

**DEVELOPMENT OF NON-ICT COMPANY PROCUREMENT AND SHIPMENT
PORTAL USING REFERENCED MOBILE APPLICATION**

**BY
CHIN PEY MIAN**

**A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION
SYSTEMS
Faculty of Information and Communication Technology
(Kampar Campus)**

JAN 2023

REPORT STATUS DECLARATION FORM

Title: DEVELOPMENT OF NON-ICT COMPANY PROCUREMENT AND
SHIPMENT PORTAL USING REFERENCED MOBILE APPLICATION

Academic Session: Jan 2023

I CHIN PEY MIAN
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)



(Supervisor's signature)

Address:

113 KAMPUNG KILANG

32700 BERUAS

PERAK

GOH HOCK GUAN

Date: 24 APRIL 2023

Date: 24/4/2023

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 24 APRIL 2023

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that CHIN PEY MIAN (ID No: 2004071) has completed this final year project entitled “DEVELOPMENT OF NON-ICT COMPANY PROCUREMENT AND SHIPMENT PORTAL USING REFERENCED MOBILE APPLICATION” under the supervision of Dr GOH HOCK GUAN (Supervisor) from the Department of Computer and Communication, Faculty of Information and Communication Technology, and Dr Kiran Adnan (Co-Supervisor)* from the Department of Digital Economy Technology , Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,




 (CHIN PEY MIAN)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**DEVELOPMENT OF NON-ICT COMPANY PROCUREMENT AND SHIPMENT PORTAL USING REFERENCED MOBILE APPLICATION**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :  _____
Name : CHIN PEY MIAN
Date : 24 APRIL 2023

ACKNOWLEDGEMENTS

I want to express my gratitude to everyone who guided me on the project. I would like to thank the company who provided me the opportunity on the participation of the project. I am very appreciating for the opportunity to learn from my supervisor, Ts Dr Goh Hock Guan, who provided me the guidance, valuable advice, and patient throughout the development of the project.

Furthermore, I would like to thank my family for the support, love, and encouragement during my studies. I would also like to thank my friends who encouraged and motivated me to accomplish the project.

ABSTRACT

Many organisations have adapted the current digital technology to transform their business models to gain competitive advantage in the marketplace. Euro Potential Sdn Bhd is a non-IT company trying to digitalise its business operations. Currently, the company operates a traditional business model which the salesperson communicating with customer through phone calls, physical meetings, or messengers. In this project, the web portal is developed to solve the problems faced by the company. The Agile methodology was implemented to match the project objectives. Agile methodology with multiple iterations planned for the project development. Each iteration will focus on the development of the module. This project has divided into several modules which are Procurement and Shipment module, Quote module, Consumption module, Project Development module and Product module. The Kanban methodology was utilised as the Agile framework to manage the development progress. The technology used to develop the system is the MERN stack. MERN stands for MongoDB, Express, React.js, Node, which are the four key technologies built as a three-tier architecture. In conclusion, the primary module of the project that will be established for management and salesperson to manage order progress. The module used to solve the company's difficulties and achieve the transformation of business operations.

TABLE OF CONTENTS

TITLE PAGE	i
REPORT STATUS DECLARATION FORM	ii
FYP THESIS SUBMISSION FORM	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xiii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement and Motivation	1
1.2 Objectives	2
1.3 Project Scope and Direction	3
1.4 Contributions	4
1.5 Report Organization	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 Review of the Technologies	5
2.1.1 Front-end Framework	5
2.1.2 Database	8
2.2 Review of the Existing Systems	10
2.2.1 Odoo	10
2.2.2 WISDM Product Enquiry Pro for WooCommerce	14
2.2.3 Order Management System from Site Impact	16
CHAPTER 3 SYSTEM METHODOLOGY	18
3.1 Design Specifications	18
3.1.1 Agile Methodologies	18

3.1.2	Kanban framework	19
3.1.3	Technologies Involved	20
3.2	System Design Diagram	22
3.2.1	System Architecture Diagram	22
3.2.2	Entity Relationship Diagram	24
3.2.3	Use Case Diagram	25
3.2.4	Use Case Description	26
3.2.5	Procurement and Shipment Module	37
3.3	Timeline	39
 CHAPTER 4 SYSTEM DESIGN		 41
4.1	System Block Diagram	41
4.2	System Flow Diagram	42
4.3	Database Design	49
4.4	User Interface Design	54
 CHAPTER 5 SYSTEM IMPLEMENTATION		 59
5.1	Hardware Setup	59
5.2	Software Setup	59
5.2.1	Visual Studio Code	59
5.2.2	Node.js	60
5.2.3	MongoDB	60
5.3	Setting and Configuration	61
5.3.1	React.js	61
5.3.2	Express.js	61
5.3.3	Configuration between React.js, Express.js and MongoDB	62
5.4	System Operation (with Screenshot)	63
5.4.1	Login and Order Module	63
5.4.2	Company Module	72
5.4.3	Consumption Module	73
5.4.4	Quote Module	76
5.4.5	Product Module	78
5.4.6	News Module	79

5.4.7 Notification Module	82
5.5 Implementation Issues and Challenges	83
5.6 Concluding Remark	84
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION	85
6.1 System Testing and Performance Metrics	85
6.2 Testing Setup and Result	86
6.3 Project Challenges	93
6.4 Objectives Evaluation	93
6.5 Concluding Remark	94
CHAPTER 7 CONCLUSION AND RECOMMENDATION	95
7.1 Conclusion	95
7.2 Recommendation	95
REFERENCES	96
APPENDIX	99
WEEKLY LOG	99
POSTER	105
PLAGIARISM CHECK RESULT	106
FYP2 CHECKLIST	111

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	User interface for the web version of mobile application	2
Figure 2.1.1	Angular Architecture Diagram	7
Figure 2.2.1.1	User Interface for the quotation list of Odoo	10
Figure 2.2.1.2	User interface for the histogram table of Odoo	11
Figure 2.2.1.3	User Interface for edit quotation on Odoo	12
Figure 2.2.1.4	Invoice template generated by Odoo	13
Figure 2.2.2.1	User Interface for order list of Product Enquiry Pro	14
Figure 2.2.2.2	User Interface for edit quotation details on Product Enquiry Pro	14
Figure 2.2.2.3	Quotation details modal for send quote on Product Enquiry Pro	15
Figure 2.2.3.1	User Interface for create new order on OMS	16
Figure 2.2.3.2	User Interface for order details on Product Enquiry Pro	16
Figure 3.1.1	Agile Methodology Model	17
Figure 3.1.2	Kanban framework for project development	20
Figure 3.2.1	System Architecture	22
Figure 3.2.2	Entity Relationship Diagram for the project	24
Figure 3.2.3	Use Case Diagram for the project	25
Figure 3.2.4	Activity Diagram for procurement and shipment module	38
Figure 3.3.1	Timeline of the system development for FYP1	39
Figure 3.3.2	Timeline of the system development for FYP	40
Figure 4.1	System Block Diagram	41
Figure 4.2.1	Login Flow Diagram	42
Figure 4.2.2	Update Order Flow Diagram	43
Figure 4.2.3	Upload File Flow Diagram	44
Figure 4.2.4	Delete File Flow Diagram	45
Figure 4.2.5	Project Management Flow Diagram	46
Figure 4.2.6	Request Quotation Flow Diagram	47
Figure 4.4.1	Order Management Wireframe	48
Figure 4.4.2	Sales Dashboard Wireframe	55

