie_max_age,
        httponly=True,
    )
```

Figure 6.33: Code Snippet of Storing JWT Tokens in Cookies

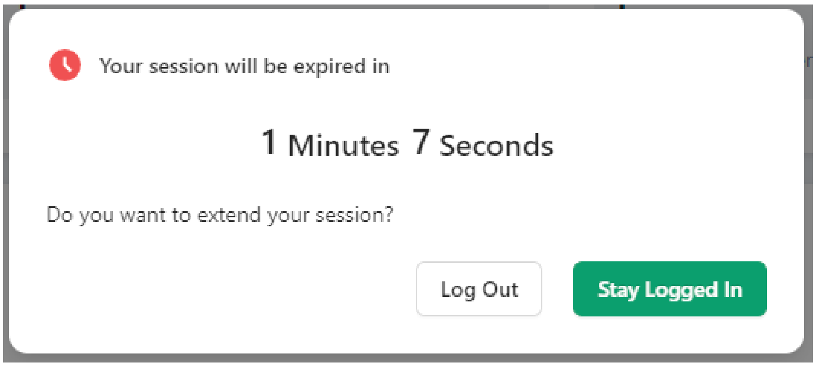
Besides, to avoid unauthorized access to the API when the access token is expired while the refresh token is valid, an interceptor function was created to refresh the access token when automatically it is expired while the refresh token is valid.



```
axios.interceptors.response.use(
  (res) => {
    return res;
  },
  async (error) => {
    const originalRequest = error.config;
    if (typeof error.response === 'undefined') {
      return Promise.reject(error);
    }

    if (
      error.response.status === 401 &&
      originalRequest.url === 'token/refresh/'
    ) {
      delete axios.defaults.headers['Authorization'];
      Cookies.remove('access_token');
      clearStorage();

      window.location.href = '';
      return Promise.reject(error);
    }

    if (
      error.response.data.code === 'token_not_valid' &&
      error.response.status === 401 &&
      error.response.statusText === 'Unauthorized' &&
      originalRequest.url !== 'token/verify/'
    ) {
      console.log('Requesting new session.');
      return await refreshTknAPI()
        .then(async (res) => {
          originalRequest.headers['Authorization'] = `JWT ${Cookies.get(
            'access_token'
          )}`;
          console.log('New session obtained.');
          return await axios(originalRequest);
        })
        .catch((err) => {
          return Promise.reject(err);
        });
```

Figure 6.34: Code Snippet of Axios Interceptor

Furthermore, a session extension modal was created to alert the admins that their session is about to expire and prompt them to extend their session or log out. If the admins choose to extend their session, the system will send a refresh token request to the API. Otherwise, the system will log out the admin. In addition to the session extension, the system will only alert the admins if the admins are idle, or else, the system will automatically refresh the token.

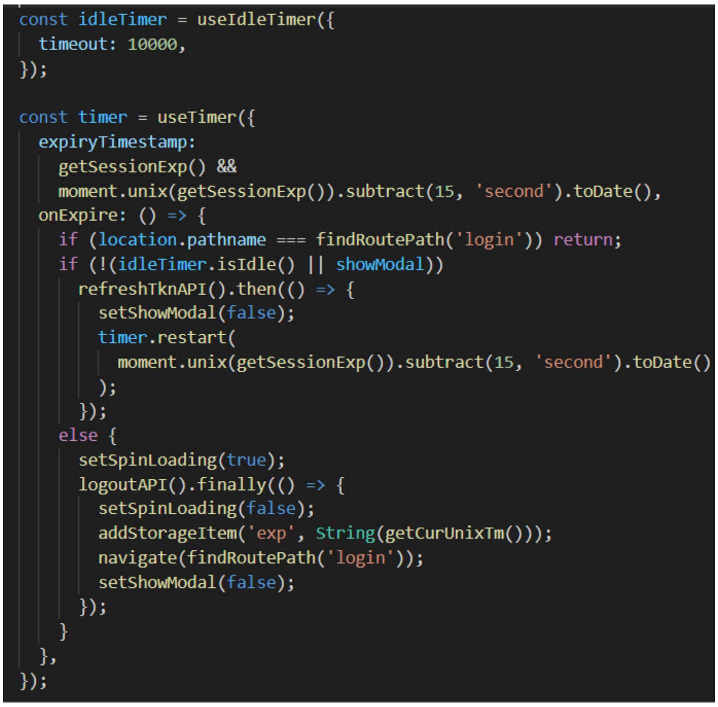


Your session will be expired in

1 Minutes 7 Seconds

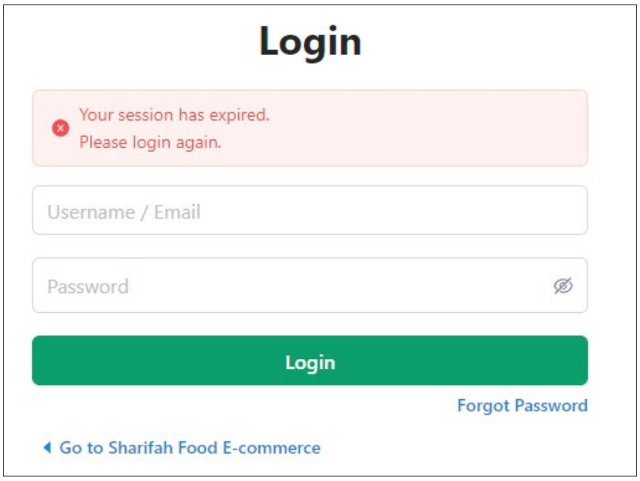
Do you want to extend your session?

Figure 6.35: Session Extension Modal



```
const idleTimer = useIdleTimer({
  timeout: 10000,
});