Figure 4.4.3	Company List Wireframe	56
Figure 4.4.4	Company Profile Wireframe	56
Figure 4.4.5	Product List Wireframe	57
Figure 4.4.6	Consumption List Wireframe	57
Figure 4.4.7	Quote List Wireframe	58
Figure 5.3.3.1	Code for MongoDB Connection using Mongoose	62
Figure 5.3.3.2	MongoDB Connection Success Message	62
Figure 5.3.3.3	Mongoose Schema for orderdb Database	63
Figure 5.3.3.4	Code for Requesting URI in React.js using Axios	63
Figure 5.4.1.1	Role-Based Login Page	64
Figure 5.4.1.2	Order Page with Successful Authentication	64
Figure 5.4.1.3	Order Modal for Order Quotation	65
Figure 5.4.1.4	Order Page	65
Figure 5.4.1.5	Order Modal for Purchase Order Confirmation	66
Figure 5.4.1.6	Updated Order Record after Confirm Purchase Order	66
Figure 5.4.1.7	Order Modal with Selected Shipment Details	67
Figure 5.4.1.8	Order Modal for Mark Order as Completed	67
Figure 5.4.1.9	Updated Order Record with Order Completed Status	68
Figure 5.4.1.10	Order Modal for Upload Files	68
Figure 5.4.1.11	Order Modal with Uploaded Files	69
Figure 5.4.1.12	File Delete Confirmation Pop Up Modal	69
Figure 5.4.1.13	Order Modal Display after File Deleted	70
Figure 5.4.1.14	Monthly Sales Dashboard	70
Figure 5.4.1.15	Order Records Display in the Chatbot	71
Figure 5.4.2.1	Company List Page	72
Figure 5.4.2.2	Customer Profile Dashboard	72
Figure 5.4.2.3	Project Modal with Record	73
Figure 5.4.3.1	Consumption List Page	73
Figure 5.4.3.2	Add Consumption Modal	74
Figure 5.4.3.3	Consumption List with Updated Record	74
Figure 5.4.3.4	Supplier View Consumption Access Modal	75
Figure 5.4.3.5	Excel File with Exported Consumption List	75
Figure 5.4.4.1	Request List Page	76

Figure 5.4.4.2	Quote List Page	76
Figure 5.4.4.3	Request Quotation Modal	77
Figure 5.4.4.4	Request List with Newly Added Record	77
Figure 5.4.5.1	Product List Page	78
Figure 5.4.6.1	News Page	79
Figure 5.4.6.2	News Modal with New Input	79
Figure 5.4.6.3	Newly Added News	80
Figure 5.4.6.4	News Viewed by Users Record	81
Figure 5.4.7.1	Hover-Over Icon Notification Modal Diagram	82
Figure 5.4.7.2	Notification List Page	83

LIST OF TABLES

Table Number	Title	Page
Table 2.1.1	Pros and Cons of React, Angular, and Vue.js	8
Table 3.1.3	Tools and Software	21
Table 3.2.4.1	Login Use Case Description	26
Table 3.2.4.2	Update Order Date Use Case Description	27
Table 3.2.4.3	Update File Use Case Description	28
Table 3.2.4.4	Manage Project Development Use Case Description	29
Table 3.2.4.5	Add New Project Development Use Case Description	30
Table 3.2.4.6	Update Project Details Use Case Description	31
Table 3.2.4.7	Request Quotation Use Case Description	32
Table 3.2.4.8	View Consumption List Use Case Description	33
Table 3.2.4.9	Add New Consumption Use Case Description	34
Table 3.2.4.10	Update Consumption Use Case Description	35
Table 3.2.4.11	Delete File Use Case Description	36
Table 4.3.1	Data Dictionary for User Entity	49
Table 4.3.2	Data Dictionary for Company Entity	49
Table 4.3.3	Data Dictionary for Product Entity	50
Table 4.3.4	Data Dictionary for Order Entity	51
Table 4.3.5	Data Dictionary for Consumption Entity	52
Table 4.3.6	Data Dictionary for Project Entity	52
Table 4.3.7	Data Dictionary for Quotation Entity	53
Table 5.1.1	Computer Specifications	59
Table 6.2.1	Login Test Case	86
Table 6.2.2	Order Module Test Case	87
Table 6.2.3	Project Management Test Case	89
Table 6.2.4	Consumption Management Test Case	90
Table 6.2.5	Quote Management Test Case	91

LIST OF ABBREVIATIONS

<i>API</i>	Application programming interface
<i>APP</i>	Application
<i>DOM</i>	Document Object Model
<i>ES6</i>	ECMAScript 6
<i>ETC</i>	Et cetera
<i>HTML</i>	Hypertext Markup Language
<i>HTTP</i>	Hypertext Transfer Protocol
<i>IT</i>	Information technology
<i>JS</i>	JavaScript
<i>JSX</i>	JavaScript Syntax Extension
<i>NPM</i>	Node Package Manager
<i>ODM</i>	Object Document Mapping
<i>RAM</i>	Random Access Memory
<i>SDN BHD</i>	Sendirian Berhad
<i>SQL</i>	Structured Query Language
<i>URI</i>	Uniform Resource Identifier
<i>URL</i>	Uniform Resource Locators

Chapter 1

Introduction

This chapter will introduce the problem that the project aims to solve with the motivation and outline the problem scope and project objectives.

1.1 Problem Statement and Motivation

With the development of technology, companies across several industries are embracing technology to enhance the working efficiency and minimise workload. Euro Potential Sdn Bhd was one of the organizations that was restructuring its operations. The company applies with the traditional business model which the salesperson communicating with customer through phone calls, physical meetings, or messengers. The phone as a communication tool may misinterpret customer requests due to loud background sounds and accent issues. To overcome this issue, a system that lets users enquiry products using text can avoid false indications to the company.

The company recently engaged a team to develop the mobile application. Although many features have been developed in the app, there are still many functions that have not been completed, especially the order and shipment modules. There is lack of the connection between two collections in the database. Each shipment can deliver several orders based on the load container. If both collections do not have relations, user unable find which order has associated with the shipment, so they are not able to arrange the shipment.

A system that could adapt to several types of devices was critical to the project's success, especially for a business software. Based on the article written by Krunal Vyas in [1], an IT consultant has justified that web portal should be the first option to grow up the digital business. Web portals are more flexible than mobile applications since mobile applications cannot be accessed directly on computers without the assistance of an external tool. The corporation realised that the mobile application did not match the business needs.

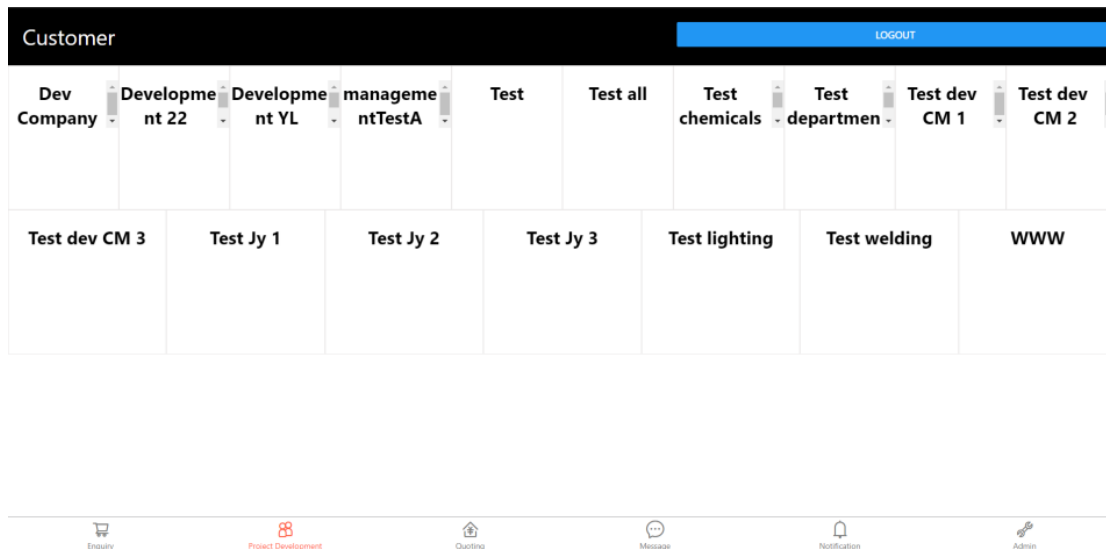


Figure 1.1 User interface for the web version of mobile application

To solve this issue, The mobile application developer has transformed the mobile application to web version using Expo. The system can now be accessible through browser, however due to the website design standards differed from the mobile application, the web version of the mobile application violated the website design principles. The Figure 1.1 has shown that the navigation bar was at the bottom, the arrangement of text did not well adjusted, the space between the letters is different. The website has an unsatisfied user experience and the organiser decided to create a web portal to support the functions with appropriate user interface.

The motivation of this project is to develop a web portal for the salesperson and management department of a non-IT company which can easily to perform the job tasks on the web portal to replace the traditional operations. The staff and customer can adapt to the new business model and shorten the time for information sharing to increase the efficiency.

1.2 Objectives

1. To develop a user-friendly web portal with customizable role-based access controls, designed to streamline the procurement and shipment process, and provide an efficient alternative to traditional business operations.
2. To develop an integrated order and shipment module that allows users to manage the entire order progress from a single platform, including order placement, shipment tracking, and delivery confirmation.

3. To ensure data consistency and accuracy between the mobile application and web portal on the same module by implementing real-time synchronization and data validation mechanisms, minimizing data discrepancies, and ensuring data integrity.

1.3 Project Scope and Direction

The scope of this project is to develop a web application using the MERN that provides a comprehensive procurement and shipment management system with additional modules for quotes, consumption, project development, and product management. The project will be developed with reference to an existing mobile application. The connections between web portal and mobile application are APIs. There are several APIs to control on different modules and connect to MongoDB. The web application will be designed to be responsive and user-friendly, ensuring a consistent user experience across devices.

The main module that helps to enhance the company operation is procurement and shipment module. The Procurement and Shipment module will include functionalities for managing order information, order tracking, and shipment tracking. The module shall replicate the existing procedures of the company and provide users with the capability to modify order details and monitor the progress of their orders. As an instance, a file upload feature will be created to manage documents such as invoices or receipts, which are typically required to be sent to customers upon order completion.

The project will include developing a robust data management system, ensuring data consistency and integrity across the web and mobile platforms. The web application will be designed to integrate seamlessly with the mobile application, allowing users to access their content and data across devices. Further functionalities shall be developed to enhance the effectiveness of the portal and promote user adoption and reliance on the platform.

The project will be developed using an Agile development methodology with regular progress updates and testing to ensure that the application meets the required specifications and quality standards. The existing mobile application will be used as a reference point throughout the development process to ensure consistency and alignment with the overall project objectives.

1.4 Contributions

As part of this project, a range of tasks should perform in accordance with the current business model of the company. Since the order process and required data will be different from the current existing system, this is significance on learn how to visualize the company's order process and design a complete set of processes. It is critical to enable the user to carry out the task without missing any necessary procedures. Furthermore, this project has provided how to verify data consistency between platforms. Users can update data on different platforms at the same time with the lower risk of data corruption. A synchronization of the data can be done between different platforms.

After the project was successfully completed, the organization was able to transform their traditional business model into a digital business. This is the opportunity to witness the information technology take the business to new heights, which the project will enable users to complete tasks more efficiently, resulting in increased productivity and reduced workload.

1.5 Report Organization

This report is organised into 7 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Discussion, Chapter 7 Conclusion. The first chapter is the introduction of this project which includes problem statement and motivation, project scope, project objectives, project contribution, and report organisation. Chapter two of the project shall comprise a literature review, which will entail a detailed analysis of existing products, with a comparative evaluation of their strengths and weaknesses. Furthermore, the chapter will conduct a comparison of various web development technologies to determine the most appropriate and effective approach for the project. The third chapter of the project shall focus on the system methodology employed in the development process, including the presentation of the entity-relationship diagram, use case diagram, and the respective descriptions. The fourth chapter introduces the system design approach, which includes the system flow diagram, wireframes, and the database design. The fifth chapter will give a comprehensive introduction to the implementation process of the system, including the operational details of the system. The sixth chapter will demonstrate the system evaluation process and present an in-depth discussion of the challenges encountered during the project's development. The final chapter summarises the project and provides recommendations for future development.

Chapter 2

Literature Review

In this chapter, three existing similar systems and web development techniques are evaluated. The assessment focuses on comparing the features and functions of systems.

2.1 Review of the Technologies

2.1.1 Front-end Framework

A development framework is a structured and standardized approach to software development that provides developers with a set of tools, libraries, and guidelines to build software applications more efficiently. It helps simplify the development process and increases consistency and quality throughout the development lifecycle. There are various types of web frameworks in the market, React.js, Vue.js, and Angular are three popular JavaScript frameworks for building web applications. This chapter will analyze these three frameworks and compare their advantages and disadvantages.

React.js is a popular open-source JavaScript library for building user interfaces [2]. It was created by Facebook and the developer community manages updates [3]. Its declarative approach, component-based architecture, and virtual DOM make it a versatile and efficient framework for web development. Developers can describe what the UI should look like and React.js will take care of updating the actual UI, making it easier to reason about the state of the application and reduce the likelihood of bugs [4]. The component-based architecture allows developers to create reusable UI elements that can be used across different parts of an application, making it easier to maintain and update the application.