const timer = useTimer({
  expiryTimestamp:
    getSessionExp() &&
    moment.unix(getSessionExp()).subtract(15, 'second').toDate(),
  onExpire: () => {
    if (location.pathname === findRoutePath('login')) return;
    if (!(idleTimer.isIdle() || showModal))
      refreshTknAPI().then(() => {
        setShowModal(false);
        timer.restart(
          moment.unix(getSessionExp()).subtract(15, 'second').toDate()
        );
      });
    else {
      setSpinLoading(true);
      logoutAPI().finally(() => {
        setSpinLoading(false);
        addStorageItem('exp', String(getCurUnixTm()));
        navigate(findRoutePath('login'));
        setShowModal(false);
      });
    }
  },
});
```

Figure 6.36: Code Snippet of Checking Idle Admin and Display Modal

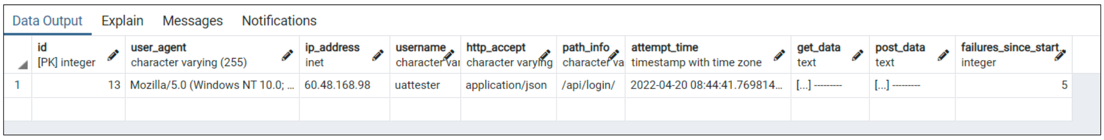


Login

Your session has expired.
Please login again.

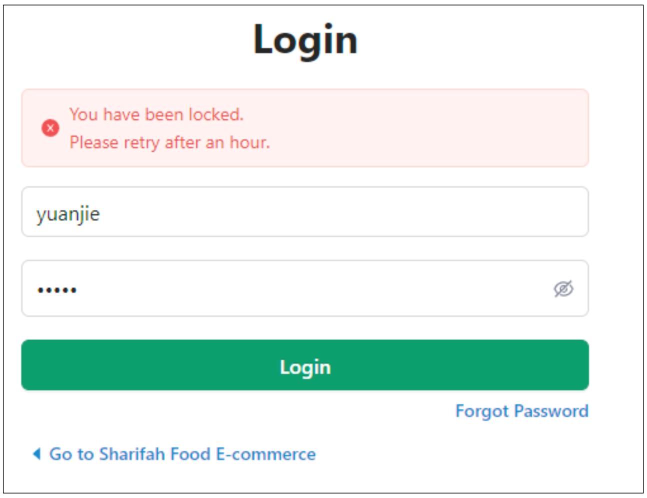
Figure 6.37: Session Expired Message After Refresh Token Expires

The system also locks the admin account and IP address after five failed attempts on login to fulfill the security requirement of the project. In order to accomplish the requirement, the django-axes library was used to check the failed attempts. All the failed attempts will be stored in a table named axes_accessattempt in the database, as shown in Figure 6.38. When the failed attempts reach five times, the system will display an error message and lock the IP address and account for 1 hour, as shown in Figure 6.39.



	id [PK] integer	user_agent character varying (255)	ip_address inet	username character vai	http_accept character varying	path_info character va	attempt_time timestamp with time zone	get_data text	post_data text	failures_since_start integer
1	13	Mozilla/5.0 (Windows NT 10.0; ...	60.48.168.98	uattester	application/json	/api/login/	2022-04-20 08:44:41.769814...	[...] ---------	[...] ---------	5

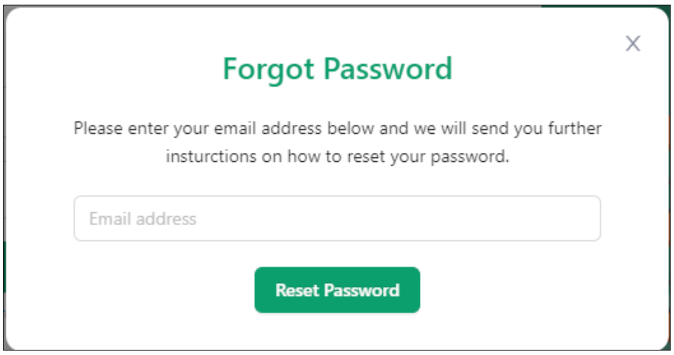
Figure 6.38: axes_accessattempt Table to Store Login Failed Attempts



You have been locked.
Please retry after an hour.

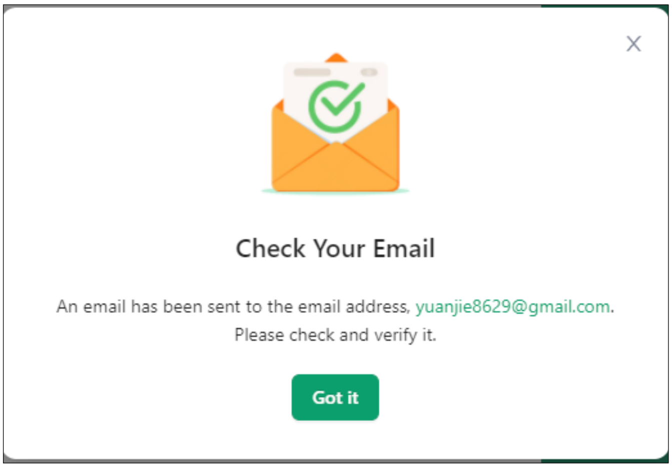
Figure 6.39: Account Locked Message

If the admin forgets the password, the admin can click the forgot password on the login page, and a modal in Figure 6.40 will be popped up to ask for the admin's email. Once the admin inputs the valid email, the system will send an email containing a reset password URL and prompt the admin to check the email, as shown in Figure 6.41. The admin would require to enter the URL to reset the password as the URL includes a reset password token tied with the admin email. After the admin inputs a new password and fulfill the password criteria, as shown in Figure 6.43, the system will update the admin password.



Please enter your email address below and we will send you further insturctions on how to reset your password.

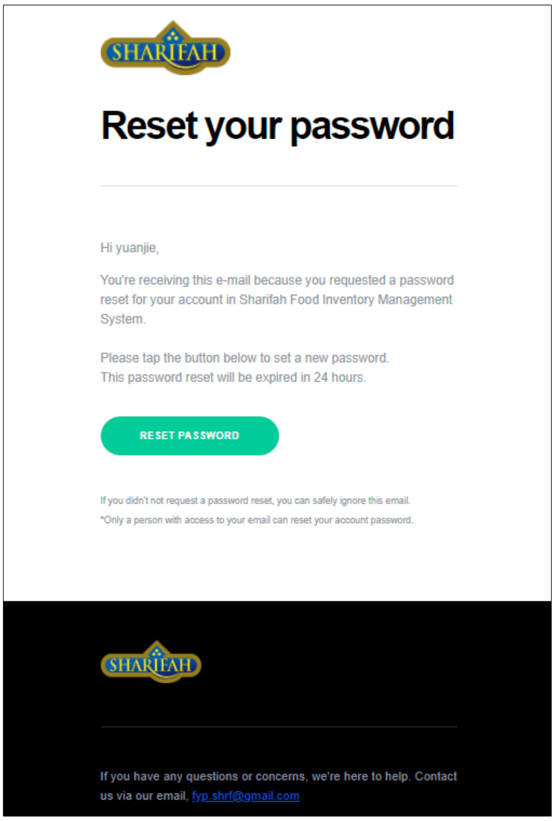
Figure 6.40: Forgot Password Modal



Check Your Email

An email has been sent to the email address, yuanjie8629@gmail.com. Please check and verify it.

Figure 6.41: Check Email Modal



Reset your password

Hi yuanjie,

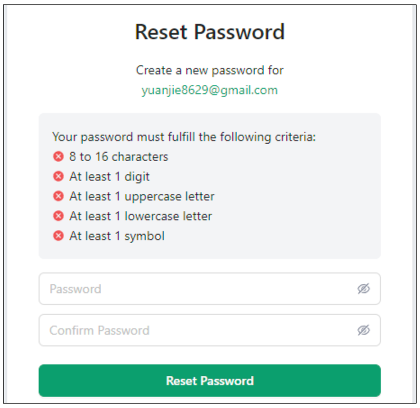
You're receiving this e-mail because you requested a password reset for your account in Sharifah Food Inventory Management System.

Please tap the button below to set a new password.
This password reset will be expired in 24 hours.

If you didn't not request a password reset, you can safely ignore this email.
*Only a person with access to your email can reset your account password.

If you have any questions or concerns, we're here to help. Contact us via our email, fyp.shrf@gmail.com

Figure 6.42: Reset Password Email



Reset Password

Create a new password for yuanjie8629@gmail.com

Your password must fulfill the following criteria:

- ☒ 8 to 16 characters
- ☒ At least 1 digit
- ☒ At least 1 uppercase letter
- ☒ At least 1 lowercase letter
- ☒ At least 1 symbol

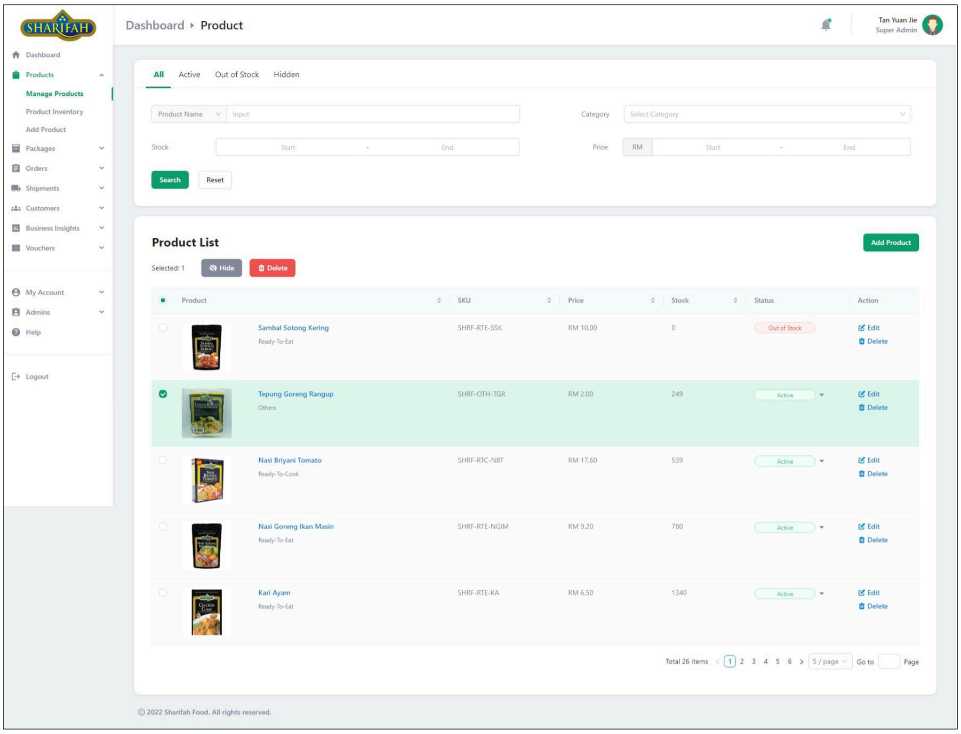
Figure 6.43: Reset Password Page

6.6.2 Product Management

Product management consists of three parts: Manage Products, Product Inventory, and Add/Edit Product.

6.6.2.1 Manage Products

The manage products page allows the admins to perform filtering and sorting by product name, sku, category, stock range, and price range. Besides, the admin can also perform bulk operations to hide or delete the products.



Product List

Selected: 1

●	Product		SKU	Price	Stock	Status	Action
○		Sambal Sotong Kering Ready-To-Eat	SHRF-RTE-SSK	RM 10.00	0	Out of Stock	Edit Delete
✓		Tepung Goreng Rangup Others	SHRF-OTH-TGR	RM 2.00	249	Active	Edit Delete
○		Nasi Briyani Tomato Ready-To-Cook	SHRF-RTC-NBT	RM 17.60	539	Active	Edit Delete
○		Nasi Goreng Ikan Masin Ready-To-Eat	SHRF-RTE-NGIM	RM 9.20	780	Active	Edit Delete
○		Kari Ayam Ready-To-Eat	SHRF-RTE-KA	RM 6.50	1340	Active	Edit Delete

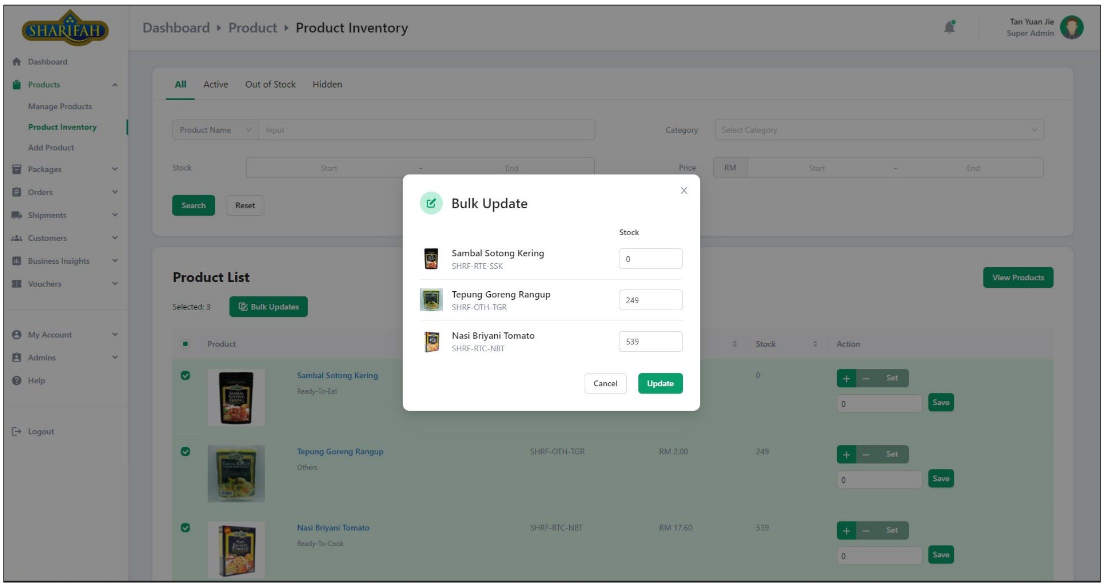
Total 26 items < 1 2 3 4 5 6 > 5 / page Go to Page

© 2022 Sharifah Food. All rights reserved.

Figure 6.44: Manage Products

6.6.2.2 Product Inventory

The product inventory page provides the admin with a straightforward way to update the product stock. Besides, the admin can also perform bulk updates on the product stocks. Also, the admin is able to filter and sort the products on the product inventory page. Figure 6.45 shows the bulk update modal that allows the admin to update many product stocks at once.



Bulk Update

	Stock
Sambal Sotong Kering SHRF-RTE-SSK	0
Tepung Goreng Rangup SHRF-OTH-TGR	249
Nasi Briyani Tomato SHRF-RTC-NBT	539

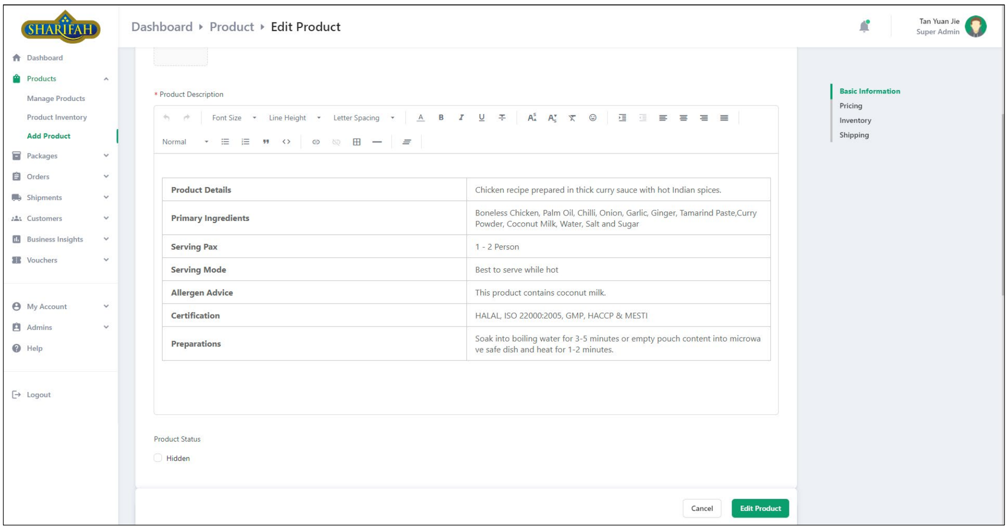
Cancel Update

Product			Stock	Action
Sambal Sotong Kering Ready-To-Eat			0	0
Tepung Goreng Rangup Others	SHRF-OTH-TGR	RM 2.00	249	0
Nasi Briyani Tomato Ready-To-Cook	SHRF-RTC-NBT	RM 17.60	539	0

Figure 6.45: Bulk Update in Product Inventory

6.6.2.3 Add/Edit Product

In order to add the product, the admin can click the ‘Add Product’ menu on the left sidebar or the ‘Add Product’ button on the manage products page. To edit a product, the admin can click on the edit button or the blue-coloured product name. The add and edit product pages are similar, except that the edit product will display all the product details of the selected product for the admin to modify. Besides, the rich text editor is used for the product description input, allowing the admin to customize the product description by adding tables, lists, emojis, etc., as shown in Figure 6.46.



Product Details	Chicken recipe prepared in thick curry sauce with hot Indian spices.
Primary Ingredients	Boneless Chicken, Palm Oil, Chilli, Onion, Garlic, Ginger, Tamarind Paste,Curry Powder, Coconut Milk, Water, Salt and Sugar
Serving Pax	1 - 2 Person
Serving Mode	Best to serve while hot
Allergen Advice	This product contains coconut milk.
Certification	HALAL, ISO 22000:2005, GMP, HACCP & MESTI
Preparations	Soak into boiling water for 3-5 minutes or empty pouch content into microwa ve safe dish and heat for 1-2 minutes.

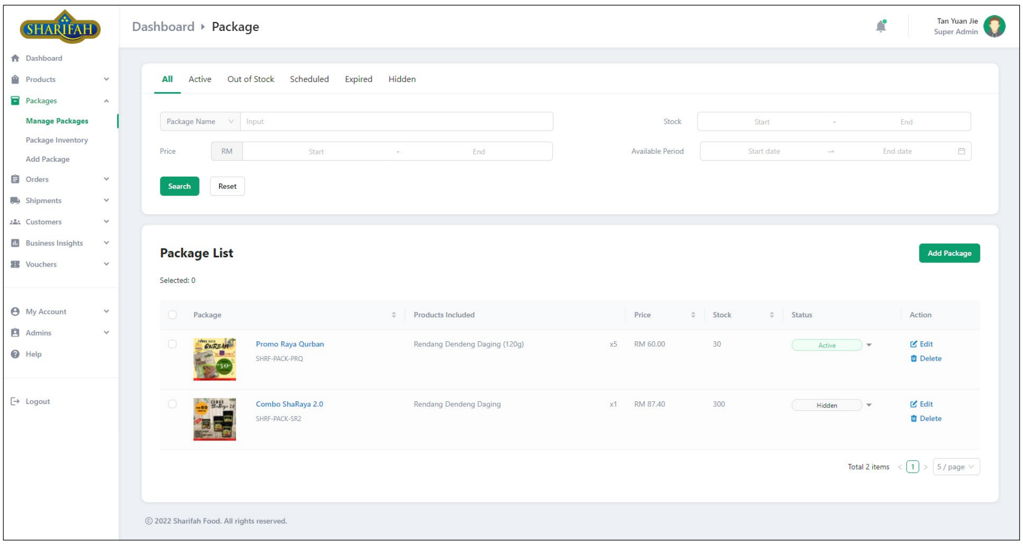
Figure 6.46: Rich Text Editor

6.6.3 Package Management

Product management contains three parts: Manage Packages, Package Inventory, and Add/Edit Package.

6.6.3.1 Manage Packages

The manage packages page is similar to the manage products page, and the only difference is the display of the data in the table. On the manage packages page, there is an additional column that displays the products included in a package, as shown in Figure 6.47. Additionally, the admin can filter and sort the packages by the available period range.



Package List

Selected: 0

□	Package	Products Included		Price	Stock	Status	Action
□	Promo Raya Qurban SHRF-PACK-PRQ	Rendang Dendeng Daging (120g)	x5	RM 60.00	30	Active	Edit Delete
□	Combo ShaRaya 2.0 SHRF-PACK-SR2	Rendang Dendeng Daging	x1	RM 87.40	300	Hidden	Edit Delete

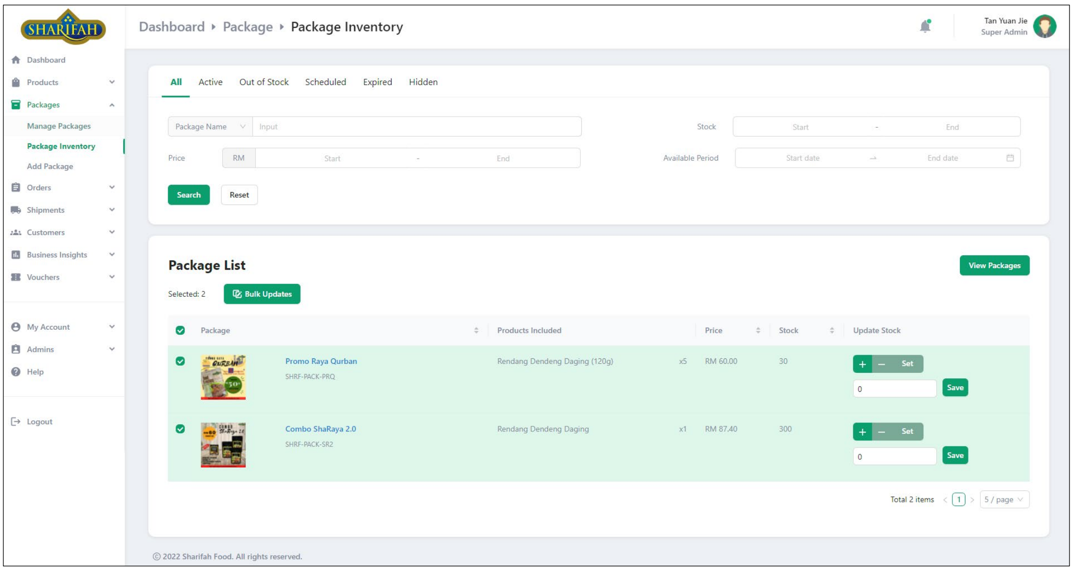
Total 2 items ‹ 1 › 5 / page

© 2022 Sharifah Food. All rights reserved.

Figure 6.47: Manage Packages

6.6.3.2 Package Inventory

The package inventory is identical to the product inventory, and the admins can update the package stocks directly on the package inventory page, as shown in Figure 6.48.

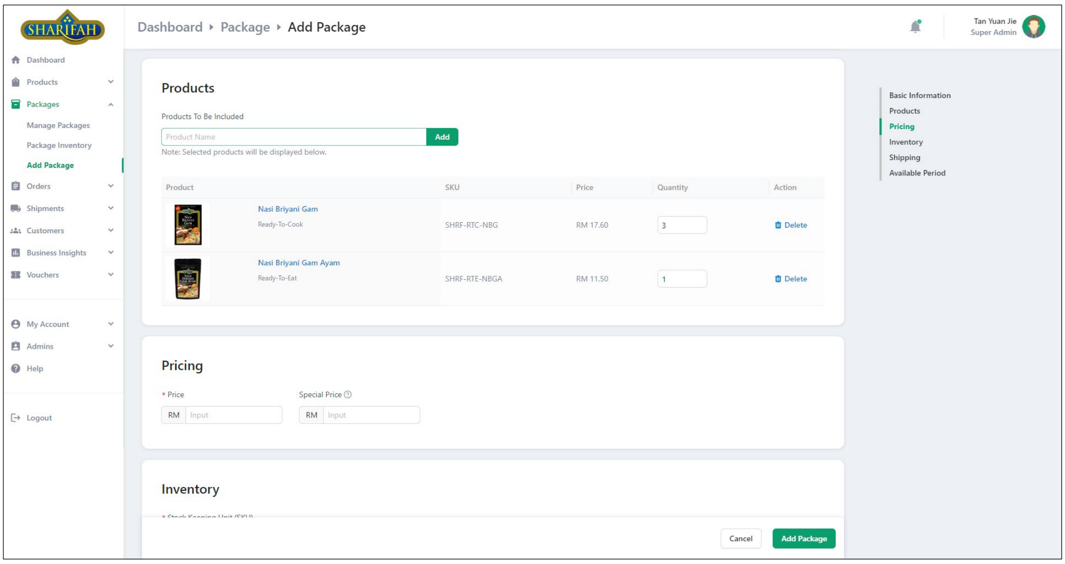


☑	Package	Products Included		Price	Stock	Update Stock
☑	Promo Raya Qurban SHRF-PACK-PRQ	Rendang Dendeng Daging (120g)	x5	RM 60.00	30	0
☑	Combo ShaRaya 2.0 SHRF-PACK-SR2	Rendang Dendeng Daging	x1	RM 87.40	300	0

Figure 6.48: Package Inventory

6.6.3.3 Add/Edit Package

Since a package contains products, the add and edit page for the package includes the input fields for the package’s items. Therefore, the admin can search and add the products to be included. Besides, the admin can also specify the quantity of the product to be included in the package, as shown in Figure 6.49.

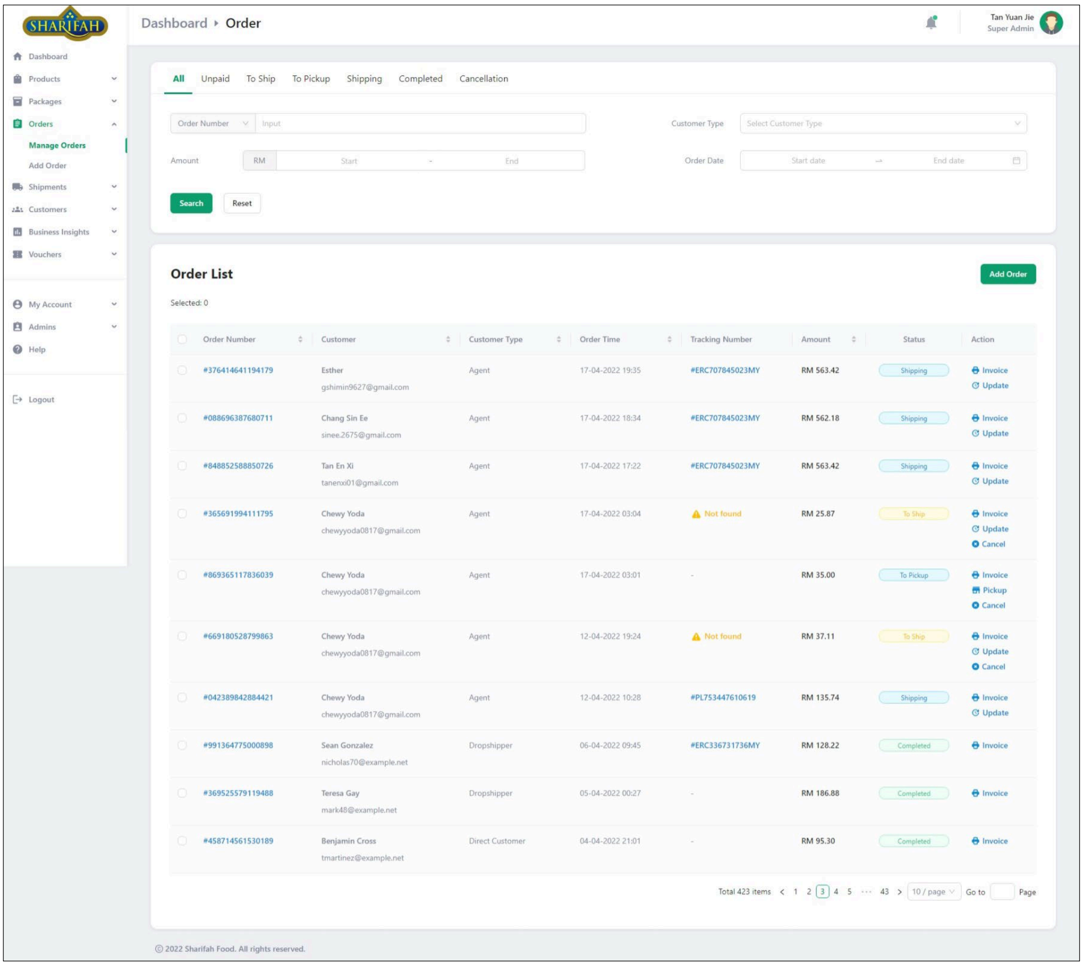


Product	SKU	Price	Quantity	Action
Nasi Briyani Gam Ready-To-Cook	SHRF-RTC-NBG	RM 17.60	3	Delete
Nasi Briyani Gam Ayam Ready-To-Eat	SHRF-RTE-NBGA	RM 11.50	1	Delete

Figure 6.49: Add/Edit Package

6.6.4 Order Management

On the order management page, which is shown in Figure 6.50, the admin can filter and sort the orders by the order number, customer name, customer email, tracking number, order amount range, customer type and order date. Additionally, the admin can also perform single or batch operations such as updating orders' tracking numbers, updating orders' pickup dates, generating invoices, and cancelling orders based on the order status. Figure 6.51 shows the modal form that allows the admin to update multiple orders' tracking numbers at once.



Order List

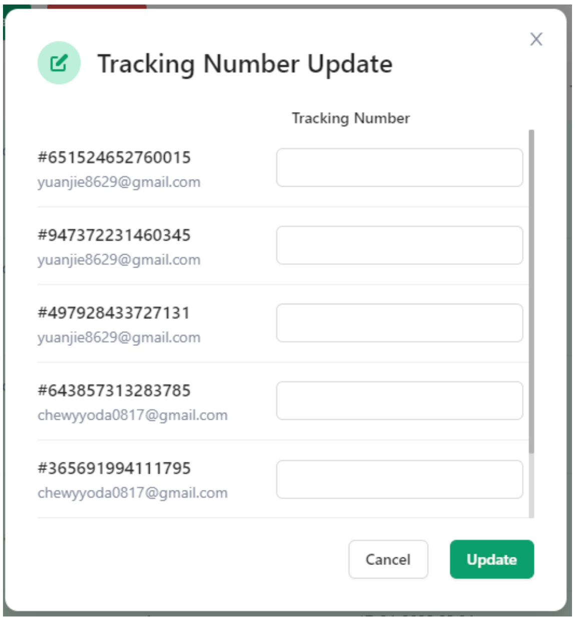
Selected: 0

Order Number	Customer	Customer Type	Order Time	Tracking Number	Amount	Status	Action
#376414641194179	Esther gshimin9627@gmail.com	Agent	17-04-2022 19:35	#ERC707845023MY	RM 563.42	Shipping	Invoice Update
#088696387680711	Chang Sin Ee sinee.2675@gmail.com	Agent	17-04-2022 18:34	#ERC707845023MY	RM 562.18	Shipping	Invoice Update
#848852588850726	Tan En Xi tanenxi01@gmail.com	Agent	17-04-2022 17:22	#ERC707845023MY	RM 563.42	Shipping	Invoice Update
#365691994111795	Chewy Yoda chewyyoda0817@gmail.com	Agent	17-04-2022 03:04	⚠ Not found	RM 25.87	To Ship	Invoice Update Cancel
#869365117836039	Chewy Yoda chewyyoda0817@gmail.com	Agent	17-04-2022 03:01	-	RM 35.00	To Pickup	Invoice Pickup Cancel
#669180528799863	Chewy Yoda chewyyoda0817@gmail.com	Agent	12-04-2022 19:24	⚠ Not found	RM 37.11	To Ship	Invoice Update Cancel
#042389842884421	Chewy Yoda chewyyoda0817@gmail.com	Agent	12-04-2022 10:28	#PL753447610619	RM 135.74	Shipping	Invoice Update
#991364775000898	Sean Gonzalez nicholas70@example.net	Dropshipper	06-04-2022 09:45	#ERC336731736MY	RM 128.22	Completed	Invoice
#369525579119488	Teresa Gay mark48@example.net	Dropshipper	05-04-2022 00:27	-	RM 186.88	Completed	Invoice
#458714561530189	Benjamin Cross tmartinez@example.net	Direct Customer	04-04-2022 21:01	-	RM 95.30	Completed	Invoice

Total 423 items < 1 2 3 4 5 ··· 43 > 10 / page Go to Page

© 2022 Sharifah Food. All rights reserved.

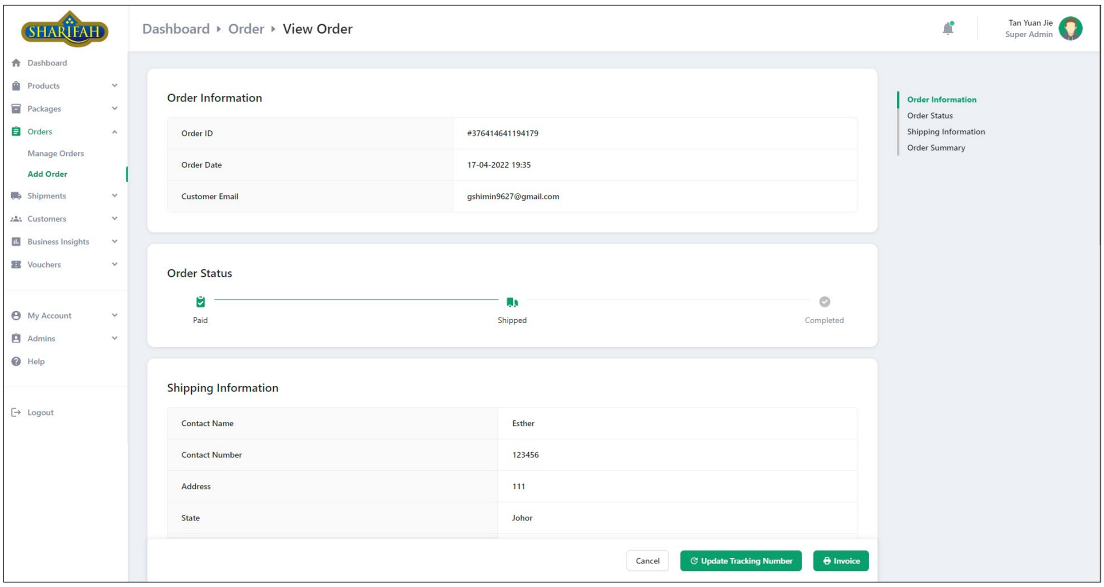
Figure 6.50: Order Management



	Tracking Number
#651524652760015 yuanjie8629@gmail.com	
#947372231460345 yuanjie8629@gmail.com	
#497928433727131 yuanjie8629@gmail.com	
#643857313283785 chewyyoda0817@gmail.com	
#365691994111795 chewyyoda0817@gmail.com	

Figure 6.51: Bulk Update Tracking Number

In terms of CRUD operations, the admin is only allowed to view the order details placed by the customer, and there is no way for the admin to edit or delete the order details. To view the order details, the admin can click the order number that appears in the leftmost column in the table. Figure 6.52 shows part of the order details page after the admin click on the order number in the order management page.



Order Information

Order ID	#376414641194179
Order Date	17-04-2022 19:35
Customer Email	gshimin9627@gmail.com

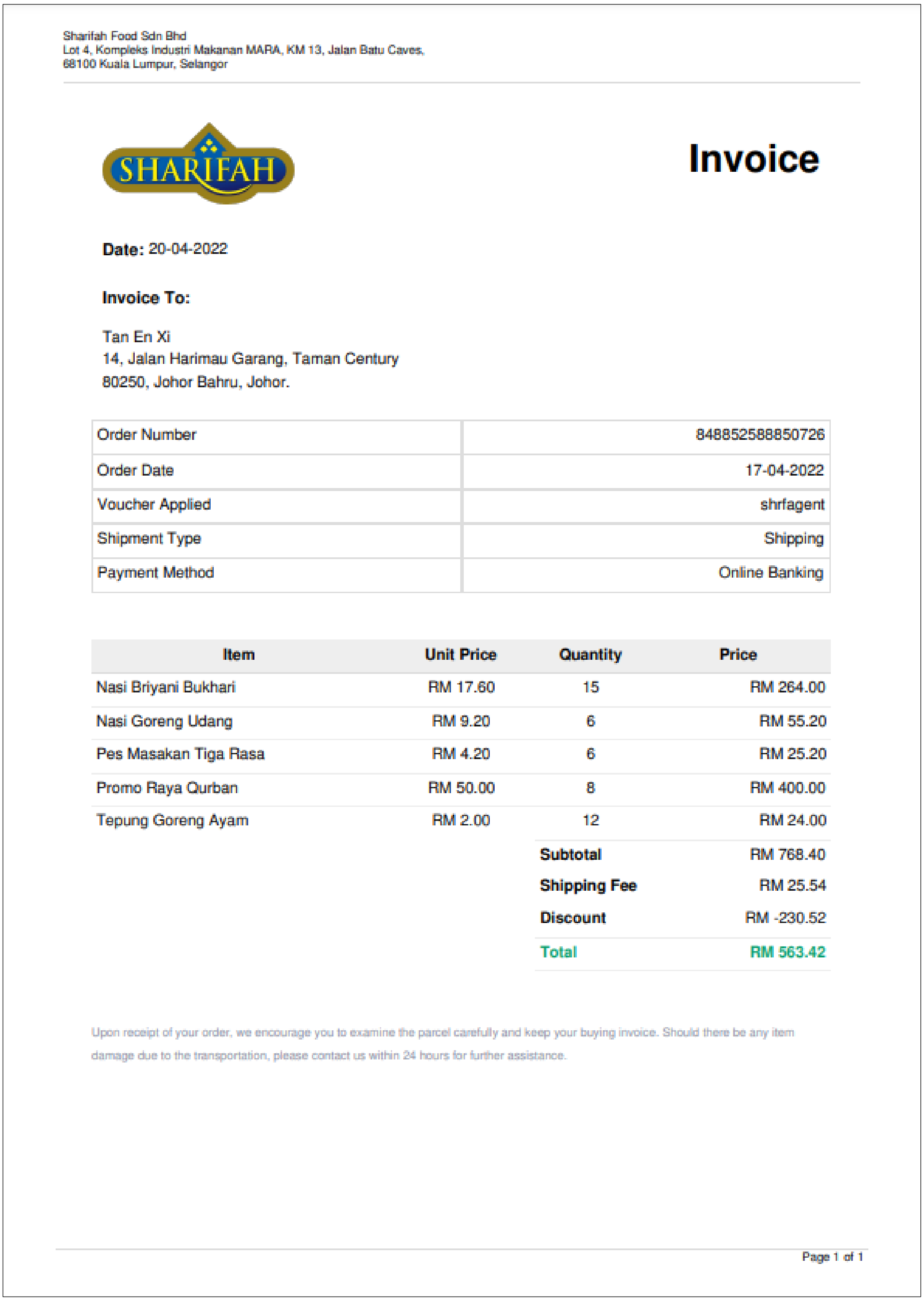
Order Status

Shipping Information

Contact Name	Esther
Contact Number	123456
Address	111
State	Johor

Figure 6.52: Part of Order Details

To generate invoices, the admin can click on the ‘Invoice’ button that appears in the ‘action’ column of the table to generate a single order invoice. If the admin wants to generate multiple invoices, the admin can select multiple orders and click the ‘Generate Invoice(s)’ button, and the invoices will be downloaded as a zip file. Figure 6.53 shows the order invoice generated by the system.



Sharifah Food Sdn Bhd
Lot 4, Kompleks Industri Makanan MARA, KM 13, Jalan Batu Caves,
68100 Kuala Lumpur, Selangor

Invoice

Date: 20-04-2022

Invoice To:

Tan En Xi
14, Jalan Harimau Garang, Taman Century
80250, Johor Bahru, Johor.

Order Number	848852588850726
Order Date	17-04-2022
Voucher Applied	shrfagent
Shipment Type	Shipping
Payment Method	Online Banking

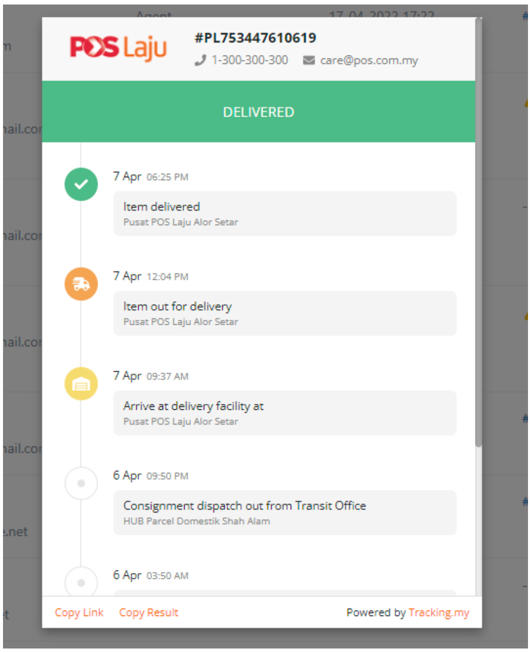
Item	Unit Price	Quantity	Price
Nasi Briyani Bukhari	RM 17.60	15	RM 264.00
Nasi Goreng Udang	RM 9.20	6	RM 55.20
Pes Masakan Tiga Rasa	RM 4.20	6	RM 25.20
Promo Raya Qurban	RM 50.00	8	RM 400.00
Tepung Goreng Ayam	RM 2.00	12	RM 24.00
		Subtotal	RM 768.40
		Shipping Fee	RM 25.54
		Discount	RM -230.52
		Total	**RM 563.42**

Upon receipt of your order, we encourage you to examine the parcel carefully and keep your buying invoice. Should there be any item damage due to the transportation, please contact us within 24 hours for further assistance.

Page 1 of 1

Figure 6.53: Order Invoice

Another feature of order management is that it allows the admin to track the order by clicking the tracking number, in which the system will display a popup that links to a third-party shipment tracking platform named Tracking.my, as shown in Figure 6.54.



POS Laju #PL753447610619
☎ 1-300-300-300 ✉ care@pos.com.my

DELIVERED

7 Apr 06:25 PM
Item delivered
Pusat POS Laju Alor Setar

7 Apr 12:04 PM
Item out for delivery
Pusat POS Laju Alor Setar

7 Apr 09:37 AM
Arrive at delivery facility at
Pusat POS Laju Alor Setar

6 Apr 09:50 PM
Consignment dispatch out from Transit Office
HUB Parcel Domestik Shah Alam

6 Apr 03:50 AM

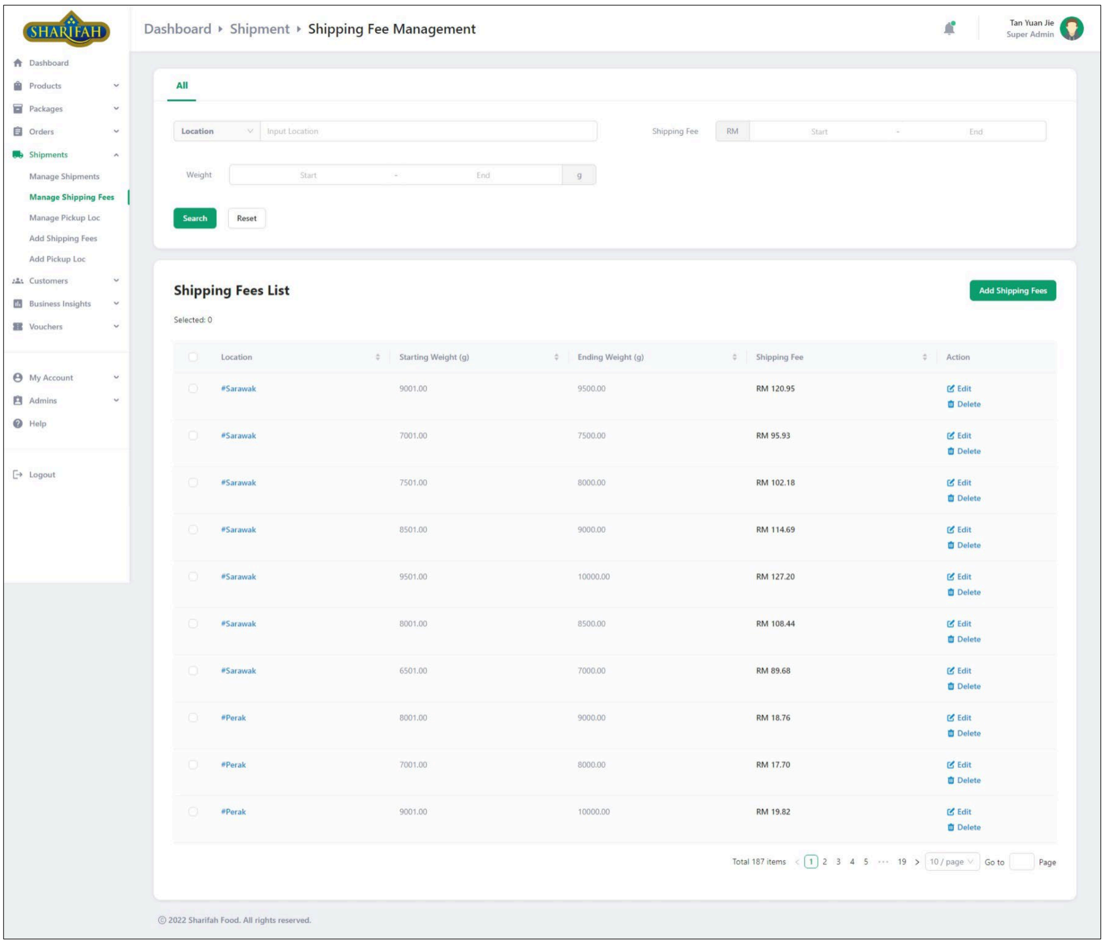
Figure 6.54: Third-party Shipment Tracking Popup

6.6.5 Shipping Fee Management

Shipping fee management can be divided into two parts: Manage Shipping Fees and Add/Edit Shipping Fees.

6.6.5.1 Manage Shipping Fees

The manage shipping fees page allows the admin to filter and sort the shipping fees based on the location, weight range, and shipping fee range. The admin can also view and delete the shipping fees in the table. Figure 6.55 shows the manage shipping fees page.



Location | Input Location

Shipping Fee: RM Start - End

Weight: Start - End g

Shipping Fees List

Selected: 0

□	Location	Starting Weight (g)	Ending Weight (g)	Shipping Fee	Action
□	#Sarawak	9001.00	9500.00	RM 120.95	Edit Delete
□	#Sarawak	7001.00	7500.00	RM 95.93	Edit Delete
□	#Sarawak	7501.00	8000.00	RM 102.18	Edit Delete
□	#Sarawak	8501.00	9000.00	RM 114.69	Edit Delete
□	#Sarawak	9501.00	10000.00	RM 127.20	Edit Delete
□	#Sarawak	8001.00	8500.00	RM 108.44	Edit Delete
□	#Sarawak	6501.00	7000.00	RM 89.68	Edit Delete
□	#Perak	8001.00	9000.00	RM 18.76	Edit Delete
□	#Perak	7001.00	8000.00	RM 17.70	Edit Delete
□	#Perak	9001.00	10000.00	RM 19.82	Edit Delete

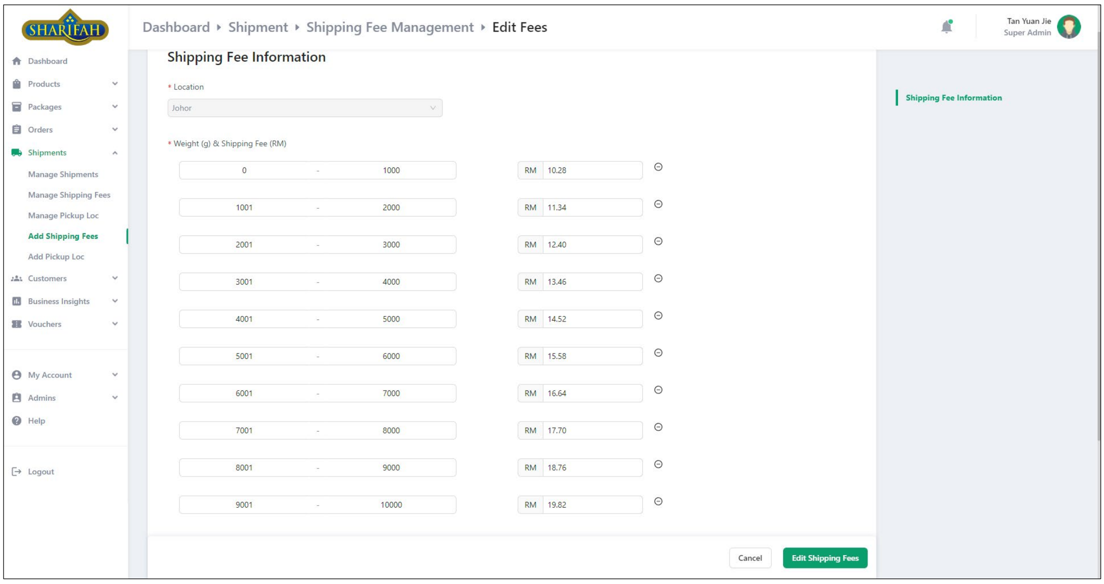
Total 187 items < 1 2 3 4 5 ··· 19 > 10 / page Go to Page

© 2022 Sharifah Food. All rights reserved.

Figure 6.55: Manage Shipping Fees

6.6.5.2 Add/Edit Shipping Fees

The admin is allowed to add and edit the shipping fees based on state. Besides, the add and edit page for shipping fees will show all the shipping fees and respective weight ranges that exist for a state to the admin so that the admin can efficiently manage the shipping fees in a particular state, as shown in Figure 6.56.



Shipping Fee Information

* Location

Johor

* Weight (g) & Shipping Fee (RM)

0	-	1000	RM	10.28
1001	-	2000	RM	11.34
2001	-	3000	RM	12.40
3001	-	4000	RM	13.46
4001	-	5000	RM	14.52
5001	-	6000	RM	15.58
6001	-	7000	RM	16.64
7001	-	8000	RM	17.70
8001	-	9000	RM	18.76
9001	-	10000	RM	19.82

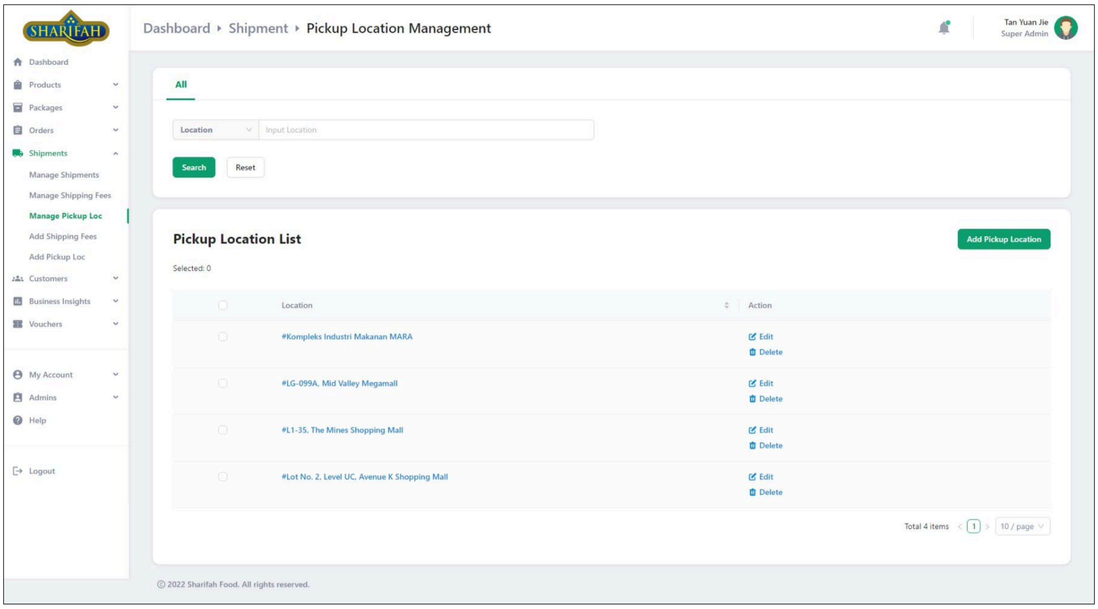
Figure 6.56: Edit Shipping Fees

6.6.6 Pickup Location Management

Pickup location management contains two parts: Manage Pickup Locations and Add/Edit Pickup Location.

6.6.6.1 Manage Pickup Locations

The manage pickup locations page allows the admin to filter and sort the pickup location. The admin can also view and delete the pickup location in the table. Figure 6.57 shows the manage pickup location page.



Location: Input Location

Pickup Location List

Selected: 0

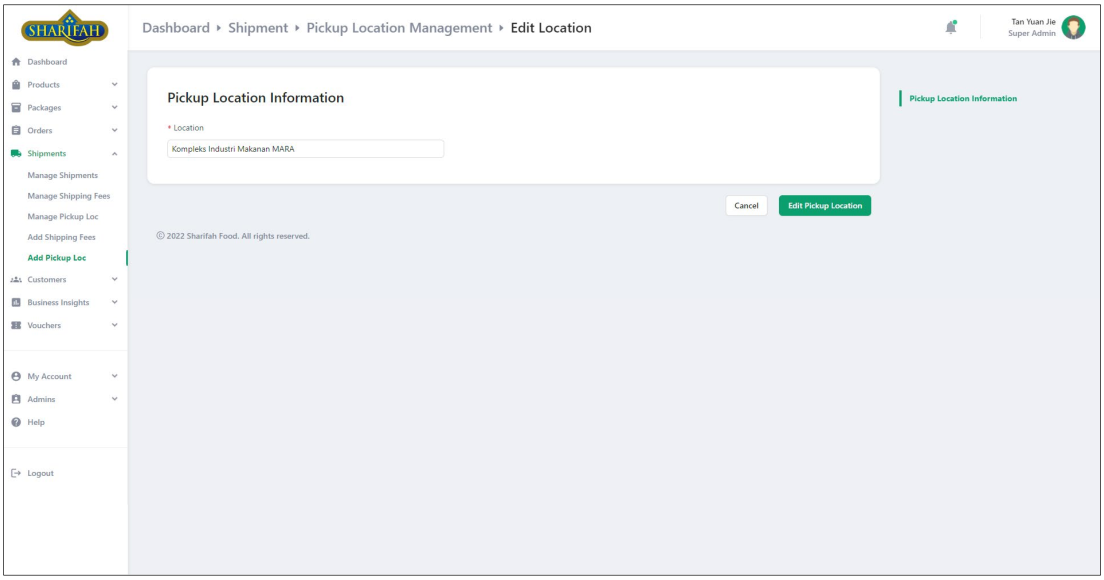
	Location	Action
□	#Kompleks Industri Makanan MARA	Edit Delete
□	#LG-099A, Mid Valley Megamall	Edit Delete
□	#L1-35, The Mines Shopping Mall	Edit Delete
□	#Lot No. 2, Level UC, Avenue K Shopping Mall	Edit Delete

Total 4 items < 1 > 10 / page

Figure 6.57: Manage Pickup Locations

6.6.6.2 Add/Edit Pickup Location

The admin is allowed to add and edit the pickup location. To add a new pickup location, the admin can click the ‘Add Pickup Loc’ menu at the left sidebar or the ‘Add Pickup Location’ on the manage pickup locations page, and the system will redirect the admin to the add pickup location page. Suppose the admin wants to edit the pickup location. In that case, the admin can click the particular location in the table. After that, the pickup location details will then be displayed on the edit pickup location page, as shown in Figure 6.58.



Pickup Location Information

* Location: Kompleks Industri Makanan MARA

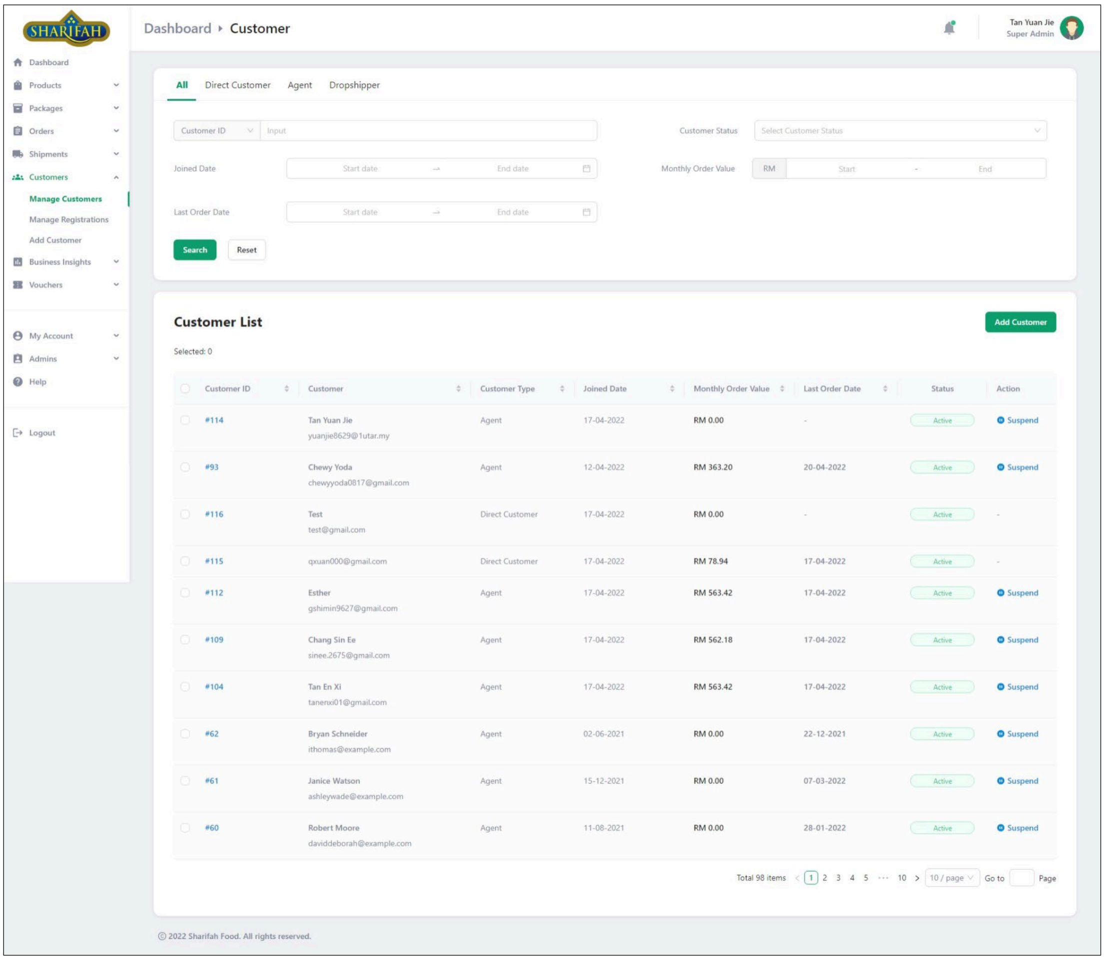
Figure 6.58: Edit Pickup Location

6.6.7 Customer Management

Customer management can be separated into 5 parts: Manage Customers, Manage Customer Registrations, Add Customers, View Customer, and View Customer Registration.

6.6.7.1 Manage Customers

The system allows the admins to view all the customers. Besides, the admin can also perform filtering and sorting on the customer list by customer id, name, email, status, joined date, last order date, and monthly order value. Figure 6.59 shows the UI of the manage customer page.



Customer List

Selected: 0

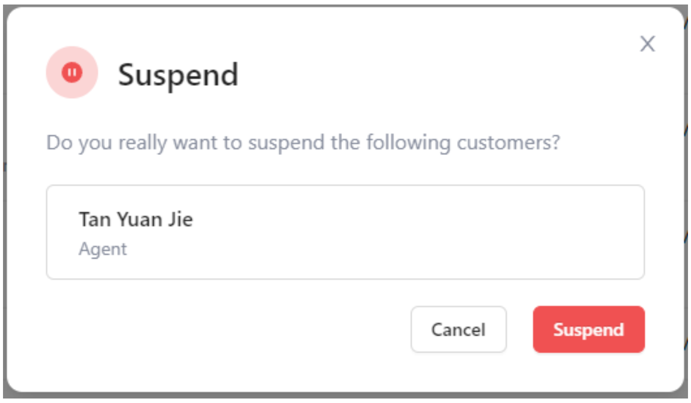
Customer ID	Customer	Customer Type	Joined Date	Monthly Order Value	Last Order Date	Status	Action
#114	Tan Yuan Jie yuanjie8629@1utar.my	Agent	17-04-2022	RM 0.00	-	Active	Suspend
#93	Chewy Yoda chewyyoda0817@gmail.com	Agent	12-04-2022	RM 363.20	20-04-2022	Active	Suspend
#116	Test test@gmail.com	Direct Customer	17-04-2022	RM 0.00	-	Active	-
#115	qxuan000@gmail.com	Direct Customer	17-04-2022	RM 78.94	17-04-2022	Active	-
#112	Esther gshimin9627@gmail.com	Agent	17-04-2022	RM 563.42	17-04-2022	Active	Suspend
#109	Chang Sin Ee sinee.2675@gmail.com	Agent	17-04-2022	RM 562.18	17-04-2022	Active	Suspend
#104	Tan En Xi tanenxi01@gmail.com	Agent	17-04-2022	RM 563.42	17-04-2022	Active	Suspend
#62	Bryan Schneider ithomas@example.com	Agent	02-06-2021	RM 0.00	22-12-2021	Active	Suspend
#61	Janice Watson ashleywade@example.com	Agent	15-12-2021	RM 0.00	07-03-2022	Active	Suspend
#60	Robert Moore daviddeborah@example.com	Agent	11-08-2021	RM 0.00	28-01-2022	Active	Suspend

Total 98 items < 1 2 3 4 5 ··· 10 > 10 / page Go to Page

© 2022 Sharifah Food. All rights reserved.

Figure 6.59: Manage Customers

In addition, the admin is able to view the monthly order value of the customers. By viewing the monthly order value, the admin is able to obtain the monthly sales of the agents and dropshippers. If the sales performance of the agent or dropshipper is poor, the admin can decide to suspend the agent or dropshipper, as shown in Figure 6.60. Besides, the admin can also activate the suspended agent or dropshipper account, as shown in Figure 6.60 and Figure 6.61. Once the admin suspends or activates an agent or a dropshipper, an email notification will be sent to the agent or dropshipper to notify them about the suspension or activation on their accounts. Figure 6.62 shows the email received by the agent or dropshipper on the suspension and activation notifications.

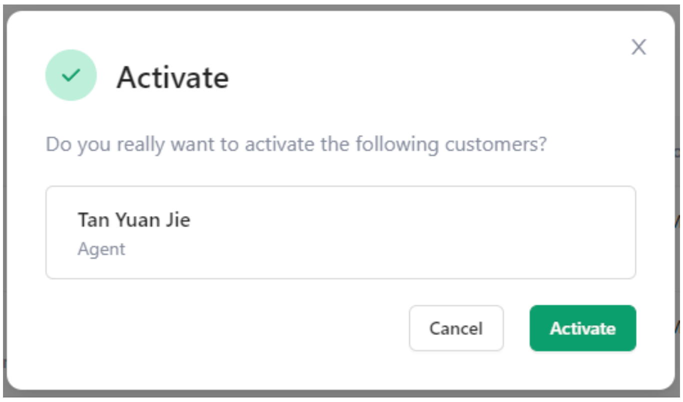


Suspend

Do you really want to suspend the following customers?

Tan Yuan Jie
Agent

Figure 6.60: Suspend Agent or Dropshipper Account

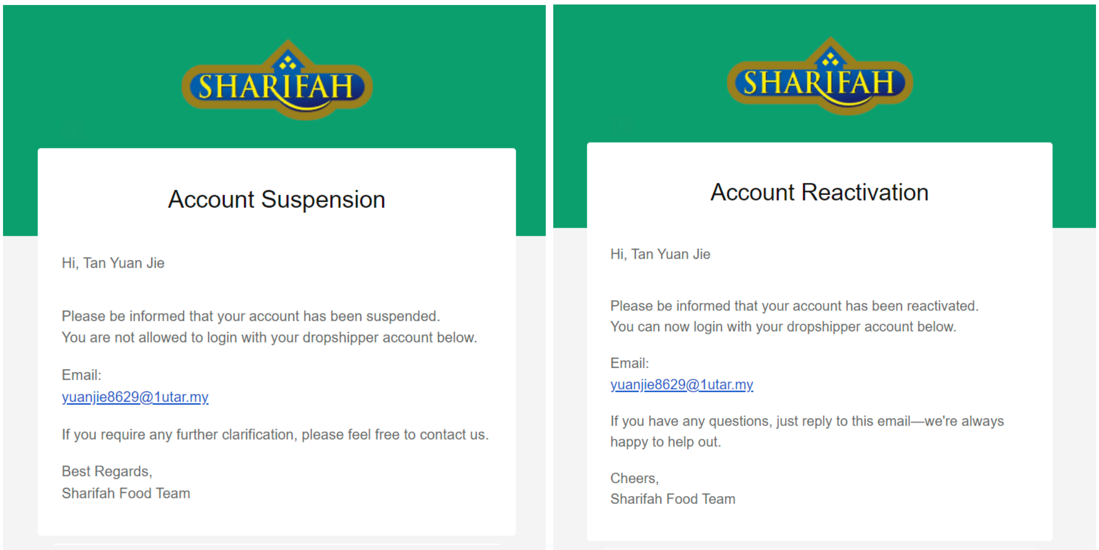


Activate

Do you really want to activate the following customers?

Tan Yuan Jie
Agent

Figure 6.61: Activate Agent or Dropshipper Account



Account Suspension

Hi, Tan Yuan Jie

Please be informed that your account has been suspended.
You are not allowed to login with your dropshipper account below.

Email:
yuanjie8629@1utar.my

If you require any further clarification, please feel free to contact us.

Best Regards,
Sharifah Food Team

Account Reactivation

Hi, Tan Yuan Jie

Please be informed that your account has been reactivated.
You can now login with your dropshipper account below.

Email:
yuanjie8629@1utar.my

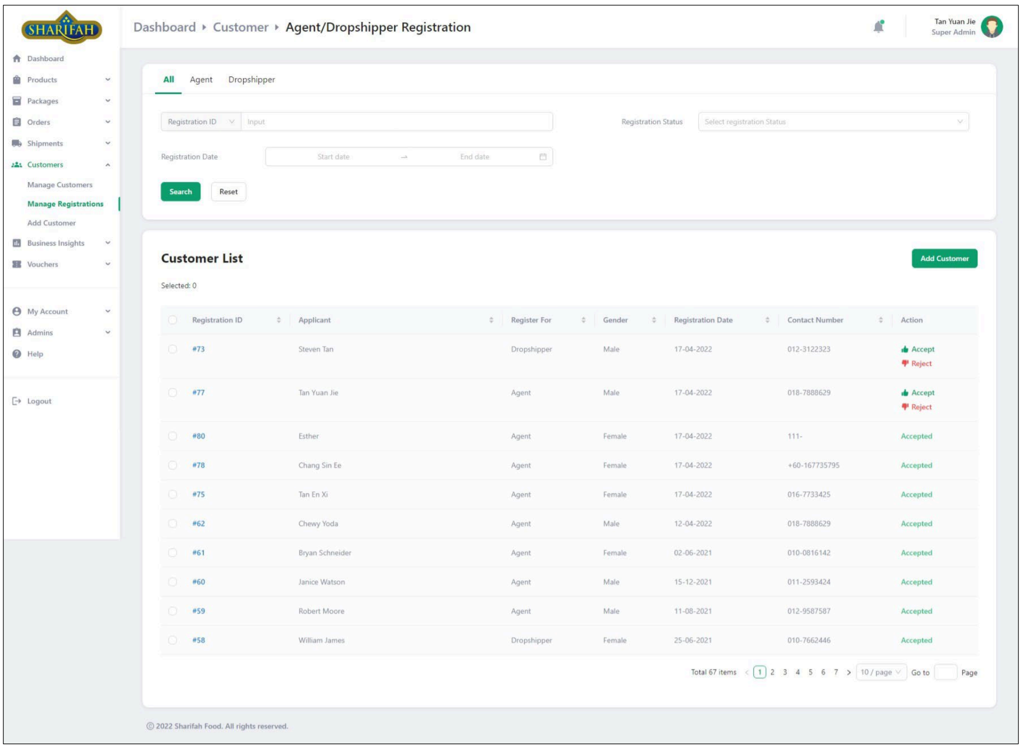
If you have any questions, just reply to this email—we're always happy to help out.

Cheers,
Sharifah Food Team

Figure 6.62: Email Notification on Suspension or Activation of Account

6.6.7.2 Manage Customer Registrations

As the customer can register for agent and dropshipper positions on the e-commerce platform, the admin would require to review and approve the customer registrations. On the manage customer registration, the admin can filter and sort the registrations by registration id, applicant name, email, contact number, registration status, and registration date, as shown in Figure 6.63.



Registration ID: Input

Registration Status: Select registration Status

Registration Date: Start date → End date

Customer List

Selected: 0

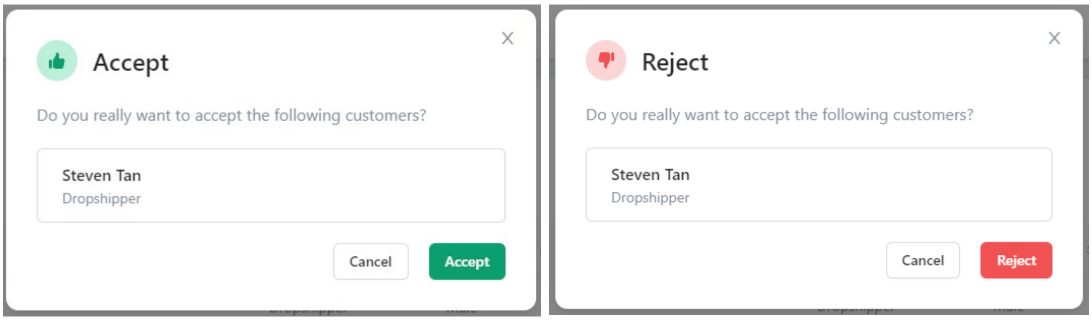
Registration ID	Applicant	Register For	Gender	Registration Date	Contact Number	Action
#73	Steven Tan	Dropshipper	Male	17-04-2022	012-3122323	Accept Reject
#77	Tan Yuan Jie	Agent	Male	17-04-2022	018-7888629	Accept Reject
#80	Esther	Agent	Female	17-04-2022	111-	Accepted
#78	Chang Sin Ee	Agent	Female	17-04-2022	+60-167735795	Accepted
#75	Tan En Xi	Agent	Female	17-04-2022	016-7733425	Accepted
#62	Chewy Yoda	Agent	Male	12-04-2022	018-7888629	Accepted
#61	Bryan Schneider	Agent	Female	02-06-2021	010-0816142	Accepted
#60	Janice Watson	Agent	Male	15-12-2021	011-2593424	Accepted
#59	Robert Moore	Agent	Male	11-08-2021	012-9587587	Accepted
#58	William James	Dropshipper	Female	25-06-2021	010-7662446	Accepted

Total 67 items < 1 2 3 4 5 6 7 > 10 / page Go to Page

© 2022 Sharifah Food. All rights reserved.

Figure 6.63: Manage Customer Registration

Besides, after some consideration, the admin can accept or reject the agent or dropshipper registration by clicking the ‘Accept’ or ‘Reject’ buttons. After the admin clicks on the ‘Accept’ or ‘Reject’ buttons, a confirmation modal will be popped up for confirmation, as shown in Figure 6.64. Once the admin accepts or rejects the customer registration, the customer will receive an email notification on the registration outcome. In addition, if the customer’s email does not exist in the system, the system will automatically create an account with a random password and send it to the customer along with the email notification, as shown in Figure 6.65. On the other hand, if the registration is rejected, the applicant would receive a rejection email, as shown in Figure 6.66.



Accept

Do you really want to accept the following customers?

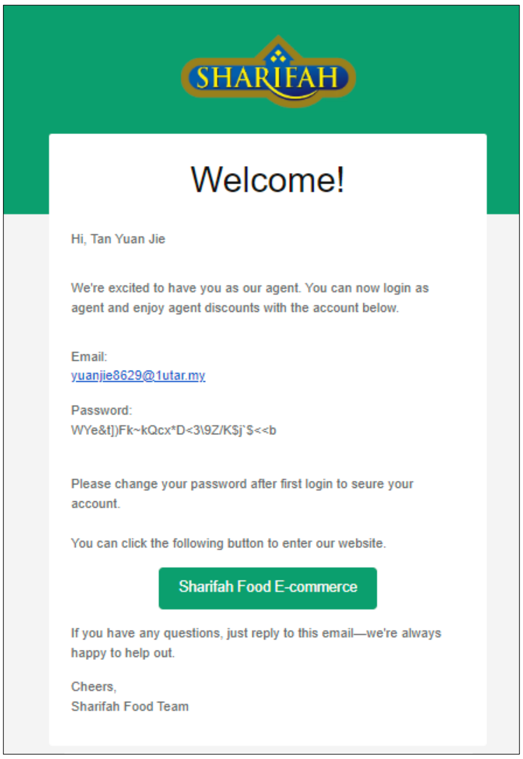
Steven Tan
Dropshipper

Reject

Do you really want to accept the following customers?

Steven Tan
Dropshipper

Figure 6.64: Confirmation Modal on Accepting or Rejecting Customer Registration



Welcome!

Hi, Tan Yuan Jie

We're excited to have you as our agent. You can now login as agent and enjoy agent discounts with the account below.

Email:
yuanjie8629@1utar.my

Password:
WYe&t])Fk~kQcx*D<3\9Z/K$j`$<<b

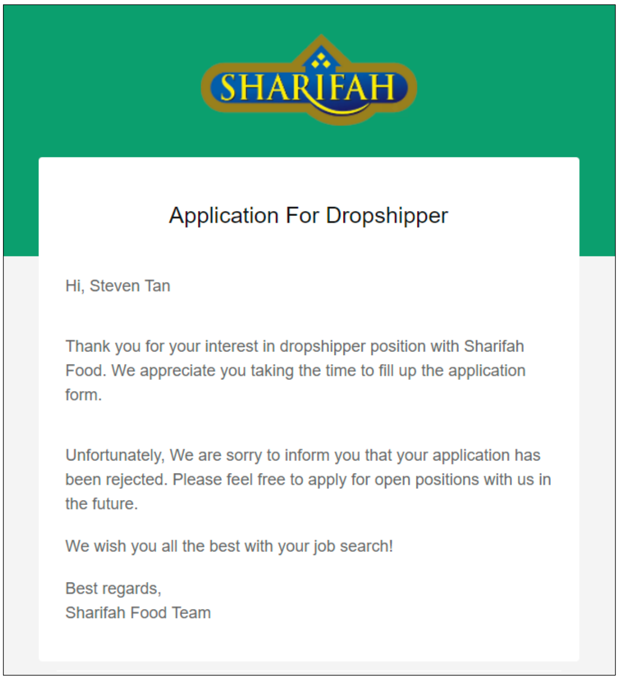
Please change your password after first login to seure your account.

You can click the following button to enter our website.

If you have any questions, just reply to this email—we're always happy to help out.

Cheers,
Sharifah Food Team

Figure 6.65: Email Notification on Success Registraion as Agent or Dropshipper



Application For Dropshipper

Hi, Steven Tan

Thank you for your interest in dropshipper position with Sharifah Food. We appreciate you taking the time to fill up the application form.

Unfortunately, We are sorry to inform you that your application has been rejected. Please feel free to apply for open positions with us in the future.

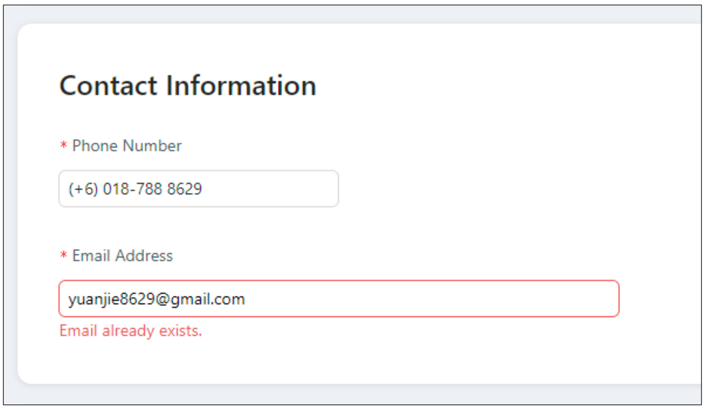
We wish you all the best with your job search!

Best regards,
Sharifah Food Team

Figure 6.66: Agent/Dropshipper Registration Rejection Email

6.6.7.3 Add Customer

Without the customer submitting the registration form on the e-commerce platform and pending approval from the admin, the admin can also add the customer as agent or dropshipper in the inventory management system by clicking the ‘Add Customer’ button on the menu at the left sidebar or the manage customer and manage customer registration pages. After that, the admin will be redirected to the add customer page and required to fill in the basic information, contact information, address, employment details and the position of the new agent or dropshipper. If the email entered exists in the system, the error message would be shown in Figure 6.67.



Contact Information

* Phone Number

(+6) 018-788 8629

* Email Address

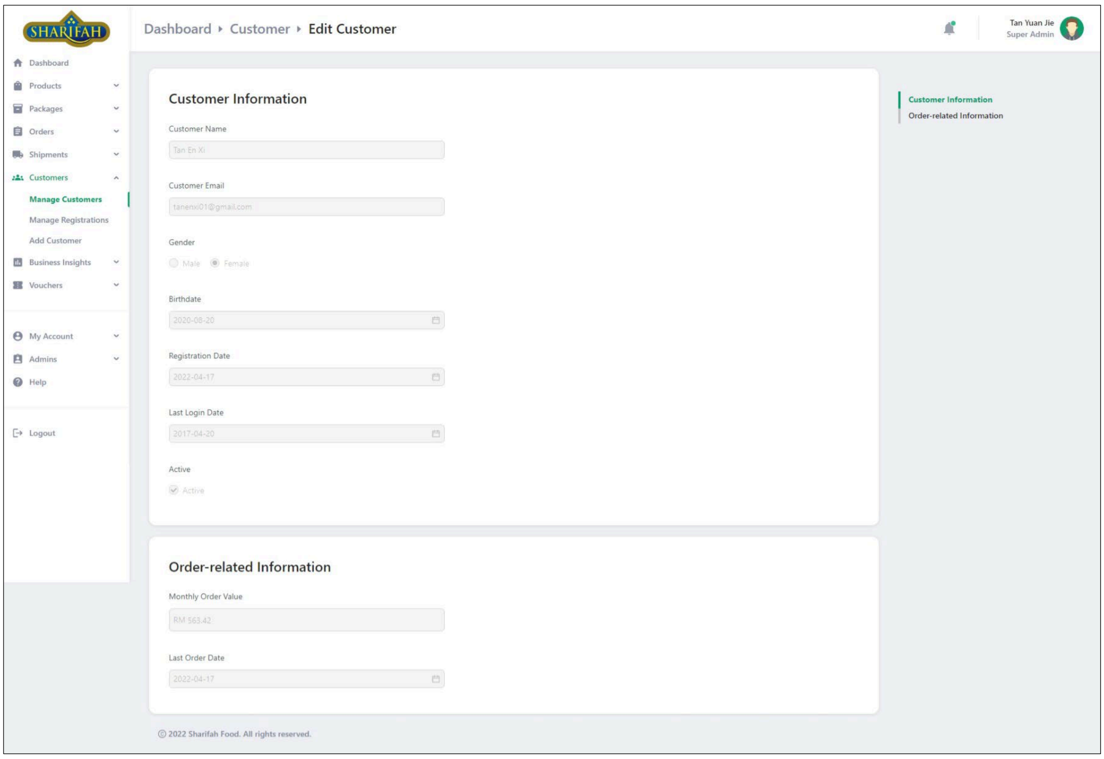
yuanjie8629@gmail.com

Email already exists.

Figure 6.67: Email Validation on Add Customer Page

6.6.7.4 View Customer

The admin can click the customer id in the table on the manage customer page to view the customer details. After the admin clicks the customer id, the system will redirect to the view customer page and display the customer details, as shown in Figure 6.68.



Customer Information

Customer Name: Tan En Xi

Customer Email: tanenxi01@gmail.com

Gender: ○ Male ● Female

Birthdate: 2020-08-20

Registration Date: 2022-04-17

Last Login Date: 2017-04-20

Active: ☑ Active

Order-related Information

Monthly Order Value: RM 563.42

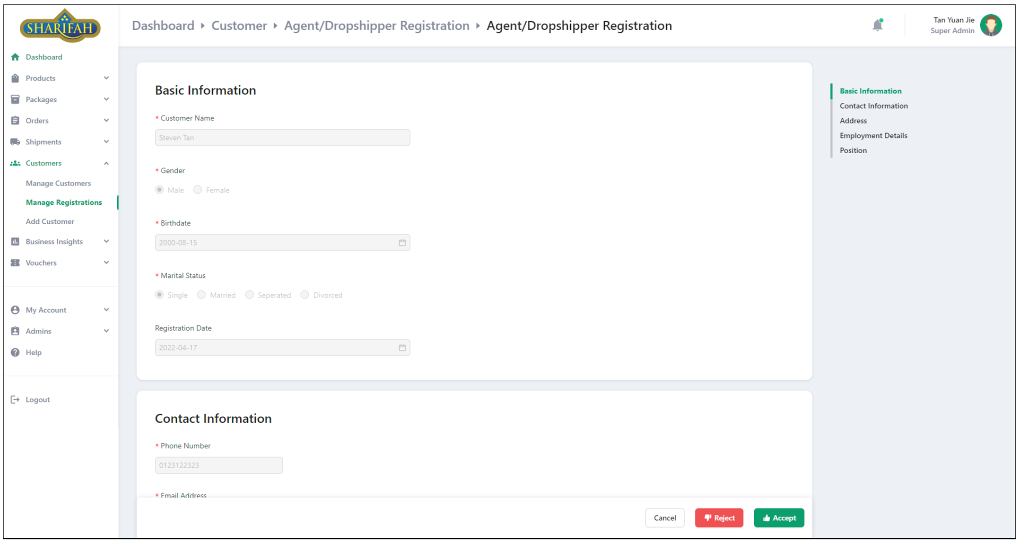
Last Order Date: 2022-04-17

© 2022 Sharifah Food. All rights reserved.

Figure 6.68: View Customer

6.6.7.5 View Customer Registration

The admin can also view the customer registration by clicking the registration id on the manage customer registration page. Once the admin clicks the registration id, the admin will be redirected to the registration details page, as shown in Figure 6.69. If the registration has not been accepted or rejected, the ‘Accept’ and ‘Reject’ buttons will also be available on the view customer registration page.



Basic Information

* Customer Name: Steven Tan

* Gender: ● Male ○ Female

* Birthdate: 2000-08-15

* Marital Status: ● Single ○ Married ○ Seperated ○ Divorced

Registration Date: 2022-04-17

Contact Information

* Phone Number: 0123122323

* Email Address

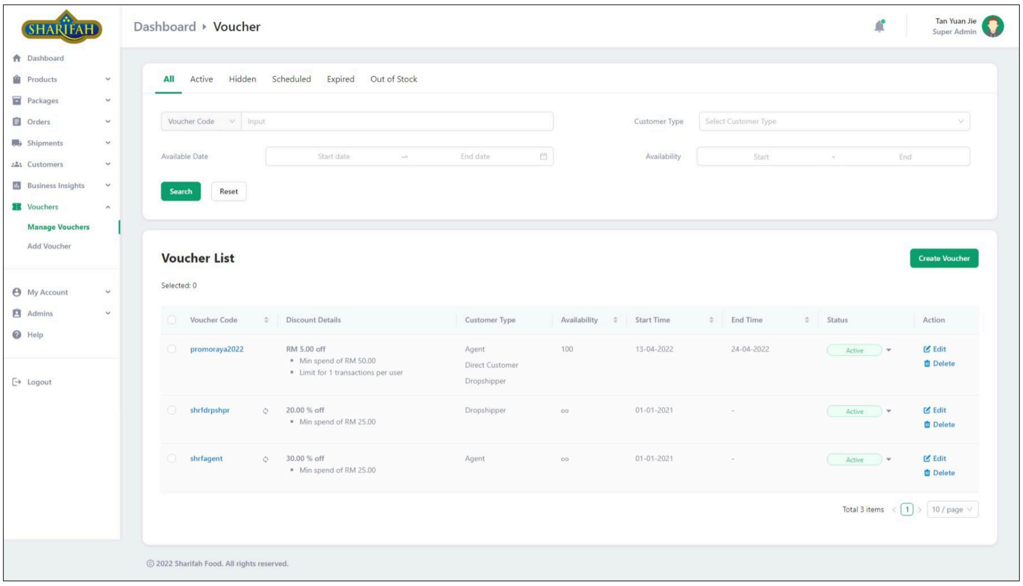
Figure 6.69: View Agent/Dropshipper Registration Details

6.6.8 Voucher Management

The voucher management can be separated into two parts: Manage Voucher and Add/Edit Voucher.

6.6.8.1 Manage Voucher

The manage voucher page allows the admins to filter and sort by voucher code, eligible customer type, available date, and availability, as shown in Figure 6.70. Besides, the admin can also perform bulk operations to hide or delete the vouchers.



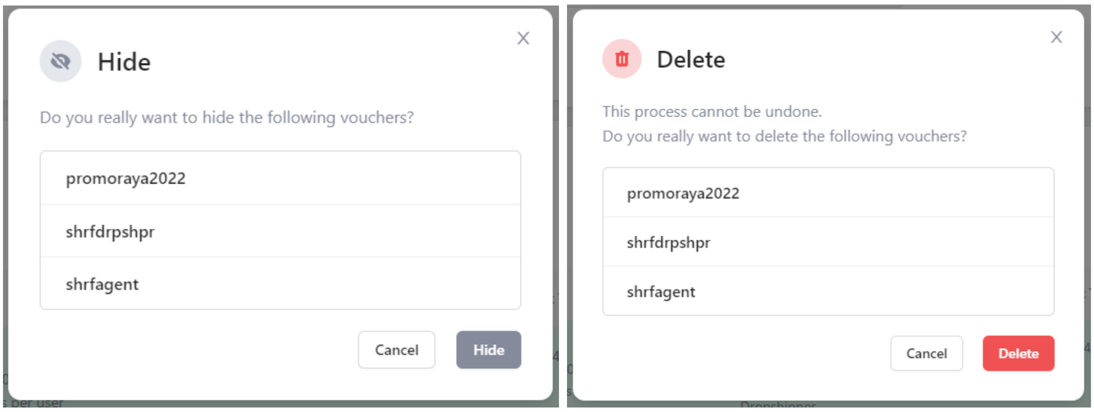
Voucher List

Selected: 0

	Voucher Code	Discount Details	Customer Type	Availability	Start Time	End Time	Status	Action
○	promoraya2022	RM 5.00 off • Min spend of RM 50.00 • Limit for 1 transactions per user	Agent Direct Customer Dropshipper	100	13-04-2022	24-04-2022	Active	Edit Delete
○	shrfdrpshpr	20.00 % off • Min spend of RM 25.00	Dropshipper	∞	01-01-2021	-	Active	Edit Delete
○	shrfagent	30.00 % off • Min spend of RM 25.00	Agent	∞	01-01-2021	-	Active	Edit Delete

Figure 6.70: Manage Voucher

If the admin wants to hide or delete the voucher, the admin can select the vouchers and click the ‘Hide’ or ‘Delete’ button, and a confirmation modal will be popped up to confirm the operation, as shown in Figure 6.71.



Hide

Do you really want to hide the following vouchers?

promoraya2022
shrfdrpshpr
shrfagent

Delete

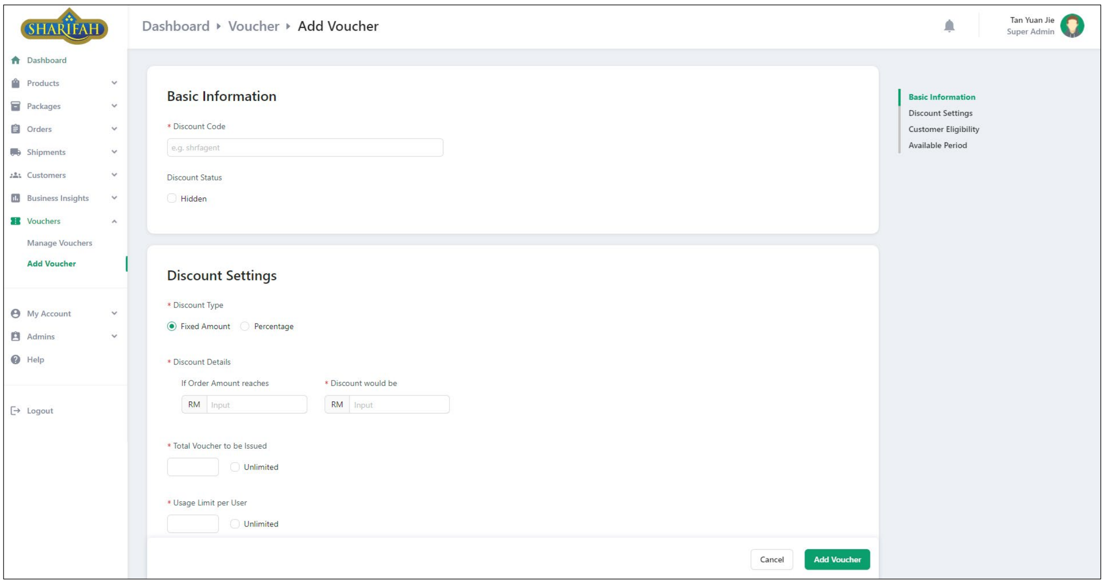
This process cannot be undone.
Do you really want to delete the following vouchers?

promoraya2022
shrfdrpshpr
shrfagent

Figure 6.71: Confirmation Modal to Hide and Delete Voucher

6.6.8.2 Add/Edit Voucher

In order to add the voucher, the admin can click the ‘Add Voucher’ menu on the left sidebar or the manage products page. To edit a product, the admin can click on the edit button or the blue-coloured voucher code. The add and edit voucher pages are the same, except that the edit voucher will display all the selected voucher details for the admin to modify. Figure 6.72 shows the part of the add voucher page.



Basic Information

* Discount Code

e.g. shrfagent

Discount Status

□ Hidden

Discount Settings

* Discount Type

● Fixed Amount ○ Percentage

* Discount Details

If Order Amount reaches: RM Input

* Discount would be: RM Input

* Total Voucher to be Issued

□ Unlimited

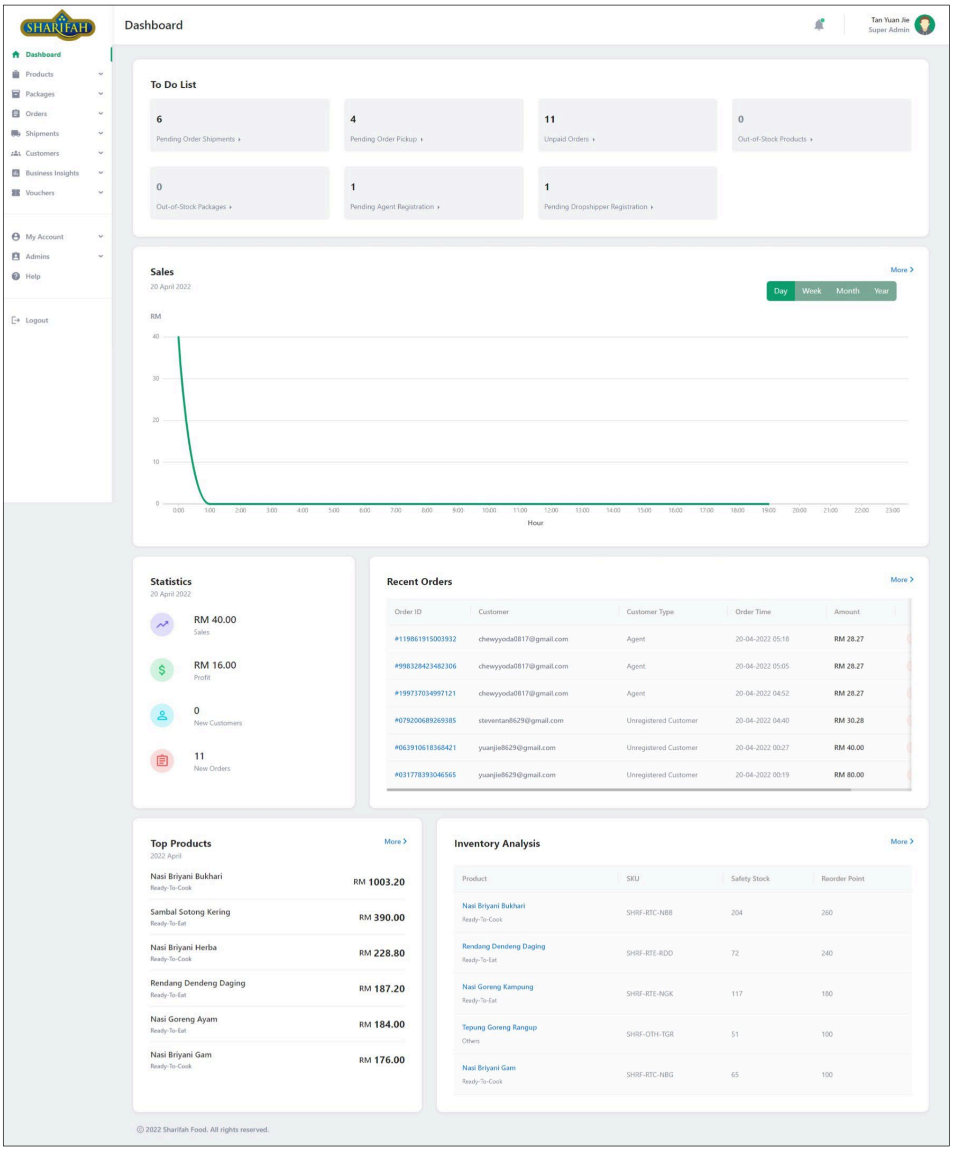
* Usage Limit per User

□ Unlimited

Figure 6.72: Part of Add Voucher Page

6.6.9 Dashboard

The dashboard shows the overview of the system, including the to-do list, sales analysis, statistics, recent orders, top products, and inventory analysis. The to-do list tells the admin the number of items that require the admin's actions. Besides, the sales analysis shows the analysis of the sales for the current date period. The admin can choose to view the sales in a day, week, month, or year range. The statistics show today's total number of sales, profit, new customers, and new orders. Moreover, the recent orders show the six most recent orders placed by the customers. The top products show the top six products that have the highest sales. Lastly, the inventory analysis shows a brief overview of the safety stock inventory analysis. Figure 6.73 shows the dashboard of the inventory management system.



Dashboard

To Do List

6
Pending Order Shipments

4
Pending Order Pickup

11
Unpaid Orders

0
Out-of-Stock Products

0
Out-of-Stock Packages

1
Pending Agent Registration

1
Pending Dropshipper Registration

Sales
20 April 2022

Statistics
20 April 2022

RM 40.00
Sales

RM 16.00
Profit

0
New Customers

11
New Orders

Recent Orders

Order ID	Customer	Customer Type	Order Time	Amount
#119861915003932	chewyyoda0817@gmail.com	Agent	20-04-2022 05:18	RM 28.27
#998328423482306	chewyyoda0817@gmail.com	Agent	20-04-2022 05:05	RM 28.27
#199737034997121	chewyyoda0817@gmail.com	Agent	20-04-2022 04:52	RM 28.27
#079200689269385	steventan8629@gmail.com	Unregistered Customer	20-04-2022 04:40	RM 30.28
#063910618368421	yuanjie8629@gmail.com	Unregistered Customer	20-04-2022 00:27	RM 40.00
#031778393046565	yuanjie8629@gmail.com	Unregistered Customer	20-04-2022 00:19	RM 80.00

Top Products
2022 April

Nasi Briyani Bukhari Ready-To-Cook	RM 1003.20
Sambal Sotong Kering Ready-To-Eat	RM 390.00
Nasi Briyani Herba Ready-To-Cook	RM 228.80
Rendang Dendeng Daging Ready-To-Eat	RM 187.20
Nasi Goreng Ayam Ready-To-Eat	RM 184.00
Nasi Briyani Gam Ready-To-Cook	RM 176.00

Inventory Analysis

Product	SKU	Safety Stock	Reorder Point
Nasi Briyani Bukhari Ready-To-Cook	SHRF-RTC-NBB	204	260
Rendang Dendeng Daging Ready-To-Eat	SHRF-RTE-RDD	72	240
Nasi Goreng Kampung Ready-To-Eat	SHRF-RTE-NGK	117	180
Tepung Goreng Rangup Others	SHRF-OTH-TGR	51	100
Nasi Briyani Gam Ready-To-Cook	SHRF-RTC-NBG	65	100

© 2022 Sharifah Food. All rights reserved.

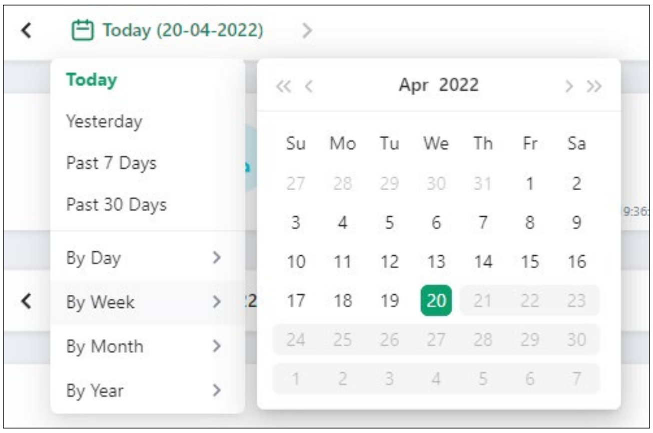
Figure 6.73: Dashboard

6.6.10 Business Insights

The business insights can be divided into two sections: Statistics and Inventory Analysis.

6.6.10.1 Statistics

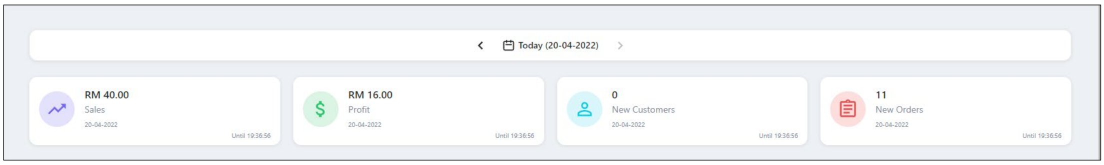
The statistics page shows the analyses that are mainly based on order sales. The statistics page has three sections: the statistics summary, key metrics analyses, and ranking list. In addition, the admin can select the date range by clicking the date range component. As we can see from Figure 6.74, the admin can select the date by day, week, month, or year. Besides, the admin can also choose for today, yesterday, past 7 days, and past 30 days to view the analyses based on the selected date range.



Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Figure 6.74: Date Range Component

The statistics summary shows the total sales, profit, new customers, and new orders based on the selected date range, as shown in Figure 6.75.


Today (20-04-2022)

RM 40.00
Sales
20-04-2022
Until 19:36:56

RM 16.00
Profit
20-04-2022
Until 19:36:56

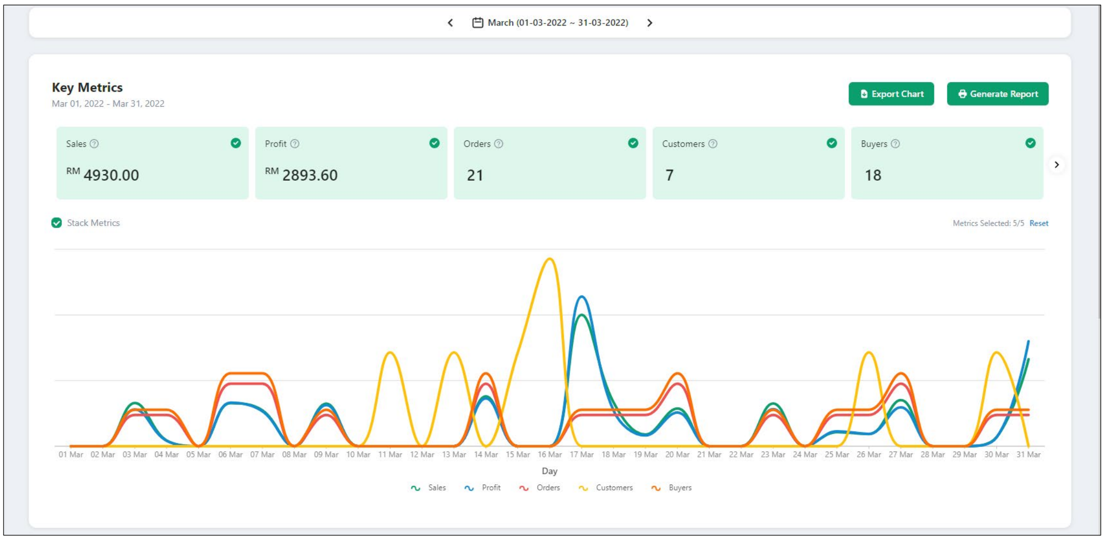
0
New Customers
20-04-2022
Until 19:36:56

11
New Orders
20-04-2022
Until 19:36:56

Figure 6.75: Statistics Summary

The key metrics analyses contain up to 8 key metrics, including the sales, profit, orders, customers, buyers, average order value, item units sold, and average basket size. The admin can either view a single key metrics analysis by checking the

“Stack Metrics” or stack the line charts of the key metrics analyses, as shown in Figure 6.76. However, the maximum number of metrics selected to be stacked is five metrics to avoid overcomplicating the line chart display.



March (01-03-2022 ~ 31-03-2022)

Key Metrics
Mar 01, 2022 - Mar 31, 2022

Export Chart | Generate Report

Sales	Profit	Orders	Customers	Buyers
RM 4930.00	RM 2893.60	21	7	18

☑ Stack Metrics

Metrics Selected: 5/5 Reset

01 Mar 02 Mar 03 Mar 04 Mar 05 Mar 06 Mar 07 Mar 08 Mar 09 Mar 10 Mar 11 Mar 12 Mar 13 Mar 14 Mar 15 Mar 16 Mar 17 Mar 18 Mar 19 Mar 20 Mar 21 Mar 22 Mar 23 Mar 24 Mar 25 Mar 26 Mar 27 Mar 28 Mar 29 Mar 30 Mar 31 Mar
Day
Sales
Profit
Orders
Customers
Buyers

Figure 6.76: Key Metrics Analyses

In addition, the admin can export the line chart to an image by clicking the “Export Chart” button. The exported chart is shown in Figure 6.77. Another feature of the key metrics analysis is that the admin can generate an Excel report by clicking the “Generate Report” button. Figure 6.78 shows the generated Excel report after clicking the button.

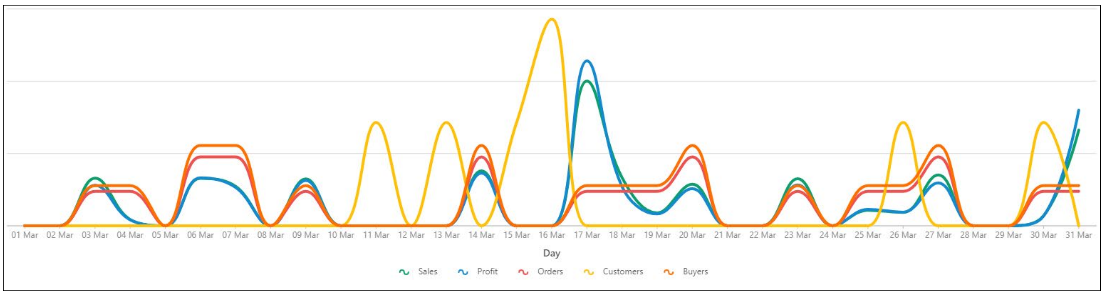
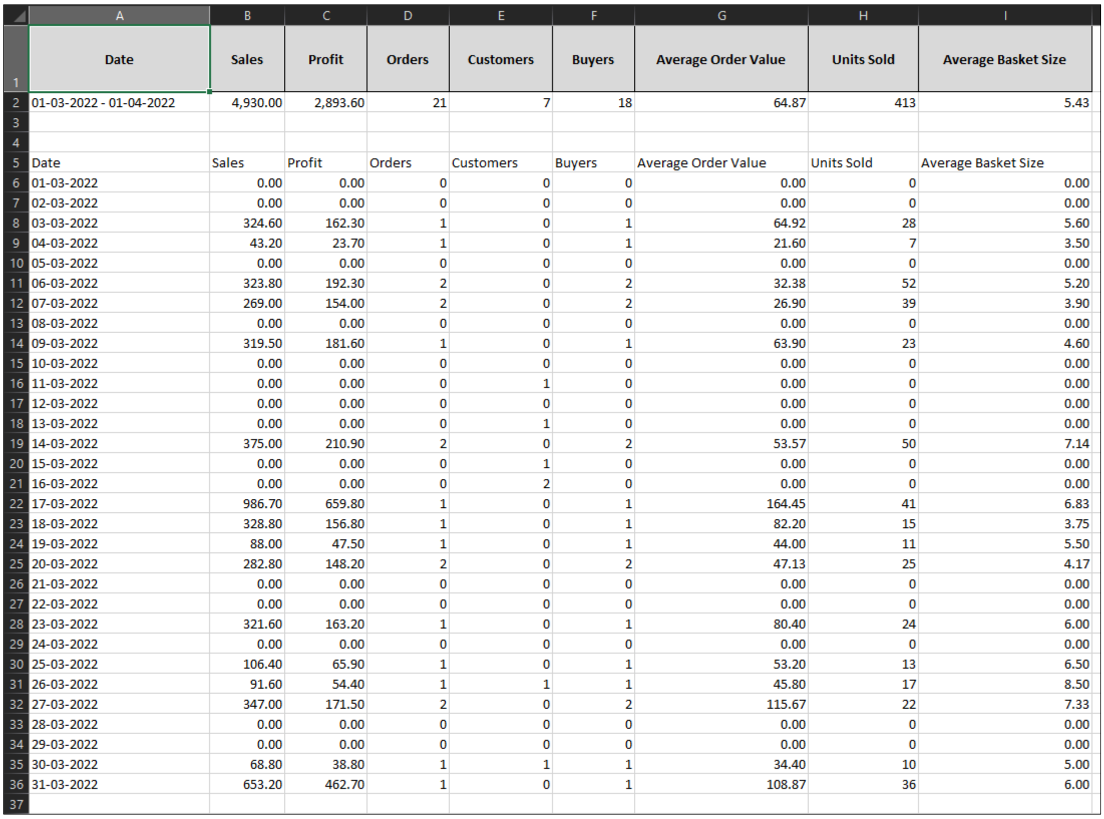


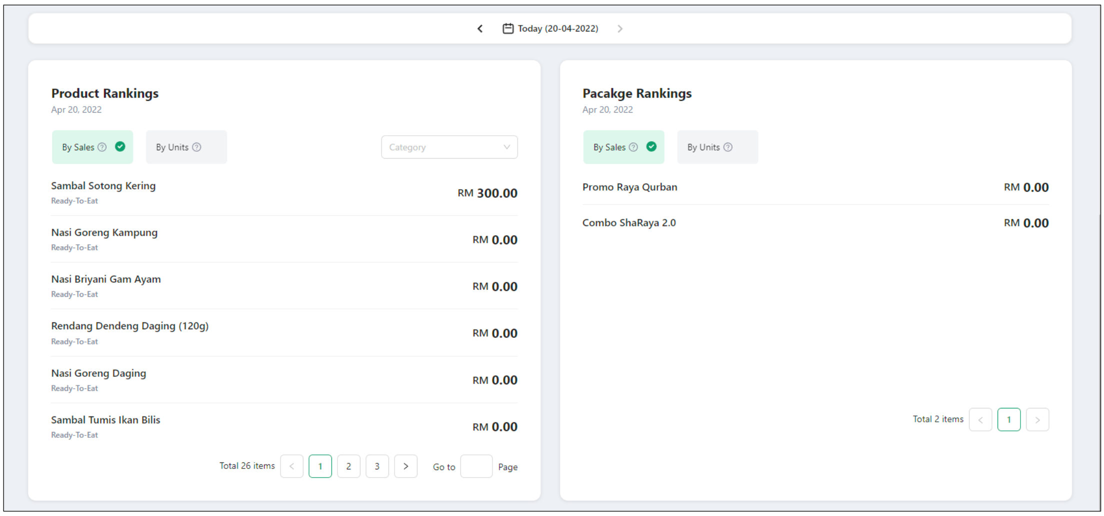
Figure 6.77: Export Chart



	A	B	C	D	E	F	G	H	I
1	**Date**	**Sales**	**Profit**	**Orders**	**Customers**	**Buyers**	**Average Order Value**	**Units Sold**	**Average Basket Size**
2	01-03-2022 - 01-04-2022	4,930.00	2,893.60	21	7	18	64.87	413	5.43
3									
4									
5	Date	Sales	Profit	Orders	Customers	Buyers	Average Order Value	Units Sold	Average Basket Size
6	01-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
7	02-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
8	03-03-2022	324.60	162.30	1	0	1	64.92	28	5.60
9	04-03-2022	43.20	23.70	1	0	1	21.60	7	3.50
10	05-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
11	06-03-2022	323.80	192.30	2	0	2	32.38	52	5.20
12	07-03-2022	269.00	154.00	2	0	2	26.90	39	3.90
13	08-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
14	09-03-2022	319.50	181.60	1	0	1	63.90	23	4.60
15	10-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
16	11-03-2022	0.00	0.00	0	1	0	0.00	0	0.00
17	12-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
18	13-03-2022	0.00	0.00	0	1	0	0.00	0	0.00
19	14-03-2022	375.00	210.90	2	0	2	53.57	50	7.14
20	15-03-2022	0.00	0.00	0	1	0	0.00	0	0.00
21	16-03-2022	0.00	0.00	0	2	0	0.00	0	0.00
22	17-03-2022	986.70	659.80	1	0	1	164.45	41	6.83
23	18-03-2022	328.80	156.80	1	0	1	82.20	15	3.75
24	19-03-2022	88.00	47.50	1	0	1	44.00	11	5.50
25	20-03-2022	282.80	148.20	2	0	2	47.13	25	4.17
26	21-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
27	22-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
28	23-03-2022	321.60	163.20	1	0	1	80.40	24	6.00
29	24-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
30	25-03-2022	106.40	65.90	1	0	1	53.20	13	6.50
31	26-03-2022	91.60	54.40	1	1	1	45.80	17	8.50
32	27-03-2022	347.00	171.50	2	0	2	115.67	22	7.33
33	28-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
34	29-03-2022	0.00	0.00	0	0	0	0.00	0	0.00
35	30-03-2022	68.80	38.80	1	1	1	34.40	10	5.00
36	31-03-2022	653.20	462.70	1	0	1	108.87	36	6.00
37									

Figure 6.78: Generated Excel Report

Lastly, the admin can view the product and package rankings by sales or units. Similar to the analyses mentioned above, the admin is able to select the date range for the rankings. Besides, pagination is applied so that the ranking list will not be too long to fit on the screen. Figure 6.79 shows the ranking lists of the product and packages.



Product Rankings

Apr 20, 2022

Sambal Sotong Kering Ready-To-Eat	RM 300.00
Nasi Goreng Kampung Ready-To-Eat	RM 0.00
Nasi Briyani Gam Ayam Ready-To-Eat	RM 0.00
Rendang Dendeng Daging (120g) Ready-To-Eat	RM 0.00
Nasi Goreng Daging Ready-To-Eat	RM 0.00
Sambal Tumis Ikan Bilis Ready-To-Eat	RM 0.00

Pacakge Rankings

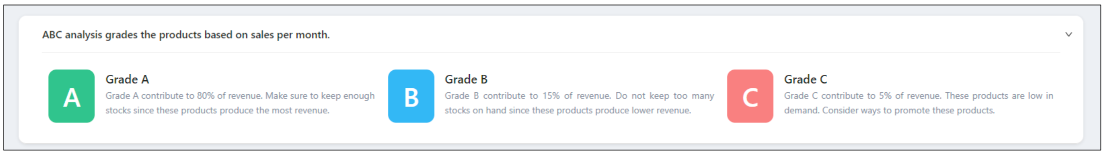
Apr 20, 2022

Promo Raya Qurban	RM 0.00
Combo ShaRaya 2.0	RM 0.00

Figure 6.79: Ranking Lists

6.6.11 Inventory Analysis

The inventory analysis page shows the analyses that are mainly based on product inventory. As discussed in CHAPTER 2, the inventory management system will provide four inventory analyses: the ABC, HML, EOQ, and SS analyses. The analyses are computed based on the sales per month. Besides, the admin cannot choose to view the inventory analyses on the current month as it will result in inaccurate analyses. In order to provide a better insight into the purpose of each inventory analysis, brief descriptions are available to the admin for viewing, as shown in Figure 6.80 and Figure 6.81. Moreover, the EOQ and SS analyses require additional parameters to compute the analysis result. Therefore, additional components are shown to display the products with missing parameters so that the admin can acknowledge those products, as shown in Figure 6.82. Figure 6.83 shows the ABC analysis page in the inventory management system.



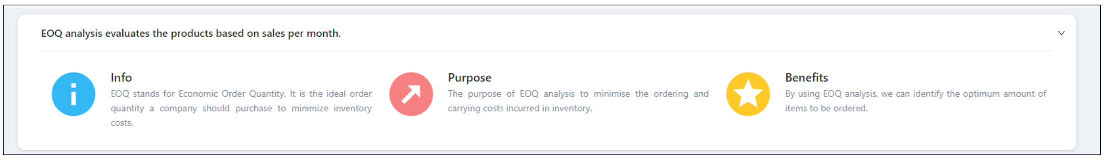
ABC analysis grades the products based on sales per month.

A Grade A
Grade A contribute to 80% of revenue. Make sure to keep enough stocks since these products produce the most revenue.

B Grade B
Grade B contribute to 15% of revenue. Do not keep too many stocks on hand since these products produce lower revenue.

C Grade C
Grade C contribute to 5% of revenue. These products are low in demand. Consider ways to promote these products.

Figure 6.80: Brief Introduction to ABC Analysis



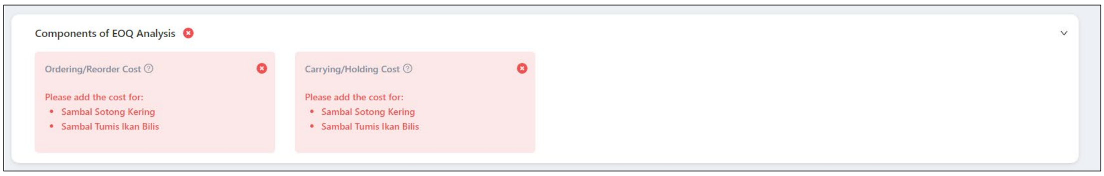
EOQ analysis evaluates the products based on sales per month.

Info
EOQ stands for Economic Order Quantity. It is the ideal order quantity a company should purchase to minimize inventory costs.

Purpose
The purpose of EOQ analysis to minimise the ordering and carrying costs incurred in inventory.

Benefits
By using EOQ analysis, we can identify the optimum amount of items to be ordered.

Figure 6.81: Brief Introduction to EOQ Analysis



Components of EOQ Analysis

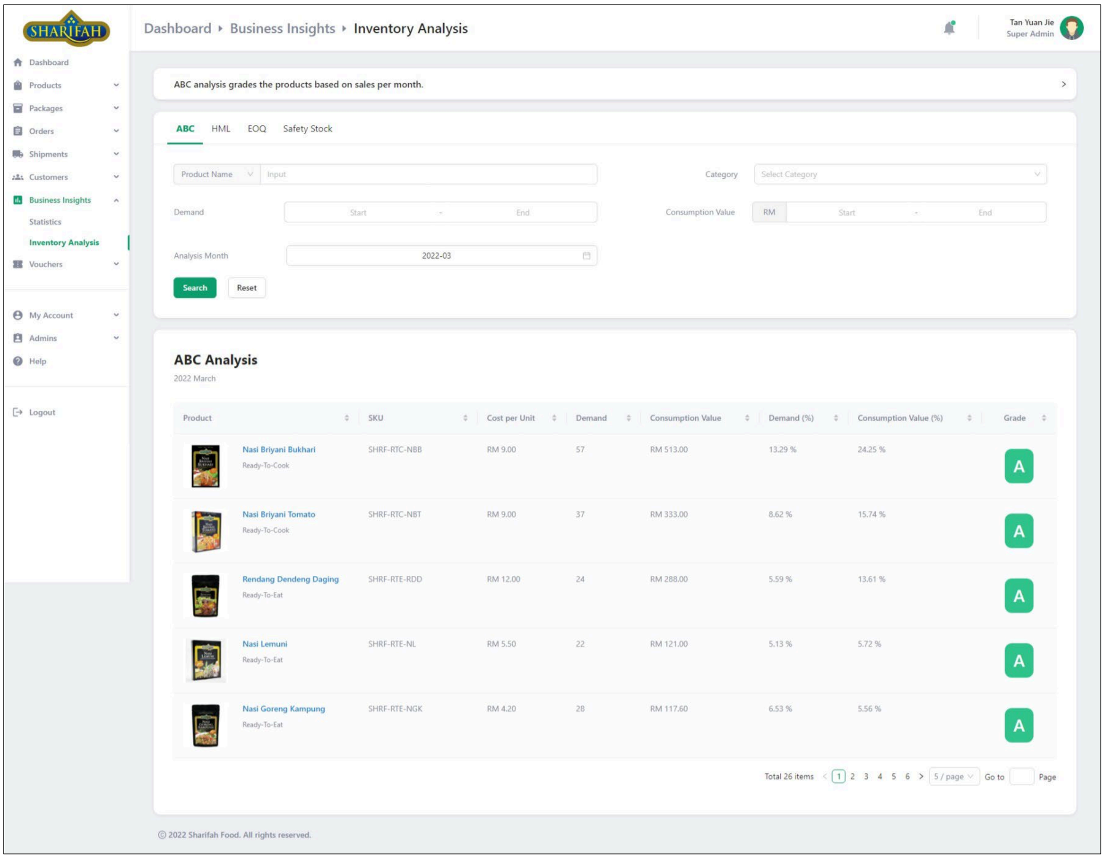
Ordering/Reorder Cost

Please add the cost for:
- Sambal Sotong Kering
- Sambal Tumis Ikan Bilis

Carrying/Holding Cost

Please add the cost for:
- Sambal Sotong Kering
- Sambal Tumis Ikan Bilis

Figure 6.82: Components of EOQ Analysis



ABC analysis grades the products based on sales per month.

Product Name: Input

Category: Select Category

Demand: Start - End

Consumption Value: RM Start - End

Analysis Month: 2022-03

ABC Analysis

2022 March

Product	SKU	Cost per Unit	Demand	Consumption Value	Demand (%)	Consumption Value (%)	Grade
Nasi Briyani Bukhari Ready-To-Cook	SHRF-RTC-NBB	RM 9.00	57	RM 513.00	13.29 %	24.25 %	A
Nasi Briyani Tomato Ready-To-Cook	SHRF-RTC-NBT	RM 9.00	37	RM 333.00	8.62 %	15.74 %	A
Rendang Dendeng Daging Ready-To-Eat	SHRF-RTE-RDD	RM 12.00	24	RM 288.00	5.59 %	13.61 %	A
Nasi Lemuni Ready-To-Eat	SHRF-RTE-NL	RM 5.50	22	RM 121.00	5.13 %	5.72 %	A
Nasi Goreng Kampung Ready-To-Eat	SHRF-RTE-NGK	RM 4.20	28	RM 117.60	6.53 %	5.56 %	A

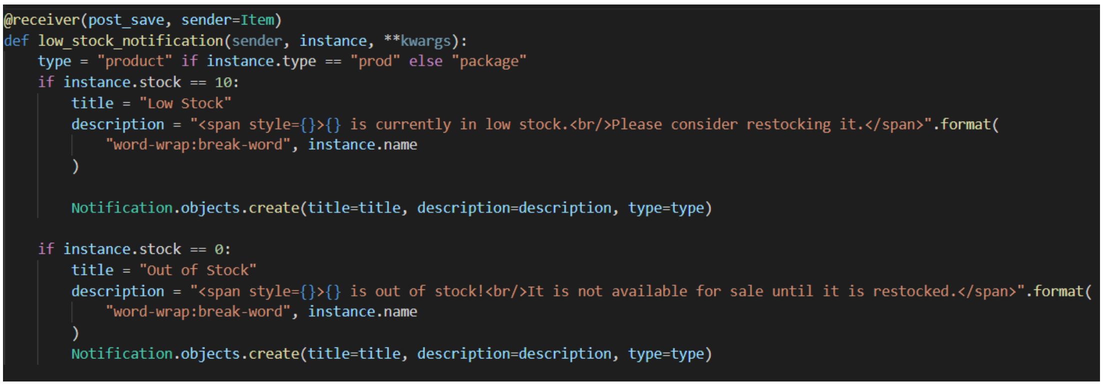
Total 26 items < 1 2 3 4 5 6 > 5 / page Go to Page

© 2022 Sharifah Food. All rights reserved.

Figure 6.83: ABC Analysis

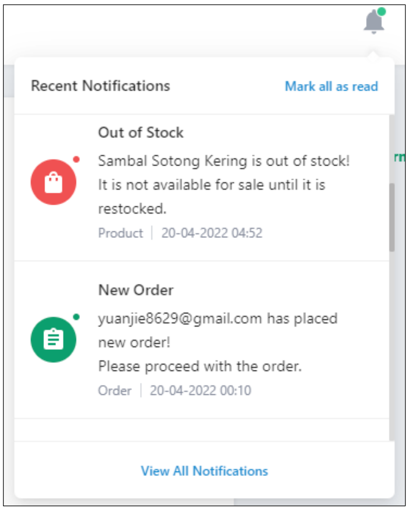
6.6.12 Notification

A notification will be generated when there is a new order, new agent or drop shipper registration and low stock items. The project uses the signal dispatchers provided by Django to automatically add the notification. Figure 6.84 shows the code snippet for adding low stock notification for the low-stock items, while Figure 6.85 shows the notification list that can be viewed in the inventory management system.



```
@receiver(post_save, sender=Item)
def low_stock_notification(sender, instance, **kwargs):
    type = "product" if instance.type == "prod" else "package"
    if instance.stock == 10:
        title = "Low Stock"
        description = "<span style={}>{} is currently in low stock.<br/>Please consider restocking it.</span>".format(
            "word-wrap:break-word", instance.name
        )

        Notification.objects.create(title=title, description=description, type=type)

    if instance.stock == 0:
        title = "Out of Stock"
        description = "<span style={}>{} is out of stock!<br/>It is not available for sale until it is restocked.</span>".format(
            "word-wrap:break-word", instance.name
        )
        Notification.objects.create(title=title, description=description, type=type)
```

Figure 6.84: Code Snippet for Low Stock Notification



Recent Notifications

Mark all as read

Out of Stock

Sambal Sotong Kering is out of stock! It is not available for sale until it is restocked.

Product | 20-04-2022 04:52

New Order

yuanjie8629@gmail.com has placed new order! Please proceed with the order.

Order | 20-04-2022 00:10

View All Notifications

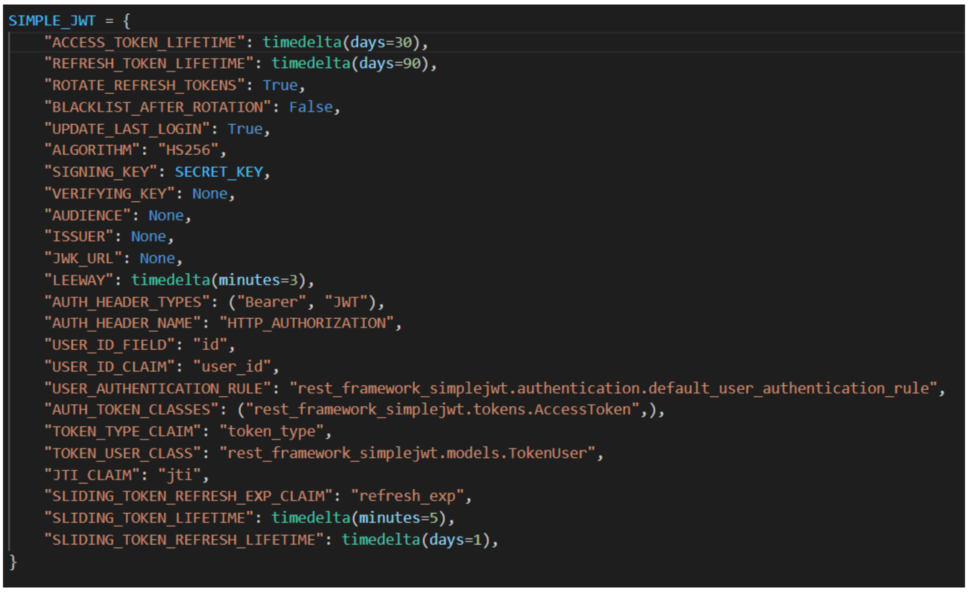
Figure 6.85: Notification List

6.7 Implementation of E-commerce Platform

This section discusses the features implemented in the e-commerce platform.

6.7.1 Authentication

The e-commerce platform's authentication is similar to the inventory management system discussed. However, since the e-commerce platform does not have strict control over security, there is no session extension modal to alert the user on the session expiring because the lifetime of the tokens in the e-commerce platform is set to be longer. Figure 6.86 shows the JWT settings on the e-commerce platform.

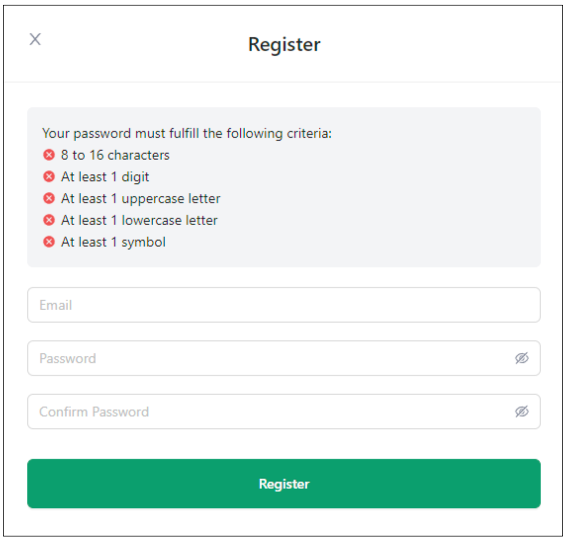