One of the key features of React.js is its use of the Virtual DOM, a lightweight, in-memory representation of the actual browser DOM [5]. By using the Virtual DOM, React.js can efficiently update the actual browser DOM, resulting in a fast and smooth user experience. The Virtual DOM is also platform-independent, making it possible to use React.js to build applications that run on multiple platforms, including the web, mobile devices, and desktop applications. Additionally, React.js uses the Virtual DOM provides a simplified programming model, allowing developers to focus on writing clean and maintainable code, without worrying about the complexities of managing the browser DOM [5]. React.js has a large ecosystem of libraries and tools that make it easy to build and maintain web applications. React.js can also be used to build mobile and desktop applications, making it a versatile framework.

However, React.js also has its weaknesses. JSX, React.js syntax extension to JavaScript, can make it difficult for developers who are not familiar with it to understand React.js code [5]. React.js is also not designed to handle other aspects of web development, which only supported for View layer in model-view controller [5]. Developers will need to use other tools or libraries to handle server-side rendering or database access such as Express.js to perform the Model and the Controller layer. The resulting bundle size can also be large due to React.js component-based architecture and virtual DOM, which can slow down the initial load time of a web application. Finally, relying on a large ecosystem of third-party libraries and tools can lead to compatibility issues and security vulnerabilities if developers use unstable or unmaintained libraries [6].

Vue.js is a popular front-end JavaScript framework used for building single-page applications and user interfaces [7]. Its strengths lie in its simplicity, ease of use, and flexibility. Vue.js offers a gentle learning curve, making it easy for developers to get started with the framework, even if they are not experienced in web development. It also provides a clear and concise documentation, which is highly praised by developers.

Vue.js possesses a further advantage in its lightweight nature. Specifically, upon minification and compression, its size ranges from 18-21KB, facilitating seamless integration with alternative libraries and frameworks [8]. Furthermore, Vue.js utilises a data binding system like React.js, whereby the changes to data are automatically reflected in the user interface. This functionality simplifies the management of application state for developers and reduces the risk of programming errors. Additionally, Vue.js employs a component-based architecture, enabling developers to establish reusable UI components. This approach streamlines application maintenance and updating processes. Notably, Vue.js proffers an expanding ecosystem of tools and libraries that enhances its interoperability with other technologies [9].

However, Vue.js has some weaknesses that developers should consider. Its ecosystem is not as large as React.js or Angular, which means that finding specific libraries and tools may be more challenging [8]. Additionally, Vue.js' performance may suffer when dealing with large-scale applications with many components, which can be mitigated by using server-side rendering or optimizing the application's architecture [8].

Angular is a popular front-end web development framework that is maintained by Google. Its strengths lie in its highly scalable architecture, strong community support, and the ability to handle complex data models. Angular offers a complete solution for building complex

and large-scale web applications, with features such as routing, dependency injection, and state management built in [10].

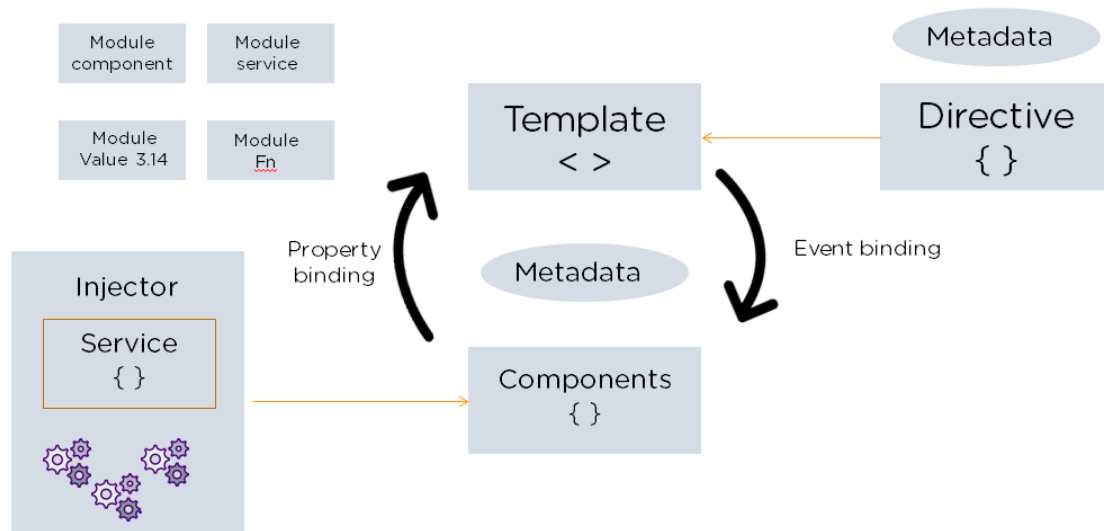


Figure 2.1.1 Angular Architecture Diagram

Source: Adapted from [11]

AngularJS is utilized in web applications to create a modular and maintainable application. The goal is to simplify development, maintenance, and testing by incorporating model-view controller (MVC) functionality in the web application development process. The Figure 2.1.1 shows the distinctive characteristic of AngularJS is its ability to enable automatic synchronization of data between model and view components, referred to as two-way data bindings. The two-way data binding allows changes made to the data model to be immediately reflected in the user interface and vice versa.

Angular's modular architecture allows developers to break down complex applications into smaller, reusable modules. This makes it easy to manage the application's complexity and scale the application as needed [11]. Angular's dependency injection system is highly praised, making it easy to access shared data models and services. This makes it easy to manage complex data models across different parts of an application and ensures that the data is always up to date and consistent [12]. Developers can inject dependencies into components and services, which helps to reduce duplication of code and improves code maintainability. Additionally, Angular offers comprehensive documentation, which makes it easier for developers to learn and get started with the framework. The Angular community is also very active, providing a wealth of resources and support for developers.

However, Angular's main weakness is its steep learning curve. Due to its intricate design and comprehensive feature set, inexperienced web developers may encounter

Bachelor of Information Systems (Honours) Business Information Systems
Faculty of Information and Communication Technology (Kampar Campus), UTAR

difficulties acquiring proficiency with this technology [12]. Additionally, Angular's performance may suffer when dealing with large-scale applications. This can be mitigated by optimizing the application's architecture, but it may require additional effort and resources. Finally, while Angular offers a large ecosystem of libraries and tools, it is not as popular as React.js or Vue.js, which means that finding specific libraries and tools may be more challenging.

Table 2.1.1 Pros and Cons of React, Angular, and Vue.js

Framework	Strengths	Weaknesses
React.js	<ul style="list-style-type: none"> - Simplistic and lightweight - Component-based architecture with virtual DOM - Large ecosystem of libraries and tools 	<ul style="list-style-type: none"> - JSX syntax can be difficult to understand - Not designed for server-side rendering or database access - Large bundle size
Vue.js	<ul style="list-style-type: none"> - Simple and easy to use - Lightweight with a small bundle size - React.js activity system for easy state management - Growing ecosystem of libraries and tools 	<ul style="list-style-type: none"> - Ecosystem not as large as React.js or Angular - Performance may suffer with large-scale applications
Angular	<ul style="list-style-type: none"> - Highly scalable architecture - Built-in features for complex applications - Active community with comprehensive documentation 	<ul style="list-style-type: none"> - Steep learning curve - Performance may suffer with large-scale applications - Ecosystem not as popular as React.js or Vue.js

Choosing a front-end framework should prioritize selecting one that fits the project requirements, rather than simply going with the most popular option. For this project, React.js was the ideal choice for creating a cross-platform system that ensures consistency of functionality between the web and mobile applications, while also being easy to maintain.