```
SIMPLE_JWT = {
    "ACCESS_TOKEN_LIFETIME": timedelta(days=30),
    "REFRESH_TOKEN_LIFETIME": timedelta(days=90),
    "ROTATE_REFRESH_TOKENS": True,
    "BLACKLIST_AFTER_ROTATION": False,
    "UPDATE_LAST_LOGIN": True,
    "ALGORITHM": "HS256",
    "SIGNING_KEY": SECRET_KEY,
    "VERIFYING_KEY": None,
    "AUDIENCE": None,
    "ISSUER": None,
    "JWK_URL": None,
    "LEEWAY": timedelta(minutes=3),
    "AUTH_HEADER_TYPES": ("Bearer", "JWT"),
    "AUTH_HEADER_NAME": "HTTP_AUTHORIZATION",
    "USER_ID_FIELD": "id",
    "USER_ID_CLAIM": "user_id",
    "USER_AUTHENTICATION_RULE": "rest_framework_simplejwt.authentication.default_user_authentication_rule",
    "AUTH_TOKEN_CLASSES": ("rest_framework_simplejwt.tokens.AccessToken",),
    "TOKEN_TYPE_CLAIM": "token_type",
    "TOKEN_USER_CLASS": "rest_framework_simplejwt.models.TokenUser",
    "JTI_CLAIM": "jti",
    "SLIDING_TOKEN_REFRESH_EXP_CLAIM": "refresh_exp",
    "SLIDING_TOKEN_LIFETIME": timedelta(minutes=5),
    "SLIDING_TOKEN_REFRESH_LIFETIME": timedelta(days=1),
}
```

Figure 6.86: JWT Authentication Settings of E-commerce Platform

As shown in Figure 6.86, the access token lifetime is set to 30 days, and the refresh token lifetime is set to 90 days. Therefore, the logged-in user can continue to stay logged in on the e-commerce platform for 90 days, provided that the cookies containing the refresh token are kept in the browser.

Besides, an additional authentication feature on an e-commerce platform is registering an account as a customer. The user can register as a customer by entering the email address and password in the registration drawer, as shown in Figure 6.87. Figure 6.88 and Figure 6.89 show the login and forgot password drawers on the e-commerce platform.



Your password must fulfill the following criteria:
- ✖ 8 to 16 characters
- ✖ At least 1 digit
- ✖ At least 1 uppercase letter
- ✖ At least 1 lowercase letter
- ✖ At least 1 symbol

Figure 6.87: Registration Drawer

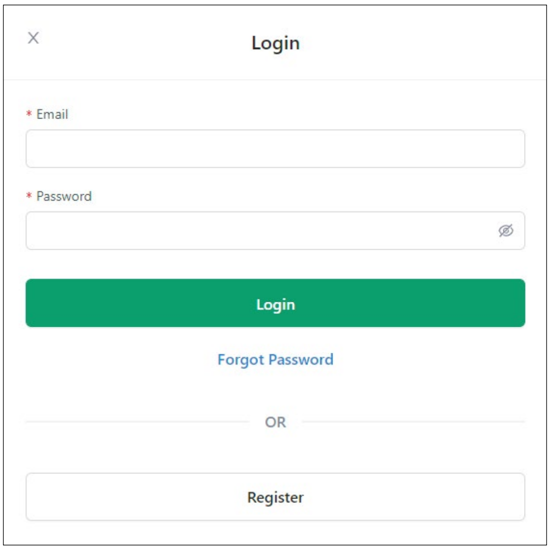
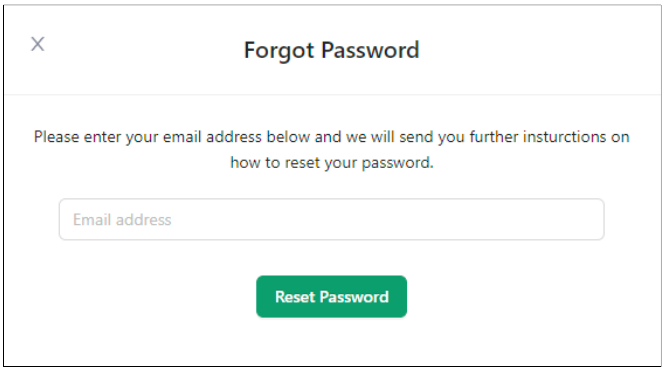


Figure 6.88: Login Drawer



Forgot Password

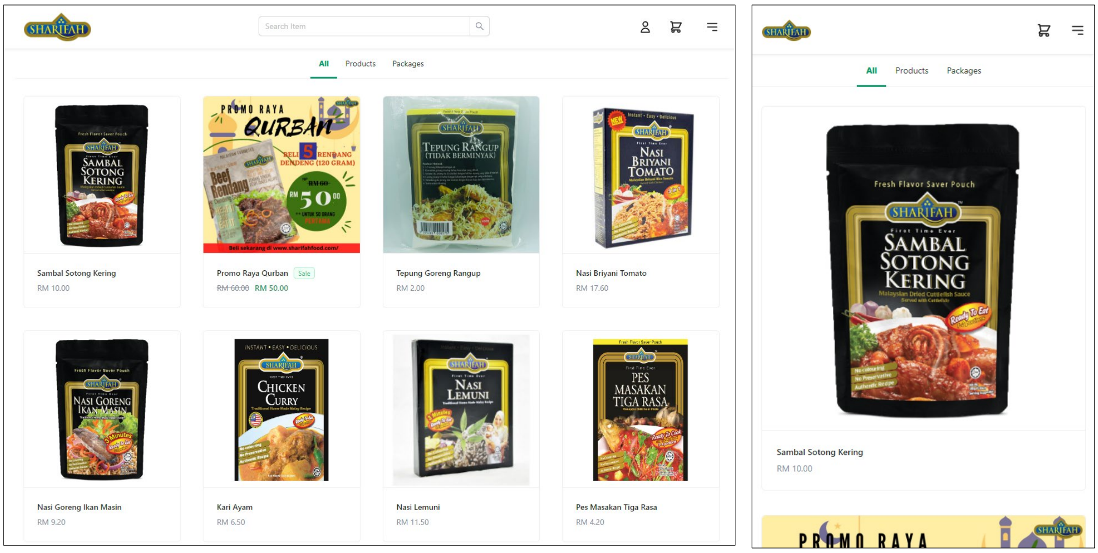
Please enter your email address below and we will send you further insturctions on how to reset your password.

Email address

Figure 6.89: Forgot Password Drawer

6.7.2 Browse Items (Products and Packages)

The users can browse a list of items available. Besides, the users can also filter the items based on item types, such as products and packages. All the available products will be listed in the products tab, and the users can filter the products based on the product category. On the other hand, all the available promotional packages will be listed in the packages tab. Additionally, the browse item list also applies pagination to avoid a lengthy item listing, thus providing better user experiences. Figure 6.90, Figure 6.91, and Figure 6.92 show the UI of browsing items, products, and packages in desktop and mobile views.



Sambal Sotong Kering
RM 10.00

Promo Raya Qurban Sale
~~RM 60.00~~ RM 50.00

Tepung Goreng Rangup
RM 2.00

Nasi Briyani Tomato
RM 17.60

Nasi Goreng Ikan Masin
RM 9.20

Kari Ayam
RM 6.50

Nasi Lemuni
RM 11.50

Pes Masakan Tiga Rasa
RM 4.20

Sambal Sotong Kering
RM 10.00

Figure 6.90: Browse Items

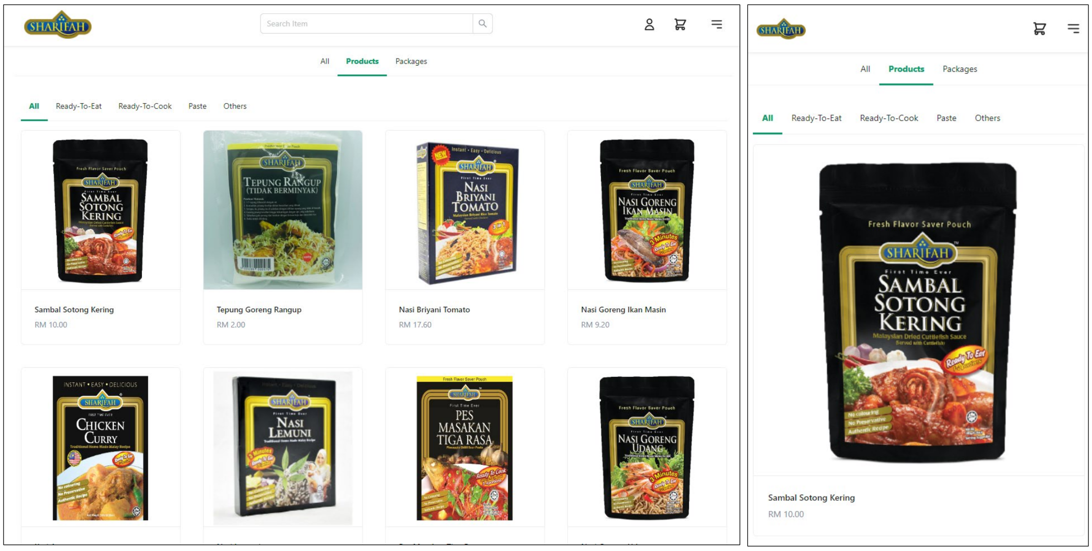


Figure 6.91: Browse Products

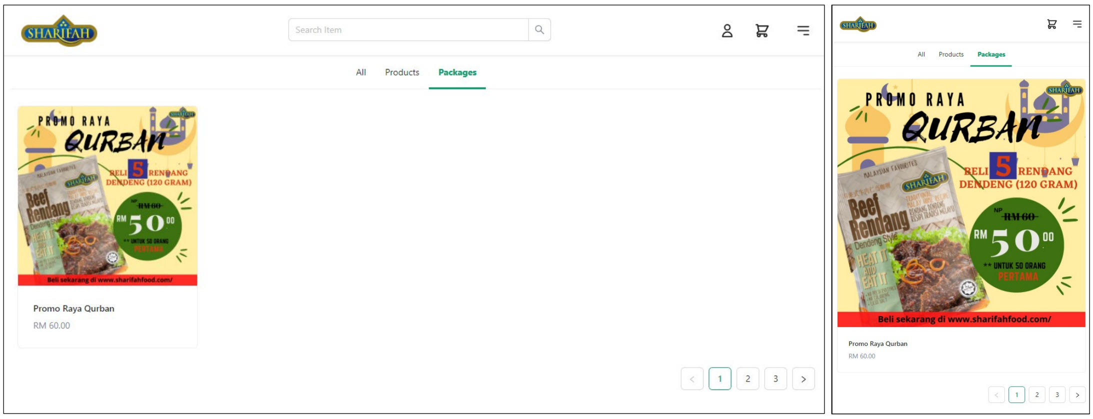


Figure 6.92: Browse Packages

6.7.3 Search Item

The e-commerce platform also provides the feature to search the items by item name. The user can click on the search items input to enter the keyword to be searched. If there is no item matched with the keyword, a “Not Found” message will be displayed in Figure 6.95. Figure 6.93 shows the search input component to enter the search keyword, while Figure 6.94 shows the sample of search results when the user searches “nasi goreng”.

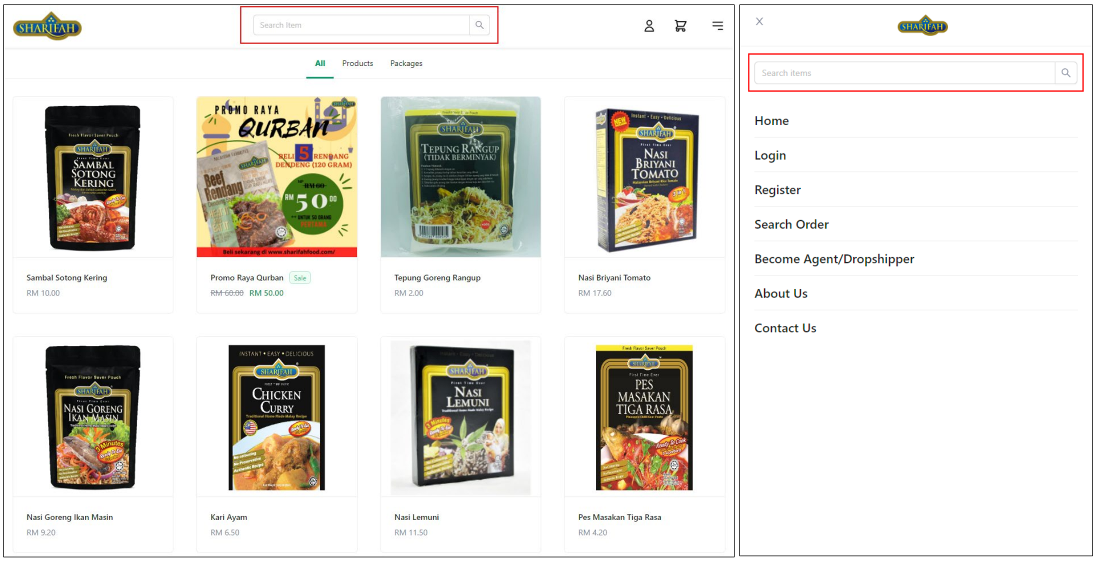


Figure 6.93: Search Item Input Component

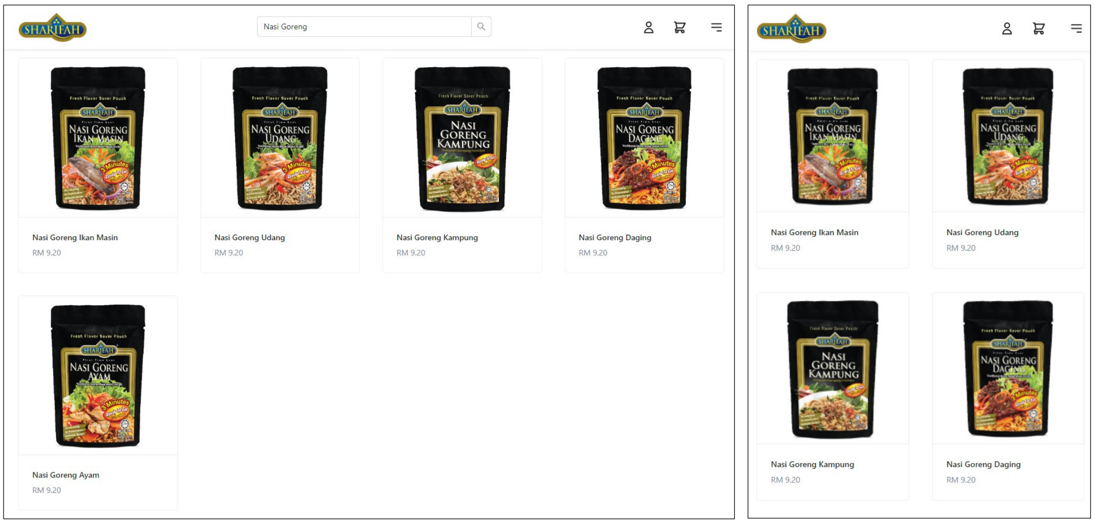


Figure 6.94: Search Item Results

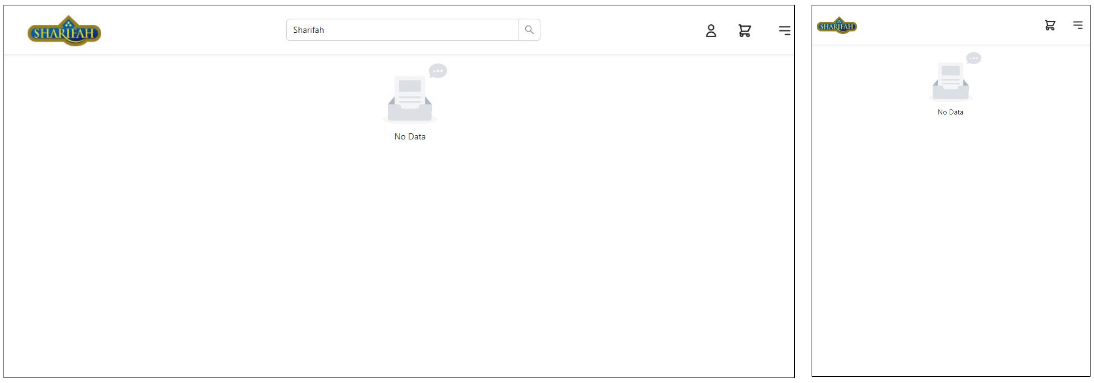
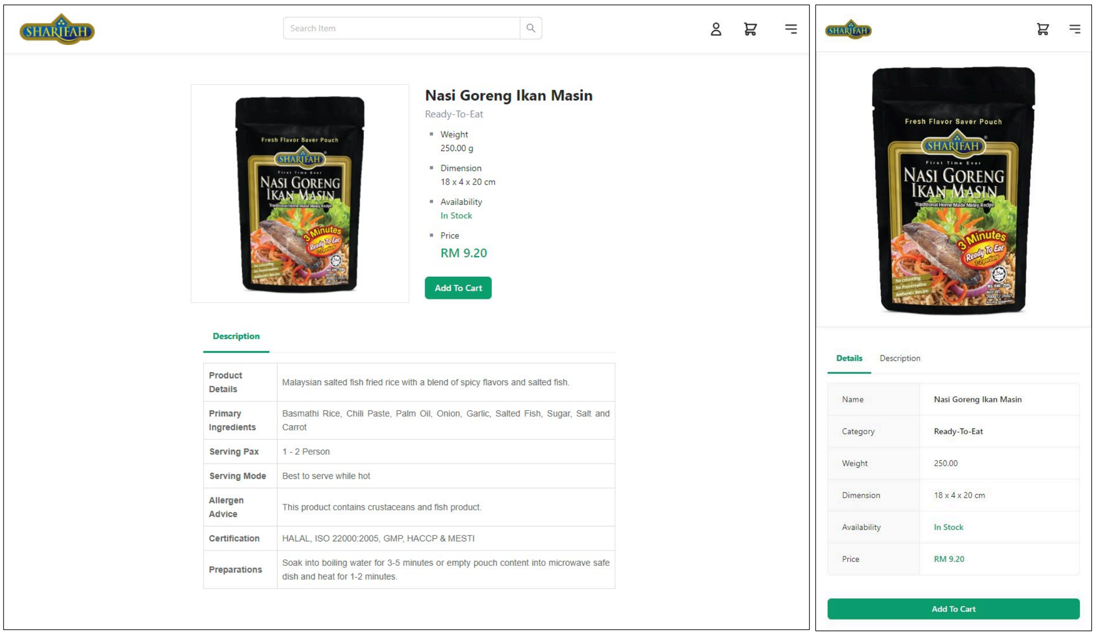


Figure 6.95: Serch Item - Not Found

6.7.4 View Item

When the user browses the list of items, the user can click on a particular item to view the item's details. Once the user clicks the item, the system will redirect the user to the item details page showing the item details, as shown in Figure 6.96. Additionally, the user can add the item to the cart by clicking the ‘Add To Cart’ button. Then, the system will check the item stock and add the item to the user’s cart. Once the item is added to the user’s cart, a successful message will be displayed, as shown in Figure 6.97. Nonetheless, if the item has already reached the maximum stock in the user’s cart, an error message will be displayed, as shown in Figure 6.98. Moreover, if the item is out of stock, the ‘Add To Cart’ button will be disabled, as demonstrated in Figure 6.99.



Nasi Goreng Ikan Masin

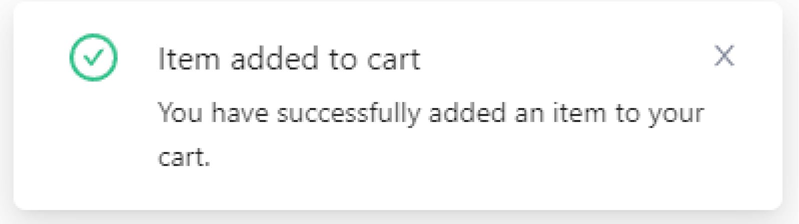
Ready-To-Eat

- Weight
  250.00 g
- Dimension
  18 x 4 x 20 cm
- Availability
  In Stock
- Price
  RM 9.20

Product Details	Malaysian salted fish fried rice with a blend of spicy flavors and salted fish.
Primary Ingredients	Basmathi Rice, Chili Paste, Palm Oil, Onion, Garlic, Salted Fish, Sugar, Salt and Carrot
Serving Pax	1 - 2 Person
Serving Mode	Best to serve while hot
Allergen Advice	This product contains crustaceans and fish product.
Certification	HALAL, ISO 22000:2005, GMP, HACCP & MESTI
Preparations	Soak into boiling water for 3-5 minutes or empty pouch content into microwave safe dish and heat for 1-2 minutes.

Name	Nasi Goreng Ikan Masin
Category	Ready-To-Eat
Weight	250.00
Dimension	18 x 4 x 20 cm
Availability	In Stock
Price	RM 9.20

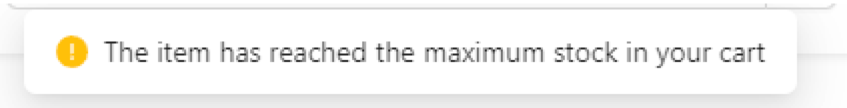
Figure 6.96: Item Details



Item added to cart

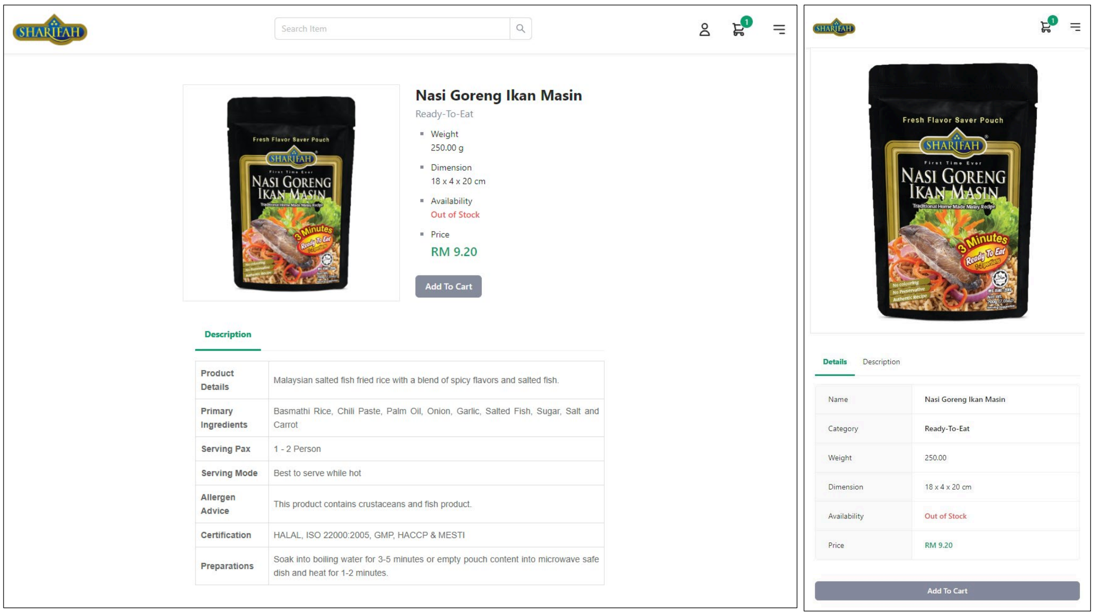
You have successfully added an item to your cart.

Figure 6.97: Add to Cart Success Message



The item has reached the maximum stock in your cart

Figure 6.98: Add to Cart Error Message



Nasi Goreng Ikan Masin

Ready-To-Eat

- Weight
  250.00 g
- Dimension
  18 x 4 x 20 cm
- Availability
  Out of Stock
- Price
  RM 9.20

Add To Cart

Description

Product Details	Malaysian salted fish fried rice with a blend of spicy flavors and salted fish.
Primary Ingredients	Basmathi Rice, Chili Paste, Palm Oil, Onion, Garlic, Salted Fish, Sugar, Salt and Carrot
Serving Pax	1 - 2 Person
Serving Mode	Best to serve while hot
Allergen Advice	This product contains crustaceans and fish product.
Certification	HALAL, ISO 22000:2005, GMP, HACCP & MESTI
Preparations	Soak into boiling water for 3-5 minutes or empty pouch content into microwave safe dish and heat for 1-2 minutes.

Details Description

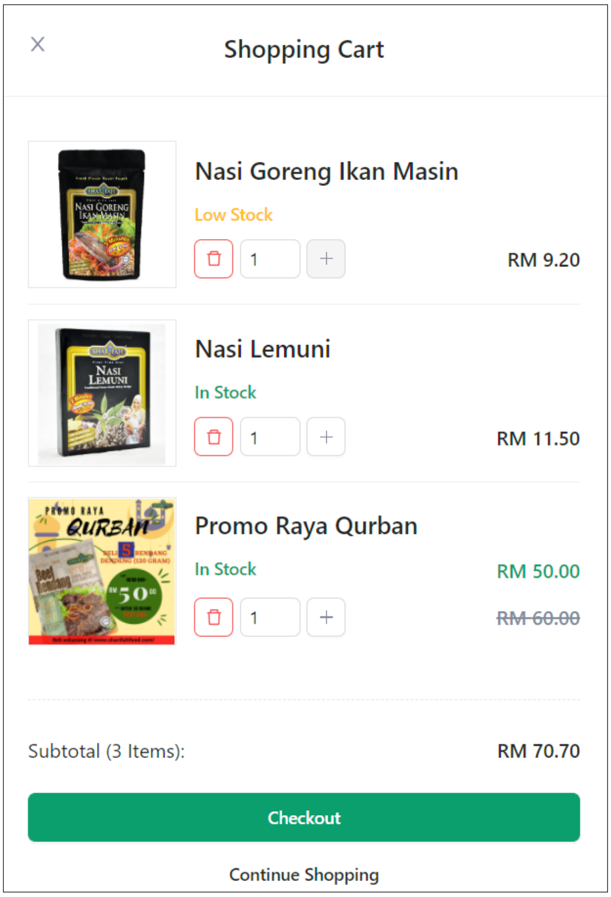
Name	Nasi Goreng Ikan Masin
Category	Ready-To-Eat
Weight	250.00
Dimension	18 x 4 x 20 cm
Availability	Out of Stock
Price	RM 9.20

Add To Cart

Figure 6.99: Out of Stock Item Details

6.7.5 View Shopping Cart

The users can view their shopping cart on the e-commerce platform anytime by clicking the shopping cart icon. Once the user clicks the shopping cart icon, the shopping cart drawer will be popped up. If there are items in the shopping cart, a list of all items and the subtotal price will be displayed, as in Figure 6.100. Alternately, a message of “There are no items in your cart” will be displayed if the shopping cart is empty, as shown in Figure 6.101.



Shopping Cart

Nasi Goreng Ikan Masin

Low Stock

1 RM 9.20

Nasi Lemuni

In Stock

1 RM 11.50

Promo Raya Qurban

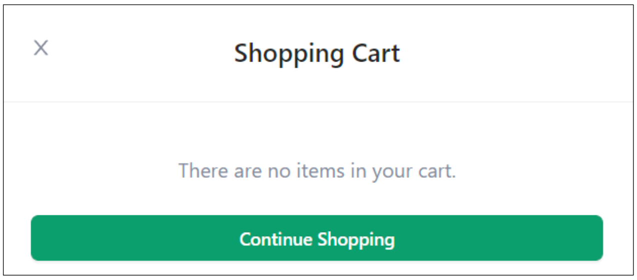
In Stock

RM 50.00

1 ~~RM 60.00~~

Subtotal (3 Items): RM 70.70

Figure 6.100: Shopping Cart



Shopping Cart

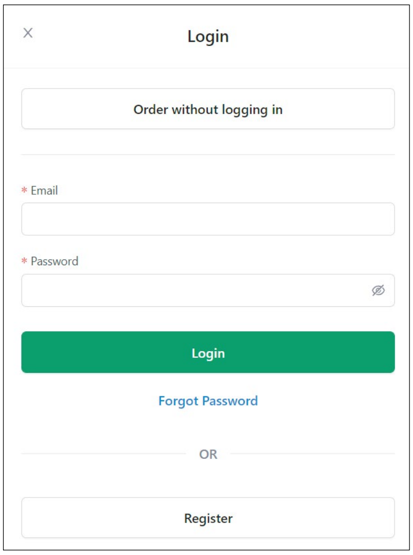
There are no items in your cart.

Figure 6.101: Shopping Cart Without Items

6.7.6 Checkout

The users can place orders by checking out their shopping carts. If the users want to check out their shopping cart, they can click the ‘Checkout’ button shown on the Shopping Cart Drawer. If the user does not log in, the system will display a login drawer to prompt the user to log in or continue without login, as shown in Figure 6.102. After that, the system will redirect the user to the checkout page. The checkout page can be divided into four components: Shipment Information, Voucher, Order

Summary, and Payment method. Figure 6.103 shows the overview of the checkout page in the e-commerce platform.



Login

Order without logging in

* Email

* Password

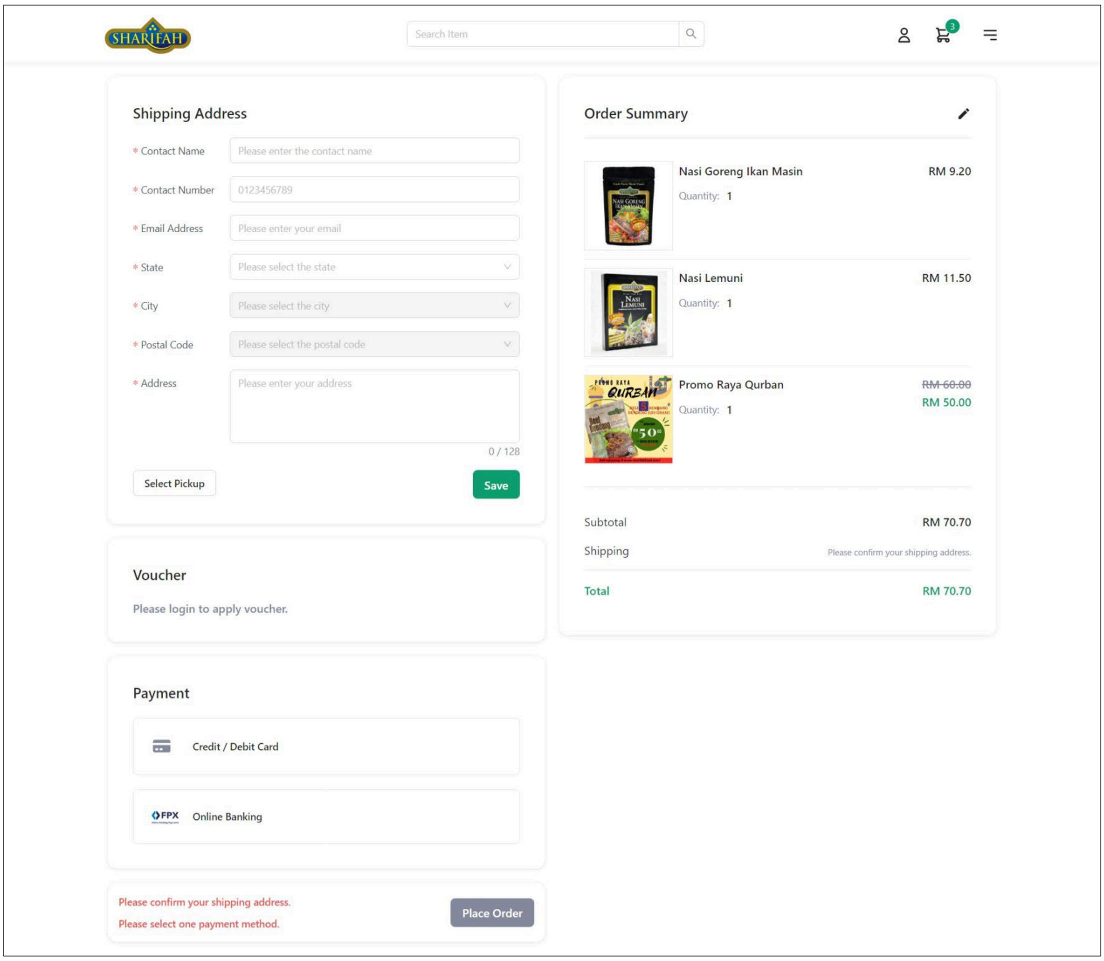
Login

Forgot Password

OR

Register

Figure 6.102: Login Drawer for Checkout



Shipping Address

* Contact Name: Please enter the contact name

* Contact Number: 0123456789

* Email Address: Please enter your email

* State: Please select the state

* City: Please select the city

* Postal Code: Please select the postal code

* Address: Please enter your address

0 / 128

Select Pickup

Save

Voucher

Please login to apply voucher.

Payment

Credit / Debit Card

Online Banking

Please confirm your shipping address.

Please select one payment method.

Place Order

Order Summary

Nasi Goreng Ikan Masin — RM 9.20 — Quantity: 1

Nasi Lemuni — RM 11.50 — Quantity: 1

Promo Raya Qurban — ~~RM 60.00~~ RM 50.00 — Quantity: 1

Subtotal: RM 70.70

Shipping: Please confirm your shipping address.

Total: RM 70.70

Figure 6.103: Checkout Page

6.7.6.1 Shipment Information

The user must fill up the shipment information to checkout an order. In addition, the user can choose the shipment method, which is by shipping or pickup. If the users want to ship their orders by shipping, they are required to fill up the shipping address information, as shown in Figure 6.104. On the other hand, the users can click the ‘Select Pickup’ button and fill up the pickup form, as shown in Figure 6.105, to pick up their orders.

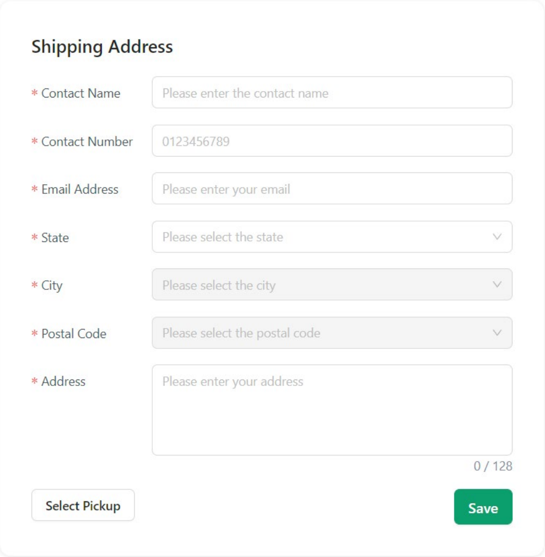


Figure 6.104: Shipping Address Form

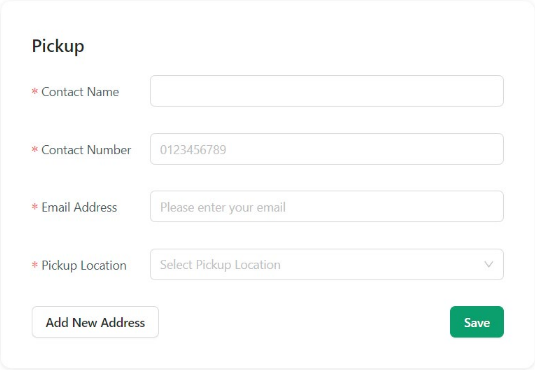
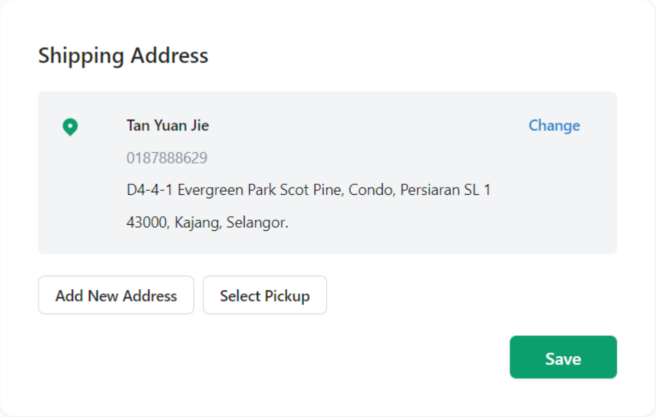


Figure 6.105: Pickup Form

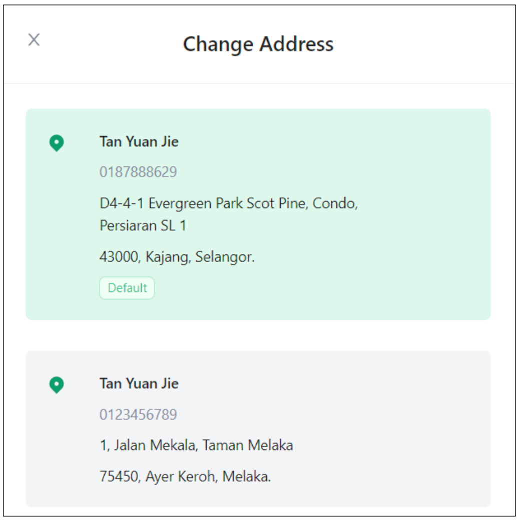
For the logged-in users, the system will automatically display their default address, as shown in Figure 6.106. Suppose the logged-in user wishes to choose pickup or another address that is not added in the address book. In that case, the user can click the ‘Add New Address’ or ‘Select Pickup’ buttons, respectively, so that the system will display the respective form for the user to fill up. Besides, if the user wants to select another address from the address book, the user can click the ‘Change’ text button, and the system will display a list of the user’s address in a drawer, as shown in the Figure 6.107. Once the user confirms the shipment information, the user must click the ‘Save’ button to confirm the shipment information.



Shipping Address

Tan Yuan Jie
0187888629
D4-4-1 Evergreen Park Scot Pine, Condo, Persiaran SL 1
43000, Kajang, Selangor.

Figure 6.106: Default Shipping Address



Change Address

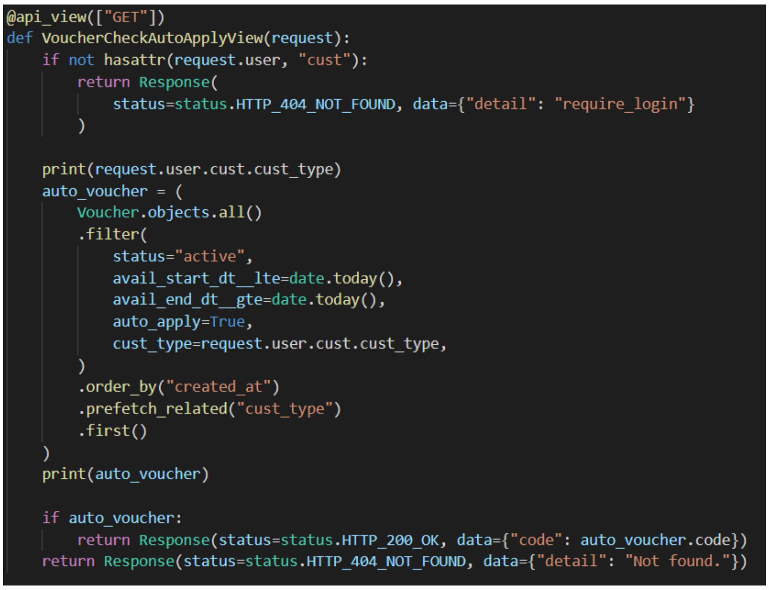
Tan Yuan Jie
0187888629
D4-4-1 Evergreen Park Scot Pine, Condo,
Persiaran SL 1
43000, Kajang, Selangor.
Default

Tan Yuan Jie
0123456789
1, Jalan Mekala, Taman Melaka
75450, Ayer Keroh, Melaka.

Figure 6.107: Change Address

6.7.6.2 Voucher

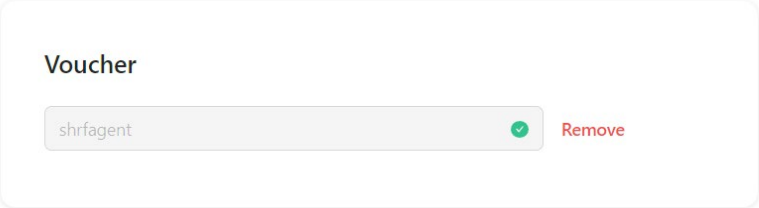
On the checkout page, the system will check if there is an auto-applied voucher applicable for the user's order and automatically adds the voucher if applicable. However, the user must be logged in to apply for the voucher. Figure 6.108 shows the code snippet to check the auto-applied voucher and return the auto-applied voucher if the auto-applied voucher is valid for the user.



```
@api_view(["GET"])
def VoucherCheckAutoApplyView(request):
    if not hasattr(request.user, "cust"):
        return Response(
            status=status.HTTP_404_NOT_FOUND, data={"detail": "require_login"}
        )

    print(request.user.cust.cust_type)
    auto_voucher = (
        Voucher.objects.all()
        .filter(
            status="active",
            avail_start_dt__lte=date.today(),
            avail_end_dt__gte=date.today(),
            auto_apply=True,
            cust_type=request.user.cust.cust_type,
        )
        .order_by("created_at")
        .prefetch_related("cust_type")
        .first()
    )
    print(auto_voucher)

    if auto_voucher:
        return Response(status=status.HTTP_200_OK, data={"code": auto_voucher.code})
    return Response(status=status.HTTP_404_NOT_FOUND, data={"detail": "Not found."})
```

Figure 6.108: Code Snippet of Checking Auto-applied Voucher

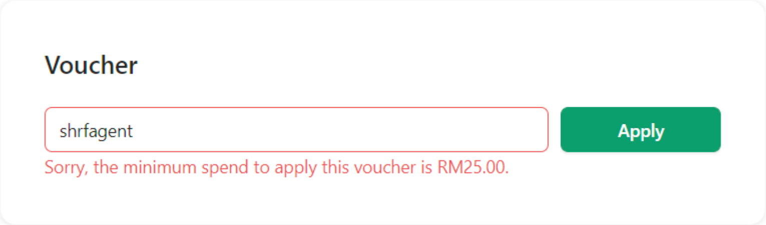
If the logged-in user successfully applies a voucher, the input will be disabled, and a 'Remove' button will be displayed, as demonstrated in Figure 6.109. On the other hand, an error message will be displayed, as shown in Figure 6.110, if the voucher entered is invalid or ineligible for the user.



Voucher

shrfagent

Figure 6.109: Voucher Applied



Voucher

shrfagent

Sorry, the minimum spend to apply this voucher is RM25.00.

Figure 6.110: Voucher Invalid or Ineligible

6.7.6.3 Order Summary

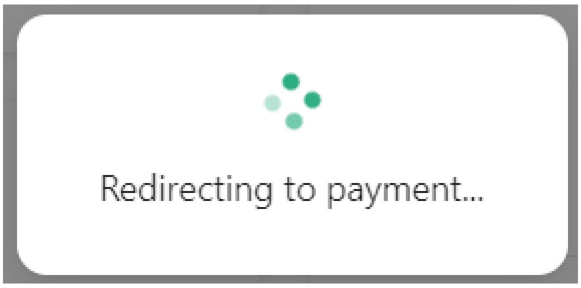
The order summary shows a list of all items to be ordered by the user. Besides, it also shows the subtotal price, shipping fee, discount, and total price to the user.

6.7.6.4 Payment Method

There are two payment methods available: Credit / Debit Card and FPX Online banking. The user must select one method to pay for the order.

6.7.7 Payment

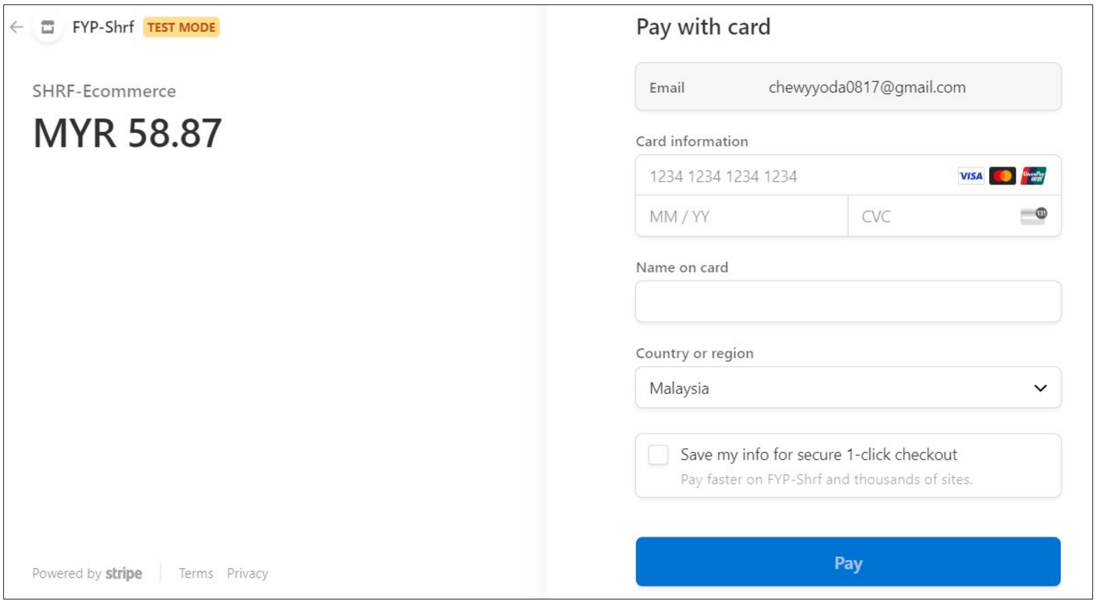
After the user fills up the shipment information and selects a payment method, the user can click 'Place Order' to make a payment. When the user places an order, the system will create a new order with unpaid status if all the items in the order are in stock. After that, the system will communicate with the third-party payment gateway, Stripe, and display a redirect message, as shown in Figure 6.111.



Redirecting to payment...

Figure 6.111: Redirecting to Payment Message

Once the communication is success, the user will be redirected to the payment gateway and fill in the payment information. Figure 6.112 shows the credit or debit card payment form after the user is redirected to the payment gateway.



FYP-Shrf TEST MODE

SHRF-Ecommerce

MYR 58.87

Powered by stripe | Terms Privacy

Pay with card

Email chewyyoda0817@gmail.com

Card information

1234 1234 1234 1234

MM / YY

CVC

Name on card

Country or region

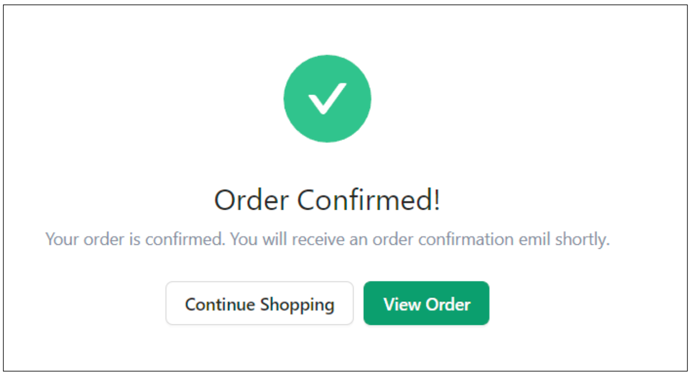
Malaysia

□ Save my info for secure 1-click checkout

Pay faster on FYP-Shrf and thousands of sites.

Figure 6.112: Credit Card Payment

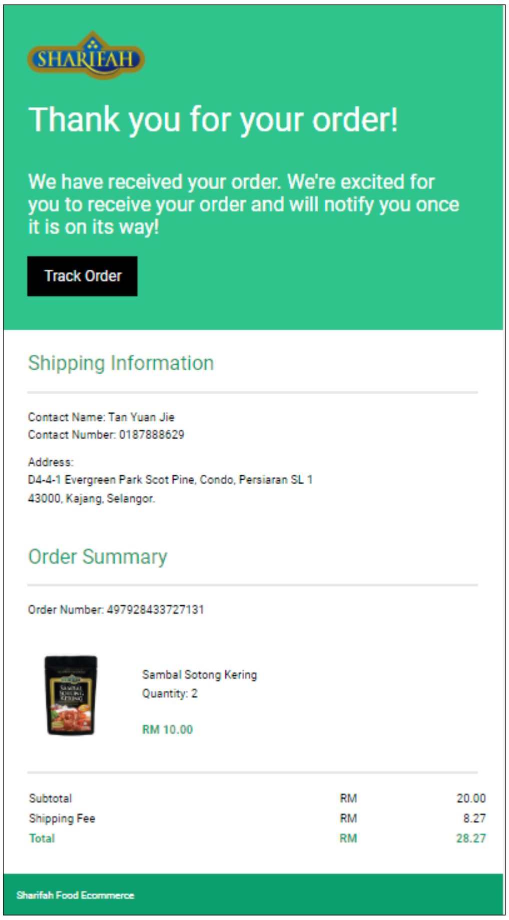
After the user successfully makes a payment, the system will inform the user that the order is confirmed, as shown in Figure 6.113. Besides, the user will also receive an order confirmation email, as shown in Figure 6.114.



Order Confirmed!

Your order is confirmed. You will receive an order confirmation emil shortly.

Figure 6.113: Order Confirmed



Thank you for your order!

We have received your order. We're excited for you to receive your order and will notify you once it is on its way!

Shipping Information

Contact Name: Tan Yuan Jie
Contact Number: 0187888629

Address:
D4-4-1 Evergreen Park Scot Pine, Condo, Persiaran SL 1
43000, Kajang, Selangor.

Order Summary

Order Number: 497928433727131

Sambal Sotong Kering
Quantity: 2

RM 10.00

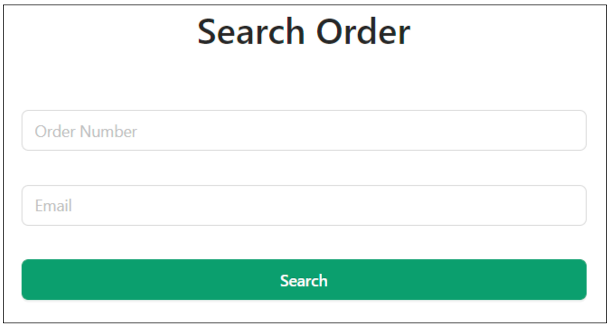
Subtotal	RM	20.00
Shipping Fee	RM	8.27
Total	RM	28.27

Sharifah Food Ecommerce

Figure 6.114: Order Confirmation Email

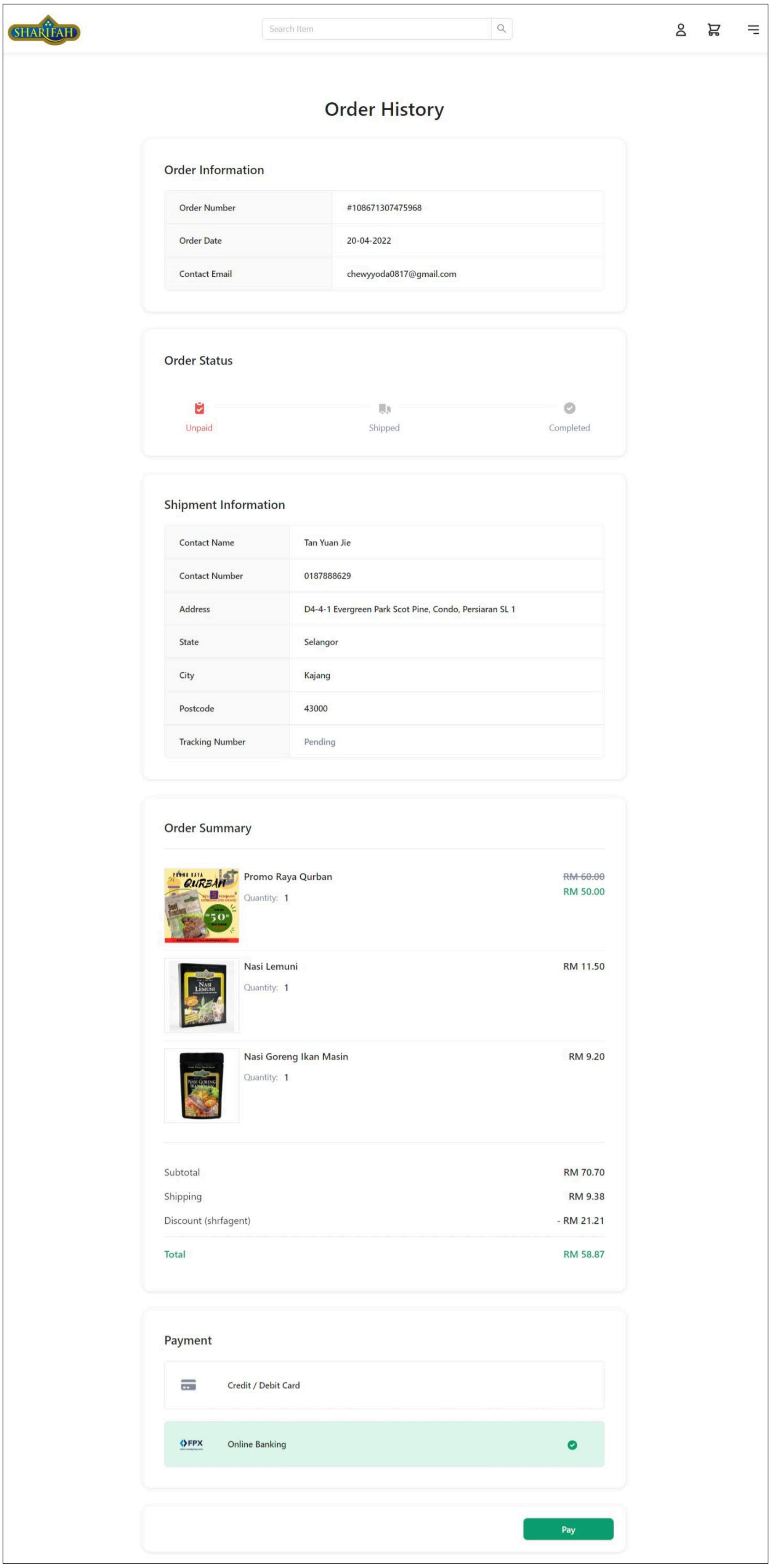
6.7.8 Search Order

The search order is an exclusive feature for unregistered users to track their orders. The user must enter the correct email and order number to view their placed order. Figure 6.115 shows the search order form. If the order number and email are valid, the user will be redirected to the order details page. Additionally, the users can choose to pay the order if they have not paid the order, as shown in Figure 6.116.



Search Order

Figure 6.115: Search Order



Order History

Order Information

Order Number	#108671307475968
Order Date	20-04-2022
Contact Email	chewyyoda0817@gmail.com

Order Status

Shipment Information

Contact Name	Tan Yuan Jie
Contact Number	0187888629
Address	D4-4-1 Evergreen Park Scot Pine, Condo, Persiaran SL 1
State	Selangor
City	Kajang
Postcode	43000
Tracking Number	Pending

Order Summary

Promo Raya Qurban Quantity: 1	~~RM 60.00~~ RM 50.00
Nasi Lemuni Quantity: 1	RM 11.50
Nasi Goreng Ikan Masin Quantity: 1	RM 9.20

Subtotal	RM 70.70
Shipping	RM 9.38
Discount (shrfagent)	- RM 21.21
Total	RM 58.87

Payment

○ Credit / Debit Card

● FPX Online Banking ✓

Figure 6.116: Order Details Page

6.7.9 Register as Agent/Dropshipper

The user can register as an agent or dropshipper by clicking the ‘Become Agent/Dropshipper’ menu on the menu drawer, as shown in Figure 6.117. After that, the user is required to fill out the registration form, as shown in Figure 6.118.

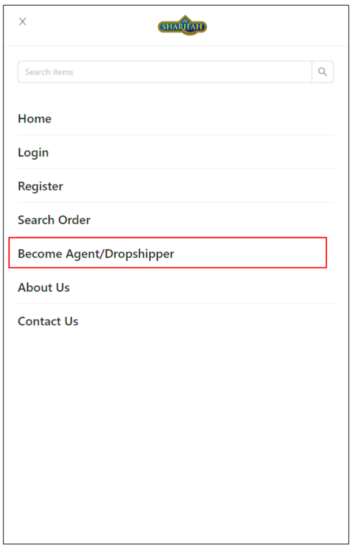
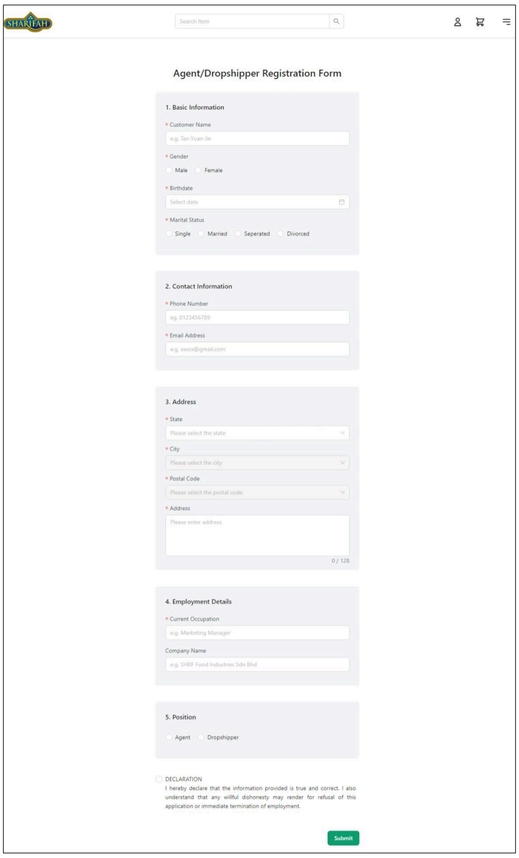


Figure 6.117: Become Agent/Dropshipper



Agent/Dropshipper Registration Form

1. Basic Information

* Customer Name
e.g. Tan Yuan Jie

* Gender
○ Male ○ Female

* Birthdate
Select date

* Marital Status
○ Single ○ Married ○ Seperated ○ Divorced

2. Contact Information

* Phone Number
eg. 0123456789

* Email Address
e.g. xxxxx@gmail.com

3. Address

* State
Please select the state

* City
Please select the city

* Postal Code
Please select the postal code

* Address
Please enter address
0 / 128

4. Employment Details

* Current Occupation
e.g. Marketing Manager

Company Name
e.g. SHRF Food Industries Sdn Bhd

5. Position

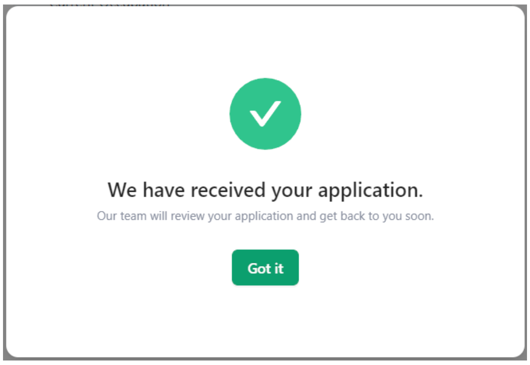
○ Agent ○ Dropshipper

□ DECLARATION
I hereby declare that the information provided is true and correct. I also understand that any willful dishonesty may render for refusal of this application or immediate termination of employment.

Submit

Figure 6.118: Agent/Dropshipper Registration Form

After the user has filled out the form, the user will receive a notification modal prompting to wait for the admin's review, as shown in Figure 6.119.



We have received your application.

Our team will review your application and get back to you soon.

Figure 6.119: Agent/Dropshipper Registration Notification Modal

6.7.10 Profile Management

The registered users can manage their profiles by clicking the user icon. After that, the system will display a drawer, as shown in Figure 6.120. The profile management can be divided into three parts: Order History, Address Book, and Account Information.

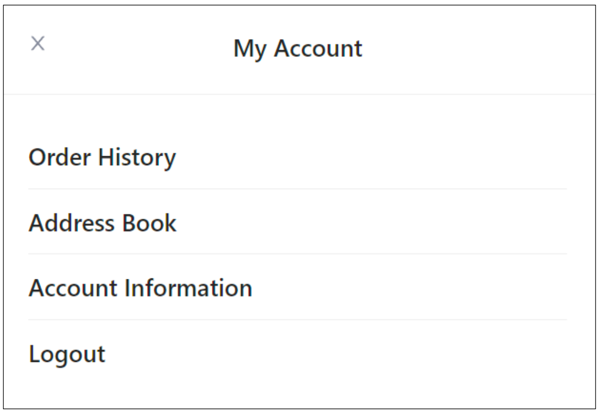
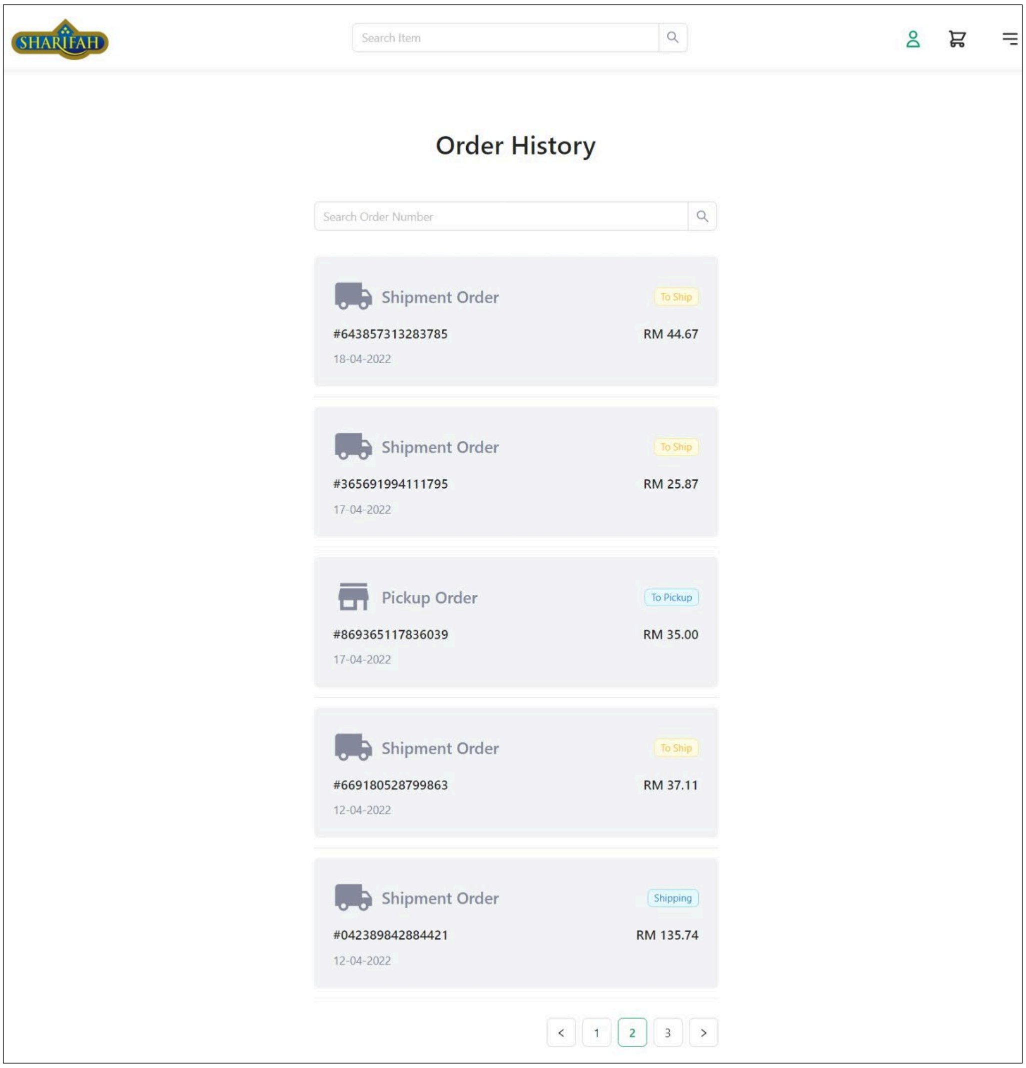


Figure 6.120: Profile Management Drawer

6.7.10.1 Order History

The registered users can view a list of all their placed orders, as shown in Figure 6.121. The user can also search for the particular orders by order number. If the registered user clicks on a particular order, the system will redirect the user to the order details, as shown in Figure 6.116.



Order History

Shipment Order To Ship
#643857313283785 RM 44.67
18-04-2022

Shipment Order To Ship
#365691994111795 RM 25.87
17-04-2022

Pickup Order To Pickup
#869365117836039 RM 35.00
17-04-2022

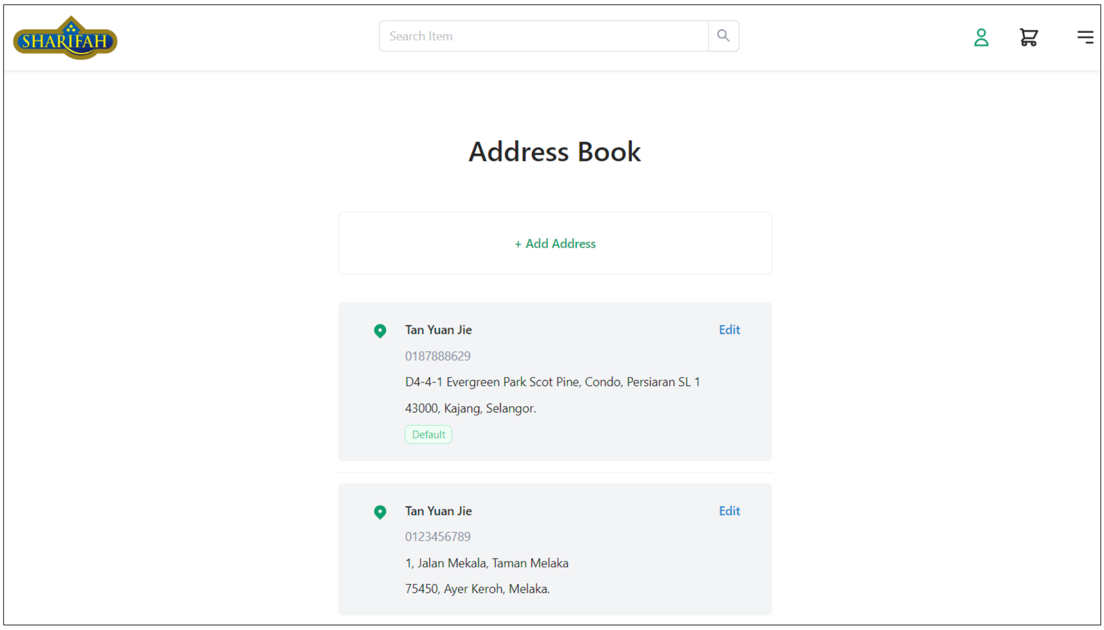
Shipment Order To Ship
#669180528799863 RM 37.11
12-04-2022

Shipment Order Shipping
#042389842884421 RM 135.74
12-04-2022

Figure 6.121: Order History

6.7.10.2 Address Book

The registered users can store their addresses in their address book, so they do not need to re-enter their address every time during checkout. On the address book page, the registered users can view a list of their addresses, as shown in Figure 6.122. Moreover, the registered users can also add a new address or edit an existing address. To add or edit an address, the registered users click on the ‘Add Address’ card button or the ‘Edit’ text button on the address card. Then, the system will display an address form in modal for desktop view or drawer for mobile view. Figure 6.123 shows the modal and drawer address forms for adding address.



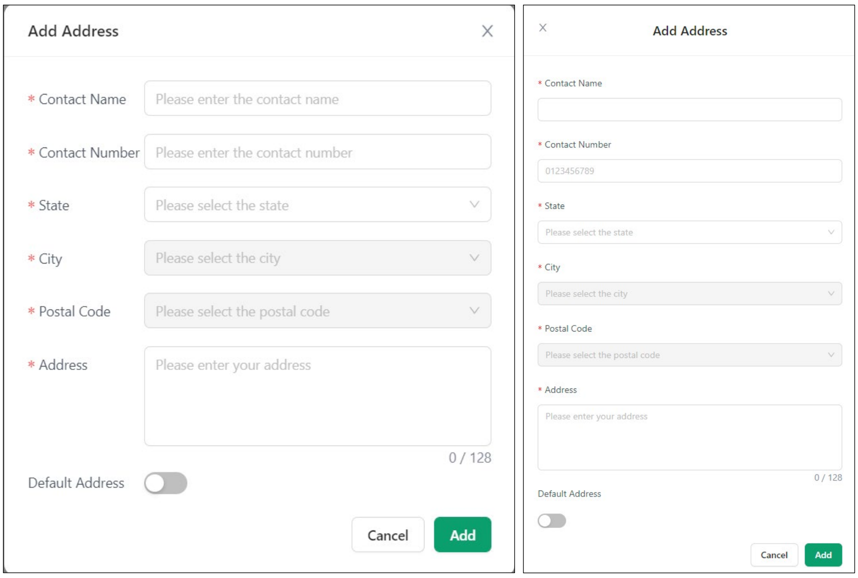
Address Book

+ Add Address

Tan Yuan Jie Edit
0187888629
D4-4-1 Evergreen Park Scot Pine, Condo, Persiaran SL 1
43000, Kajang, Selangor.
Default

Tan Yuan Jie Edit
0123456789
1, Jalan Mekala, Taman Melaka
75450, Ayer Keroh, Melaka.

Figure 6.122: Address Book


Add Address

* Contact Name: Please enter the contact name
* Contact Number: Please enter the contact number
* State: Please select the state
* City: Please select the city
* Postal Code: Please select the postal code
* Address: Please enter your address
0 / 128
Default Address

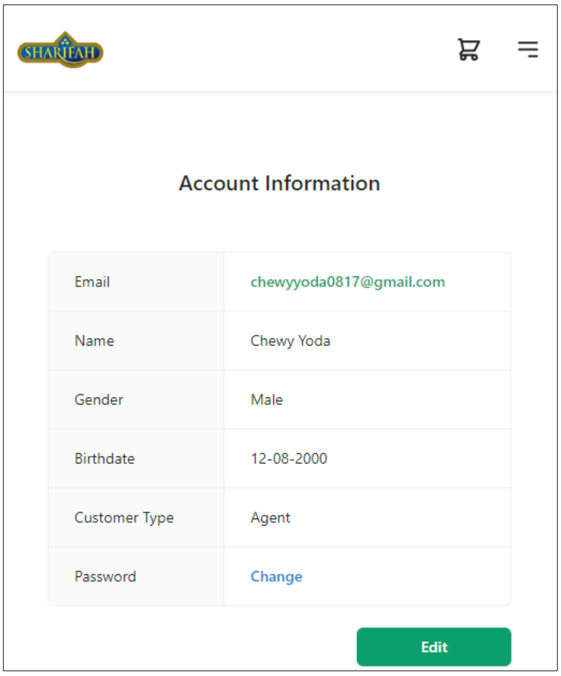
Add Address

* Contact Name
* Contact Number: 0123456789
* State: Please select the state
* City: Please select the city
* Postal Code: Please select the postal code
* Address: Please enter your address
0 / 128
Default Address

Figure 6.123: Add Address Modal and Drawer

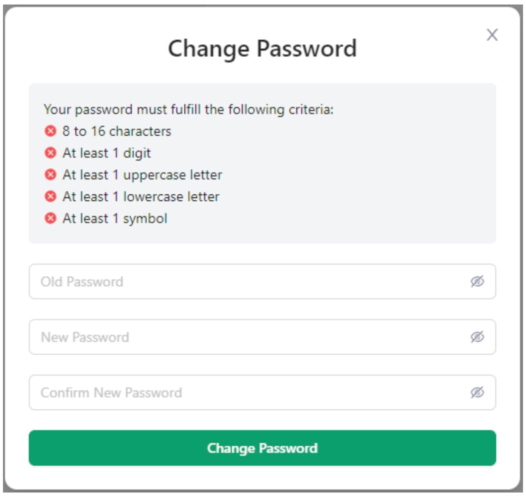
6.7.10.3 Account Information

The registered users can also manage their account information. When the registered users click the ‘Account Information’ on the profile management drawer, the system will redirect the users to the account information page, displaying all the user's information, as shown in Figure 6.124. The user can change the password by clicking the ‘Change’ text button on the password row, and a change password modal will be displayed, as demonstrated in Figure 6.125. If the user wants to edit account information, the user can click the edit button and the edit modal form will be displayed to the user, as shown in Figure 6.126.



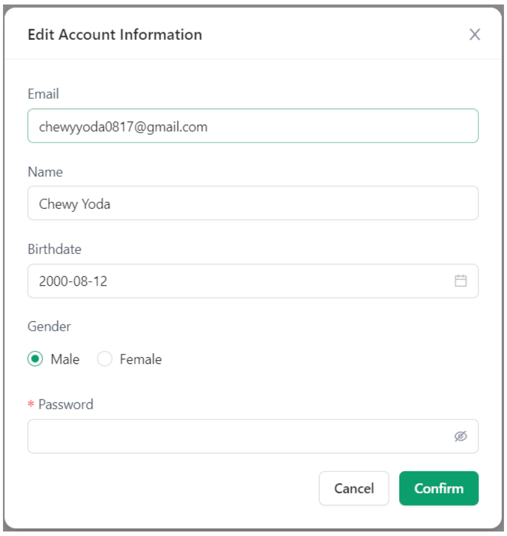
Email	chewyyoda0817@gmail.com
Name	Chewy Yoda
Gender	Male
Birthdate	12-08-2000
Customer Type	Agent
Password	Change

Figure 6.124: Account Information



Your password must fulfill the following criteria:
- ☒ 8 to 16 characters
- ☒ At least 1 digit
- ☒ At least 1 uppercase letter
- ☒ At least 1 lowercase letter
- ☒ At least 1 symbol

Figure 6.125: Change Password



Edit Account Information

Email

chewyyoda0817@gmail.com

Name

Chewy Yoda

Birthdate

2000-08-12

Gender

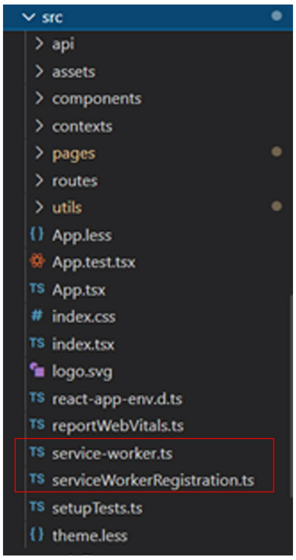
● Male ○ Female

* Password

Figure 6.126: Edit Account Information

6.7.11 PWA Setup

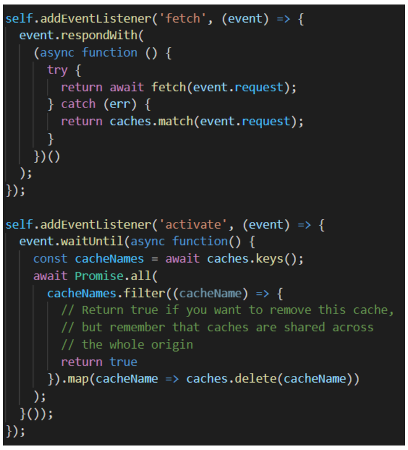
The project uses PWA to provide a native-like mobile application to the customer of Sharifah Food. In order to set up PWA, the front-end application of the e-commerce platform is built with the PWA template from Create React App. After that, the build files are shown in Figure 6.127.