2.1.2 Database

A database is a structured collection of data that can be accessed, managed, and updated easily.

When it comes to web development, databases play a crucial role in storing and retrieving data

from web applications. They provide a way to store large amounts of data in an organized and efficient manner, and make it possible to search, filter, and manipulate that data in real-time. The review of the database was conducted based on scalability, performance, and flexibility.

MySQL is a traditional relational database management system (RDBMS) that has been around for decades. It is designed to handle large volumes of structured data, making it a popular choice for applications that require complex queries and transactions. MySQL can scale vertically by adding more powerful hardware, or horizontally by sharing the database across multiple servers. MySQL is considered one of the most secure database management systems that provides a range of security features to ensure the safety of sensitive data stored in the database [13]. These security features include encryption, user authentication, and access control mechanisms [14]. By using these features, MySQL helps to prevent unauthorized access, data theft, and other security breaches that can harm the confidentiality, integrity, and availability of data. While MySQL can handle a large amount of data, it may not be as scalable as some other database management systems, particularly when it comes to handling large volumes of data and complex transactions [15].

MongoDB is a NoSQL database management system that is designed to handle large volumes of unstructured data. It provides a flexible and scalable data model, distributed architecture, and automatic sharding to handle large volumes of data and high traffic loads. Sharding allows a MongoDB deployment to be divided into smaller parts called shards, which can be distributed across multiple servers or nodes [16]. This helps distribute the load and ensures that data is evenly distributed across the system. However, the unstructured data stored in database without joins functionalities can lead to data redundancy. For large projects with huge request traffic, it requires a high volume of storage. To address this issue, MongoDB provides several features to help manage data redundancy and optimize storage space. One such feature is indexing, which allows developers to define indexes on specific fields in a collection. Indexing can be a limitation when the implementation is set incorrectly and results in poor performance [17].

Firestore is a cloud-native NoSQL document database that is designed to be highly scalable and highly available. It is part of Google Cloud Platform and is designed to work seamlessly with other Google Cloud services. Developers can easily adopt Google services to enhance the value of their projects. Google Firebase is a serverless platform that is simple to implement and requires low-capability equipment [18]. However, it's worth noting that Firebase's cloud storage is heavily reliant on network connectivity, and a weak connection may

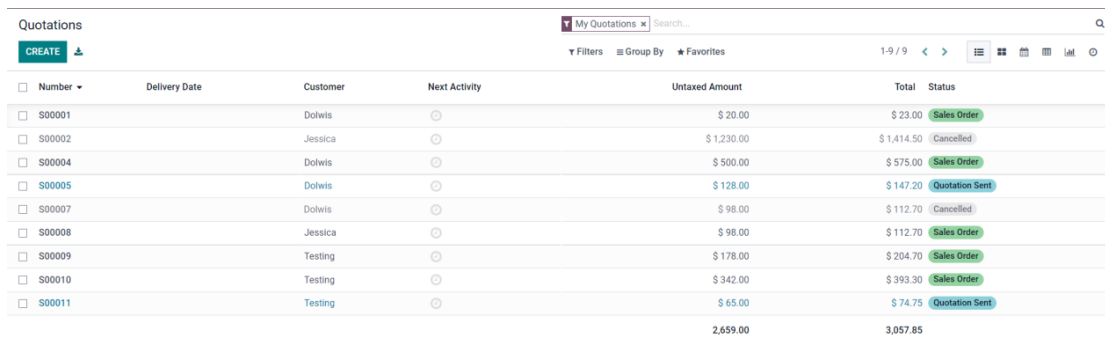
lead to lower performance of traffic [19]. Additionally, Firebase may not be the best option for supporting complex queries. Despite this, Firebase remains an excellent choice for developers looking for a simple, low-maintenance solution for their projects.

In general, both MongoDB and Firestore are designed to be highly scalable and flexible, making them ideal for applications that need to handle large volumes of unstructured data. MySQL, on the other hand, is more suited to applications that require complex queries and transactions and can benefit from the advantages of a traditional RDBMS.

2.2 Review of the Existing Systems

2.2.1 Odoo

Odoo is a strongly business management software that provide widely tools for company to integrate with the services based on their business need. It provided mobile application and software as a service that user can use the service through different devices. User can manage inventory, project, process the deals, or accounting using Odoo modules. This research focusing on sales module that provide by Odoo platform.



Number	Delivery Date	Customer	Next Activity	Untaxed Amount	Total	Status
S00001		Dolwis		\$ 20.00	\$ 23.00	Sales Order
S00002		Jessica		\$ 1,230.00	\$ 1,414.50	Cancelled
S00004		Dolwis		\$ 500.00	\$ 575.00	Sales Order
S00005		Dolwis		\$ 128.00	\$ 147.20	Quotation Sent
S00007		Dolwis		\$ 98.00	\$ 112.70	Cancelled
S00008		Jessica		\$ 98.00	\$ 112.70	Sales Order
S00009		Testing		\$ 178.00	\$ 204.70	Sales Order
S00010		Testing		\$ 342.00	\$ 393.30	Sales Order
S00011		Testing		\$ 65.00	\$ 74.75	Quotation Sent
				2,659.00	3,057.85	

Figure 2.2.1.1 User Interface for the quotation list of Odoo

Source: Adapted from [20]

The sales module provided high adaptive features which the view of quotations can be change to different template such as Kanban, Calendar, Pivot and Graph. User can have different perspective and use the templates based on their preferences. Even though there have several templates to view the quotations, but the main purpose of the templates was summarised the quotations. For example, Calendar summarised the records based on the creation date, user easy to track the quotation of the day.