```
src
  api
  assets
  components
  contexts
  pages
  routes
  utils
  App.less
  App.test.tsx
  App.tsx
  index.css
  index.tsx
  logo.svg
  react-app-env.d.ts
  reportWebVitals.ts
  service-worker.ts
  serviceWorkerRegistration.ts
  setupTests.ts
  theme.less
```

Figure 6.127: PWA Build Files

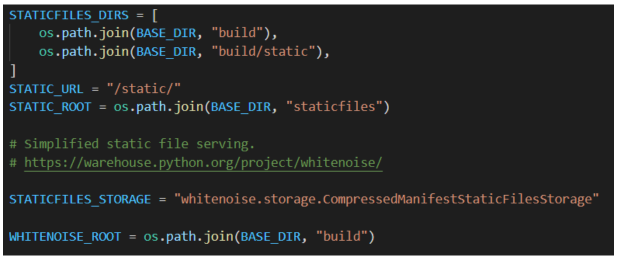
In addition, some functions have been added to the service-worker.ts file to apply the network first cache strategy for the PWA. Figure 6.128 shows the code snippets to apply the network first cache strategy on the PWA.



```
self.addEventListener('fetch', (event) => {
  event.respondWith(
    (async function () {
      try {
        return await fetch(event.request);
      } catch (err) {
        return caches.match(event.request);
      }
    })()
  );
});

self.addEventListener('activate', (event) => {
  event.waitUntil(async function() {
    const cacheNames = await caches.keys();
    await Promise.all(
      cacheNames.filter((cacheName) => {
        // Return true if you want to remove this cache,
        // but remember that caches are shared across
        // the whole origin
        return true
      }).map(cacheName => caches.delete(cacheName))
    );
  }());
});
```

Figure 6.128: Code Snippet for Network First Cache Strategy

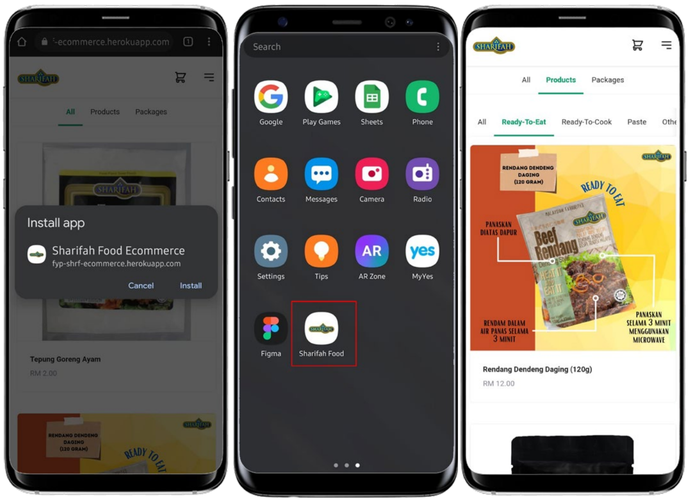
Lastly, the back-end server’s static files were configured so that the PWA could be able to cache for offline access. Figure 6.129 shows the server configurations on the static files.



```
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "build"),
    os.path.join(BASE_DIR, "build/static"),
]
STATIC_URL = "/static/"
STATIC_ROOT = os.path.join(BASE_DIR, "staticfiles")

# Simplified static file serving.
# https://warehouse.python.org/project/whitenoise/

STATICFILES_STORAGE = "whitenoise.storage.CompressedManifestStaticFilesStorage"

WHITENOISE_ROOT = os.path.join(BASE_DIR, "build")
```

Figure 6.129: Code Snippet of Static Files Settings

After all, the mobile application is able to install the e-commerce platform using the browser, as shown in Figure 6.130.



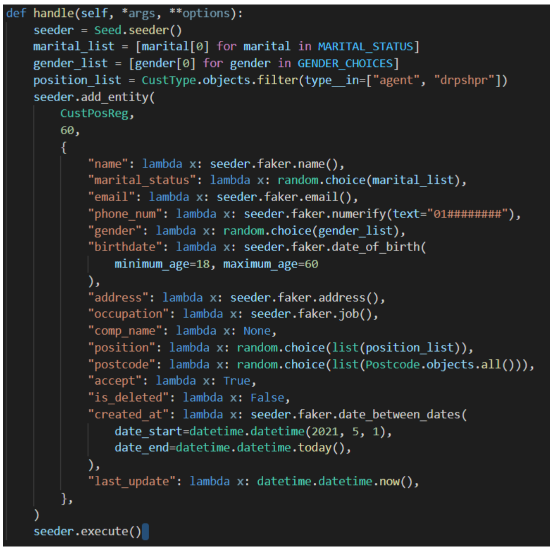
Install app

Sharifah Food Ecommerce
fyp-shrf-ecommerce.herokuapp.com

Figure 6.130: PWA as Mobile Application

6.8 Data Seeding

In order to populate the database with data for analysis purposes, data seeding is performed. The project utilizes the django-seed library to seed some sample data into the database. In addition, some management commands have been developed to perform the data seeding process. Figure 6.131 shows the data seeding management command to populate the agent and dropshipper data.



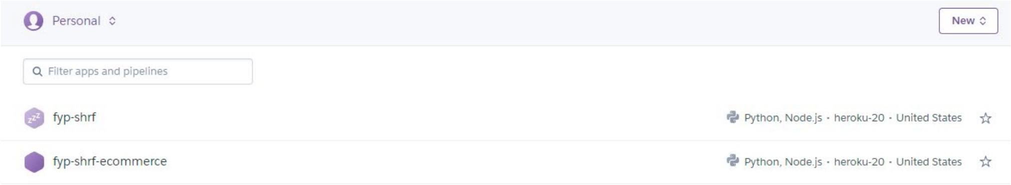
```python
def handle(self, *args, **options):
    seeder = Seed.seeder()
    marital_list = [marital[0] for marital in MARITAL_STATUS]
    gender_list = [gender[0] for gender in GENDER_CHOICES]
    position_list = CustType.objects.filter(type__in=["agent", "drpshpr"])
    seeder.add_entity(
        CustPosReg,
        60,
        {
            "name": lambda x: seeder.faker.name(),
            "marital_status": lambda x: random.choice(marital_list),
            "email": lambda x: seeder.faker.email(),
            "phone_num": lambda x: seeder.faker.numerify(text="01########"),
            "gender": lambda x: random.choice(gender_list),
            "birthdate": lambda x: seeder.faker.date_of_birth(
                minimum_age=18, maximum_age=60
            ),
            "address": lambda x: seeder.faker.address(),
            "occupation": lambda x: seeder.faker.job(),
            "comp_name": lambda x: None,
            "position": lambda x: random.choice(list(position_list)),
            "postcode": lambda x: random.choice(list(Postcode.objects.all())),
            "accept": lambda x: True,
            "is_deleted": lambda x: False,
            "created_at": lambda x: seeder.faker.date_between_dates(
                date_start=datetime.datetime(2021, 5, 1),
                date_end=datetime.datetime.today(),
            ),
            "last_update": lambda x: datetime.datetime.now(),
        },
    )
    seeder.execute()
```

Figure 6.131: Code Snippet of Data Seeding for Agent/Dropshipper

In addition, around 350 orders are seeded to populate the orders data from 2021 to 2022. After the data seeding process is finished, the sales and inventory analyses can be easily tested and verified.

6.9 System Deployment

The inventory management system and the e-commerce platform are deployed and managed through Heroku. Figure 6.132 shows the apps deployed in the Heroku server. The fyp-shrf app refers to the inventory management system, while the fyp-shrf-ecommerce app refers to the e-commerce platform.

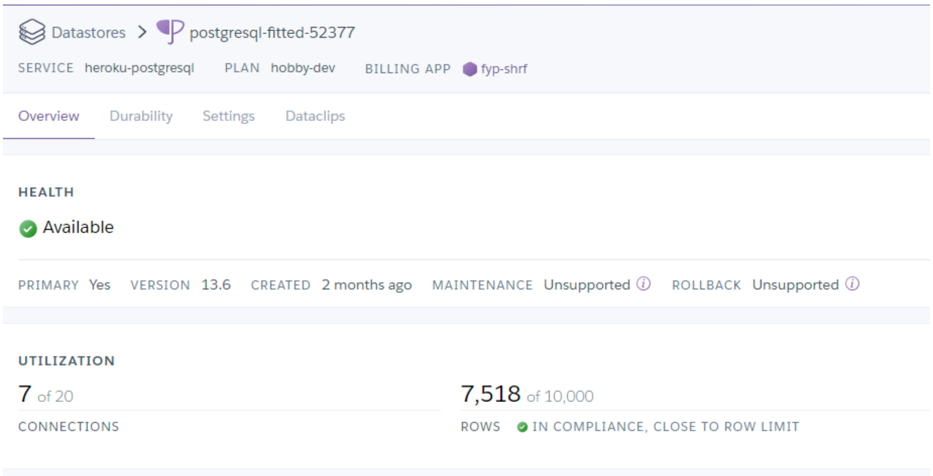


fyp-shrf — Python, Node.js · heroku-20 · United States

fyp-shrf-ecommerce — Python, Node.js · heroku-20 · United States

Figure 6.132: Heroku Apps

Besides, some Heroku's add-ons were in the project, including the Heroku Postgres, Heroku Redis, and Heroku Scheduler. The Heroku Postgres is a Postgres database provided by Heroku, and it provides free database access with a total of 1 GB and a limit of 10,000 rows. Thus, both the inventory management system and e-commerce platform share the Heroku Postgres as a database. Figure 6.133 shows the Heroku Postgres database used by the project.



Datastores > postgresql-fitted-52377

SERVICE heroku-postgresql PLAN hobby-dev BILLING APP fyp-shrf

HEALTH

Available

PRIMARY Yes VERSION 13.6 CREATED 2 months ago MAINTENANCE Unsupported ⓘ ROLLBACK Unsupported ⓘ

UTILIZATION

7 of 20

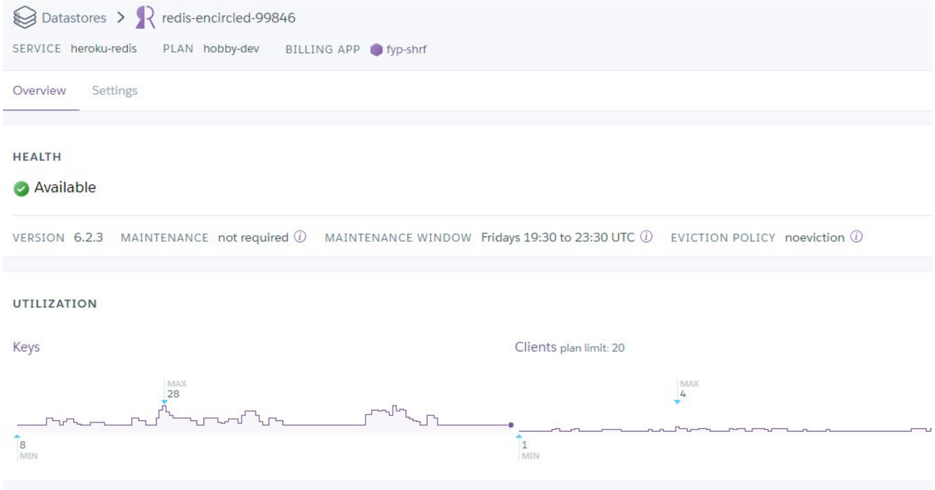
CONNECTIONS

7,518 of 10,000

ROWS ● IN COMPLIANCE, CLOSE TO ROW LIMIT

Figure 6.133: Heroku Postgres

Additionally, Heroku also provides its Redis server as Heroku Redis. The Heroku Redis provides free 25Mb memory for the Redis server. Therefore, both the inventory management system and e-commerce platform are connected to two Heroku Redis as the caching servers, respectively. Figure 6.134 shows the Heroku Redis server used by the inventory management system.



Datastores > redis-encircled-99846

SERVICE heroku-redis PLAN hobby-dev BILLING APP fyp-shrf

HEALTH

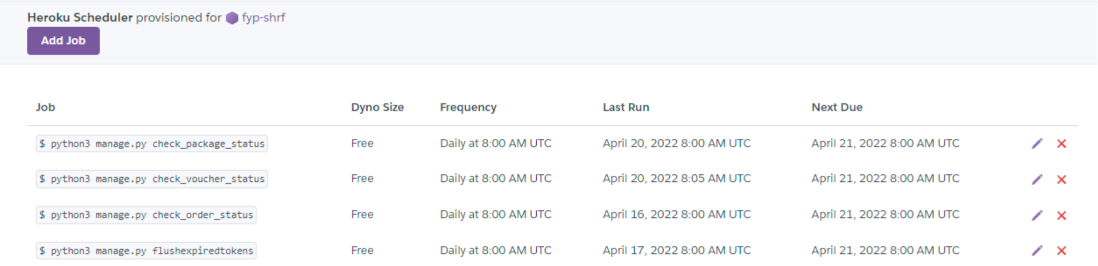
Available

VERSION 6.2.3 MAINTENANCE not required ⓘ MAINTENANCE WINDOW Fridays 19:30 to 23:30 UTC ⓘ EVICTION POLICY noeviction ⓘ

UTILIZATION

Figure 6.134: Heroku Redis

Lastly, Heroku Scheduler was used to run the daily scheduled tasks. There are several tasks that require daily checking. For instance, the system would require checking the available period of the promotional packages and vouchers so that the scheduled packages and vouchers can be available or expired in the e-commerce platform based on the date scheduled. Moreover, the system would also need to check for the unpaid orders to be cancelled after 3 days. Since the system does not implement auto shipment tracking for orders, the system would require marking the order as completed after 14 days of shipping. Thus, there are three management commands for the scheduled tasks created and added to the Heroku Scheduler so that the commands can be executed automatically on a daily basis. Figure 6.135 shows the scheduled commands to be executed by Heroku Scheduler.



Heroku Scheduler provisioned for fyp-shrf

Job	Dyno Size	Frequency	Last Run	Next Due
`$ python3 manage.py check_package_status`	Free	Daily at 8:00 AM UTC	April 20, 2022 8:00 AM UTC	April 21, 2022 8:00 AM UTC
`$ python3 manage.py check_voucher_status`	Free	Daily at 8:00 AM UTC	April 20, 2022 8:05 AM UTC	April 21, 2022 8:00 AM UTC
`$ python3 manage.py check_order_status`	Free	Daily at 8:00 AM UTC	April 16, 2022 8:00 AM UTC	April 21, 2022 8:00 AM UTC
`$ python3 manage.py flushexpiredtokens`	Free	Daily at 8:00 AM UTC	April 17, 2022 8:00 AM UTC	April 21, 2022 8:00 AM UTC

Figure 6.135: Heroku Scheduler

6.10 Summary

In conclusion, this chapter has discussed the detailed system implementation. All the features to be included in the inventory management system and e-commerce platform are fully implemented. Besides, some system enhancements were also performed to move the systems toward the enterprise level. The system enhancements include the usage of model inheritance in solving the polymorphic relationships, soft delete design pattern, API pagination, data audit trail, and database transaction management and concurrency control. Moreover, the performance of the systems was also boosted with the optimization of the database query and caching. The features implemented in the inventory management system and e-commerce platform were also discussed throughout the chapter. Lastly, data seeding was carried out, followed by Heroku's system deployment.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

Software testing is the process of verifying the software system to ensure that the system meets the expected requirements and the system is bug-free. In this project, four types of testing were performed, including unit testing, integration testing, system usability testing, and user acceptance testing. This chapter discusses the test objectives, test scope, test entry, exit criteria, unit testing, integration testing, usability testing, and user acceptance testing.

7.2 Test Objectives

The objectives of the testing are as follows:

1. To assure that the system meets all the expectations as detailed in the requirements.
2. To identify and expose issues, bugs, defects, and associated risks, ensuring that all the known issues are addressed appropriately before the end of the project.

7.3 Test Scope

The scope of the testing is to perform unit testing, integration testing, system usability testing, and user acceptance testing on the inventory management system and the e-commerce platform.

The modules to be tested in the inventory management system are as follows:

1. Login module
2. Reset password module
3. Product module
4. Package module
5. Order module
6. Shipping fee module
7. Pickup location module
8. Customer module
9. Agent and Dropshipper Registration Module
10. Voucher module

The modules to be tested in the e-commerce platform are as follows:

1. Register module
2. Login module
3. Reset password module
4. Item module
5. Cart module
6. Address module
7. Order module
8. Payment module
9. Shipping fee module
10. Pickup location module
11. Agent and Dropshipper Registration Module
12. Voucher module
13. Customer module

7.4 Test Level

The testing will include three test levels, which are as follows:

1. Unit Testing
2. Integration Testing
3. User Acceptance Testing

7.5 Test Entry

1. The structures of the systems are completed.
2. The environment for testing is ready.
3. Necessary tools for the test are acquired.

7.6 Exit Criteria

1. All the planned tests are performed.
2. The pass rate of all tests is more than 95%.
3. All the high-priority errors and failures are fixed.

7.7 Unit Testing

7.7.1 Inventory Management System

a. Login Module

Table 7.1: Inventory Management System Unit Test Case - Admin Login

Test Case ID	INV001	**Module Name**	Login Module	
Test Title	Unit Test Case for Admin Login			
Pre-condition	1. Admin has an active account.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter valid credentials	1. Enter username or email 2. Enter password 3. Click Login button	- Valid username or email - Valid password	Redirect to dashboard page	Pass
Enter empty credentials	1. Enter username or email 2. Enter password 3. Click Login button	- Empty username or email - Empty password	Display error message	Pass
Enter invalid credentails	1. Enter username or email 2. Enter password 3. Click Login button	- Invalid username or email - Valid password	Display error message	Pass
Enter invalid password	1. Enter username or email 2. Enter password 3. Click Login button	- Valid username or email - Invalid password	Display error message	Pass

Table 7.1 (Continued)

Enter invalid username or email	1. Enter username or email 2. Enter password 3. Click Login button	- Invalid username or email - Valid password	Display error message	Pass
Failed Attempts >= 5	1. Enter username or email 2. Enter invalid password 3. Click Login button 4. Repeat steps 1 to 3 for 5 times.	- Valid username or email - Invalid password	IP address and account locked	Pass

b. Reset Password Module

Table 7.2: Inventory Management System Unit Test Case - Admin Password Reset

Test Case ID	INV002	**Module Name**	Reset Password Module	
Test Title	Unit Test Case for Admin Password Reset			
Pre-condition	1. Admin has an active account.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Request reset password with valid email	1. Click "Forgot Password" button 2. Enter email address 3. Enter "Reset Password" button	- Valid email	Send an email with reset password link	Pass
Request reset password with invalid email	1. Click "Forgot Password" button 2. Enter email address 3. Enter "Reset Password" button	- Invalid email	Display error message	Pass

Table 7.2 (Continued)

Reset password with valid token	1. Click reset password link in email received. 2. Redirect to reset password page 3. Enter new password 4. Enter confirm password	-Valid token -Valid new password -Valid confirm password	Update new password	Pass
Reset password with invalid token	1. Click reset password link in email received. 2. Redirect to reset password page	-Invalid token	Redirect to login page	Pass
Reset password with invalid new password	1. Click reset password link in email received. 2. Redirect to reset password page 3. Enter new password 4. Enter confirm password	-Valid token - Invalid new password -Valid confirm password	Display error message	Pass
Reset password with invalid confirm password	1. Click reset password link in email received. 2. Redirect to reset password page 3. Enter new password 4. Enter confirm password	-Valid token - Valid new password -Invalid confirm password	Display error message	Pass

This unit test case is reused in the e-commerce platform

c. Product Module

Table 7.3: Inventory Management System Unit Test Case - View Products

<table>
<tr><th>Test Case ID</th><th>INV003</th><th colspan="2">Module Name</th><th>Product Module</th></tr>
<tr><td><b>Test Title</b></td><td colspan="4">Unit Test Case for Viewing Products</td></tr>
<tr><td><b>Pre-condition</b></td><td colspan="4">1. Admin has logged in to the system.
2. There is at least one product in the system.</td></tr>
<tr><td><b>Test Case Summary</b></td><td><b>Test Steps</b></td><td><b>Test Data</b></td><td><b>Expected Result</b></td><td><b>Status</b></td></tr>
<tr><td>Display all products</td><td>1. Click “Manage Products” sider menu
2. Redirect to product management page</td><td>-</td><td>A list of products is shown</td><td>Pass</td></tr>
<tr><td>Display products by status</td><td>1. Click “Manage Products” sider menu
2. Redirect to product management page</td><td>-</td><td>A list of products with respect to the status is shown</td><td>Pass</td></tr>
<tr><td>Display products based on ordering</td><td>1. Click “Manage Products” sider menu
2. Redirect to product management page
3. Click the table column with the sort icon</td><td>-</td><td>A list of products ordered by the table column is shown</td><td>Pass</td></tr>
<tr><td>Display products with filters</td><td>1. Click “Manage Products” sider menu
2. Redirect to product management page
3. Enter inputs to filter such as product name, sku, stock, category and price</td><td>-</td><td>A list of products filtered by the inputs is shown</td><td>Pass</td></tr>
</table>

Table 7.4: Inventory Management System Unit Test Case - Add Product

Test Case ID	INV004	**Module Name**	Product Module	
Test Title	Unit Test Case for Adding Product			
Pre-condition	1. Admin has logged in to the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter all valid required fields	1. Click "Add Product" sider menu 2. Redirect to add product page 3. Enter all required information 4. Upload valid thumbnail 5. Click "Add Product" button	name, category, thumbnail, description, price, cost_per_unit, sku, stock, weight, length, width, height	The product is added to the system	Pass
Enter duplicated item SKU	1. Click "Add Product" sider menu 2. Redirect to add product page 3. Enter all required information 4. Upload valid thumbnail 5. Click "Add Product" button	name, category, thumbnail, description, price, cost per unit, sku, stock, weight, length, width, height	Display "SKU already exists" message	Pass

Table 7.4 (Continued)

Do not enter requried fields	1. Click “Add Product” sider menu 2. Redirect to add product page 4. Click “Add Product” button	-	Display error message	Pass

Table 7.5: Inventory Management System Unit Test Case - Edit Product

Test Case ID	INV005	**Module Name**	Product Module	
Test Title	Unit Test Case for Editing Product			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one product in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Edit with valid information	1. Click “Manage Products” sider menu 2. Click “Edit” button on a product 3. Edit name, category and price 4. Click “Edit Product” button	name, category, price	Selected product updated with name, category and price	Pass
Edit with empty required fields	1. Click “Manage Products” sider menu 2. Click “Edit” button on a product 3. Redirect to the edit product page 3. Erase name 4. Click “Edit Product” button	-	Display error message	Pass

Table 7.5 (Continued)

Update status to “Hidden”	1. Click “Manage Products” sider menu 2. Select at least one product in the table 3. Click “Hide” button 4. Click “Confirm” button on the modal popup	status	Product status updated to “hidden”	Pass
Add product stock	1. Click “Product Inventory” sider menu 2. Click “+” button on the “Action” column table 3. Enter 10 in the input on the “Action” column table 4. Click the “Save” button	stock	Product added 10 more stocks on top of the existing stock	Pass
Reduce product stock	1. Click “Product Inventory” sider menu 2. Click “-” button on the “Action” column table 3. Enter 5 in the input on the “Action” column table 4. Click the “Save” button	stock	Product reduced 5 stocks on top of the existing stock	Pass
Set product stock	1. Click “Product Inventory” sider menu 2. Click “Set” button on the “Action” column table 3. Enter 20 in the input on the “Action” column table 4. Click the “Save” button	stock	Product stock updated to 20	Pass

Table 7.6: Inventory Management System Unit Test Case - Delete Product

Test Case ID	INV006	**Module Name**	Product Module	
Test Title	Unit Test Case for Deleting Product			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one product in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Delete a product	1. Click “Manage Products” sider menu 2. Click “Delete” button on a product 3. Click “Delete” button on the modal popup	-	Selected product updated is_deleted to true	Pass

d. Package Module

Table 7.7: Inventory Management System Unit Test Case - View Package

Test Case ID	INV007	**Module Name**	Package Module	
Test Title	Unit Test Case for Viewing Packages			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one package in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display all packages	1. Click “Manage Packages” sider menu 2. Redirect to package management page	-	A list of packages is shown	Pass
Display packages by status	1. Click “Manage Packages” sider menu 2. Redirect to package management page	-	A list of packages with respect to the status is shown	Pass

Table 7.7 (Continued)

Display packages based on ordering	1. Click "Manage Packages" sider menu 2. Redirect to package management page 3. Click the table column with the sort icon	-	A list of packages ordered by the table column is shown	Pass
Display packages with filters	1. Click "Manage Packages" sider menu 2. Redirect to package management page 3. Enter inputs to filter such as package name, sku, stock, available period and price	-	A list of packages filtered by the inputs is shown	Pass

Table 7.8: Inventory Management System Unit Test Case - Add Package

Test Case ID	INV008	**Module Name**	Package Module	
Test Title	Unit Test Case for Adding Package			
Pre-condition	1. Admin has logged in to the system. 2. There are at least one product in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter all valid required fields	1. Click "Add Package" sider menu 2. Redirect to add package page 3. Enter all required information 4. Upload valid thumbnail 5. Click "Add Package" button	name, thumbnail, description, products, price, sku, stock, weight, length, width, height, avail_start_dt	The package is added to the system	Pass
Enter duplicated item SKU	1. Click "Add Package" sider menu 2. Redirect to add package page 3. Enter all required information 4. Upload valid thumbnail 5. Click "Add Package" button	name, thumbnail, description, products, price, sku, stock, weight, length, width, height, avail_start_dt	Display "SKU already exists" message	Pass

Table 7.8 (Continued)

Do not enter requried fields	1. Click "Add Package" sider menu 2. Redirect to add package page 4. Click "Add Package" button	-	Display error message	Pass

Table 7.9: Inventory Management System Unit Test Case - Edit Package

Test Case ID	INV009	**Module Name**	Package Module	
Test Title	Unit Test Case for Editing Package			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one package in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Edit with valid information	1. Click "Manage Packages" sider menu 2. Click "Edit" button on a package 3. Redirect to the edit package page 4. Edit name and price 5. Click "Edit Package" button	name, price	Selected package updated with name and price	Pass
Edit with empty required fields	1. Click "Manage Packages" sider menu 2. Click "Edit" button on a package 3. Redirect to the edit package page 4. Erase name 5. Click "Edit Package" button	-	Display error message	Pass

Table 7.9 (Continued)

Update status to “Hidden”	1. Click “Manage Packages” sider menu 2. Select at least one product in the table 3. Click “Hide” button 4. Click “Confirm” button on the modal popup	status	Package status updated to “hidden”	Pass
Add package stock	1. Click “Package Inventory” sider menu 2. Click “+” button on the “Action” column table 3. Enter 10 in the input on the “Action” column table 4. Click the “Save” button	stock	Package added 10 more stocks on top of the existing stock	Pass
Reduce package stock	1. Click “Package Inventory” sider menu 2. Click “-” button on the “Action” column table 3. Enter 5 in the input on the “Action” column table 4. Click the “Save” button	stock	Package reduced 5 stocks on top of the existing stock	Pass
Set package stock	1. Click “Package Inventory” sider menu 2. Click “Set” button on the “Action” column table 3. Enter 20 in the input on the “Action” column table 4. Click the “Save” button	stock	Package stock updated to 20	Pass

Table 7.10: Inventory Management System Unit Test Case - Delete Package

Test Case ID	INV010	**Module Name**	Package Module	
Test Title	Unit Test Case for Deleting Product			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one package in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Delete a package	1. Click "Manage Package" sider menu 2. Click "Delete" button on a package 3. Click "Delete" button on the modal popup	-	Selected package updated is_deleted to true	Pass

e. Order Module

Table 7.11: Inventory Management System Unit Test Case - View Orders

Test Case ID	INV011	**Module Name**	Order Module	
Test Title	Unit Test Case for Viewing Orders			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one order in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display all orders	1. Click "Manage Orders" sider menu 2. Redirect to order management page	-	A list of orders is shown	Pass
Display orders by status	1. Click "Manage Orders" sider menu 2. Redirect to order management page	-	A list of orders with respect to the status is shown	Pass

Table 7.11 (Continued)

Display orders based on ordering	1. Click “Manage Orders” sider menu 2. Redirect to order management page 3. Click the table column with the sort icon	-	A list of orders ordered by the table column is shown	Pass
Display orders with filters	1. Click “Manage Orders” sider menu 2. Redirect to order management page 3. Enter inputs to filter such as order number, customer name, customer email, tracking number, order amount, customer type, order date	-	A list of orders filtered by the inputs is shown	Pass
Display order details	1. Click “Manage Orders” sider menu 2. Redirect to order management page 3. Click the order number of the selected order 4. Redirect to view order page	-	The order details of the selected order is shown	Pass

Table 7.12: Inventory Management System Unit Test Case - Update Order

Test Case ID	INV012	**Module Name**	Order Module	
Test Title	Unit Test Case for Updating Order			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one order in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Update tracking number	1. Click “Manage Orders” sider menu 2. Redirect to order management page 3. Click the “Update” button on the selected order 4. Enter tracking number on the modal popup 5. Click “Update” on the modal popup	-	Order tracking number is updated, and status is set to “To Ship”	Pass
Confirm pickup	1. Click “Manage Orders” sider menu 2. Redirect to order management page 3. Click the “Pickup” button on the selecteed order 4. Click “Confirm” on the modal popup	-	Order pickup date is updated to now, and status is set to “Completed”	Pass

Table 7.12 (Continued)

Cancel order	1. Click “Manage Orders” sider menu 2. Redirect to order management page 3. Click the “Cancel” button on an order 4. Enter tracking number on the modal popup 5. Click “Update” on the modal popup	-	Order status is set to “Cancelled”	Pass

Table 7.13: Inventory Management System Unit Test Case - Generate Order Invoice

Test Case ID	INV013	**Module Name**	Order Module	
Test Title	Unit Test Case for Generating Order Invoice			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one order in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Generate single invoice	1. Click “Manage Orders” sider menu 2. Redirect to order management page 3. Click the “Invoice” button on an order 4. Click “Generate” on the modal popup	-	The invoice is downloaded as pdf file.	Pass

Table 7.13 (Continued)

Generate multiple invoices	1. Click “Manage Orders” sider menu 2. Redirect to order management page 3. Select multiple orders 3. Click the “Generate Invoice(s)” button 4. Click “Generate” on the modal popup	-	The invoices is held in a zip file, and the zip file is downloaded.	Pass

f. Shipping Fee Module

Table 7.14:Inventory Management System Unit Test Case - View Shipping Fees

Test Case ID	INV014	**Module Name**	Shipping Fee Module	
Test Title	Unit Test Case for Viewing Shipping Fees			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one shipping fee record in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display all shipping fees	1. Click “Manage Shipping Fees” sider menu 2. Redirect to shipping fees management page	-	A list of shipping fees is shown	Pass
Display shipping fees based on ordering	1. Click “Manage Shipping Fees” sider menu 2. Redirect to shipping fees management page 3. Click the table column with the sort icon	-	A list of shipping fees ordered by the table column is shown	Pass

Table 7.14 (Continued)

Display shipping fees with filters	1. Click “Manage Shipping Fees” sider menu 2. Redirect to shipping fees management page 3. Enter inputs to filter such as location, weight range and shipping fee range	-	A list of shipping fees filtered by the inputs is shown	Pass

Table 7.15: Inventory Management System Unit Test Case - Add Shipping Fees

Test Case ID	INV015	**Module Name**	Shipping Fee Module	
Test Title	Unit Test Case for Adding Shipping Fees			
Pre-condition	1. Admin has logged in to the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter all valid required fields	1. Click “Add Shipping Fees” sider menu 2. Redirect to add shipping fee page 3. Enter all required information 4. Click “Add Shipping Fees” button	location, weight_start, weight_end and ship_fee	The shipping fee records are added to the system	Pass
Do not enter requried fields	1. Click “Add Shipping Fees” sider menu 2. Redirect to add shipping fees page 4. Click “Add Shipping Fees” button	-	Display error message	Pass

Table 7.16: Inventory Management System Unit Test Case - Edit Shipping Fees

Test Case ID	INV016	**Module Name**	Shipping Fee Module	
Test Title	Unit Test Case for Editing Shipping Fees			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one shipping fee in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Edit with valid information	1. Click "Manage Shipping Fees" sider menu 2. Click "Edit" button on the selected shipping fee 3. Redirect to the edit shipping fee page 4. Edit one shipping fee in the shipping fee list 5. Click "Edit Shipping Fees" button	weight_start, weight_end and ship_fee	All the shipping fees of the selected location updated with new weight_start, weight_end and ship_fee	Pass
Edit with empty required fields	1. Click "Manage Shipping Fees" sider menu 2. Click "Edit" button on the selected shipping fee 3. Redirect to the edit shipping fee page 4. Erase all shipping fees in the shipping fee list 5. Click "Edit Shipping Fees" button	-	Display error message	Pass

Table 7.17: Inventory Management System Unit Test Case - Delete Shipping Fee

Test Case ID	INV017	**Module Name**	Shipping Fee Module	
Test Title	Unit Test Case for Deleting Shipping Fee			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one shipping fee in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Delete a shipping fee	1. Click “Manage Shipping Fees” sider menu 2. Click “Delete” button on a shipping fee 3. Click “Delete” button on the modal popup	-	Selected shipping fee updated is_deleted to true	Pass

g. Pickup Location Module

Table 7.18: Inventory Management System Unit Test Case - View Pickup Locations

Test Case ID	INV018	**Module Name**	Pickup Location Module	
Test Title	Unit Test Case for Viewing Pickup Locations			
Pre-condition	1. Admin has logged in to the system. 3. There is at least one pickup location in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display all pickup locations	1. Click “Manage Pickup Loc” sider menu 2. Redirect to pickup location management page	-	A list of pickup locations is shown	Pass
Display pickup locations based on ordering	1. Click “Manage Pickup Loc” sider menu 2. Redirect to pickup location management page 3. Click the “location” column	-	A list of pickup locations ordered by the location	Pass

Table 7.18 (Continued)

Display pickup locations with filters	1. Click "Manage Pickup Loc" sider menu 2. Redirect to pickup location management page 3. Enter location to filter	-	A list of pickup locations filtered by the location is shown	Pass

Table 7.19: Inventory Management System Unit Test Case - Add Pickup Location

Test Case ID	INV019	**Module Name**	Pickup Location Module	
Test Title	Unit Test Case for Adding Pickup Location			
Pre-condition	1. Admin has logged in to the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter all valid required fields	1. Click "Add Pickup Loc" sider menu 2. Redirect to add pickup location page 3. Enter all required information 4. Click "Add Pickup Location" button	location	The pickup location is added to the system	Pass
Do not enter requried fields	1. Click "Add Pickup Loc" sider menu 2. Redirect to add pickup location page 4. Click "Add Pickup Loc" button	-	Display error message	Pass

Table 7.20: Inventory Management System Unit Test Case - Edit Pickup Location

<table>
<tr><td><b>Test Case ID</b></td><td>INV020</td><td colspan="2"><b>Module Name</b></td><td>Pickup Location Module</td></tr>
<tr><td><b>Test Title</b></td><td colspan="4">Unit Test Case for Editing Pickup Location</td></tr>
<tr><td><b>Pre-condition</b></td><td colspan="4">1. Admin has logged in to the system.
2. There is at least one pickup location in the system.</td></tr>
<tr><td><b>Test Case Summary</b></td><td><b>Test Steps</b></td><td><b>Test Data</b></td><td><b>Expected Result</b></td><td><b>Status</b></td></tr>
<tr><td>Edit with valid information</td><td>1. Click “Manage Pickup Loc” sider menu
2. Click “Edit” button on the selected pickup location
3. Redirect to the edit pickup location page
4. Edit location
5. Click “Edit Pickup Location” button</td><td>location</td><td>The selected pickup location is updated with new location</td><td>Pass</td></tr>
<tr><td>Edit with empty required fields</td><td>1. Click “Manage Pickup Loc” sider menu
2. Click “Edit” button on the selected pickup location
3. Redirect to the edit pickup location page
4. Erase location
5. Click “Edit Pickup Location” button</td><td>-</td><td>Display error message</td><td>Pass</td></tr>
</table>

Table 7.21: Inventory Management System Unit Test Case - Delete Pickup Location

Test Case ID	INV021	**Module Name**	Pickup Location Module	
Test Title	Unit Test Case for Deleting Pickup Location			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one pickup location in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Delete a pickup location	1. Click “Manage Pickup Location” sider menu 2. Click “Delete” button on the selected pickup location 3. Click “Delete” button on the modal popup	-	Selected pickup location updated is_deleted to true	Pass

h. Customer Module

Table 7.22: Inventory Management System Unit Test Case - View Customers

Test Case ID	INV022	**Module Name**	Customer Module	
Test Title	Unit Test Case for Viewing Customers			
Pre-condition	1. Admin has logged in to the system. 1. There is at least one pickup location in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display all customers	1. Click “Manage Customers” sider menu 2. Redirect to customer management page	-	A list of customers is shown	Pass
Display customers based on ordering	1. Click “Manage Customers” sider menu 2. Redirect to customer management page 3. Click the table column with the sort icon	-	A list of customers ordered by the table column	Pass

Table 7.22 (Continued)

Display customers with filters	1. Click "Manage Customers" sider menu 2. Redirect to customer management page 3. Enter inputs to filter such as customer id, name, email, joined date, status, monthly order value and last order date	-	A list of customers filtered by the inputs is shown	Pass

Table 7.23: Inventory Management System Unit Test Case - Update Customer Status

Test Case ID	INV024	**Module Name**	Customer Module	
Test Title	Unit Test Case for Updating Customer Status			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one customer in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Suspend Customer	1. Click "Manage Customers" sider menu 2. Redirect to customer management page 3. Click the "Suspend" button on the selected customer 4. Click "Suspend" on the modal popup	-	Customer's is_active field is set to "false". Email notification is sent.	Pass

Table 7.23 (Continued)

Activate Customer	1. Click "Manage Customers" sider menu 2. Redirect to customer management page 3. Click the "Activate" button on the selected customer 4. Click "Activate" on the modal popup	-	Customer's is_active field is set to "true". Email notification is sent.	Pass

i. Agent and Dropshipper Registration Module

Table 7.24: Inventory Management System Unit Test Case - View Agent or Dropshipper Registrations

Test Case ID	INV025	**Module Name**	Agent and Dropshipper Registration Module	
Test Title	Unit Test Case for Viewing Agent or Dropshipper Registrations			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one registration in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display all agent or dropshipper registrations	1. Click "Manage Registrations" sider menu 2. Redirect to registration management page	-	A list of agent or dropshipper registrations is shown	Pass
Display agent or dropshipper registration based on ordering	1. Click "Manage Registrations" sider menu 2. Redirect to registration management page 3. Click the table column with the sort icon	-	A list of agent or dropshipper registrations ordered by the table column	Pass

Table 7.24 (Continued)

Display agent or dropshipper registration with filters	1. Click "Manage Registrations" sider menu 2. Redirect to registration management page 3. Enter inputs to filter such as registration id, applicant name, applicant email, contact number, registration status and registration date	-	A list of agent or dropshipper registrations filtered by the inputs is shown	Pass

Table 7.25: Inventory Management System Unit Test Case - Add Customer

Test Case ID	INV023	**Module Name**	Agent and Dropshipper Registration Module Module	
Test Title	Unit Test Case for Adding Customer			
Pre-condition	1. Admin has logged in to the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter all valid required fields	1. Click "Add Customer" sider menu 2. Redirect to add customer page 3. Enter all required information 4. Click "Add Customer" button	name, gender, birthdate, marital_status, phone_num, email, state, city, postcode, address, occupation, comp_name, position, status	The customer is added to the system	Pass

Table 7.25 (Continued)

Enter duplicated email	1. Click "Add Customer" sider menu 2. Redirect to add customer page 3. Enter all required information and duplicated email 4. Click "Add Customer" button	name, gender, birthdate, marital_status, phone_num, email, state, city, postcode, address, occupation, comp_name, position, status	Display "email already exists" message	Pass
Do not enter requried fields	1. Click "Add Customer" sider menu 2. Redirect to add customer page 4. Click "Add Pickup Loc" button	-	Display error message	Pass

Table 7.26: Inventory Management System Unit Test Case - Update Agent or Dropshipper Registration Status

Test Case ID	INV026	**Module Name**	Agent and Dropshipper Registration Module	
Test Title	Unit Test Case for Updating Agent or Dropshipper Registration Status			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one registration in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Accept Registration	1. Click "Manage Registrations" sider menu 2. Redirect to registration management page 3. Click the "Accept" button on the selected registration 4. Click "Accept" on the modal popup	-	Registration's accept field set to 'true' Email notification is sent.	Pass
Accept Registration	1. Click "Manage Registrations" sider menu 2. Redirect to registration management page 3. Click the "Reject" button on the selected registration 4. Click "Reject" on the modal popup	-	Registration's accept field set to 'false' Email notification is sent.	Pass

j. Voucher Module

Table 7.27: Inventory Management System Unit Test Case - View Vouchers

Test Case ID	INV027	**Module Name**	Voucher Module	
Test Title	Unit Test Case for Viewing Vouchers			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one product in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display all vouchers	1. Click "Manage Vouchers" sider menu 2. Redirect to voucher management page	-	A list of vouchers is shown	Pass
Display vouchers by status	1. Click "Manage Vouchers" sider menu 2. Redirect to voucher management page	-	A list of vouchers with respect to the status is shown	Pass
Display vouchers based on ordering	1. Click "Manage Vouchers" sider menu 2. Redirect to voucher management page 3. Click the table column with the sort icon	-	A list of vouchers ordered by the table column is shown	Pass
Display vouchers with filters	1. Click "Manage Vouchers" sider menu 2. Redirect to voucher management page 3. Enter inputs to filter such as voucher code, customer type, available date and availability	-	A list of vouchers filtered by the inputs is shown	Pass

Table 7.28: Inventory Management System Unit Test Case - Add Voucher

Test Case ID	INV028	**Module Name**	Voucher Module	
Test Title	Unit Test Case for Adding Voucher			
Pre-condition	1. Admin has logged in to the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter all valid required fields	1. Click “Add Voucher” sider menu 2. Redirect to add voucher page 3. Enter all required information 4. Click “Add Voucher” button	code, type,discount, total_amt, usage_limit, cust_type, avail_start_dt	The voucher is added to the system	Pass
Enter duplicated voucher code	1. Click “Add Voucher” sider menu 2. Redirect to add voucher page 3. Enter all required information 4. Click “Add Voucher” button	code, type,discount, total_amt, usage_limit, cust_type, avail_start_dt	Display “Voucher code already exists” message	Pass
Do not enter requried fields	1. Click “Add Voucher” sider menu 2. Redirect to add voucher page 4. Click “Add Voucher” button	-	Display error message	Pass

Table 7.29: Inventory Management System Unit Test Case - Edit Voucher

Test Case ID	INV029	**Module Name**	Voucher Module	
Test Title	Unit Test Case for Editing Voucher			
Pre-condition	1. Admin has logged in to the system. 2. There is at least one voucher in the system.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Edit with valid information	1. Click “Manage Vouchers” sider menu 2. Click “Edit” button on the selected voucher 3. Edit discount, type 4. Click “Edit Voucher” button	discount and type	Selected voucher updated with discount and type	Pass
Edit with empty required fields	1. Click “Manage Vouchers” sider menu 2. Click “Edit” button on a voucher 3. Redirect to the edit voucher page 3. Erase voucher code 4. Click “Edit Voucher” button	-	Display error message	Pass
Update status to “Hidden”	1. Click “Manage Vouchers” sider menu 2. Select at least one voucher in the table 3. Click “Hide” button 4. Click “Hide” button on the modal popup	status	Voucher status updated to “hidden”	Pass

Table 7.30: Inventory Management System Unit Test Case - Delete Voucher

<table>
<tr><td><b>Test Case ID</b></td><td>INV030</td><td colspan="2"><b>Module Name</b></td><td>Voucher Module</td></tr>
<tr><td><b>Test Title</b></td><td colspan="4">Unit Test Case for Deleting Voucher</td></tr>
<tr><td><b>Pre-condition</b></td><td colspan="4">1. Admin has logged in to the system.
2. There is at least one voucher in the system.</td></tr>
<tr><td><b>Test Case Summary</b></td><td><b>Test Steps</b></td><td><b>Test Data</b></td><td><b>Expected Result</b></td><td><b>Status</b></td></tr>
<tr><td>Delete a voucher</td><td>1. Click “Manage Vouchers” sider menu
2. Click “Delete” button on selected voucher
3. Click “Delete” button on the modal popup</td><td>-</td><td>Selected voucher updated is_deleted to true</td><td>Pass</td></tr>
</table>

7.7.2 E-commerce Platform

a. Register Module

Table 7.31: E-commerce Platform Unit Test Case - Register User

Test Case ID	ECOM001	**Module Name**	Register Module
Test Title	Unit Test Case for User Registration		
Pre-condition	-		

Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter email and password	1. Enter email 2. Enter password 3. Enter confirm password 3. Click Register button	- Valid email - Valid password - Valid confirm password	Create new user in the system	Pass
Enter empty email and password	1. Click Register button	-	Display error message	Pass
Password does not meet criteria	1. Enter email 2. Enter password 3. Enter confirm password 3. Click Register button	- Valid email - Invalid password - Valid confirm password	Display error message	Pass
Password and confirm password not matched	1. Enter email 2. Enter password 3. Enter confirm password 3. Click Register button	- Valid email - Valid password - Invalid confirm password	Display error message	Pass

Table 7.31 (Continued)

Enter duplicated email	1. Enter email 2. Enter password 3. Enter confirm password 3. Click Register button	- Valid email - Valid password - Invalid confirm password	Display “email already exists” message	Pass
Enter invalid email	1. Enter email 2. Enter password 3. Enter confirm password 3. Click Register button	- Invalid email - Valid password - Invalid confirm password	Display error message	Pass

b. Login Module

Table 7.32: E-commerce Platform Unit Test Case - User Login

Test Case ID	ECOM002	**Module Name**	Login Module	
Test Title	Unit Test Case for User Login			
Pre-condition	1. User has an active account.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter valid credentials	1. Enter username or email 2. Enter password 3. Click Login button	- Valid username or email - Valid password	Redirect to home page	Pass
Enter empty credentials	1. Enter username or email 2. Enter password 3. Click Login button	- Empty username or email - Empty password	Display error message	Pass

Table 7.32 (Continued)

Enter invalid credentails	1. Enter username or email 2. Enter password 3. Click Login button	- Invalid username or email - Valid password	Display error message	Pass
Enter invalid password	1. Enter username or email 2. Enter password 3. Click Login button	- Valid username or email - Invalid password	Display error message	Pass
Enter invalid username or email	1. Enter username or email 2. Enter password 3. Click Login button	- Invalid username or email - Valid password	Display error message	Pass

c. Item Module

Table 7.33: E-commerce Platform Unit Test Case - View Items

Test Case ID	ECOM003	**Module Name**	Item Module	
Test Title	Unit Test Case for Viewing Items			
Pre-condition	-			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display all active items	1. Enter the home page	-	A list of active items is shown	Pass

Table 7.33 (Continued)

Search items	1. Click the search item input 2. Enter item name keyword 3. Click search icon	-	A list of items whose name contains the keyword is shown	Pass
Display all active products	1. Enter the home page 2. Click “Product” tab	-	A list of active products is shown	Pass
Display all active packages	1. Enter the home page 2. Click “Packages” tab	-	A list of active packages is shown	Pass
Display item details	1. Enter the home page 2. Click on an item 3. Redirect to the item details page		The selected item details are shown	Pass

d. Cart Module

Table 7.34: E-commerce Platform Unit Test Case - Retrieve Cart

Test Case ID	ECOM004	**Module Name**	Cart Module	
Test Title	Unit Test Case for Retrieving Cart			
Pre-condition	-			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
User has cart	1. Enter the home page 2. Click the shopping cart icon	-	A list of cart items is shown	Pass
User has no cart	1. Enter the home page 2. Click the shopping cart icon	-	"No item in your cart" message is shown	Pass
User's cart contains out-of-stock items	1. Enter the home page 2. Click the shopping cart icon	-	"Please remove out of stock items" message is shown	Pass

Table 7.35: E-commerce Platform Unit Test Case - Update Cart

Test Case ID	ECOM005	**Module Name**	Cart Module	
Test Title	Unit Test Case for Updating Cart			
Pre-condition	-			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Item quantity is less than item stock	1. Enter the home page 2. Click the shopping cart icon 3. Click "+" button on a cart item	-	Added 1 more quantity on top of the existing item cart	Pass
Item quantity is more or equal to item stock	1. Enter the home page 2. Click the shopping cart icon 3. Click "+" button on a cart item	-	Display "item reached maximum stock" message	Pass
Remove item cart	1. Enter the home page 2. Click the shopping cart icon 3. Enter 0 on the quantity input of an item	-	Remove the item from the cart	Pass

e. Address Module

Table 7.36: E-commerce Platform Unit Test Case - Retrieve Address

Test Case ID	ECOM006	Module Name	Address Module	
Test Title	Unit Test Case for Retrieving Address			
Pre-condition	1. User is logged in.			
Test Case Summary	Test Steps	Test Data	Expected Result	Status
User has no address	1. Enter the checkout page	-	Display "No Address Foun" message	Pass
User has address	2. Enter the checkout page	-	Display the default address on the shipment information	Pass

Table 7.37: E-commerce Platform Unit Test Case - Add Address

Test Case ID	ECOM007	Module Name	Address Module	
Test Title	Unit Test Case for Adding Address			
Pre-condition	1. User is logged in.			
Test Case Summary	Test Steps	Test Data	Expected Result	Status
Enter all valid required fields	1. Enter address book page. 2. Click "Add Address" button 3. Enter all required information 4. Click "Add" button	contact_name, contact_num, state, city, postcode, address, default	The address is added to the system	Pass

Table 7.37 (Continued)

Do not enter requried fields	1. Enter address book page. 2. Click “Add Address” button 3. Click “Add” button	-	Display error message	Pass

Table 7.38: E-commerce Platform Unit Test Case - Edit Address

Test Case ID	ECOM008	**Module Name**	Address Module	
Test Title	Unit Test Case for Editing Address			
Pre-condition	1. User is logged in.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Edit with valid information	1. Enter address book page. 2. Click “Edit” text button on an address card 3. Edit address and contact_num 4. Click “Edit” button	address and contact_num	Selected address updated with address and contact_num	Pass
Edit with empty required fields	1. Enter address book page. 2. Click “Edit” text button on an address card 3. Erase contact_name 4. Click “Edit” button	-	Display error message	Pass

Table 7.39: E-commerce Platform Unit Test Case - Delete Address

Test Case ID	ECOM009	**Module Name**	Address Module	
Test Title	Unit Test Case for Deleting Address			
Pre-condition	1. User is logged in.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Delete an address	1. Enter address book page. 2. Click "Edit" text button on an address card 3. Click "Delete" button	-	The selected address is deleted	Pass

f. Order Module

Table 7.40: E-commerce Platform Unit Test Case - Add Order

Test Case ID	ECOM010	**Module Name**	Order Module	
Test Title	Unit Test Case for Adding Order			
Pre-condition	1. User's shopping cart contains items. 2. Shipment information is provided. 3. Payment method is selected.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Add Order	1. Enter the checkout page. 2. Fill in shipment information. 3. Select payment method. 4. Click "Place Order"	contact_name, contact_num, postcode, address, email, cart	New order is created with unpaid status	Pass

Table 7.41: E-commerce Platform Unit Test Case - View Orders

Test Case ID	ECOM011	**Module Name**	Order Module	
Test Title	Unit Test Case for Viewing Order			
Pre-condition	-			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display order history of user	1. Click user icon. 2. Click "Order History" menu tab 3. Redirect to order history page	-	A list of all orders placed by the users is shown	Pass
Display order details of user	1. Click user icon. 2. Click "Order History" menu tab 3. Redirect to order history page 4. Click one of the orders	-	A list of all orders placed by the users is shown	Pass
Search order (logged-in user)	1. Click user icon. 2. Click "Order History" menu tab 3. Redirect to order history page 4. Click the search order input 5. Enter order number keyword 6. Click the search icon	id (order number)	The order details are shown	Pass

Table 7.41 (Continued)

Search order (user without logged in)	1. Click the hamburger menu icon 2. Click “Search Order” menu tab 3. Enter email 4. Enter order number	valid email and id (order number)	The order details are shown	Pass
Search order with invalid order number (user without logged in)	1. Click the hamburger menu icon 2. Click “Search Order” menu tab 3. Enter email 4. Enter order number	- Valid email - Invalid id (order number)	Display error message	Pass
Search order with invalid email (user without logged in)	1. Click the hamburger menu icon 2. Click “Search Order” menu tab 3. Enter email 4. Enter order number	- Inalid email - Valid id (order number)	Display error message	Pass

g. Payment Module

Table 7.42: E-commerce Platform Unit Test Case - Pay Order

Test Case ID	ECOM012	**Module Name**	Payment Module	
Test Title	Unit Test Case for Paying Order			
Pre-condition	1. An order has been placed.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Pay order by credit or debit card	1. Redirected to payment page provided by the payment gateway 2. Enter card number of 4242 4242 4242 4242 3. Enter card expiry date of 12/31 4. Enter CVC number of 123 5. Enter name on card 6. Click "Pay" button.	card number, card expiry date, card CVC number, name on card	Payment successful	Pass
Pay order by FPX online banking	1. Redirected to payment page provided by the payment gateway 2. Select bank 3. Enter name. 4. Click "Pay" button. 5. Redirected to payment testing page. 6. Click "Authorize test payment".	bank, name	Payment successful	Pass

Table 7.42 (Continued)

Failed to pay the order by FPX online banking	1. Redirected to payment page provided by the payment gateway 2. Select bank 3. Enter name. 4. Click “Pay” button. 5. Redirected to payment testing page. 6. Click “Fail test payment” .	bank, name	Redirect back to the payment page	Pass
Return back to e-commerce platform without complete payment	1. Redirected to payment page provided by the payment gateway 2. Click the return button	-	Display “Payment Failed” message	Pass

h. Shipping Fee Module

Table 7.43: E-commerce Platform Unit Test Case - Retrieve Shipping Fee

Test Case ID	ECOM013	**Module Name**	Shipping Fee Module	
Test Title	Unit Test Case for Retrieving Shipping Fee			
Pre-condition	1. The user chooses the shipping method when checking out the order 2. Subtotal price and the total weight of cart items are known.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Calculate shipping fee	1. Enter checkout page. 2. Select Shipping Address. 3. Fill in shipment information. 4. Click “Save” button	subtotal_price, cart, state	The shipping fee is calculated correctly	Pass

i. Pickup Location Module

Table 7.44: E-commerce Platform Unit Test Case - Retrieve Pickup Locations

Test Case ID	ECOM014	**Module Name**	Pickup Location Module	
Test Title	Unit Test Case for Retrieving Pickup Location			
Pre-condition	1. User chooses the pickup method when checking out the order			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display list of pickup locations	1. Enter checkout page. 2. On the shipment information component, click “Select Pickup”	-	A list of pickup locations is displayed	Pass

j. Agent and Dropshipper Registration Module

Table 7.45: E-commerce Platform Unit Test Case - Add Agent or Dropshipper Registration

Test Case ID	ECOM015	**Module Name**	Agent and Dropshipper Registration Module	
Test Title	Unit Test Case for Adding Agent or Dropshipper Registration			
Pre-condition	-			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Enter all valid required fields	1. Enter the registration page. 2. Fill in all the required information 3. Click "Submit" button	-	Display success message. Add the registration into the system and notification.	Pass
Do not enter the requried fields	1. Enter the registration page. 2. Click "Submit" button	-	Display error message	Pass

k. Voucher Module

Table 7.46: E-commerce Platform Unit Test Case - Check Voucher

Test Case ID	ECOM016	**Module Name**	Voucher Module	
Test Title	Unit Test Case for Adding Agent or Dropshipper Registration			
Pre-condition	1. User is logged in.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Valid and eligible voucher code	1. Enter the checkout page. 2. Fill in the valid voucher code in the voucher code input. 3. Click "Apply"	code	Display success message. Calculate discounted total price	Pass
Invalid voucher code	1. Enter the checkout page. 2. Fill in the invalid voucher code in the voucher code input. 3. Click "Apply"	code	Display error message	Pass

I. Customer Module

Table 7.47: E-commerce Platform Unit Test Case - Retrieve Account Information

Test Case ID	ECOM017	**Module Name**	Customer Module	
Test Title	Unit Test Case for Retrieving Account Information			
Pre-condition	1. User is logged in.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Display account information	1. Click user icon. 2. Click "Account Information" menu tab 3. Redirect to account information page	-	The account information of the user is displayed	Pass

Table 7.48: E-commerce Platform Unit Test Case - Edit Account Information

Test Case ID	ECOM018	**Module Name**	Customer Module	
Test Title	Unit Test Case for Retrieving Account Information			
Pre-condition	1. User is logged in.			
Test Case Summary	**Test Steps**	**Test Data**	**Expected Result**	**Status**
Edit with valid information	1. Click user icon. 2. Click "Account Information" menu tab 3. Redirect to account information page 4. Click "Edit" button 5. Edit name, birthdate, gender. 6. Enter password.	name, birthdate, gender - Valid password	User's account information is updated with name, birthdate and gender entered	Pass