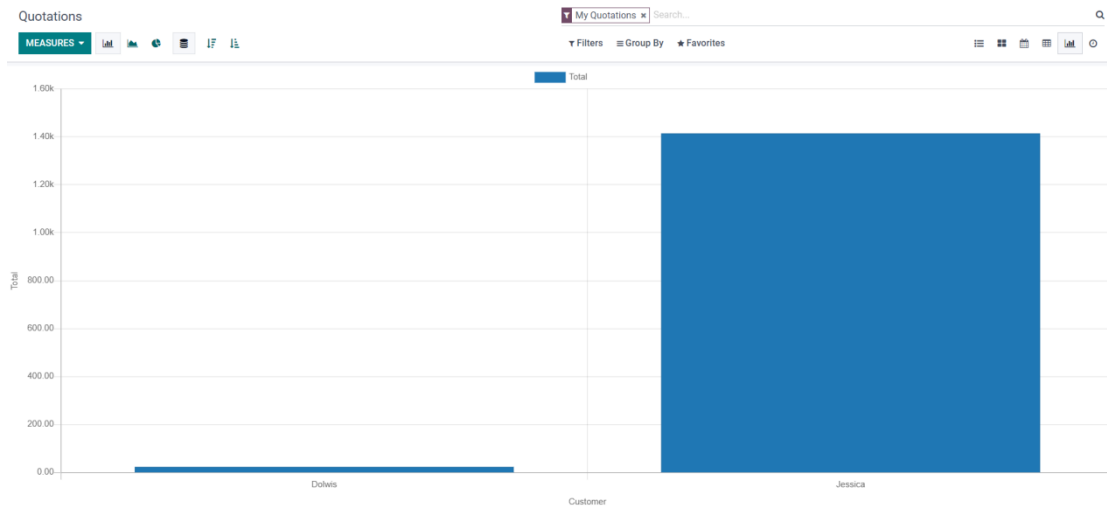


Figure 2.2.1.2 User interface for the histogram table of Oddo

Source: Adapted from [20]

The Figure 2.2.1.2 shown the statistic of total amount ordered by each customer. It helps company can analysis the sales and make further decision. The list template was the main template to keep track the quotations on computer. For the mobile version, Kanban as known as list view which more suitable for tracking quotations. The table list cannot fit the records in one screen at mobile phone. It allows to export the quotations as excel file based on the records display on the table list, but this function only works for web version. The table list can be customised which to hide and show some fields.

Quotations / New

SAVE DISCARD

SEND BY EMAIL CONFIRM

QUOTATION QUOTATION SENT SALES ORDER

Customer Preview

New

Customer: Testing, Other Address
123, Develop, XX Develop, 00990, DDVV.
Testing
Iph 30020

Expiration: _____
Pricelist: Public Pricelist (USD)
Payment Terms: _____

Order Lines Optional Products Other Info

Product	Description	Quantity	Unit Price	Taxes	Subtotal
Add a product	Add a section	Add a note			
Terms and conditions...					Total: 0.00

Figure 2.2.1.3 User Interface for edit quotation on Odoo

Source: Adapted from [20]

Although it allows customer to accept or reject the quotation, it not able to let customer enquiry product, the order flow was started on sales create a quotation and the quotation will send through email to the customer. Customers need to click the link provided on the email to make decision on accept or reject the quotation. There are provided a feedback form to let salesperson understand the reason of customer. The status will be updated based on the action taken by customer. The mailbox used as the main communication platform for salesperson informing customer and send the documents.

<p style="font-size: 24px; margin: 0;">Order # S00009</p> <p>Salesperson: Jessica</p>	<p>Testing 123, Develop, XX Develop, 00990, DDVV. Testing Ipoh 30020</p>																									
<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Description</th> <th style="text-align: left;">Quantity</th> <th style="text-align: left;">Unit Price</th> <th style="text-align: left;">Taxes</th> <th style="text-align: left;">Amount</th> </tr> </thead> <tbody> <tr> <td>Testing</td> <td>1.00 Units</td> <td>178.00</td> <td>15.00%</td> <td>\$ 178.00</td> </tr> <tr> <td colspan="4" style="border-top: 1px solid black;">Untaxed Amount</td> <td>\$ 178.00</td> </tr> <tr> <td colspan="4" style="border-top: 1px solid black;">Tax 15%</td> <td>\$ 26.70</td> </tr> <tr> <td colspan="4" style="border-top: 1px solid black;">Total</td> <td>\$ 204.70</td> </tr> </tbody> </table>	Description	Quantity	Unit Price	Taxes	Amount	Testing	1.00 Units	178.00	15.00%	\$ 178.00	Untaxed Amount				\$ 178.00	Tax 15%				\$ 26.70	Total				\$ 204.70	<p>Signature</p> <p style="font-size: 36px; font-family: cursive; margin: 0;"><i>Testing</i></p> <p>Testing</p>
Description	Quantity	Unit Price	Taxes	Amount																						
Testing	1.00 Units	178.00	15.00%	\$ 178.00																						
Untaxed Amount				\$ 178.00																						
Tax 15%				\$ 26.70																						
Total				\$ 204.70																						

Figure 2.2.1.4 Invoice template generated by Odoo

Source: Adapted from [20]

The order flow is simple and quick but is more suitable for those company just startup the business without mature business model. The invoice template can simply create by generating based on the quotation details, but the layout not able to do customisation. The company might not be easy to adapt with new invoice template.

2.2.2 WISDM Product Enquiry Pro for WooCommerce

ID	Status	Item	Customer Name	Customer Email	Enquiry Date	Total	Order #
1528		1	John Doe	john.doe@example.com	2022-08-26 03:31:40	\$150.00	-
1527		1	John Doe	john.doe@example.com	2020-10-05 10:17:26	\$158.00	-
1526		1	John Doe	john.doe@example.com	2020-10-05 09:38:43	\$79.00	-
1525		1	admin	emma.smith@example.com	2020-10-05 09:38:45	\$1,500.00	-
1524		1	admin	mrunmayee.kulkarni@wisdmilabs.com	2020-10-01 05:27:35	\$1,500.00	-
1523		1	John Doe	john.doe@example.com	2020-09-23 10:29:39	\$599.00	-
1522		1	quoteup_admin	mrunmayee.kulkarni@wisdmilabs.com	2020-09-22 07:18:56	\$599.00	-
1521		1	John	john.doe@example.com	2020-09-21 12:06:00	\$4,000.00	-
1520		1	quoteup_admin	shalesh.vishwakarma@wisdmilabs.com	2020-09-17 12:23:15	\$12,000.00	-
1519		2	John Doe	john.doe@example.com	2020-08-14 09:32:39	\$1,324.00	-

Figure 2.2.2.1 User Interface for order list of Product Enquiry Pro

Source: Adapted from [21]

WISDM provided WooCommerce a product enquiry and quote request plugin that able to let customer enquiry the product and let administrator manage the request. The Figure 2.2.2.1 shows the main page of product enquiry system for user handling orders. Administrator handling the request by generate the quotation after customer send the product enquiry. When admin click for the details, there will show the customer data to make admin have further understanding on status customer.

Customer Data

Client's Full Name: John Doe
 Client's Email Address: john.doe@example.com
 Client's IP Address: 113.25.188.254
 Billing Date: Aug 26, 2022
 Terms and conditions accepted: yes

Product Details

Item	Variations	SKU	Price	New Price	Quantity	Amount(\$)
Black Hoodie	-	-	\$150.00	150	1	\$150.00
Total						\$150.00

Search for a product: Add Product(s)
 Expiration Date: September 24, 2022
 Show Old Price in Quotation Show Price Difference in Quotation

Figure 2.2.2.2 User Interface for edit quotation details on Product Enquiry Pro

Source: Adapted from [22]

Since there have default price when customer enquiry, admin should adjust it and the system will calculate the percentage on the two different amounts. Admins need to set the expiration date to avoid user accept the quotation out of the given date. It can generate a Bachelor of Information Systems (Honours) Business Information Systems Faculty of Information and Communication Technology (Kampar Campus), UTAR

quotation form in PDF file when administrator finished the procedure and send it to customer via email. Customer can see the quotation details on the site but not allow to export as PDF file. Either customer accept or reject, the status will update to the admin panel to let admin take the next action. Admin can check the icon to identify the status, but it required the time to let user recognise the icon.

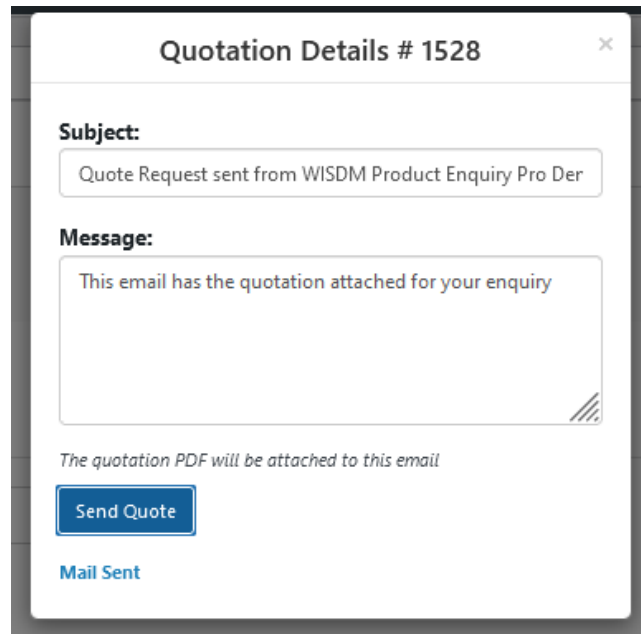


Figure 2.2.2.3 Quotation details modal for send quote on Product Enquiry Pro

Source: Adapted from [22]

Instead of other system, it allows enquiries for multiple products at the same order. It has a nice feature for customer save a copy when they submit the enquiry form. Customer can select whether they need a copy that send through email. This helps customer who prefer to have a backup and avoid the records missing. For admin panel, the edit or delete record option will only appear on the specific row that mouse hovering. It nice for those users might checking with wrong row and clicking to wrong records.

2.2.3 Order Management System from Site Impact

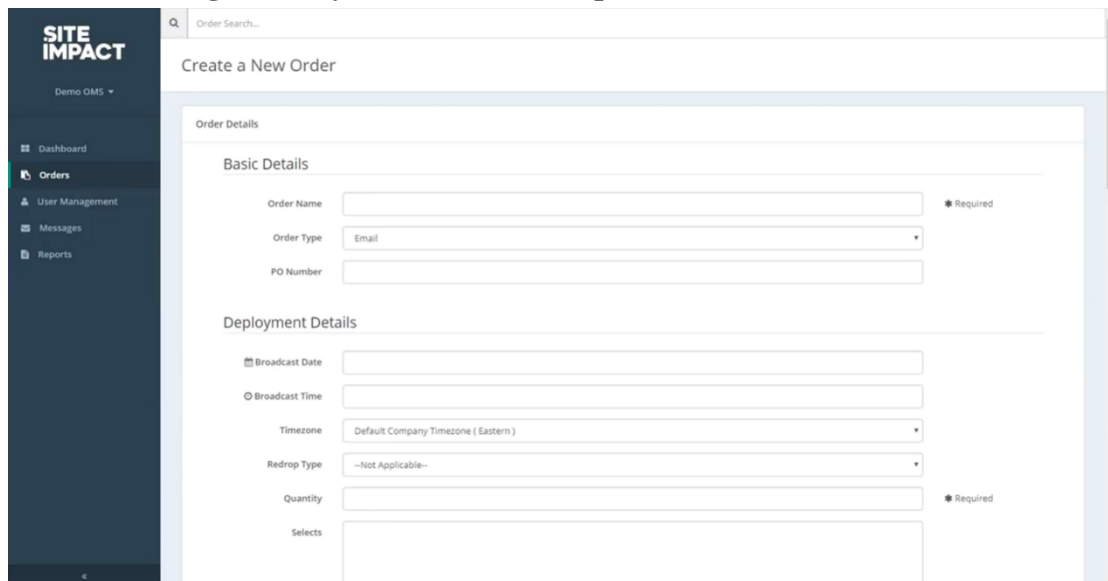


Figure 2.2.3.1 User Interface for create new order on OMS

Source: Adapted from [23]

Site Impact provided an order management system for company manage the order systematically. It starts the flow from admin create a new order by inserting the order details. There are several forms that need to key in such as deployment details, email testing details and additional details. After save the details, it will redirect to a new template that showing with the details inserted.

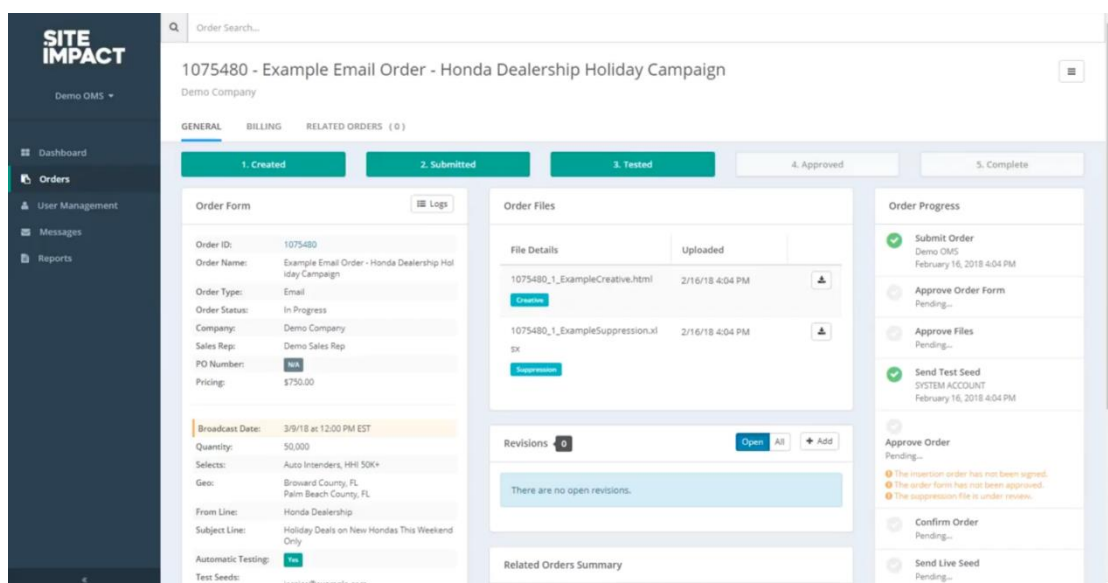


Figure 2.2.3.2 User Interface for order details on Product Enquiry Pro

Source: Adapted from [23]

Each order has an order details page that displays order details and tracks the progress of the order. The template has present nicely to let user get the information and understand the step of order clearly. It has an order progress bar that records the progress that has been made. Each progress displays the date issue and issuer. The step bar visualises the entire steps of the order, letting the user understand the steps of the order.