Table 7.48 (Continued)

Enter with invalid password	1. Click user icon. 2. Click “Account Information” menu tab 3. Redirect to account information page 4. Click “Edit” button 5. Edit name, birthdate, gender. 6. Enter password.	name, birthdate, gender - Valid password	Display Error Message	Pass
Change Password with valid inputs	1. Click user icon. 2. Click “Account Information” menu tab 3. Redirect to account information page 4. Click “Change” button on the “Password” rows 5. Enter old password. 6. Enter new password. 7. Enter confirm new password.	- Valid old password - Valid new password - Valid confirm new password	User’s account password is updated to the new password.	Pass
Change Password with an invalid old password	1. Click user icon. 2. Click “Account Information” menu tab 3. Redirect to account information page 4. Click “Change” button on the “Password” rows 5. Enter old password. 6. Enter new password. 7. Enter confirm new password.	- Invalid old password - Valid new password - Valid confirm new password	Display Error Message	Pass

Table 7.48 (Continued)

Change Password with an invalid new password	1. Click user icon. 2. Click "Account Information" menu tab 3. Redirect to account information page 4. Click "Change" button on the "Password" rows 5. Enter old password. 6. Enter new password. 7. Enter confirm new password.	- Valid old password - Invalid new password - Valid confirm new password	Display Error Message	Pass
Change Password with an invalid confirm new password	1. Click user icon. 2. Click "Account Information" menu tab 3. Redirect to account information page 4. Click "Change" button on the "Password" rows 5. Enter old password. 6. Enter new password. 7. Enter confirm new password.	- Valid old password - Valid new password - Invalid confirm new password	Display Error Message	Pass

7.8 Integration Testing

7.8.1 Inventory Management System

a. Product and Image

Table 7.49: Inventory Management System Integration Test Cases - Product and Image

No	Test Title	Test Step	Expected Result	Status
1	Add images to the product	1. Click "Add Product" sider menu 2. Redirect to add product page 3. Enter all required information 4. Upload valid thumbnail and images 5. Click "Add Product" button	The images are uploaded to Cloudinary and linked to the product	Pass
2	Edit images to the product	1. Click "Manage Products" sider menu 2. Click "Edit" button on the selected product 3. Redirect to edit product page 4. Upload new images 5. Click "Edit Product" button	The new images are uploaded to Cloudinary and linked to the product	Pass
3	Remove images to the product	1. Click "Manage Products" sider menu 2. Click "Edit" button on the selected product 3. Redirect to edit product page 4. Remove existing images 5. Click "Edit Product" button	The removed images are deleted and invalidated on the Cloudinary, and the links between the images and product are removed.	Pass

b. Package and Image

Table 7.50: Inventory Management System Integration Test Cases - Package and Image

No	Test Title	Test Step	Expected Result	Status
1	Add images to the package	1. Click “Add Package” sider menu 2. Redirect to add package page 3. Enter all required information 4. Upload valid thumbnail and images 5. Click “Add Package” button	The images are uploaded to Cloudinary and linked to the package	Pass
2	Edit images to the package	1. Click “Manage Packages” sider menu 2. Click “Edit” button on the selected package 3. Redirect to edit package page 4. Upload new images 5. Click “Edit Package” button	The new images are uploaded to Cloudinary and linked to the package	Pass
3	Remove images to the package	1. Click “Manage Packages” sider menu 2. Click “Edit” button on the selected package 3. Redirect to edit package page 4. Remove existing images 5. Click “Edit Package” button	The removed images are deleted and invalidated on the Cloudinary, and the links between the images and package are removed.	Pass

c. Order and Item (Analysis Module)

Table 7.51: Inventory Management System Integration Test Cases - Order and Item

No	Test Title	Test Step	Expected Result	Status
1	View product rankings	1. Enter the statistics page in business insights 2. Scroll to the product rankings card 3. Change the date range to “By Year” and select 2021 4. Select “By Sales” card in the product ranking card	A descending ordered list of products with their total sales for the year 2021 is shown	Pass
2	View package rankings	1. Enter the statistics page in business insights 2. Scroll to the package rankings card 3. Change the date range to “By Year” and select 2021 4. Select “By Sales” card in the package ranking card	A descending ordered list of packages with their total sales for the year 2021 is shown	Pass
3	View key metrics analysis	1. Enter the statistics page in business insights 2. Scroll to the key metrics card 3. Change the date range to “By Year” and select 2021 4. Check the stack metrics checkbox 5. Randomly select five key metrics	A line chart of five key metrics is displayed	Pass

Table 7.51 (Continued)

4	Generate Sales Analysis Report	1. Enter the statistics page in business insights 2. Scroll to the key metrics card 3. Change the date range to “By Year” and select 2021 4. Click “Generate Report”	An Excel report is downloaded and is able to open. The data in the sales report is accurate.	Pass

d. **Customer, Agent or Dropshipper Registration and Customer Type**

Table 7.52: Inventory Management System Integration Test Cases - Customer, Agent or Dropshipper Registration and Customer Type

No	Test Title	Test Step	Expected Result	Status
1	Accept Registration (New email)	1. Click “Manage Registrations” sider menu 2. Redirect to the registration management page 3. Click the “Reject” button on the selected registration 4. Click “Reject” on the modal popup	Registration’s accept field set to ‘true” A new customer account with random password is created with the email. Email notification is sent.	Pass
1	Accept Registration (existing email)	1. Click “Manage Registrations” sider menu 2. Redirect to the registration management page 3. Click the “Reject” button on the selected registration 4. Click “Reject” on the modal popup	Registration’s accept field set to ‘true” The cust_type of the existing customer account is updated to the position selected for the registration. Email notification is sent.	Pass

7.8.2 E-commerce Platform

a. Cart and Item

Table 7.53: E-commerce Platform Integration Test Cases - Cart and Item

No	Test Title	Test Step	Expected Result	Status
1	Compute subtotal price	1. Enter the home page 2. Click an item 3. Redirect to the item details page 4. Click “Add To Cart” 5. Repeat steps 1 to 4 several times to add multiple items to cart 6. Click the shopping cart icon	Display a list of items added to the cart and the subtotal price of the cart	Pass

b. Cart and Voucher

Table 7.54: E-commerce Platform Integration Test Cases - Cart and Voucher

No	Test Title	Test Step	Expected Result	Status
1	Compute discounted price	1. Ensure the shopping cart contains items to be checked out 2. Enter the checkout page. 3. Apply valid voucher.	Display a list of items added to the cart and the discount amount on the order summary	Pass

c. Cart, Item and Shipping Fee

Table 7.55: E-commerce Platform Integration Test Cases - Cart, Item and Shipping Fee

No	Test Title	Test Step	Expected Result	Status
1	Compute shipping fee	1. Ensure the shopping cart contains items to be checked out 2. Enter the checkout page. 3. Fill out the shipping address form. 4. Click "Save" button	Display a list of items added to the cart and the shipping fee on the order summary	Pass
2	Compute shipping fee (registered user with default address)	1. Ensure the shopping cart contains items to be checked out 2. Enter the checkout page. 3. Click "Save" button on the shipment information	Display a list of items added to the cart and the shipping fee on the order summary	Pass

d. Cart, Item and Order

Table 7.56: E-commerce Platform Integration Test Cases - Cart, Item and Shipping Fee

No	Test Title	Test Step	Expected Result	Status
1	Add Order Item	1. Ensure the shopping cart contains items to be checked out 2. Enter the checkout page. 3. Fill out the shipping address form. 4. Select payment method. 5. Click the "Place Order" button	Create a new order that contains the checked out cart items	Pass

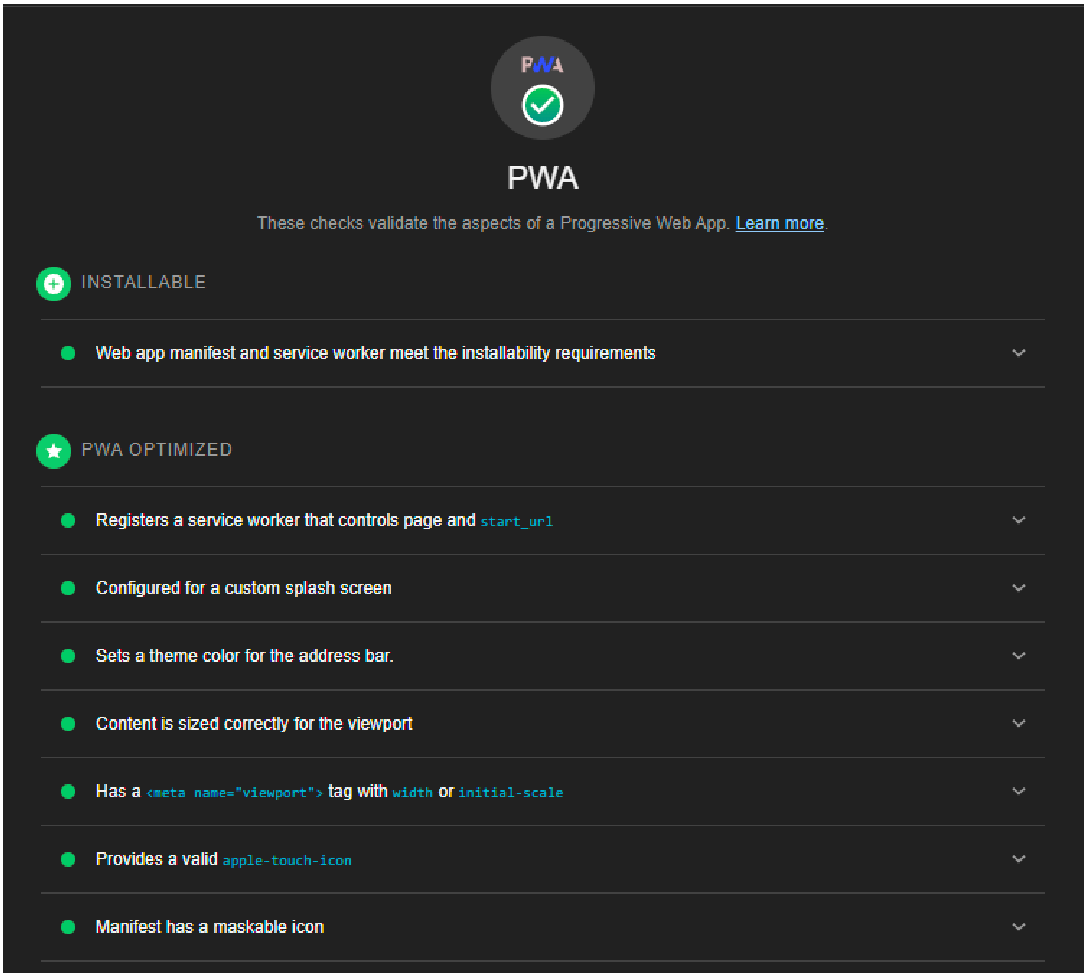
e. Cart, Item, Order, Voucher, Shipping Fee and Payment

Table 7.57: E-commerce Platform Integration Test Cases - Cart, Item, Order, Voucher, Shipping Fee and Payment

No	Test Title	Test Step	Expected Result	Status
1	Compute the total price of the order	1. Ensure the shopping cart contains items to be checked out 2. Enter the checkout page. 3. Fill out the shipping address form. 4. Apply voucher. 5. Select payment method. 6. Click "Place Order" button	Create an order with the correct total price of the order.	Pass
2	Order Payment	1. Ensure the shopping cart contains items to be checked out 2. Enter the checkout page. 3. Fill out the shipping address form. 4. Apply voucher. 5. Select payment method. 6. Click "Place Order" button 7. Redirect to payment page 8. Enter payment details 9. Click "Pay" button	The payment is successful, and the order status is updated to "To Ship" or "To Pickup". Order confirmation email received.	Pass

7.9 Lighthouse Testing

Lighthouse was used to test and analyse the PWA system. In addition, it also provides a set of metrics to guide the developers in building a PWA system. As shown in Figure 7.1, the E-commerce PWA has fulfilled all the audits from Lighthouse.



PWA

These checks validate the aspects of a Progressive Web App. Learn more.

INSTALLABLE

- ● Web app manifest and service worker meet the installability requirements

PWA OPTIMIZED

- ● Registers a service worker that controls page and `start_url`
- ● Configured for a custom splash screen
- ● Sets a theme color for the address bar.
- ● Content is sized correctly for the viewport
- ● Has a `<meta name="viewport">` tag with `width` or `initial-scale`
- ● Provides a valid `apple-touch-icon`
- ● Manifest has a maskable icon

Figure 7.1: PWA Test by Lighthouse

7.10 System Usability Testing

There are five participants invited to conduct the system usability testing. The purpose of performing the system usability testing is to evaluate the ease of use of the systems. Besides, it helps to improve the overall user experience by revealing the areas of confusion when using the system.

Each participant conducted the system usability testing along with user acceptance testing. Besides, the system usability testing was held remotely using Google Meet. During the testing, the participants were briefed on the nature of the inventory management system and the e-commerce platform developed. Besides, the participants also received the informed consent form and test scenarios, which contained a series of tasks to be performed. The participants were first asked to sign the consent form agreeing to participate in the system usability testing. Once the participants had signed the consent form, they then started performing the test scenarios. At the same time, their behaviours and interactions with the systems were observed. After the participants completed all the tests, the user satisfaction survey forms were sent to the participants to fill up. All the informed consent forms are in **Appendix C**.

7.10.1 Test Scenario

All the participants were requested to complete the test scenarios during the system usability testing. Table 7.58 and Table 7.59 show the test scenarios provided to the participants during the system usability testing.

Table 7.58: System Usability Test Scenarios for E-commerce platform

E-commerce Platform
Scenario 1 – Register and Login
Imagine you are a frequent buyer of Sharifah Food, and you wish to register an account on the Sharifah Food E-commerce platform to place an order through the platform. You are required to register yourself as a new user and log in to the application. What would you do to create a new account and log in with the account?
Scenario 2 – Edit Profile
You wish to view and edit your account information. What would you do to change your name, gender, and birthdate?
Scenario 3 – Add Address
You are likely to use the same address repeatedly to place an order, and you would like to save your address in advance so that you do not have to enter it when placing orders. What would you do?
Scenario 4 – Register as an Agent or Dropshipper
You wish to become an agent or dropshipper for Sharifah Food. However, you would require to fill out the registration form on the website to apply for an agent or dropshipper. What would you do?
Scenario 5 – Browse and View Products and Packages, Add to Shopping Cart
As you have become an agent or a dropshipper of Sharifah Food, you want to add stocks so that you can resell the goods to your customers. What would you do to add the following products and packages to your cart?

Item Type	**Item Name**	**Quantity**
Product	Nasi Briyani Bukhari	15
Product	Tepung Goreng Ayam	12
Product	Nasi Goreng Udang	6
Product	Pes Masakan Tiga Rasa	6
Package	Promo Raya Qurban	8

Table 7.58 (Continued)

Scenario 6 – Checkout
After adding the items to your cart, you would like to place an order. What would you do? *For payment, please enter the following details:

Payment Method	**Details**
Debit / Credit Card	Card Number: 4242 4242 4242 4242 Expiry Date: 12/30 CVC: 123 Name: UAT Country: Malaysia
Online Banking (FPX)	FPX Bank: (any) Name: UAT After that, click "Authorize Payment"

Scenario 7 – Track Order
After completing the payment, you wish to track and check your order. What would you do?

Table 7.58 (Continued)

Scenario 8 – Install Progressive Web Application (PWA) in Mobile

As a frequent customer of Sharifah Food, it would be troublesome to use a laptop or computer to access the website to place an order.

To install the platform on your mobile, you are required to follow the steps below:

1) On your mobile, open your browser.
2) Go to the following URL:
   https://fyp-shrf-ecommerce.herokuapp.com/
3) Did you see the popup saying ‘Add Sharifah Food to Home Screen’?
   If yes, click the popup and install the app.
   If no, go to the menu of your browser and search for “Install App” or “Add to Home Screen”.
4) After the installation, please screenshot the app on your mobile’s home screen.
5) Enter the app on your mobile’s home screen and verify if it is working.

Table 7.59: System Usability Test Scenarios for Inventory Management System

Inventory Management System

Scenario 1 – Login

Imagine you are an admin of Sharifah Food, and you wish to log in to the Sharifah Food Inventory Management System. What would you do?

The login details are as follows:

Username	uattester
Password	Testing@123

Table 7.59 (Continued)

Scenario 2 – Add Product

Sharifah Food has introduced a new product. As an admin, you are required to add the product into the system so that it can be available on the e-commerce platform for sales. What would you do?

The details of the product are as follows:

Product Name	Pes Sambal Tumis
Product Category	Paste
Product Thumbnail	https://tinyurl.com/ycybsmx6 (Please download the image first, and upload it to the website)
Product Description	Malaysian style sambal paste
Price	4.20
Cost per Unit	1.50
Ordering/Reorder Cost	85.00
Carrying/Holding Cost	0.18
Stock Keeping Unit (SKU)	SHRF-PES-PST
Stock Quantity	100
Average Lead Time	7
Maximum Lead Time	10
Weight (g)	120
Length (cm)	18
Width (cm)	3
Height (cm)	16

After adding the product, how do you view the product details?

Scenario 3 – Update Product Status

Once you have added the product, the company notifies you to hide the product you just added. What would you do?

Scenario 4 – Delete Product

If you wish to delete the product named **Nasi Lemuni**, what would you do?

Table 7.59 (Continued)

Scenario 5 – Manage Product Inventory

Sharifah Food is currently restocking some of its products, which should be reflected in the system. Hence, you are required to add the stock for the following products:

Product	Stock
Kari Ayam	+260
Nasi Goreng Ikan Masin	+120
Nasi Briyani Tomato	+60

Scenario 6 – Add Packages

Sharifah Food would like to add a new package launched in May 2022. As an admin, you are required to add the following packages. What would you do?

Package Name	Combo Nak Rasa
Package Thumbnail	https://tinyurl.com/mr2nj97s (Please download the image first, and upload to the website)
Package Description	Taste Test Package that includes various Malaysian Food.
Products	1. 1 x Nasi Briyani Gam Ayam 2. 1 x Rendang Dendeng Daging 3. 1 x Sambal Tumis Ikan Bilis 4. 1 x Sambal Sotong Kering 5. 1 x Kari Ayam
Price	54.30
Special Price	50.00
Stock Keeping Unit (SKU)	SHRF-PACK-CNR
Stock Quantity	60
Weight (g)	120
Length (cm)	18
Width (cm)	3
Height (cm)	16
Available Start Time	01 May 2022
Available End Time	01 June 2022

After adding the package, how do you view the package details?

Table 7.59 (Continued)

<table>
<tr><th>Scenario 7 – Manage Package Inventory</th></tr>
<tr><td>Sharifah Food wants to reduce the package stock, which should be reflected in the system. Hence, you are required to reduce the stock for the following packages:
<table>
<tr><th>Package</th><th>Stock</th></tr>
<tr><td>Promo Raya Qurban</td><td>-6</td></tr>
</table>
</td></tr>
<tr><th>Scenario 8 – Delete Package</th></tr>
<tr><td>If you wish to delete the package named <b>Combo ShaRaya 2.0</b>, what would you do?</td></tr>
<tr><th>Scenario 9 – Manage Orders</th></tr>
<tr><td><b>If the order you just placed in the e-commerce is by shipping</b>:
Imagine you have shipped the new order you just placed and wish to update the status and tracking number of your order. What would you do?
<table>
<tr><th>Tracking Number</th><td>ERC707845023MY</td></tr>
</table>

Are you able to track the order with the tracking number after adding the tracking number?
<b>If the order you just placed in the e-commerce is by pickup</b>:
Imagine you have picked the new order you just placed and wish to update your order’s status. What would you do?

After that, you wish to view the order details. What should you do?</td></tr>
<tr><th>Scenario 10 – Generate Invoice</th></tr>
<tr><td>After updating the order’s status, you wish to generate the order invoice. What would you do?</td></tr>
</table>

Table 7.59 (Continued)

Scenario 11 – Add Shipping Fee

What would you do if you wish to add the following shipping fees in Johor?

Weight Range (g)	**Shipping Fee**
0 - 1000	RM 10.28
1001 - 2000	RM 11.34
2001 - 3000	RM 12.40
3001 - 4000	RM 13.46
4001 - 5000	RM 14.52

Subsequence weight = 1000g

Subsequence fee = RM 1.00

Scenario 12 – Edit Shipping Fee

The courier has updated the shipping fee for Johor, you are informed to reflect the shipping fee in the system. What would you do?

The details of the shipping fees are as follows:

Weight Range (g)	**Shipping Fee**
0 - 1000	RM 10.80
1001 - 2000	RM 12.00
2001 - 3000	RM 13.20
3001 - 4000	RM 13.64
4001 - 5000	RM 14.80

Scenario 13 – Delete Shipping Fee

You wish to delete the shipping fee in Johor with weights ranging from 1001 to 2000. What would you do?

Table 7.59 (Continued)

<table>
<tr><th><b>Scenario 14 – Add Pickup Location</b></th></tr>
<tr><td>You wish to add a new pickup location named <b>LG-099A, Mid Belley Megamall</b>.</td></tr>
<tr><td><b>Scenario 15 – Edit Pickup Location</b></td></tr>
<tr><td>There is a typo for the pickup location that was just added, and you are required to correct the typo by editing the pickup location. What would you do?
The correct pickup location is <b>LG-099A, Mid Valley Megamall</b>.</td></tr>
<tr><td><b>Scenario 16 – Delete Pickup Location</b></td></tr>
<tr><td>You wish to delete the pickup location named <b>L1-35, The Mines Shopping Mall</b>. What would you do?</td></tr>
<tr><td><b>Scenario 17 – Add Customer</b></td></tr>
<tr><td>You wish to add the following customer as dropshipper. What would you do?
<table>
<tr><td><b>Customer Name</b></td><td>Kevin Thum Thian Liat</td></tr>
<tr><td><b>Gender</b></td><td>Male</td></tr>
<tr><td><b>Birthdate</b></td><td>1998-03-25</td></tr>
<tr><td><b>Marital Status</b></td><td>Single</td></tr>
<tr><td><b>Phone Number</b></td><td>0143256785</td></tr>
<tr><td><b>Email Address</b></td><td>chewyyoda8629@gmail.com</td></tr>
<tr><td><b>State</b></td><td>Wilayah Persekutuan Kuala Lumpur</td></tr>
<tr><td><b>City</b></td><td>Kuala Lumpur</td></tr>
<tr><td><b>Postal Code</b></td><td>56000</td></tr>
<tr><td><b>Address</b></td><td>30 Jalan Angsana 1 Taman Bukit Angsana Cheras</td></tr>
<tr><td><b>Current Occupation</b></td><td>Student</td></tr>
<tr><td><b>Customer’s Position</b></td><td>Dropshipper</td></tr>
<tr><td><b>Status</b></td><td>Active</td></tr>
</table>
After that, you wish to view the customer details of Leah Mcgrath. What should you do?</td></tr>
</table>

Table 7.59 (Continued)

Scenario 18 – Accept Customer Registration
After deep consideration, as an admin, you are willing to accept the registration of an applicant named **Tan Yuan Jie**, who applied for an agent position. What would you do?
Scenario 19 – Reject Customer Registration
After deep consideration, as an admin, you want to reject the registration of an applicant named **Steven Tan**, who applied for a dropshipper position. What would you do?
Scenario 20 – Suspend Customer
The agent, **Tan Yuan Jie,** currently has a poor recent sales performance. If you wish to **suspend** his agent position, what would you do?
Scenario 21 – Activate Customer
You would like to give **Tan Yuan Jie** another chance. Hence, if you wish to **activate** his account back, what would you do?

Table 7.59 (Continued)

Scenario 22 – Add Voucher

Mother’s Day is coming soon. Sharifah Food would like to offer a discount for Mother’s Day. As an admin, you are instructed to **add a new voucher** so that the customer can apply during checkout on the e-commerce platform. What would you do?

The details of the voucher are as follows:

Discount Code	motherday2022
Discount Status	Hidden
Discount Type	Fixed Amount
Discount Details	If Order Amount reaches **RM 100.00** Discount would be **RM 15.00**
Total Voucher to be Issued	300
Usage Limit per User	1
Applicable To	Direct Customer
Available Start Time	Today

Scenario 23 – Edit Voucher

After the voucher is added, change the voucher discount details to the following:

Discount Details	If Order Amount reaches **RM 100.00** Discount would be **RM 20.00**

Scenario 24 – Delete Voucher

You wish to delete the voucher with code ‘**promoraya2022**’. What would you do?

Table 7.59 (Continued)

Scenario 25 – Business Insights and Generate Report
It’s been almost half a year, and Sharifah Food wants to hold a meeting to boost the company’s sales. As an admin, you are required to view the **sales analysis (Key Metrics)** and generate a report of the sales for **This Year (2022)**. What would you do?
Scenario 26 – Inventory Analysis
You wish to know the **Safety Stock (SS)** analysis in **January 2022**. What would you do?

7.10.2 System Usability Testing Result

Based on the results listed in **Appendix E**, the System Usability Scale (SUS) can be computed to obtain the usability score for the inventory management system and e-commerce platform.

According to Sauro (2011), there are four steps to calculate the SUS score. The four steps are as follows:

1. Subtract 1 for the odd-numbered user response items.
2. Subtract 5 for the even-numbered user response items.
3. All the values now are only scaled from 0 to 4, in which 4 is the most satisfactory response.
4. Sum up the converted values for each user response item and multiply the total by 2.5, to convert the range of results from 0 – 40 to 0 – 100.

Besides, Alathas (2018) proposed the general guideline on the SUS score interpretation, as shown in Table 7.60. The project will apply the SUS score interpretation to rate the systems developed.

Table 7.60: SUS Score Interpretation (Alathas, 2018)

SUS Score	Grade	Adjective Rating
> 80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
< 51	E	Awful

7.10.2.1 SUS Score for Inventory Management System

Table 7.61: SUS Score for Inventory Management System

Participants	Usability Score per Questions										Total	Percentage
	1	2	3	4	5	6	7	8	9	10		
1	3	3	3	3	4	4	3	4	3	3	33	82.5%
2	4	3	4	4	4	4	4	4	4	4	39	97.5%
3	3	3	3	2	3	3	3	4	3	3	30	75%
4	4	4	4	4	4	4	3	4	3	4	38	95%
5	4	4	3	3	4	3	2	3	4	3	33	82.5%
Average SUS Score												**86.5%**
Grade												**A**
Adjective Rating												**Excellent**

7.10.2.2 SUS Score for E-commerce Platform

Table 7.62: SUS Score for E-commerce Platform

Participants	Usability Score per Questions										Total	Percentage
	1	2	3	4	5	6	7	8	9	10		
1	3	3	4	4	3	3	4	4	4	4	36	90%
2	4	4	4	4	4	4	4	4	4	4	40	100%
3	4	4	4	3	3	3	4	4	3	4	36	90%
4	4	4	4	4	3	4	4	4	4	4	39	97.5%
5	4	4	4	4	4	3	4	4	4	4	39	97.5%
Average SUS Score												**95%**
Grade												**A**
Adjective Rating												**Excellent**

7.11 User Acceptance Testing

User Acceptance Testing (UAT) is the last stage of testing that is performed by the client or end-user to verify the systems before the system moves to the production environment. In this project, there were 5 participants involved in the UAT for both the inventory management system and e-commerce platform. The UAT was conducted remotely through Google Meet. In addition, the UAT was performed cooperatively with system usability testing. The results of UAT are in **Appendix F**.

7.11.1 UAT Test Cases

Table 7.63: UAT Test Cases Template

Testing Date			
Testing Start Time		Testing End Time	
Tester Name			

E-commerce Platform			
Test Module	**Test Scenario**	**Status**	**Comments**
Register	1. Able to create an account with a new email.		
Login	1. Able to log in with the created account.		
Item	1. Able to browse items.		
	2. Able to search items.		
	3. Able to browse products.		
	4. Able to browse packages.		
	5. Able to view item details.		
Cart	1. Able to add an item to cart.		
	2. Able to view shopping cart.		
	3. Able to add quantity to the item in cart.		
	4. Able to reduce quantity to the item in cart.		

Table 7.63 (Continued)

	5. Able to set the quantity to the item in cart.		
	6. Able to remove an item in cart.		
	7. Able to checkout a cart with items.		
Address	1. Able to add new address in the address book.		
	2. Able to view address list in the address book.		
	3. Able to edit an address in the address book.		
	4. Able to delete an address in the address book.		
Pickup Location	1. Able to view and select pickup locations.		
Shipping Fee	1. Able to get shipping fee for shipping address,		
Order	1. Able to select shipping or pickup.		
	2. Able to place order after filling shipping information and payment method.		
	3. Able to view order history.		
	4. Able to search orders.		
	5. Able to view order details.		
Payment	1. Able to select payment method.		
	2. Able to be redirected to payment page.		
	3. Able to make payment.		
Agent and Dropshipper Registration	1. Able to submit registration form.		
	2. Able to get popup notification after submitting form.		
Voucher	1. Able to automatically apply voucher.		
	2. Able to remove voucher.		

Table 7.63 (Continued)

	3. Able to apply voucher.		
Customer	1. Able to view account information.		
	2. Able to edit account information.		
Inventory Management System			
Test Module	**Test Scenario**	**Status**	**Comments**
Login	1. Able to log in with the admin account.		
Product	1. Able to add product.		
	2. Able to view product list.		
	3. Able to edit product.		
	4. Able to hide product.		
	5. Able to delete product.		
	6. Able to update product inventory stock.		
	7. Able to filter product.		
	8. Able to sort product.		
Package	1. Able to add package.		
	2. Able to add products into package.		
	3. Able to view package list.		
	4. Able to edit package.		
	5. Able to hide package.		
	6. Able to delete package.		
	7. Able to update package inventory stock.		
	8. Able to filter package.		
	9. Able to sort package.		
Order	1. Able to view a list of orders.		
	2. Able to filter orders.		
	3. Able to sort orders.		
	4. Able to view order details.		

Table 7.63 (Continued)

	5. Able to update tracking number.		
	6. Able to update pickup for order.		
	7. Able to cancel order.		
	8. Able to generate order invoice.		
	9. Able to view the order tracking.		
Shipping Fee	1. Able to view a list of shipping fees.		
	2. Able to filter shipping fees.		
	3. Able to sort shipping fees.		
	4. Able to view shipping fees details.		
	5. Able to update shipping fees.		
	6. Able to delete shipping fees.		
Pickup Location	1. Able to view a list of pickup locations.		
	2. Able to filter pickup locations.		
	3. Able to sort pickup locations.		
	4. Able to view pickup location details.		
	5. Able to update pickup location.		
	6. Able to delete pickup location.		
Customer	1. Able to view a list of customers.		
	2. Able to filter customers.		
	3. Able to sort customers.		
	4. Able to view customer details.		
	5. Able to suspend customer.		
	6. Able to activate customer.		

Table 7.63 (Continued)

Agent and Dropshipper Registration	1. Able to view a list of registrations.		
	2. Able to filter registrations.		
	3. Able to sort registrations.		
	5. Able to accept registration.		
	6. Able to reject registration.		
	7. Able to add new agent or dropshipper.		
Voucher	1. Able to add voucher.		
	2. Able to view voucher list.		
	3. Able to edit voucher.		
	4. Able to hide voucher.		
	5. Able to delete voucher.		
	8. Able to filter vocher.		
	9. Able to sort vocher.		
Analysis	1. Able to view product rankings.		
	2. Able to view package rankings.		
	3. Able to view statistics summary.		
	4. Able to view key metrics analysis.		
	5. Able to generate sales report.		
	6. Able to export chart.		
	7. Able to view ABC analysis.		
	8. Able to view HML analysis.		
	9. Able to view EOQ analysis.		
	10. Able to view SS analysis.		

7.11.2 Improvement From UAT

After performing UAT, some confusion and suggestions were received from the participants. There are three issues found during the UAT. The three issues are listed as follows:

1. The e-commerce platform users cannot set quantity when adding the item to the shopping cart on the item details page.
2. The search dropdown input component in the inventory management system is not obvious to let the admins know that they can select different fields to search.
3. The table column name of ‘Action’ on the product and package inventory management page is ambiguous, and the user does not know the inputs used to update inventory stock.

In addition to the issues found, some rectifications have made to the UI of the inventory management system and the e-commerce platform. The issue of the search dropdown input component in the inventory management system was rectified by making the search field bold and enhancing the placeholder to indicate the search field. Figure 7.2 and Figure 7.3 show the before and after rectification of the search dropdown input issue.

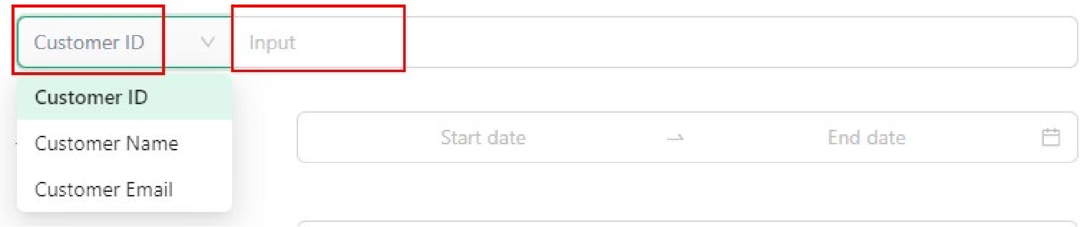


Figure 7.2: Search Dropdown Input Before Rectification

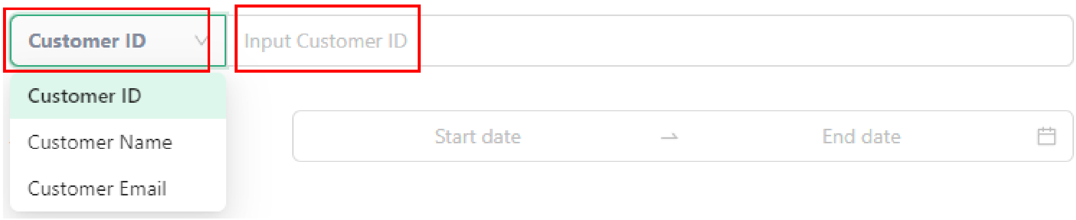


Figure 7.3: Search Dropdown Input After Rectification

In terms of the issue of ambiguous table column “Action”, the solution is to change the column name to “Update Inventory Stock”. Figure 7.4 and Figure 7.5 show the before and after rectification of the ambiguous “Action” column name.

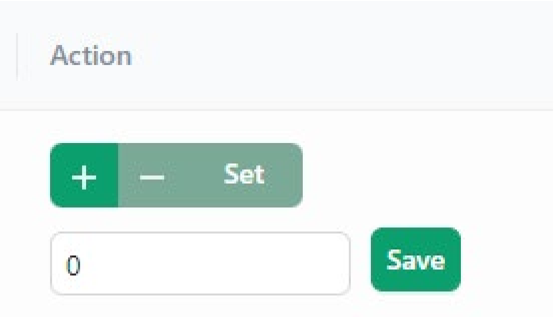


Figure 7.4: Inventory Management “Action” column Before Rectification

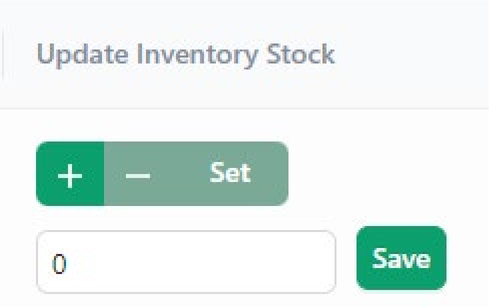


Figure 7.5: Inventory Management “Action” column After Rectification

In terms of the issue of adding quantity input for the add to cart feature on item details, it will become future work as it requires modification on both the back-end server and front-end application.

7.12 Summary

All the tests, including unit, integration, system usability, and user acceptance, have been performed. There was a total of 5 participants involved in the system usability testing and user acceptance testing. Additionally, all the participants signed the consent form before the implementation. The SUS scores for both the inventory management system and e-commerce platform have also been computed, which are 86.5% and 95%, respectively. Besides, all the unit, integration, and user acceptance tests have passed. The PWA was also tested with Lighthouse, and it fulfills all Lighthouse audits. Since the exit criteria and the project's objective have been met, the system testing life cycle of the project is marked as completed.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

The inventory management system and e-commerce platform have been successfully developed, tested and deployed in accordance with the requirements, marking the completion of the project. Besides, the project has also successfully attained all the objectives stated, which include the following:

a. To analyse and determine the most appropriate software development tools and methodology in developing the inventory management system that runs on both web and mobile platforms by 30 July 2021.
b. To design a prototype of a web-based computerised food inventory management system integrated with an e-commerce platform by 22 August 2021.
c. To develop a web and mobile computerised food inventory management system integrated with an e-commerce platform by April 2022.
d. To evaluate the system's functionalities through performing unit, integration, system usability, and user acceptance testing by achieving a 95 percent pass rate on all tests before May 2022.

The inventory management system is believed to solve the problems encountered by Sharifah Food, especially when it comes to sales and inventory management. The staff does not require to monitor the inventory manually, avoiding the occurrence of inaccurate inventory records. Besides, by integrating with the e-commerce platform, the inventory management system can have further order and customer management controls. For instance, the inventory management system can track each transaction that occurred in the e-commerce system and notify the admin of to-do lists. Additionally, with the system's sales and inventory analysis feature, the admin can view up to 8 sales key metrics and 4 inventory analyses with a single click. Furthermore, the admin can have better control over the Multi-level Marketing system. The admin can accept or reject the admin or dropshipper registrations, or even suspend the existing agent or dropshipper, so that they cannot access the e-commerce platform with the agent or dropshipper discounts.

8.2 Limitations

Although the project has been successfully completed, there are several limitations discovered during the entire system implementation phase. The limitations of the inventory management system include the following:

a. **Lack of role-based access control on the admin**

   The current inventory management system does not allow any user to register as an admin. Besides, there is no way for the super admin to manage the admins.

b. **The absence of automated order tracking feature**

   The automated order tracking feature allows a real-time update on the order status. However, the current inventory management system does not support the automated order tracking feature and will only update the order status to "completed" 14 days after the order has been shipped.

On the other hand, the limitations discovered in the e-commerce platform are as follows:

a. **Missing reviews and comments**

   Most users would like to read reviews and comments on the products they are interested in before purchasing online. Nonetheless, the e-commerce platform does not include the reviews and comments feature.

b. **Lack of push notification**

   One of the benefits of building a PWA application is that it supports push notifications. Besides, the company can share promotions and update the order status to the customers through push notifications. However, the current e-commerce platform only supports email notifications.

c. **Offline Access**

   The project has tested the offline capability of the e-commerce PWA platforms. Nonetheless, it was found that the application can only work offline if the user does not refresh the application after offline. Once the user refreshes the application when offline, the application will show unavailable message.

8.3 Future Work

Despite the fact that the systems fulfil all the specifications and core features, there is still room for enhancement. The enhancements that can be covered for the inventory management system in the future work are as follows:

a. **Implementation of Admin Management and Authorization**

   The authorization allows the company to specify different admin levels with different permission in accessing the system. This allows the company to have more controls, especially in handling sensitive data.

b. **Implementation of Automated Order Shipment Tracking System**

   The automated order shipment tracking system allows the system to report the latest order status in real-time, reducing the possibility of fraudulent order cancellations or refunds.

In terms of the e-commerce platform, the enhancements that can be covered in the future work include the following:

a. **Implementation of reviews and comments feature**

   By developing the reviews and comments feature, the customers can rate and comment on the goods they purchased. This could probably help the customers to make informed and more confident purchases from the company, thereby increasing the company's trustworthiness.

b. **Allow better mobile experiences**

   The future work enhancement can focus on the enhancement of the PWA application. Since the project only includes a fundamental PWA configuration, it can be enhanced in caching, push notification and offline experiences.

c. **Enhancement on add to cart feature**

   As identified in the system usability testing and UAT, major users are concerned about the ability to set the quantity on the item details page before adding to the cart. Besides, it was also found that some users take a long time figuring out adding the cart item quantity. Therefore, resolving these issues will definitely improve the user experience during the order purchase process.

REFERENCES

Abbas, Q., Shafiq, H., Ahmad, I. & Tharanidharan, S., 2016. *Concurrency Control in Distributed Database System.* Coimbatore, Amaze college of Animation and Technology.

Abisoye, O. A., Boboye, F. & Abisoye, B. O., 2013. Design of a Computerized Inventory Management System for Supermarkets. *International Journal of Science and Research (IJSR),* 2(9), p. 340.

Adhena, T. H., 2020. Design of Computerized Inventory Management System the Case of MPC. *Global Scientific Journals,* 8(10), pp. 2030 - 2062.

Afolabi, O. J., Onifade, M. K. & Odeyinka, F. O., 2017. Evaluation of the Role of Inventory Management in Logistics Chain of an Organisation. *LOGI – Scientific Journal on Transport and Logistics,* 8(2), pp. 1 - 11.

Alathas, H., 2018. *How to Measure Product Usability with the System Usability Scale (SUS) Score.* [Online]
Available at: https://uxplanet.org/how-to-measure-product-usability-with-the-system-usability-scale-sus-score-69f3875b858f
[Accessed 19 April 2022].

Amron, M. T., Hassan, S. b. A., Hudin, M. S. S. & Janom, N., 2018. Inventory Control System Using Waterfall Development Model. *International Journal of Engineering & Technology,* 7(4), pp. 138 - 141.

Anon., 2011. *Design and Implementation of an Automated Inventory Management System,* Nkozi: Uganda Martyrs University.

Anulika, N. C., Idoko , N. A., Chukwuwike, A. J. & Emeka, O. N., 2020. Design and Optimization of An Inventory Management System for Central Stores. *International Journal of Interdisciplinary Innovative Research &Development (IJIIRD),* 5(1), pp. 119 - 125.

Ara, R. & Rahim, A., 2018. AN ONLINE BASED INVENTORY MANAGEMENT SYSTEM IMPLEMENTATION IN PRINTING BUSINESS. *Journal of Emerging Technologies and Innovative Research,* 5(11), pp. 176 - 179.

Bales, A. B., Fox, R. & VanNevel, M. R., 2017. Managing Library Software Development: A Case Study In Developing An Inventory Management System For Off-Site Storage Using An Outside Contractor. *Library Leadership & Management,* 31(4), pp. 1 - 19.

Barjtya, S., Sharma, A. & Rani, U., 2017. A detailed study of Software Development Life Cycle (SDLC) Models. *International Journal Of Engineering And Computer Science,* 6(7), pp. 22097 - 22100.

Batra, N. & Kapil, A. K., 2010. *Concurrency Control Algorithms and its Variants: A Survey.* Mullana, M. M. Engineering College.

Bermudez, R. S. & Cantos, M. C. G., 2012. Development and Validation of Flora Inventory Management System. *MSEUF Research Studies,* 14(1), pp. 4 - 13.

Biswas, S. K. et al., 2017. Analysis of Different Inventory Control Techniques: A Case Study in a Retail Shop. *Journal of Supply Chain Management System,* 6(3), pp. 35 - 45.

Chan, S. W. et al., 2017. *Factors Influencing the Effectiveness of Inventory Management in Manufacturing SMEs.* s.l., IOP Conference Series: Materials Science and Engineering, pp. 1 - 8.

Choudhary, S. K. & Gupta, N., 2014. Development of the Inventory Management SYstem for ERP Implementation in Manufacturing Industry. *IOSR Journal of Mechanical and Civil Engineering,* 11(6), pp. 19 - 29.

Christopher, G. B. & Kabari, L. G., 2020. Hybridzed concurrency control technique for transaction processing in distributed database system. *International Journal of Computer Science and Mobile Computing,* 9(9), pp. 118 - 127.

Chuang, H. H.-C. & Oliva, R., 2015. Inventory record inaccuracy: Causes and labor effects. *Journal of Operations Management,* 39 - 40(1), pp. 63 - 78.

Clarke, M., 2021. *How to calculate safety stock and reorder point.* [Online] Available at: https://practicaldatascience.co.uk/data-science/how-to-calculate-safety-stock-and-reorder-point
[Accessed 11 August 2021].

Cruz, G. M. T. D., Dumaplin, T. B., Gaerlan, P. V. T. & Junio, G. O. M., 2019. Inventory Management System with POS. *International Journal of AdvancedResearch in Science, Engineering and Technology,* 6(4), pp. 8739 - 8745.

Dahiwale, V. & Sangode, P. B., 2019. A COMPARATIVE STUDY OF THE INVENTORY MANAGEMENT TOOLS OF TEXTILE MANUFACTURING FIRMS. *International Journal of Research in Humanities, Arts and Literature,* 7(4), pp. 335 - 344.

Dennis, A., Wixom, B. H. & Tegarden, D., 2015. *Systems analysis and design: An object-oriented approach with UML.* 5 ed. New York: John Wiley & Sons, Inc..

Dora, S. K. & Dubey, P., 2013. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) ANALYTICAL COMPARISON AND SURVEY ON TRADITIONAL AND AGILE METHODOLOGY. *National Monthly Refereed Journal of Research in Science & Technology,* 2(8), pp. 22 - 30.

Ecwid, 2021. *Start Selling on a Website.* [Online] Available at: https://www.ecwid.com/
[Accessed 3 July 2021].

Egwoh, A. Y. & Nonyelum, O. F., 2017. A SOFTWARE SYSTEM DEVELOPMENT LIFE CYCLE MODEL FOR IMPROVED STUDENTS' COMMUNICATION AND COLLABORATION. *International Journal of Computer Science & Engineering Survey,* 8(4), pp. 1 - 10.

Eveline, C. et al., 2019. Effects of Inventory Management Techniques on Procurement Performance: An Empirical Study. *International Journal of Innovative Research and Development,* 8(8), pp. 141 - 150.

Gajalakshmi, P., 2016. SOFTWARE DEVELOPMENT LIFECYCLE MODEL (SDLC) INCORPORATED WITH RELEASE MANAGEMENT. *International Research Journal of Engineering and Technology,* 3(4), pp. 1536 - 1543.

Ghuman, S. S., 2016. Concurrency Control in DBMS- A Review. *International Journal of Computer Science and Mobile Computing,* 5(5), pp. 599 - 602.

Gohil, J. A. & Dolia, P. M., 2016. Study and Comparative Analysis of Basic Pessimistic and Optimistic Concurrency Control Methods for Database Management System. *International Journal of Advanced Research in Computer and Communication Engineering,* 5(1), pp. 178 - 186.

Gupta, M. K., Arora, R. K. & Bhati, B. S., 2018. Study of Concurrency Control Techniques in Distributed DBMS. *International Journal of Machine Learning and Networked Collaborative Engineering,* 2(4), pp. 180 - 187.

Hamadi, N. B., 2018. Nfra Web Based Inventory, Purchase and Sales Control System. *American Journal of Information Science and Technology,* 2(2), pp. 24 - 35.

Ilias, J. et al., 2018. AT-THOYYIB SHOP INVENTORY MANAGEMENT SYSTEM. *Acta Informatica Malaysia (AIM),* 2(2), pp. 12 - 16.

Imeokparia, L., 2013. Inventory Management System and Performance of Food and Beverages Companies in Nigeria. *IOSR Journal of Mathematics,* 6(1), pp. 24 - 30.

Irene, N., 2011. *DESIGN AND IMPLEMENTATION OF AN AUTOMATEDINVENTORY MANAGEMENT SYSTEM,* Nkozi: Uganda Martyrs University.

Jadhav, P. & Jaybhaye, M., 2020. A Manufacturing Industry Case Study: ABC and HML Analysis for Inventory Management. *International Journal of Research in Engineering, Science and Management,* 3(9), pp. 146 - 149.

James Ng'ang'a, K., 2013. An Assessment of the Factors Influencing Effectiveness of Inventory Control; Ministry of State for Provincial Administration and Internal Security, Nairobi - Kenya. *International Journal of Business and Commerce,* 3(1), pp. 33 - 53.

Kaur, K., Salome, S. & Muthiah, . S., 2016. HARNESSING THE POWER OF MOBILE TECHNOLOGY: A LOOK AT MALAYSIAN MOBILE COMMERCE LANDSCAPE. *Infrastructure University Kuala Lumpur Research Journal,* 4(1), pp. 41 - 46.

Khabbazi, M. R. et al., 2013. Inventory System and Functionality Evaluation for Production Logistics. *Journal of Theoretical and Applied Information Technology,* 54(3), pp. 377 - 387.

Khobragade, P., Selokar, . R., Maraskolhe, R. & Talmale, M., 2018. Research paper on Inventory management system. *International Research Journal of Engineering and Technology (IRJET),* 5(4), pp. 252 - 254.

Kumar, Y. et al., 2016. HML Analysis for Inventory Management-Case Study of Steel Plant. *International Journal for Research in Applied Science & Engineering Technology,* 4(3), pp. 521 - 526.

Laar, D. S., Konjaang, J. K. & Tankia, B. A., 2015. Design and Development of a Sales Management System for SMEs in Northern Ghana. *International Journal of Innovative Research in Advanced Engineering,* 5(2), pp. 66 - 77.

Lirios, J. R. S., Liwag, M. A. F., Pedrosa, N. C. & Ng, G., 2018. A Problem Oriented Approach to Implementing an Auto Sales and Inventory System through Agile Methodology. *Innovatus,* 1(1).

Liu, N. & Zhou, J. F., 2015. Relational Database's Transaction Operation and the Concurrent Control. *International Journal of Database Theory and Application,* 8(2), pp. 259 - 266.

Malleswari, N. D., Kumar, P. M., Sathvika, D. & Kumar, A. B., 2018. A Study on SDLC For Water Fall and Agile. *International Journal of Engineering & Technology,* 7(2), pp. 10 - 13.

Mathur, S. & Malik, S., 2010. Advancements in the V-Model. *International Journal of Computer Applications,* 1(12), pp. 30 - 34.

Mat, T. Z. T. & Kadir, . S. F. A., 2016. Inventory Management System in Small and Medium Enterprises in Malaysia. *International Journal of Engineering and Management Research,* 6(2), pp. 731 - 742.

Meng, M., Steinhardt, S. & Schubert, A., 2018. Application Programming Interface Documentation: What Do Software Developers Want?. *Journal of Technical Writing and Communicatio,* 48(3), pp. 295 - 330.

Müller, J., 2021. *Smartphone users in Malaysia 2015-2025.* [Online] Available at: https://www.statista.com/statistics/494587/smartphone-users-in-malaysia/
[Accessed 1 July 2021].

Munassar, N. M. A. & Govardhan, A., 2010. A Comparison Between Five Models Of Software Engineering. *IJCSI International Journal of Computer Science Issues,* 7(5), pp. 94 - 101.

Muyumba, T. & Phiri, J., 2017. A Web based Inventory Control System using Cloud Architecture and Barcode Technology for Zambia Air Forc. *International Journal of Advanced Computer Science and Applications (IJACSA),* 8(11), pp. 132 - 142.

Nadkarni, R. & Ghewari, A., 2016. An Inventory Control using ABC Analysis and FSN Analysis. *nternational Journal of Engineering, Business and Enterprise Applications,* 16(124), pp. 24 - 28.

Nasseri, M. & Jameii, S. M., 2017. *Concurrency Control Methods in Distributed Database: A Review and Comparison.* Tehran, Islamic Azad University, pp. 200 - 205.

Ngugi, E. N., Kimutai, G. & Kibet, Y., 2019. Effects of Inventory Management Systems on Performance of Manufacturing Companies in Eldoret Town, Kenya. *The Strategic Journal of Business & Change Management,* 6(2), pp. 1431 - 1445.

Odoo, 2021. *Amazing employees deserve amazing software.* [Online] Available at: https://www.odoo.com/
[Accessed 3 July 2021].

Oluwole, A., 2019. *DESIGN AND IMPLEMENTATION OF ACOMPUTERIZED STOCK MANAGEMENT SYSTEM,* s.l.: Akinsanya Oluwole.

Quasim, T., 2013. An Efficient Approach For Concurrency Control In Distributed Database System. *Indian Streams Research Journal,* 3(9).

Regulwar, G. B. et al., 2010. Variations in V Model for Software Development. *International Journal of Advanced Research in Computer Science,* 1(2), pp. 133 -138.

Salve, S. M., Samreen, S. N. & Valmik, N. K., 2018. A Comparative Study on Software Development Life Cycle Models. *International Research Journal of Engineering and Technology (IRJET),* 5(2), pp. 696 - 700.

Samuel, K. S. & Ondiek, G. O., 2014. Inventory Management Automation and The Performance of Supermarkets in Western Kenya. *International Journal of Research in Management & Business Studies,* 1(4), pp. 9 - 18.

Saqqa, S. A., Sawalha, S. & AbdelNabi, H., 2020. Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies,* 14(11), pp. 246 - 270.

Sauro, J., 2011. *Measuring Usability with the System Usability Scale (SUS).* [Online] Available at: https://measuringu.com/sus/#:~:text=What%20is%20a%20Good%20SUS,through%20a%20process%20called%20normalizing.
[Accessed 19 4 2022].

Shabani, A., Maroti, G., Leeuw, S. d. & Dullaert, W., 2021. Inventory record inaccuracy and store-level performance. *International Journal of Production Economics,* Volume 235, pp. 1 - 16.

Sharifah Food, n.d. *Sharifah.* [Online] Available at: https://www.sharifahfood.com/
[Accessed 26 June 2021].

Shinde, V., Singh, A., Wayal, K. & Vadhavinde, V., 2018. Web App Store for Inventory Management and Stock Report in Android App Using Centralized Database. *International Journal for Research in Engineering Application & Management (IJREAM),* pp. 69 - 73.

Shopify, 2021. *Build an online business—no matter what business you're in.* [Online] Available at: https://www.shopify.my/
[Accessed 3 July 2021].

Singh, S. R. & Kumar, S., 2016. An Overview of World Wide Web Protocol (Hypertext Transfer Protocol and Hypertext Transfer Protocol Secure). *International*

Journal of Advanced Research in Computer Science and Software Engineering, 6(5), pp. 396 - 399.

Solanki, R. S., 2018. An overview of Concurrency Control Techniques in Distributed Database. *International Journal for Research in Applied Science & Engineering Technology,* 6(1), pp. 1279 - 1285.

Sommerville, I., 2016. *Software Engineering.* 10 ed. Harlow: Pearson Education Limited.

Sporta, F. O., 2018. Effect of Inventory Control Techniques on Organization's Performance at Kenya Medical Supplies Agencies. *THE INTERNATIONAL JOURNAL OF BUSINESS & MANAGEMENT,* 6(3), pp. 62 - 76.

Stitch Labs, 2019. *Safety Stock, Reorder Point & Lead Time: How to Calculate With Formulas.* [Online]
Available at: https://www.stitchlabs.com/learning-center/safety-stock-reorder-point-lead-time-calculate-formulas/
[Accessed 11 August 2021].

Sutanto, A. P., 2019. *WEB BASED INVENTORY MANAGEMENT SYSTEM,* Surakarta: UNIVERSITAS MUHAMMADIYAH SURAKARTA.

Tapado, B. M. & Delluza, M. E. T., 2016. Equipment Inventory Management System (EIMS). *International Journal of Humanities and Social Sciences,* 8(2), pp. 76 - 83.

Tarika, B., 2019. A Review on Models of Software Development Life Cycle. *ITS Journal of Data Research,* Volume 3, pp. 61 - 68.

Tom, J. V., Jayakumar, A. & Sijo, M. T., 2013. Analysis of Inventory Control Techniques; A Comparative Study. *International Journal of Scientific and Research Publications,* 3(3), pp. 1 - 6.

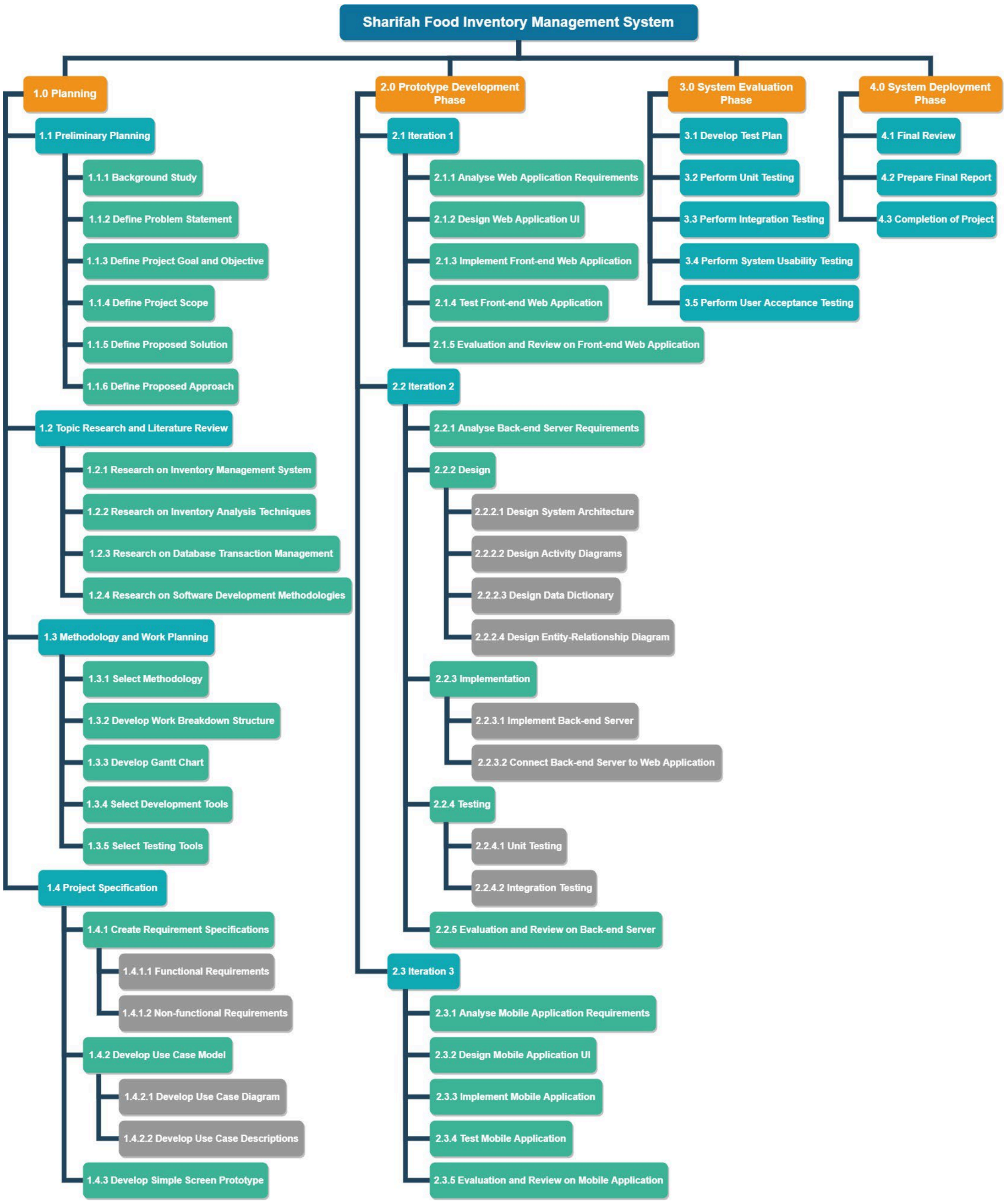
Vrat, P., 2014. Selective Inventory Management. In: *Materials Management.* New Delhi: Springer Texts in Business and Economics, pp. 37 - 49.

WooCommerce, 2021. *Build exactly the eCommerce website you want.* [Online]
Available at: https://woocommerce.com/
[Accessed 3 July 2021].

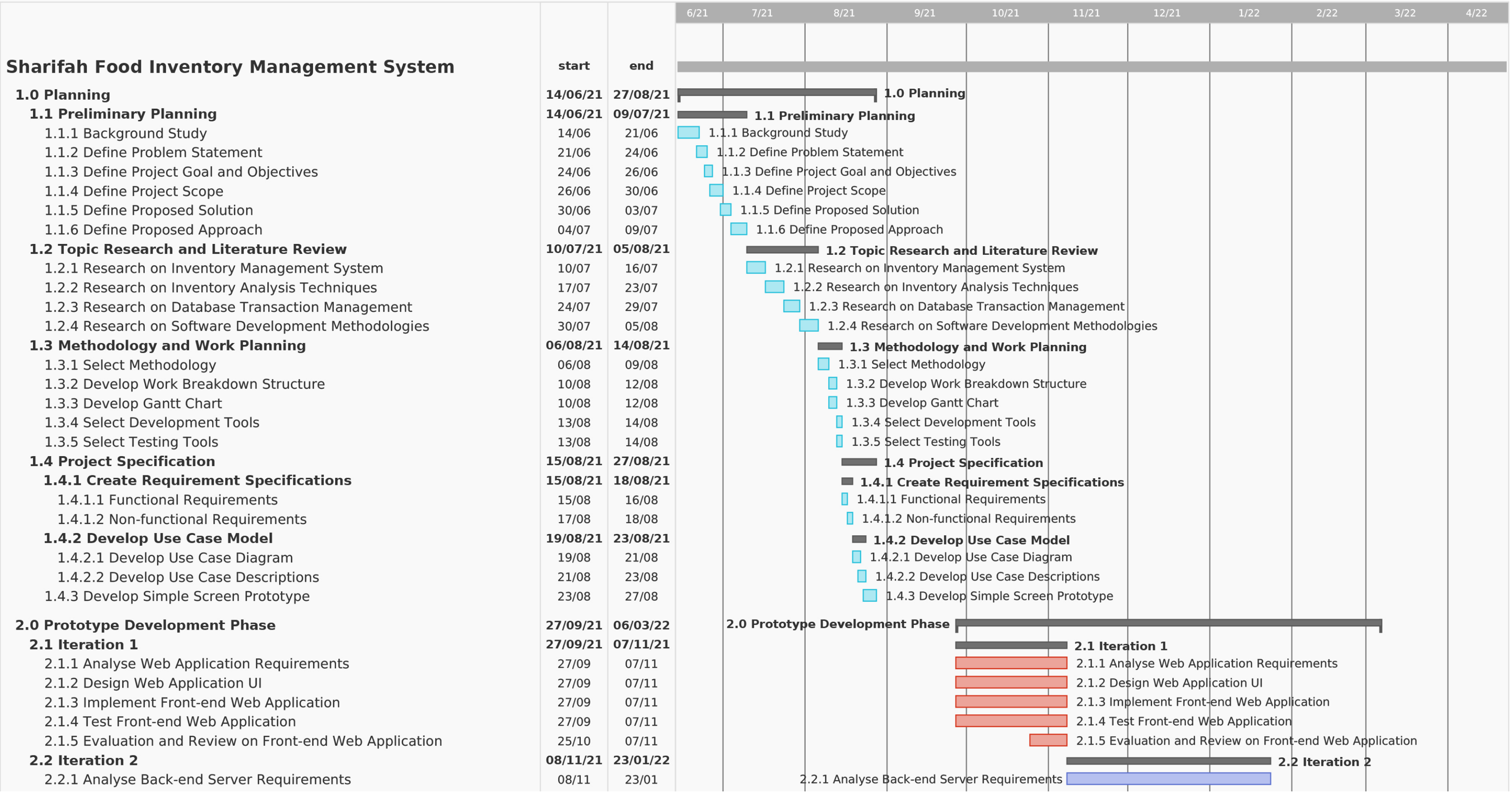
Zoho Inventory Team, n.d. *Elevate Your Inventory Management to the Cloud. Start Living the Easy Life.* [Online]
Available at: https://www.zoho.com/inventory/articles/elevate-your-inventory-management-to-the-cloud.html
[Accessed 3 July 2021].

APPENDICES

Appendix A: Work Breakdown Structure (WBS)


Sharifah Food Inventory Management System
1.0 Planning
1.1 Preliminary Planning
1.1.1 Background Study
1.1.2 Define Problem Statement
1.1.3 Define Project Goal and Objective
1.1.4 Define Project Scope
1.1.5 Define Proposed Solution
1.1.6 Define Proposed Approach
1.2 Topic Research and Literature Review
1.2.1 Research on Inventory Management System
1.2.2 Research on Inventory Analysis Techniques
1.2.3 Research on Database Transaction Management
1.2.4 Research on Software Development Methodologies
1.3 Methodology and Work Planning
1.3.1 Select Methodology
1.3.2 Develop Work Breakdown Structure
1.3.3 Develop Gantt Chart
1.3.4 Select Development Tools
1.3.5 Select Testing Tools
1.4 Project Specification
1.4.1 Create Requirement Specifications
1.4.1.1 Functional Requirements
1.4.1.2 Non-functional Requirements
1.4.2 Develop Use Case Model
1.4.2.1 Develop Use Case Diagram
1.4.2.2 Develop Use Case Descriptions
1.4.3 Develop Simple Screen Prototype
2.0 Prototype Development Phase
2.1 Iteration 1
2.1.1 Analyse Web Application Requirements
2.1.2 Design Web Application UI
2.1.3 Implement Front-end Web Application
2.1.4 Test Front-end Web Application
2.1.5 Evaluation and Review on Front-end Web Application
2.2 Iteration 2
2.2.1 Analyse Back-end Server Requirements
2.2.2 Design
2.2.2.1 Design System Architecture
2.2.2.2 Design Activity Diagrams
2.2.2.3 Design Data Dictionary
2.2.2.4 Design Entity-Relationship Diagram
2.2.3 Implementation
2.2.3.1 Implement Back-end Server
2.2.3.2 Connect Back-end Server to Web Application
2.2.4 Testing
2.2.4.1 Unit Testing
2.2.4.2 Integration Testing
2.2.5 Evaluation and Review on Back-end Server
2.3 Iteration 3
2.3.1 Analyse Mobile Application Requirements
2.3.2 Design Mobile Application UI
2.3.3 Implement Mobile Application
2.3.4 Test Mobile Application
2.3.5 Evaluation and Review on Mobile Application
3.0 System Evaluation Phase
3.1 Develop Test Plan
3.2 Perform Unit Testing
3.3 Perform Integration Testing
3.4 Perform System Usability Testing
3.5 Perform User Acceptance Testing
4.0 System Deployment Phase
4.1 Final Review
4.2 Prepare Final Report
4.3 Completion of Project

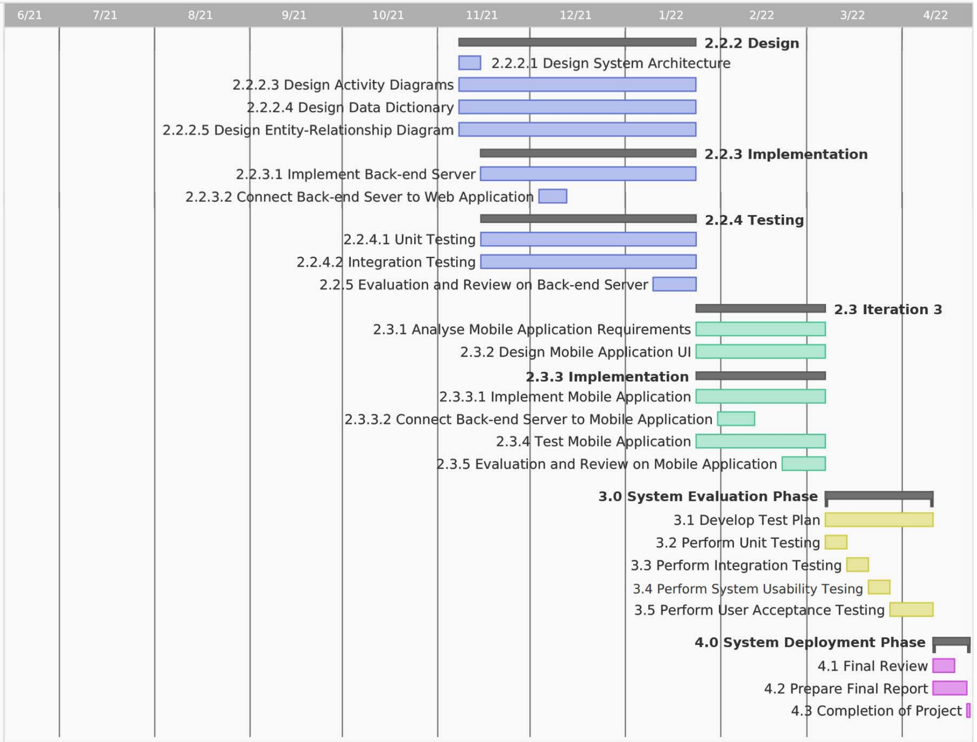
Appendix B: Gantt Chart



Sharifah Food Inventory Management System	start	end
1.0 Planning	**14/06/21**	**27/08/21**
1.1 Preliminary Planning	**14/06/21**	**09/07/21**
1.1.1 Background Study	14/06	21/06
1.1.2 Define Problem Statement	21/06	24/06
1.1.3 Define Project Goal and Objectives	24/06	26/06
1.1.4 Define Project Scope	26/06	30/06
1.1.5 Define Proposed Solution	30/06	03/07
1.1.6 Define Proposed Approach	04/07	09/07
1.2 Topic Research and Literature Review	**10/07/21**	**05/08/21**
1.2.1 Research on Inventory Management System	10/07	16/07
1.2.2 Research on Inventory Analysis Techniques	17/07	23/07
1.2.3 Research on Database Transaction Management	24/07	29/07
1.2.4 Research on Software Development Methodologies	30/07	05/08
1.3 Methodology and Work Planning	**06/08/21**	**14/08/21**
1.3.1 Select Methodology	06/08	09/08
1.3.2 Develop Work Breakdown Structure	10/08	12/08
1.3.3 Develop Gantt Chart	10/08	12/08
1.3.4 Select Development Tools	13/08	14/08
1.3.5 Select Testing Tools	13/08	14/08
1.4 Project Specification	**15/08/21**	**27/08/21**
1.4.1 Create Requirement Specifications	**15/08/21**	**18/08/21**
1.4.1.1 Functional Requirements	15/08	16/08
1.4.1.2 Non-functional Requirements	17/08	18/08
1.4.2 Develop Use Case Model	**19/08/21**	**23/08/21**
1.4.2.1 Develop Use Case Diagram	19/08	21/08
1.4.2.2 Develop Use Case Descriptions	21/08	23/08
1.4.3 Develop Simple Screen Prototype	23/08	27/08
2.0 Prototype Development Phase	**27/09/21**	**06/03/22**
2.1 Iteration 1	**27/09/21**	**07/11/21**
2.1.1 Analyse Web Application Requirements	27/09	07/11
2.1.2 Design Web Application UI	27/09	07/11
2.1.3 Implement Front-end Web Application	27/09	07/11
2.1.4 Test Front-end Web Application	27/09	07/11
2.1.5 Evaluation and Review on Front-end Web Application	25/10	07/11
2.2 Iteration 2	**08/11/21**	**23/01/22**
2.2.1 Analyse Back-end Server Requirements	08/11	23/01

Appendix B (Continued)

2.2.2 Design	**08/11/21**	**23/01/22**
2.2.2.1 Design System Architecture	08/11	14/11
2.2.2.2 Design Activity Diagrams	08/11	23/01
2.2.2.3 Design Data Dictionary	08/11	23/01
2.2.2.4 Design Entity-Relationship Diagram	08/11	23/01
2.2.3 Implementation	**15/11/21**	**23/01/22**
2.2.3.1 Implement Back-end Server	15/11	23/01
2.2.3.2 Connect Back-end Sever to Web Application	04/12	12/12
2.2.4 Testing	**15/11/21**	**23/01/22**
2.2.4.1 Unit Testing	15/11	23/01
2.2.4.2 Integration Testing	15/11	23/01
2.2.5 Evaluation and Review on Back-end Server	10/01	23/01
2.3 Iteration 3	**24/01/22**	**06/03/22**
2.3.1 Analyse Mobile Application Requirements	24/01	06/03
2.3.2 Design Mobile Application UI	24/01	06/03
2.3.3 Implementation	**24/01/22**	**06/03/22**
2.3.3.1 Implement Mobile Application	24/01	06/03
2.3.3.2 Connect Back-end Server to Mobile Application	31/01	11/02
2.3.4 Test Mobile Application	24/01	06/03
2.3.5 Evaluation and Review on Mobile Application	21/02	06/03
3.0 System Evaluation Phase	**07/03/22**	**10/04/22**
3.1 Develop Test Plan	07/03	10/04
3.2 Perform Unit Testing	07/03	13/03
3.3 Perform Integration Testing	14/03	20/03
3.4 Perform System Usability Tesing	21/03	27/03
3.5 Perform User Acceptance Testing	28/03	10/04
4.0 System Deployment Phase	**11/04/22**	**22/04/22**
4.1 Final Review	11/04	17/04
4.2 Prepare Final Report	11/04	21/04
4.3 Completion of Project	22/04	22/04



Appendix C: Informed Consent Form on Usability Testing and User Acceptance Testing

#Participant 1:

Informed Consent Form

This study is being conducted by Tan Yuan Jie, a software engineering student from Universiti Tunku Abdul Rahman. This study aims to evaluate the feasibility and usability of the developed systems under the title of "Sharifah Food Inventory Management System".

You have been invited to participate in this research study which will consist of the following:

1. User Acceptance Test (UAT)
2. System Usability Test

By signing this form, you acknowledged that:

- You have agreed to participate in the study conducted.
- You have understood the use and release of your information.
- You have understood that your information is for research purposes only.
- Your signature does not represent a granting of any legal rights.

Date : 17/04/2022
Name : Tan En Xi
Signature : ***TAN EN XI***

Thank you for your participation.

#Participant 2:

Informed Consent Form

This study is being conducted by Tan Yuan Jie, a software engineering student from Universiti Tunku Abdul Rahman. This study aims to evaluate the feasibility and usability of the developed systems under the title of "Sharifah Food Inventory Management System".

You have been invited to participate in this research study which will consist of the following:

3. User Acceptance Test (UAT)
4. System Usability Test

By signing this form, you acknowledged that:

- You have agreed to participate in the study conducted.
- You have understood the use and release of your information.
- You have understood that your information is for research purposes only.
- Your signature does not represent a granting of any legal rights.

Date	: 17.04.2022
Name	: Chang Sin Ee
Signature	: SINEE

Thank you for your participation.

#Participant 3:

<u>Informed Consent Form</u>

This study is being conducted by Tan Yuan Jie, a software engineering student from Universiti Tunku Abdul Rahman. This study aims to evaluate the feasibility and usability of the developed systems under the title of "Sharifah Food Inventory Management System".

You have been invited to participate in this research study which will consist of the following:

5. User Acceptance Test (UAT)
6. System Usability Test

By signing this form, you acknowledged that:

- You have agreed to participate in the study conducted.
- You have understood the use and release of your information.
- You have understood that your information is for research purposes only.
- Your signature does not represent a granting of any legal rights.

Date : 17/4/2022

Name : Goh Shi Min

Signature : GOH

Thank you for your participation.

#Participant 4:

Informed Consent Form

This study is being conducted by Tan Yuan Jie, a software engineering student from Universiti Tunku Abdul Rahman. This study aims to evaluate the feasibility and usability of the developed systems under the title of "Sharifah Food Inventory Management System".

You have been invited to participate in this research study which will consist of the following:

7. User Acceptance Test (UAT)
8. System Usability Test

By signing this form, you acknowledged that:

- You have agreed to participate in the study conducted.
- You have understood the use and release of your information.
- You have understood that your information is for research purposes only.
- Your signature does not represent a granting of any legal rights.

Date	: 17-04-2022
Name	: GOH QI XUAN
Signature	: GOH QI XUAN

Thank you for your participation.

#Participant 5:

Informed Consent Form

This study is being conducted by Tan Yuan Jie, a software engineering student from Universiti Tunku Abdul Rahman. This study aims to evaluate the feasibility and usability of the developed systems under the title of "Sharifah Food Inventory Management System".

You have been invited to participate in this research study which will consist of the following:

9. User Acceptance Test (UAT)
10. System Usability Test

By signing this form, you acknowledged that:

- You have agreed to participate in the study conducted.
- You have understood the use and release of your information.
- You have understood that your information is for research purposes only.
- Your signature does not represent a granting of any legal rights.

Date : 17/4/2022

Name : Seow Kai Sheng

Signature : KAI

Thank you for your participation.

Appendix D: Sample of User Satisfaction Survey Form

Section A: Tester Information					
Name		Gender		Age	
Email		Occupation			

Section B: Survey on Management System
1. Do you use administrative management systems before?
☐ Yes ☐ No
2. If yes, list out the administrative management systems you have used before.

Section C: Survey on E-commerce Platform		
1. Do you shop online?		
☐ Yes ☐ No		
2. How often do you shop online?		
☐ Never ☐ Daily ☐ Several times a week ☐ Weekly ☐ Several times a month		
3. List out three e-commerce platforms that you have used before.		

Section D: Important Note
1. Please do not provide any confidential information, such as your bank account and credit card number.
2. Please follow the instructions stated in the UAT Test Scenarios file.
3. Please fill up Section E and F after completing UAT for the e-commerce platform and Inventory Management System.
4. The System can be accessed through the following links:
Inventory Management System: https://fyp-shrf.herokuapp.com/
E-commerce Platform: https://fyp-shrf-ecommerce.herokuapp.com/

Section E: User Satisfaction Survey on E-commerce Platform					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the platform to place orders on Sharifah Food products.	□	□	□	□	□
2. I found the platform unnecessarily complex.	□	□	□	□	□
3. I thought the platform was easy to use.	□	□	□	□	□
4. I think I would need the support of a technical person to be able to use the platform.	□	□	□	□	□
5. I found the platform was easily moved through without much backtracking or data re-entry.	□	□	□	□	□
6. I thought there was too much inconsistency in the platform.	□	□	□	□	□
7. I would imagine that most people would learn about this platform very quickly.	□	□	□	□	□
8. I found the platform very awkward to use.	□	□	□	□	□
9. I felt very confident using the platform.	□	□	□	□	□
10. I needed to learn a lot of things before I could get going with the platform.	□	□	□	□	□

Please comment on the following questions.	
What did you like the best about the platform?	
What did you like the least about the platform?	
What would you say if you were to describe the platform to a colleague in a sentence or two?	
Do you have any other final comments or questions?	

Section F: User Satisfaction Survey on Inventory Management System					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the system for administrative management.	□	□	□	□	□
2. I found the system unnecessarily complex.	□	□	□	□	□
3. I thought the system was easy to use.	□	□	□	□	□
4. I think I would need the support of a technical person to be able to use the system.	□	□	□	□	□
5. I found the system was easily moved through without much backtracking or data re-entry.	□	□	□	□	□
6. I thought there was too much inconsistency in the system.	□	□	□	□	□
7. I would imagine that most people would learn about this system very quickly.	□	□	□	□	□
8. I found the system very awkward to use.	□	□	□	□	□
9. I felt very confident using the system.	□	□	□	□	□
10. I needed to learn a lot of things before I could get going with the system.	□	□	□	□	□

Please comment on the following questions.	
What did you like the best about the system?	
What did you like the least about the system?	
What would you say if you were to describe the system to a colleague in a sentence or two?	
Do you have any other final comments or questions?	

Appendix E: Results of User Satisfaction Survey Form

#Participant 1:

Section A: Tester Information					
Name	Tan En Xi	Gender	Female	Age	21
Email	tanenxi01@gmail.com	Occupation	Student (QS)		

Section B: Survey on Management System
1. Do you use administrative management systems before?
☑ Yes ☐ No
2. If yes, list out the administrative management systems you have used before.
Counter system

Section C: Survey on E-commerce Platform		
1. Do you shop online?		
☑ Yes ☐ No		
2. How often do you shop online?		
☐ Never ☐ Daily ☐ Several times a week ☐ Weekly ☑ Several times a month		
3. List out three e-commerce platforms that you have used before.		
Lazada	Shopee	Uniqlo

Section D: Important Note
1. Please do not provide any confidential information, such as your bank account and credit card number.
2. Please follow the instructions stated in the UAT Test Scenarios file.
3. Please fill up Section E and F after completing UAT for the e-commerce platform and Inventory Management System.
4. The System can be accessed through the following links:
Inventory Management System: https://fyp-shrf.herokuapp.com/
E-commerce Platform: https://fyp-shrf-ecommerce.herokuapp.com/

Section E: User Satisfaction Survey on E-commerce Platform					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the platform to place orders on Sharifah Food products.	□	□	□	☑	□
2. I found the platform unnecessarily complex.	□	☑	□	□	□
3. I thought the platform was easy to use.	□	□	□	□	☑
4. I think I would need the support of a technical person to be able to use the platform.	☑	□	□	□	□
5. I found the platform was easily moved through without much backtracking or data re-entry.	□	□	□	☑	□
6. I thought there was too much inconsistency in the platform.	□	☑	□	□	□
7. I would imagine that most people would learn about this platform very quickly.	□	□	□	□	☑
8. I found the platform very awkward to use.	☑	□	□	□	□
9. I felt very confident using the platform.	□	□	□	□	☑
10. I needed to learn a lot of things before I could get going with the platform.	☑	□	□	□	□

Please comment on the following questions.	
What did you like the best about the platform?	Simple layout
What did you like the least about the platform?	Add to cart
What would you say if you were to describe the platform to a colleague in a sentence or two?	An E-commerce platform that is easy to use.
Do you have any other final comments or questions?	It is better to place the quantity after pressing the 'Add to Cart' button immediately instead of editing it on the Shopping Cart.

Section F: User Satisfaction Survey on Inventory Management System					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the system for administrative management.	□	□	□	☑	□
2. I found the system unnecessarily complex.	□	☑	□	□	□
3. I thought the system was easy to use.	□	□	□	☑	□
4. I think I would need the support of a technical person to be able to use the system.	□	☑	□	□	□
5. I found the system was easily moved through without much backtracking or data re-entry.	□	□	□	□	☑
6. I thought there was too much inconsistency in the system.	☑	□	□	□	□
7. I would imagine that most people would learn about this system very quickly.	□	□	□	☑	□
8. I found the system very awkward to use.	☑	□	□	□	□
9. I felt very confident using the system.	□	□	□	☑	□
10. I needed to learn a lot of things before I could get going with the system.	□	☑	□	□	□

Please comment on the following questions.	
What did you like the best about the system?	Business Insights
What did you like the least about the system?	Manage Customers
What would you say if you were to describe the system to a colleague in a sentence or two?	An Inventory Management System that has good design.
Do you have any other final comments or questions?	None

#Participant 2:

Section A: Tester Information					
Name	Chang Sin Ee	Gender	F	Age	27
Email	sinee.2675@gmail.com	Occupation	Analyst		

Section B: Survey on Management System	
1. Do you use administrative management systems before?	
☐ Yes	☑ No
2. If yes, list out the administrative management systems you have used before.	

Section C: Survey on E-commerce Platform		
1. Do you shop online?		
☑ Yes	☐ No	
2. How often do you shop online?		
☐ Never ☐ Daily ☑ Several times a week ☐ Weekly ☐ Several times a month		
3. List out three e-commerce platforms that you have used before.		
Shopee	Lazada	TaoBao

Section D: Important Note
1. Please do not provide any confidential information, such as your bank account and credit card number.
2. Please follow the instructions stated in the UAT Test Scenarios file.
3. Please fill up Section E and F after completing UAT for the e-commerce platform and Inventory Management System.
4. The System can be accessed through the following links:
Inventory Management System: https://fyp-shrf.herokuapp.com/
E-commerce Platform: https://fyp-shrf-ecommerce.herokuapp.com/

Section E: User Satisfaction Survey on E-commerce Platform					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the platform to place orders on Sharifah Food products.	□	□	□	□	☑
2. I found the platform unnecessarily complex.	☑	□	□	□	□
3. I thought the platform was easy to use.	□	□	□	□	☑
4. I think I would need the support of a technical person to be able to use the platform.	☑	□	□	□	□
5. I found the platform was easily moved through without much backtracking or data re-entry.	☑	□	□	□	□
6. I thought there was too much inconsistency in the platform.	☑	□	□	□	□
7. I would imagine that most people would learn about this platform very quickly.	□	□	□	□	☑
8. I found the platform very awkward to use.	☑	□	□	□	□
9. I felt very confident using the platform.	□	□	□	□	☑
10. I needed to learn a lot of things before I could get going with the platform.	☑	□	□	□	□

Please comment on the following questions.	
What did you like the best about the platform?	-
What did you like the least about the platform?	-
What would you say if you were to describe the platform to a colleague in a sentence or two?	-
Do you have any other final comments or questions?	-

Section F: User Satisfaction Survey on Inventory Management System					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the system for administrative management.	□	□	□	□	☑
2. I found the system unnecessarily complex.	□	☑	□	□	□
3. I thought the system was easy to use.	□	□	□	□	☑
4. I think I would need the support of a technical person to be able to use the system.	☑	□	□	□	□
5. I found the system was easily moved through without much backtracking or data re-entry.	□	□	□	□	☑
6. I thought there was too much inconsistency in the system.	☑	□	□	□	□
7. I would imagine that most people would learn about this system very quickly.	□	□	□	□	☑
8. I found the system very awkward to use.	☑	□	□	□	□
9. I felt very confident using the system.	□	□	□	□	☑
10. I needed to learn a lot of things before I could get going with the system.	☑	□	□	□	□

Please comment on the following questions.	
What did you like the best about the system?	-
What did you like the least about the system?	-
What would you say if you were to describe the system to a colleague in a sentence or two?	-
Do you have any other final comments or questions?	-

#Participant 3:

Section A: Tester Information					
Name	Goh Shi Min	Gender	Female	Age	26
Email	Gshimin9627@gmail.com	Occupation	Auditor		

Section B: Survey on Management System
1. Do you use administrative management systems before?
☐ Yes ☑ No
2. If yes, list out the administrative management systems you have used before.

Section C: Survey on E-commerce Platform		
1. Do you shop online?		
☑ Yes ☐ No		
2. How often do you shop online?		
☐ Never ☐ Daily ☐ Several times a week ☐ Weekly ☑ Several times a month		
3. List out three e-commerce platforms that you have used before.		
Shopee	Lazada	Taobao

Section D: Important Note
1. Please do not provide any confidential information, such as your bank account and credit card number.
2. Please follow the instructions stated in the UAT Test Scenarios file.
3. Please fill up Section E and F after completing UAT for the e-commerce platform and Inventory Management System.
4. The System can be accessed through the following links:
Inventory Management System: https://fyp-shrf.herokuapp.com/
E-commerce Platform: https://fyp-shrf-ecommerce.herokuapp.com/

Section E: User Satisfaction Survey on E-commerce Platform					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the platform to place orders on Sharifah Food products.	□	□	□	□	☑
2. I found the platform unnecessarily complex.	☑	□	□	□	□
3. I thought the platform was easy to use.	□	□	□	□	☑
4. I think I would need the support of a technical person to be able to use the platform.	□	☑	□	□	□
5. I found the platform was easily moved through without much backtracking or data re-entry.	□	□	□	☑	□
6. I thought there was too much inconsistency in the platform.	□	☑	□	□	□
7. I would imagine that most people would learn about this platform very quickly.	□	□	□	□	☑
8. I found the platform very awkward to use.	☑	□	□	□	□
9. I felt very confident using the platform.	□	□	□	☑	□
10. I needed to learn a lot of things before I could get going with the platform.	☑	□	□	□	□

Please comment on the following questions.	
What did you like the best about the platform?	User friendly, the flow for shopping is enjoyable
What did you like the least about the platform?	Adding product to the cart if quantity needed is more than one
What would you say if you were to describe the platform to a colleague in a sentence or two?	User friendly
Do you have any other final comments or questions?	None

Section F: User Satisfaction Survey on Inventory Management System					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the system for administrative management.	□	□	□	☑	□
2. I found the system unnecessarily complex.	□	☑	□	□	□
3. I thought the system was easy to use.	□	□	□	☑	□
4. I think I would need the support of a technical person to be able to use the system.	□	□	☑	□	□
5. I found the system was easily moved through without much backtracking or data re-entry.	□	□	□	☑	□
6. I thought there was too much inconsistency in the system.	□	☑	□	□	□
7. I would imagine that most people would learn about this system very quickly.	□	□	□	☑	□
8. I found the system very awkward to use.	☑	□	□	□	□
9. I felt very confident using the system.	□	□	□	☑	□
10. I needed to learn a lot of things before I could get going with the system.	□	☑	□	□	□
Please comment on the following questions.					

What did you like the best about the system?	Easy to use
What did you like the least about the system?	Minimize deviation is allowed when searching for things
What would you say if you were to describe the system to a colleague in a sentence or two?	User friendly
Do you have any other final comments or questions?	None

#Participant 4:

Section A: Tester Information					
Name	GOH QI XUAN	Gender	MALE	Age	24
Email	qxuan000@gmail.com	Occupation	IT ASSISTANT		

Section B: Survey on Management System
1. Do you use administrative management systems before?
☐ Yes ☑ No
2. If yes, list out the administrative management systems you have used before.

Section C: Survey on E-commerce Platform		
1. Do you shop online?		
☑ Yes ☐ No		
2. How often do you shop online?		
☐ Never ☐ Daily ☐ Several times a week ☐ Weekly ☑ Several times a month		
3. List out three e-commerce platforms that you have used before.		
Shopee	Lazada	Zalora

Section D: Important Note
1. Please do not provide any confidential information, such as your bank account and credit card number.
2. Please follow the instructions stated in the UAT Test Scenarios file.
3. Please fill up Section E and F after completing UAT for the e-commerce platform and Inventory Management System.
4. The System can be accessed through the following links:
Inventory Management System: https://fyp-shrf.herokuapp.com/
E-commerce Platform: https://fyp-shrf-ecommerce.herokuapp.com/

Section E: User Satisfaction Survey on E-commerce Platform					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the platform to place orders on Sharifah Food products.	□	□	□	□	☑
2. I found the platform unnecessarily complex.	☑	□	□	□	□
3. I thought the platform was easy to use.	□	□	□	□	☑
4. I think I would need the support of a technical person to be able to use the platform.	☑	□	□	□	□
5. I found the platform was easily moved through without much backtracking or data re-entry.	□	□	□	☑	□
6. I thought there was too much inconsistency in the platform.	☑	□	□	□	□
7. I would imagine that most people would learn about this platform very quickly.	□	□	□	□	☑
8. I found the platform very awkward to use.	☑	□	□	□	□
9. I felt very confident using the platform.	□	□	□	□	☑
10. I needed to learn a lot of things before I could get going with the platform.	☑	□	□	□	□

Please comment on the following questions.	
What did you like the best about the platform?	Friendly user interface design
What did you like the least about the platform?	-
What would you say if you were to describe the platform to a colleague in a sentence or two?	-
Do you have any other final comments or questions?	-

Section F: User Satisfaction Survey on Inventory Management System					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the system for administrative management.	□	□	□	□	☑
2. I found the system unnecessarily complex.	☑	□	□	□	□
3. I thought the system was easy to use.	□	□	□	□	☑
4. I think I would need the support of a technical person to be able to use the system.	☑	□	□	□	□
5. I found the system was easily moved through without much backtracking or data re-entry.	□	□	□	□	☑
6. I thought there was too much inconsistency in the system.	☑	□	□	□	□
7. I would imagine that most people would learn about this system very quickly.	□	□	□	☑	□
8. I found the system very awkward to use.	☑	□	□	□	□
9. I felt very confident using the system.	□	□	□	☑	□
10. I needed to learn a lot of things before I could get going with the system.	☑	□	□	□	□

Please comment on the following questions.	
What did you like the best about the system?	The management interface shows all the important data to the seller with a simple and clear design
What did you like the least about the system?	-
What would you say if you were to describe the system to a colleague in a sentence or two?	-
Do you have any other final comments or questions?	-

#Participant 5:

Section A: Tester Information					
Name	Seow Kai Sheng	Gender	Male	Age	22
Email	kai4112000@1utar.my	Occupation	Student (SE)		

Section B: Survey on Management System	
1. Do you use administrative management systems before?	
☐ Yes	☑ No
2. If yes, list out the administrative management systems you have used before.	
-	

Section C: Survey on E-commerce Platform		
1. Do you shop online?		
☑ Yes	☐ No	
2. How often do you shop online?		
☐ Never ☐ Daily ☐ Several times a week ☐ Weekly ☑ Several times a month		
3. List out three e-commerce platforms that you have used before.		
Shopee	Lazada	-

Section D: Important Note
1. Please do not provide any confidential information, such as your bank account and credit card number.
2. Please follow the instructions stated in the UAT Test Scenarios file.
3. Please fill up Section E and F after completing UAT for the e-commerce platform and Inventory Management System.
4. The System can be accessed through the following links:
Inventory Management System: https://fyp-shrf.herokuapp.com/
E-commerce Platform: https://fyp-shrf-ecommerce.herokuapp.com/

Section E: User Satisfaction Survey on E-commerce Platform					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the platform to place orders on Sharifah Food products.	□	□	□	□	☑
2. I found the platform unnecessarily complex.	☑	□	□	□	□
3. I thought the platform was easy to use.	□	□	□	□	☑
4. I think I would need the support of a technical person to be able to use the platform.	☑	□	□	□	□
5. I found the platform was easily moved through without much backtracking or data re-entry.	□	□	□	□	☑
6. I thought there was too much inconsistency in the platform.	□	☑	□	□	□
7. I would imagine that most people would learn about this platform very quickly.	□	□	□	□	☑
8. I found the platform very awkward to use.	☑	□	□	□	□
9. I felt very confident using the platform.	□	□	□	□	☑
10. I needed to learn a lot of things before I could get going with the platform.	☑	□	□	□	□

Please comment on the following questions.	
What did you like the best about the platform?	Responsive and simple UI.
What did you like the least about the platform?	Add To Cart.
What would you say if you were to describe the platform to a colleague in a sentence or two?	An e-commerce platform to place orders for Sharifah Food.
Do you have any other final comments or questions?	Maybe can add input to allow user to enter quantity on the item details page

Section F: User Satisfaction Survey on Inventory Management System					
Please rate the following statements.					
1 = Strongly Disagree 5 = Strongly Agree	1	2	3	4	5
1. I think that I would like to use the system for administrative management.	□	□	□	□	☑
2. I found the system unnecessarily complex.	☑	□	□	□	□
3. I thought the system was easy to use.	□	□	□	☑	□
4. I think I would need the support of a technical person to be able to use the system.	□	☑	□	□	□
5. I found the system was easily moved through without much backtracking or data re-entry.	□	□	□	□	☑
6. I thought there was too much inconsistency in the system.	□	☑	□	□	□
7. I would imagine that most people would learn about this system very quickly.	□	□	☑	□	□
8. I found the system very awkward to use.	□	☑	□	□	□
9. I felt very confident using the system.	□	□	□	□	☑
10. I needed to learn a lot of things before I could get going with the system.	□	☑	□	□	□

Please comment on the following questions.	
What did you like the best about the system?	Attractive UI.
What did you like the least about the system?	-
What would you say if you were to describe the system to a colleague in a sentence or two?	An inventory management system with good UI design.
Do you have any other final comments or questions?	Can make the search dropdown input more obvious so that the user knows that it is a dropdown.

Appendix F: Results of UAT

#Participant 1:

Testing Date	17/4/2022		
Testing Start Time	5:15 P.M.	Testing End Time	5:56 P.M.
Tester Name	Tan En Xi		
E-commerce Platform			
Test Module	**Test Scenario**	**Status**	**Comments**
Register	1. Able to create an account with a new email.	Pass	-
Login	1. Able to log in with the created account.	Pass	-
Item	1. Able to browse items.	Pass	-
	2. Able to search items.	Pass	-
	3. Able to browse products.	Pass	-
	4. Able to browse packages.	Pass	-
	5. Able to view item details.	Pass	-
Cart	1. Able to add an item to cart.	Pass	-
	2. Able to view shopping cart.	Pass	-
	3. Able to add quantity to the item in cart.	Pass	-
	4. Able to reduce quantity to the item in cart.	Pass	-
	5. Able to set quantity to the item in cart.	Pass	-
	6. Able to remove an item in cart.	Pass	-
	7. Able to checkout a cart with items.	Pass	-
Address	1. Able to add new address in the address book.	Pass	Search city and postcode take time, because cannot key in.

	2. Able to view address list in the address book.	Pass	-
	3. Able to edit an address in the address book.	Pass	-
	4. Able to delete an address in the address book.	Pass	-
Pickup Location	1. Able to view and select pickup locations.	Pass	-
Shipping Fee	1. Able to get shipping fee for shipping address,	Pass	-
Order	1. Able to select shipping or pickup.	Pass	-
	2. Able to place order after filling shipping information and payment method.	Pass	-
	3. Able to view order history.	Pass	-
	4. Able to search orders.	Pass	-
	5. Able to view order details.	Pass	-
Payment	1. Able to select payment method.	Pass	-
	2. Able to be redirected to payment page.	Pass	-
	3. Able to make payment.	Pass	Received order confirmation email.
Agent and Dropshipper Registration	1. Able to submit registration form.	Pass	-
	2. Able to get popup notification after submitting form.	Pass	Received email.
Voucher	1. Able to automatically apply voucher.	Pass	-
	2. Able to remove voucher.	Pass	-
	3. Able to apply voucher.	Pass	-
Customer	1. Able to view account information.	Pass	-
	2. Able to edit account information.	Pass	-

Inventory Management System			
Test Module	**Test Scenario**	**Status**	**Comments**
Login	1. Able to log in with the admin account.	Pass	-
Product	1. Able to add product.	Pass	-
	2. Able to view product list.	Pass	-
	3. Able to edit product.	Pass	-
	4. Able to hide product.	Pass	-
	5. Able to delete product.	Pass	-
	6. Able to update product inventory stock.	Pass	-
	7. Able to filter product.	Pass	-
	8. Able to sort product.	Pass	-
Package	1. Able to add package.	Pass	-
	2. Able to add products into package.	Pass	-
	3. Able to view package list.	Pass	-
	4. Able to edit package.	Pass	-
	5. Able to hide package.	Pass	-
	6. Able to delete package.	Pass	-
	7. Able to update package inventory stock.	Pass	-
	8. Able to filter package.	Pass	-
	9. Able to sort package.	Pass	-
Order	1. Able to view a list of orders.	Pass	-
	2. Able to filter orders.	Pass	-
	3. Able to sort orders.	Pass	-
	4. Able to view order details.	Pass	-
	5. Able to update tracking number.	Pass	-
	6. Able to update pickup for order.	Pass	-
	7. Able to cancel order.	Pass	-

	8. Able to generate order invoice.	Pass	can download as pdf
	9. Able to view the order tracking.	Pass	-
Shipping Fee	1. Able to view a list of shipping fees.	Pass	-
	2. Able to filter shipping fees.	Pass	-
	3. Able to sort shipping fees.	Pass	-
	4. Able to view shipping fees details.	Pass	-
	5. Able to update shipping fees.	Pass	-
	6. Able to delete shipping fees.	Pass	-
Pickup Location	1. Able to view a list of pickup locations.	Pass	-
	2. Able to filter pickup locations.	Pass	-
	3. Able to sort pickup locations.	Pass	-
	4. Able to view pickup location details.	Pass	-
	5. Able to update pickup location.	Pass	-
	6. Able to delete pickup location.	Pass	-
Customer	1. Able to view a list of customers.	Pass	-
	2. Able to filter customers.	Pass	Did not notice the input is dropdown.
	3. Able to sort customers.	Pass	-
	4. Able to view customer details.	Pass	-
	5. Able to suspend customer.	Pass	-
	6. Able to activate customer.	Pass	-
Agent and Dropshipper Registration	1. Able to view a list of registrations.	Pass	-
	2. Able to filter registrations.	Pass	-
	3. Able to sort registrations.	Pass	-

	4. Able to view registration details.	Pass	-
	5. Able to accept registration.	Pass	-
	6. Able to reject registration.	Pass	-
	7. Able to add new agent or dropshipper.	Pass	-
Voucher	1. Able to add voucher.	Pass	-
	2. Able to view voucher list.	Pass	-
	3. Able to edit voucher.	Pass	-
	4. Able to hide voucher.	Pass	-
	5. Able to delete voucher.	Pass	-
	8. Able to filter voucher.	Pass	-
	9. Able to sort voucher.	Pass	-
Analysis	1. Able to view product rankings.	Pass	-
	2. Able to view package rankings.	Pass	-
	3. Able to view statistics summary.	Pass	-
	4. Able to view key metrics analysis.	Pass	-
	5. Able to generate sales report.	Pass	can download as excel file.
	6. Able to export chart.	Pass	-
	7. Able to view ABC analysis.	Pass	-
	8. Able to view HML analysis.	Pass	-
	9. Able to view EOQ analysis.	Pass	-
	10. Able to view SS analysis.	Pass	-

#Participant 2:

Testing Date	17/4/2022		
Testing Start Time	6:15 P.M.	Testing End Time	6:45 P.M.
Tester Name	Chang Sin Ee		

E-commerce Platform			
Test Module	**Test Scenario**	**Status**	**Comments**
Register	1. Able to create an account with a new email.	Pass	-
Login	1. Able to log in with the created account.	Pass	-
Item	1. Able to browse items.	Pass	-
	2. Able to search items.	Pass	-
	3. Able to browse products.	Pass	-
	4. Able to browse packages.	Pass	-
	5. Able to view item details.	Pass	-
Cart	1. Able to add an item to cart.	Pass	-
	2. Able to view shopping cart.	Pass	-
	3. Able to add quantity to the item in cart.	Pass	-
	4. Able to reduce quantity to the item in cart.	Pass	-
	5. Able to set quantity to the item in cart.	Pass	-
	6. Able to remove an item in cart.	Pass	-
	7. Able to checkout a cart with items.	Pass	-
Address	1. Able to add new address in the address book.	Pass	-
	2. Able to view address list in the address book.	Pass	-
	3. Able to edit an address in the address book.	Pass	-

	4. Able to delete an address in the address book.	Pass	-
Pickup Location	1. Able to view and select pickup locations.	Pass	-
Shipping Fee	1. Able to get shipping fee for shipping address,	Pass	-
Order	1. Able to select shipping or pickup.	Pass	-
	2. Able to place order after filling shipping information and payment method.	Pass	-
	3. Able to view order history.	Pass	-
	4. Able to search orders.	Pass	-
	5. Able to view order details.	Pass	-
Payment	1. Able to select payment method.	Pass	-
	2. Able to be redirected to payment page.	Pass	-
	3. Able to make payment.	Pass	Received order confirmation email.
Agent and Dropshipper Registration	1. Able to submit registration form.	Pass	-
	2. Able to get popup notification after submitting form.	Pass	Received email.
Voucher	1. Able to automatically apply voucher.	Pass	-
	2. Able to remove voucher.	Pass	-
	3. Able to apply voucher.	Pass	-
Customer	1. Able to view account information.	Pass	-
	2. Able to edit account information.	Pass	-
Inventory Management System			
Test Module	**Test Scenario**	**Status**	**Comments**
Login	1. Able to log in with the admin account.	Pass	-

Product	1. Able to add product.	Pass	-
	2. Able to view product list.	Pass	-
	3. Able to edit product.	Pass	Take time to find the link to the edit product page.
	4. Able to hide product.	Pass	Hide through edit product.
	5. Able to delete product.	Pass	-
	6. Able to update product inventory stock.	Pass	-
	7. Able to filter product.	Pass	-
	8. Able to sort product.	Pass	-
Package	1. Able to add package.	Pass	-
	2. Able to add products into package.	Pass	-
	3. Able to view package list.	Pass	-
	4. Able to edit package.	Pass	-
	5. Able to hide package.	Pass	-
	6. Able to delete package.	Pass	-
	7. Able to update package inventory stock.	Pass	-
	8. Able to filter package.	Pass	-
	9. Able to sort package.	Pass	-
Order	1. Able to view a list of orders.	Pass	-
	2. Able to filter orders.	Pass	-
	3. Able to sort orders.	Pass	-
	4. Able to view order details.	Pass	-
	5. Able to update tracking number.	Pass	-
	6. Able to update pickup for order.	Pass	-
	7. Able to cancel order.	Pass	-
	8. Able to generate order invoice.	Pass	can download as pdf

	9. Able to view the order tracking.	Pass	-
Shipping Fee	1. Able to view a list of shipping fees.	Pass	-
	2. Able to filter shipping fees.	Pass	-
	3. Able to sort shipping fees.	Pass	-
	4. Able to view shipping fees details.	Pass	-
	5. Able to update shipping fees.	Pass	-
	6. Able to delete shipping fees.	Pass	-
Pickup Location	1. Able to view a list of pickup locations.	Pass	-
	2. Able to filter pickup locations.	Pass	-
	3. Able to sort pickup locations.	Pass	-
	4. Able to view pickup location details.	Pass	-
	5. Able to update pickup location.	Pass	-
	6. Able to delete pickup location.	Pass	-
Customer	1. Able to view a list of customers.	Pass	-
	2. Able to filter customers.	Pass	-
	3. Able to sort customers.	Pass	-
	4. Able to view customer details.	Pass	-
	5. Able to suspend customer.	Pass	-
	6. Able to activate customer.	Pass	-
Agent and Dropshipper Registration	1. Able to view a list of registrations.	Pass	-
	2. Able to filter registrations.	Pass	-
	3. Able to sort registrations.	Pass	-
	4. Able to view registration details.	Pass	-
	5. Able to accept registration.	Pass	-
	6. Able to reject registration.	Pass	-

	7. Able to add new agent or dropshipper.	Pass	-
Voucher	1. Able to add voucher.	Pass	-
	2. Able to view voucher list.	Pass	-
	3. Able to edit voucher.	Pass	-
	4. Able to hide voucher.	Pass	-
	5. Able to delete voucher.	Pass	-
	8. Able to filter voucher.	Pass	-
	9. Able to sort voucher.	Pass	-
Analysis	1. Able to view product rankings.	Pass	-
	2. Able to view package rankings.	Pass	-
	3. Able to view statistics summary.	Pass	-
	4. Able to view key metrics analysis.	Pass	-
	5. Able to generate sales report.	Pass	can download as excel file.
	6. Able to export chart.	Pass	-
	7. Able to view ABC analysis.	Pass	-
	8. Able to view HML analysis.	Pass	-
	9. Able to view EOQ analysis.	Pass	-
	10. Able to view SS analysis.	Pass	-

#Participant 3:

Testing Date	17/4/2022		
Testing Start Time	7:30 P.M.	Testing End Time	8:03 P.M.
Tester Name	Goh Shi Min		

E-commerce Platform

Test Module	Test Scenario	Status	Comments
Register	1. Able to create an account with a new email.	Pass	-
Login	1. Able to log in with the created account.	Pass	-
Item	1. Able to browse items.	Pass	-
	2. Able to search items.	Pass	-
	3. Able to browse products.	Pass	-
	4. Able to browse packages.	Pass	-
	5. Able to view item details.	Pass	-
Cart	1. Able to add an item to cart.	Pass	-
	2. Able to view shopping cart.	Pass	-
	3. Able to add quantity to the item in cart.	Pass	-
	4. Able to reduce quantity to the item in cart.	Pass	-
	5. Able to set quantity to the item in cart.	Pass	-
	6. Able to remove an item in cart.	Pass	-
	7. Able to checkout a cart with items.	Pass	-
Address	1. Able to add new address in the address book.	Pass	-
	2. Able to view address list in the address book.	Pass	-
	3. Able to edit an address in the address book.	Pass	-

	4. Able to delete an address in the address book.	Pass	-
Pickup Location	1. Able to view and select pickup locations.	Pass	-
Shipping Fee	1. Able to get shipping fee for shipping address,	Pass	-
Order	1. Able to select shipping or pickup.	Pass	-
	2. Able to place order after filling shipping information and payment method.	Pass	-
	3. Able to view order history.	Pass	-
	4. Able to search orders.	Pass	-
	5. Able to view order details.	Pass	-
Payment	1. Able to select payment method.	Pass	-
	2. Able to be redirected to payment page.	Pass	-
	3. Able to make payment.	Pass	Received order confirmation email.
Agent and Dropshipper Registration	1. Able to submit registration form.	Pass	-
	2. Able to get popup notification after submitting form.	Pass	Received email.
Voucher	1. Able to automatically apply voucher.	Pass	-
	2. Able to remove voucher.	Pass	-
	3. Able to apply voucher.	Pass	-
Customer	1. Able to view account information.	Pass	-
	2. Able to edit account information.	Pass	-
Inventory Management System			
Test Module	**Test Scenario**	**Status**	**Comments**
Login	1. Able to log in with the admin account.	Pass	-

Product	1. Able to add product.	Pass	-
	2. Able to view product list.	Pass	-
	3. Able to edit product.	Pass	-
	4. Able to hide product.	Pass	-
	5. Able to delete product.	Pass	-
	6. Able to update product inventory stock.	Pass	Update through edit page. Does not understand the meaning of the ‘Action’ column in product inventory page
	7. Able to filter product.	Pass	-
	8. Able to sort product.	Pass	-
Package	1. Able to add package.	Pass	-
	2. Able to add products into package.	Pass	-
	3. Able to view package list.	Pass	-
	4. Able to edit package.	Pass	-
	5. Able to hide package.	Pass	-
	6. Able to delete package.	Pass	-
	7. Able to update package inventory stock.	Pass	-
	8. Able to filter package.	Pass	-
	9. Able to sort package.	Pass	-
Order	1. Able to view a list of orders.	Pass	-
	2. Able to filter orders.	Pass	-
	3. Able to sort orders.	Pass	-
	4. Able to view order details.	Pass	-
	5. Able to update tracking number.	Pass	-
	6. Able to update pickup for order.	Pass	-
	7. Able to cancel order.	Pass	-

	8. Able to generate order invoice.	Pass	can download as pdf
	9. Able to view the order tracking.	Pass	-
Shipping Fee	1. Able to view a list of shipping fees.	Pass	-
	2. Able to filter shipping fees.	Pass	-
	3. Able to sort shipping fees.	Pass	-
	4. Able to view shipping fees details.	Pass	-
	5. Able to update shipping fees.	Pass	-
	6. Able to delete shipping fees.	Pass	-
Pickup Location	1. Able to view a list of pickup locations.	Pass	-
	2. Able to filter pickup locations.	Pass	-
	3. Able to sort pickup locations.	Pass	-
	4. Able to view pickup location details.	Pass	-
	5. Able to update pickup location.	Pass	-
	6. Able to delete pickup location.	Pass	-
Customer	1. Able to view a list of customers.	Pass	-
	2. Able to filter customers.	Pass	Do not notice search dropdown (directly search customer name by customer id)
	3. Able to sort customers.	Pass	-
	4. Able to view customer details.	Pass	-
	5. Able to suspend customer.	Pass	-
	6. Able to activate customer.	Pass	-
	1. Able to view a list of registrations.	Pass	-
	2. Able to filter registrations.	Pass	-

Agent and Dropshipper Registration	3. Able to sort registrations.	Pass	-
	4. Able to view registration details.	Pass	-
	5. Able to accept registration.	Pass	-
	6. Able to reject registration.	Pass	-
	7. Able to add new agent or dropshipper.	Pass	-
Voucher	1. Able to add voucher.	Pass	-
	2. Able to view voucher list.	Pass	-
	3. Able to edit voucher.	Pass	-
	4. Able to hide voucher.	Pass	-
	5. Able to delete voucher.	Pass	-
	8. Able to filter voucher.	Pass	-
	9. Able to sort voucher.	Pass	-
Analysis	1. Able to view product rankings.	Pass	-
	2. Able to view package rankings.	Pass	-
	3. Able to view statistics summary.	Pass	-
	4. Able to view key metrics analysis.	Pass	-
	5. Able to generate sales report.	Pass	can download as excel file.
	6. Able to export chart.	Pass	-
	7. Able to view ABC analysis.	Pass	-
	8. Able to view HML analysis.	Pass	-
	9. Able to view EOQ analysis.	Pass	-
	10. Able to view SS analysis.	Pass	-

#Participant 4:

Testing Date	17/4/2022		
Testing Start Time	9:05 P.M.	Testing End Time	9:34 P.M.
Tester Name	Goh Qi Xuan		
E-commerce Platform			
Test Module	**Test Scenario**	**Status**	**Comments**
Register	1. Able to create an account with a new email.	Pass	-
Login	1. Able to log in with the created account.	Pass	-
Item	1. Able to browse items.	Pass	-
	2. Able to search items.	Pass	-
	3. Able to browse products.	Pass	-
	4. Able to browse packages.	Pass	-
	5. Able to view item details.	Pass	-
Cart	1. Able to add an item to cart.	Pass	-
	2. Able to view shopping cart.	Pass	-
	3. Able to add quantity to the item in cart.	Pass	-
	4. Able to reduce quantity to the item in cart.	Pass	-
	5. Able to set quantity to the item in cart.	Pass	-
	6. Able to remove an item in cart.	Pass	-
	7. Able to checkout a cart with items.	Pass	-
Address	1. Able to add new address in the address book.	Pass	-
	2. Able to view address list in the address book.	Pass	-
	3. Able to edit an address in the address book.	Pass	-

	4. Able to delete an address in the address book.	Pass	-
Pickup Location	1. Able to view and select pickup locations.	Pass	-
Shipping Fee	1. Able to get shipping fee for shipping address,	Pass	-
Order	1. Able to select shipping or pickup.	Pass	-
	2. Able to place order after filling shipping information and payment method.	Pass	-
	3. Able to view order history.	Pass	-
	4. Able to search orders.	Pass	-
	5. Able to view order details.	Pass	-
Payment	1. Able to select payment method.	Pass	-
	2. Able to be redirected to payment page.	Pass	-
	3. Able to make payment.	Pass	Received order confirmation email.
Agent and Dropshipper Registration	1. Able to submit registration form.	Pass	-
	2. Able to get popup notification after submitting form.	Pass	Received email.
Voucher	1. Able to automatically apply voucher.	Pass	-
	2. Able to remove voucher.	Pass	-
	3. Able to apply voucher.	Pass	-
Customer	1. Able to view account information.	Pass	-
	2. Able to edit account information.	Pass	-
Inventory Management System			
Test Module	**Test Scenario**	**Status**	**Comments**
Login	1. Able to log in with the admin account.	Pass	-

Product	1. Able to add product.	Pass	-
	2. Able to view product list.	Pass	-
	3. Able to edit product.	Pass	-
	4. Able to hide product.	Pass	-
	5. Able to delete product.	Pass	-
	6. Able to update product inventory stock.	Pass	-
	7. Able to filter product.	Pass	-
	8. Able to sort product.	Pass	-
Package	1. Able to add package.	Pass	-
	2. Able to add products into package.	Pass	-
	3. Able to view package list.	Pass	-
	4. Able to edit package.	Pass	-
	5. Able to hide package.	Pass	-
	6. Able to delete package.	Pass	-
	7. Able to update package inventory stock.	Pass	-
	8. Able to filter package.	Pass	-
	9. Able to sort package.	Pass	-
Order	1. Able to view a list of orders.	Pass	-
	2. Able to filter orders.	Pass	-
	3. Able to sort orders.	Pass	-
	4. Able to view order details.	Pass	-
	5. Able to update tracking number.	Pass	-
	6. Able to update pickup for order.	Pass	-
	7. Able to cancel order.	Pass	-
	8. Able to generate order invoice.	Pass	can download as pdf
	9. Able to view the order tracking.	Pass	-
Shipping Fee	1. Able to view a list of shipping fees.	Pass	-
	2. Able to filter shipping fees.	Pass	-

	3. Able to sort shipping fees.	Pass	-
	4. Able to view shipping fees details.	Pass	-
	5. Able to update shipping fees.	Pass	-
	6. Able to delete shipping fees.	Pass	-
Pickup Location	1. Able to view a list of pickup locations.	Pass	-
	2. Able to filter pickup locations.	Pass	-
	3. Able to sort pickup locations.	Pass	-
	4. Able to view pickup location details.	Pass	-
	5. Able to update pickup location.	Pass	-
	6. Able to delete pickup location.	Pass	-
Customer	1. Able to view a list of customers.	Pass	-
	2. Able to filter customers.	Pass	-
	3. Able to sort customers.	Pass	-
	4. Able to view customer details.	Pass	-
	5. Able to suspend customer.	Pass	-
	6. Able to activate customer.	Pass	-
Agent and Dropshipper Registration	1. Able to view a list of registrations.	Pass	-
	2. Able to filter registrations.	Pass	-
	3. Able to sort registrations.	Pass	-
	4. Able to view registration details.	Pass	-
	5. Able to accept registration.	Pass	-
	6. Able to reject registration.	Pass	-
	7. Able to add new agent or dropshipper.	Pass	-
Voucher	1. Able to add voucher.	Pass	-
	2. Able to view voucher list.	Pass	-

	3. Able to edit voucher.	Pass	-
	4. Able to hide voucher.	Pass	-
	5. Able to delete voucher.	Pass	-
	8. Able to filter voucher.	Pass	-
	9. Able to sort voucher.	Pass	-
Analysis	1. Able to view product rankings.	Pass	-
	2. Able to view package rankings.	Pass	-
	3. Able to view statistics summary.	Pass	-
	4. Able to view key metrics analysis.	Pass	-
	5. Able to generate sales report.	Pass	can download as excel file.
	6. Able to export chart.	Pass	-
	7. Able to view ABC analysis.	Pass	-
	8. Able to view HML analysis.	Pass	-
	9. Able to view EOQ analysis.	Pass	-
	10. Able to view SS analysis.	Pass	-

#Participant 5:

Testing Date	17/4/2022		
Testing Start Time	10:05 P.M.	Testing End Time	10:38 P.M.
Tester Name	Seow Kai Sheng		
E-commerce Platform			
Test Module	**Test Scenario**	**Status**	**Comments**
Register	1. Able to create an account with a new email.	Pass	-
Login	1. Able to log in with the created account.	Pass	-
Item	1. Able to browse items.	Pass	-
	2. Able to search items.	Pass	-
	3. Able to browse products.	Pass	-
	4. Able to browse packages.	Pass	-
	5. Able to view item details.	Pass	-
Cart	1. Able to add an item to cart.	Pass	-
	2. Able to view shopping cart.	Pass	-
	3. Able to add quantity to the item in cart.	Pass	-
	4. Able to reduce quantity to the item in cart.	Pass	-
	5. Able to set quantity to the item in cart.	Pass	-
	6. Able to remove an item in cart.	Pass	-
	7. Able to checkout a cart with items.	Pass	-
Address	1. Able to add new address in the address book.	Pass	-
	2. Able to view address list in the address book.	Pass	-
	3. Able to edit an address in the address book.	Pass	-

	4. Able to delete an address in the address book.	Pass	-
Pickup Location	1. Able to view and select pickup locations.	Pass	-
Shipping Fee	1. Able to get shipping fee for shipping address,	Pass	-
Order	1. Able to select shipping or pickup.	Pass	-
	2. Able to place order after filling shipping information and payment method.	Pass	-
	3. Able to view order history.	Pass	-
	4. Able to search orders.	Pass	-
	5. Able to view order details.	Pass	-
Payment	1. Able to select payment method.	Pass	-
	2. Able to be redirected to payment page.	Pass	-
	3. Able to make payment.	Pass	Received order confirmation email.
Agent and Dropshipper Registration	1. Able to submit registration form.	Pass	-
	2. Able to get popup notification after submitting form.	Pass	Received email.
Voucher	1. Able to automatically apply voucher.	Pass	-
	2. Able to remove voucher.	Pass	-
	3. Able to apply voucher.	Pass	-
Customer	1. Able to view account information.	Pass	-
	2. Able to edit account information.	Pass	-
Inventory Management System			
Test Module	**Test Scenario**	**Status**	**Comments**
Login	1. Able to log in with the admin account.	Pass	-

Product	1. Able to add product.	Pass	-
	2. Able to view product list.	Pass	-
	3. Able to edit product.	Pass	-
	4. Able to hide product.	Pass	-
	5. Able to delete product.	Pass	-
	6. Able to update product inventory stock.	Pass	-
	7. Able to filter product.	Pass	-
	8. Able to sort product.	Pass	-
Package	1. Able to add package.	Pass	-
	2. Able to add products into package.	Pass	-
	3. Able to view package list.	Pass	-
	4. Able to edit package.	Pass	-
	5. Able to hide package.	Pass	-
	6. Able to delete package.	Pass	-
	7. Able to update package inventory stock.	Pass	-
	8. Able to filter package.	Pass	-
	9. Able to sort package.	Pass	-
Order	1. Able to view a list of orders.	Pass	-
	2. Able to filter orders.	Pass	-
	3. Able to sort orders.	Pass	-
	4. Able to view order details.	Pass	-
	5. Able to update tracking number.	Pass	-
	6. Able to update pickup for order.	Pass	-
	7. Able to cancel order.	Pass	-
	8. Able to generate order invoice.	Pass	can download as pdf
	9. Able to view the order tracking.	Pass	-
Shipping Fee	1. Able to view a list of shipping fees.	Pass	-
	2. Able to filter shipping fees.	Pass	-

	3. Able to sort shipping fees.	Pass	-
	4. Able to view shipping fees details.	Pass	-
	5. Able to update shipping fees.	Pass	-
	6. Able to delete shipping fees.	Pass	-
Pickup Location	1. Able to view a list of pickup locations.	Pass	-
	2. Able to filter pickup locations.	Pass	-
	3. Able to sort pickup locations.	Pass	-
	4. Able to view pickup location details.	Pass	-
	5. Able to update pickup location.	Pass	-
	6. Able to delete pickup location.	Pass	-
Customer	1. Able to view a list of customers.	Pass	-
	2. Able to filter customers.	Pass	search dropdown is not obvious
	3. Able to sort customers.	Pass	-
	4. Able to view customer details.	Pass	-
	5. Able to suspend customer.	Pass	-
	6. Able to activate customer.	Pass	-
Agent and Dropshipper Registration	1. Able to view a list of registrations.	Pass	-
	2. Able to filter registrations.	Pass	-
	3. Able to sort registrations.	Pass	-
	4. Able to view registration details.	Pass	-
	5. Able to accept registration.	Pass	-
	6. Able to reject registration.	Pass	-
	7. Able to add new agent or dropshipper.	Pass	-
Voucher	1. Able to add voucher.	Pass	-

	2. Able to view voucher list.	Pass	-
	3. Able to edit voucher.	Pass	-
	4. Able to hide voucher.	Pass	-
	5. Able to delete voucher.	Pass	-
	8. Able to filter voucher.	Pass	-
	9. Able to sort voucher.	Pass	-
Analysis	1. Able to view product rankings.	Pass	-
	2. Able to view package rankings.	Pass	-
	3. Able to view statistics summary.	Pass	-
	4. Able to view key metrics analysis.	Pass	-
	5. Able to generate sales report.	Pass	can download as excel file.
	6. Able to export chart.	Pass	-
	7. Able to view ABC analysis.	Pass	-
	8. Able to view HML analysis.	Pass	-
	9. Able to view EOQ analysis.	Pass	-
	10. Able to view SS analysis.	Pass	-