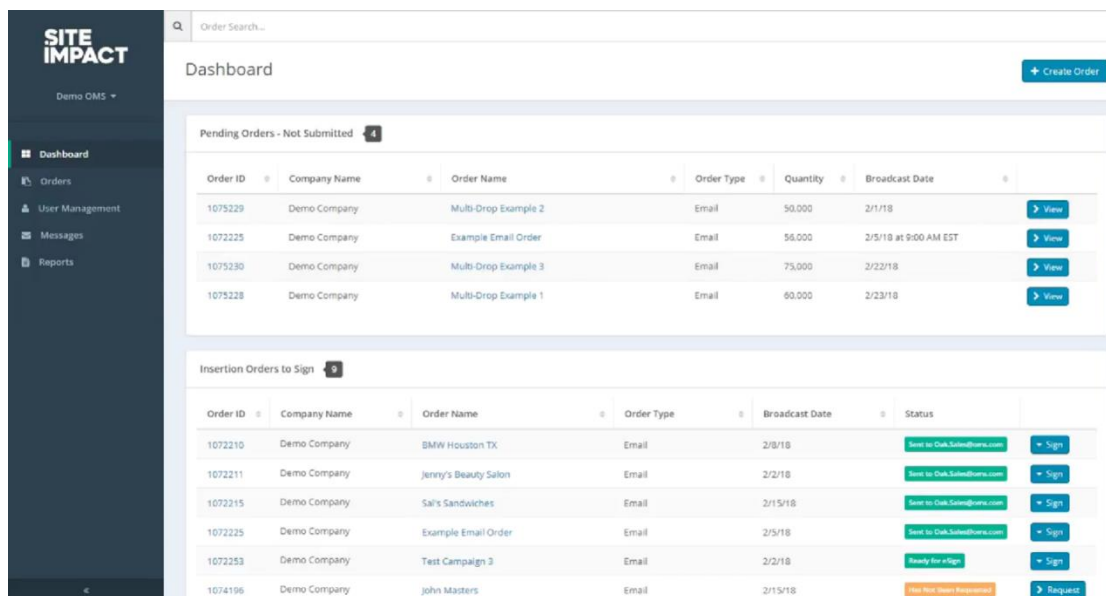


Figure 2.2.3.3 User Interface for dashboard on Product Enquiry Pro

Source: Adapted from [22]

On the dashboard, the order separated by table accordingly based on different progress. This arrangement is ideal for users looking for records. All required actions will be stated on the navigation buttons. Customer can quickly to understand the action of next step instead of retrieving the memory or figuring out the task.

Chapter 3

System Methodology

This chapter clarifies the use of project development methodology and the use of development tools. The system architecture diagram visualises the methods for system development.

3.1 Design Specifications

3.1.1 Agile Methodologies

The methodology applied in the project was agile methodology. Compared to traditional methodology, agile methodology provided more flexible and efficient way to deliver product. The traditional methodology used a linear approach where processes obtain in a predetermined order. The system requirements must plan on the initial stage, and it is hard to phrase back the previous phase after going to next stage. The deliverables may differ from the client's expectations because the customer is involved at both the initial and last stages. It is difficult, time-consuming, and expensive to go back and make the necessary corrections.

Agile methodology is an iterative approach to project management that is more adaptable and flexible. It is responding to the demands of the client or project rather than predicting the product from initial phase. According to the Agile Manifesto in [24], the Agile methodology emphasizes software development over comprehensive documentation. It facilitated the generation of new ideas and implement it throughout the development phase to enhance system feasibility. It compensates with the possibility that some requirements may have gone undetected throughout the planning process. Client collaboration was prioritised over contract negotiation in the Agile Manifesto. It enhanced customer engagement by encouraging them to be involved in all phases to ensure that the flow is as envisioned.

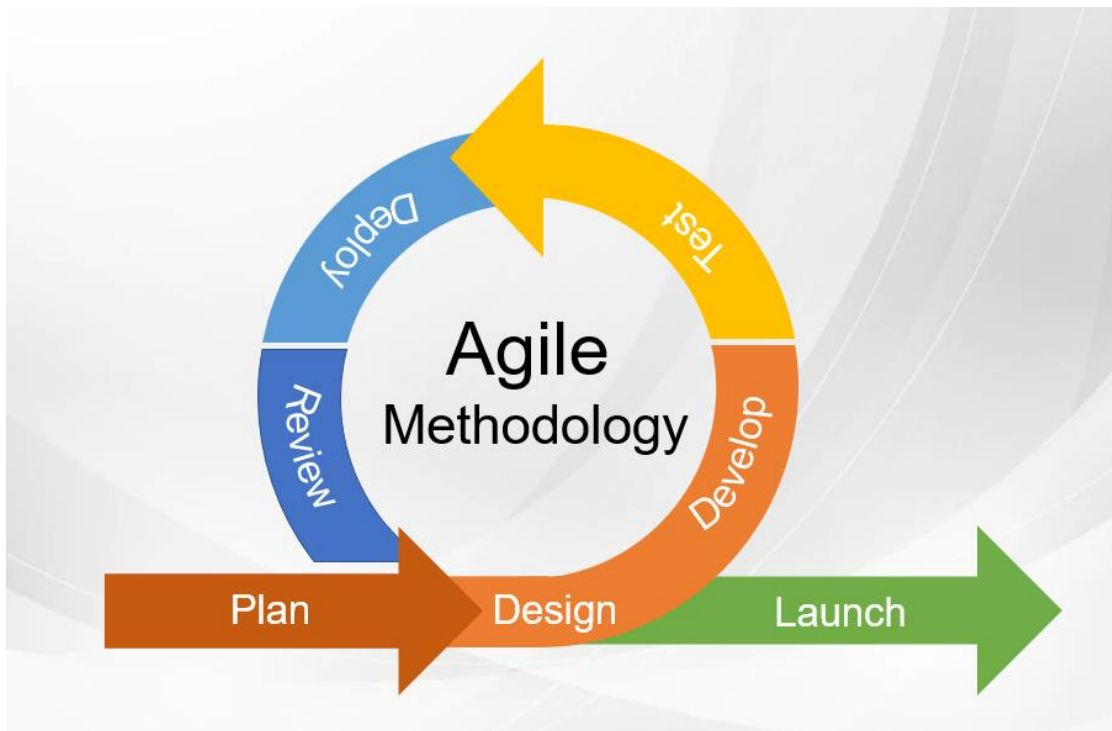


Figure 3.1.1 Agile Methodology Model

3.1.2 Kanban framework

There are several types of agile methodologies and Kanban is a framework for implementing agile software development. It can visualize the progress of a project by managing an individual's workflow. The diagram shows the Kanban framework for project development. It is divided into three categories, To Do, In Progress, and Done. By using the framework, it is easy to manage tasks and focus on ongoing tasks. Once the task is complete, it can easily drag and drop the task to the completed state. The tasks to be performed can be dragged and dropped into progress.

CHAPTER 3

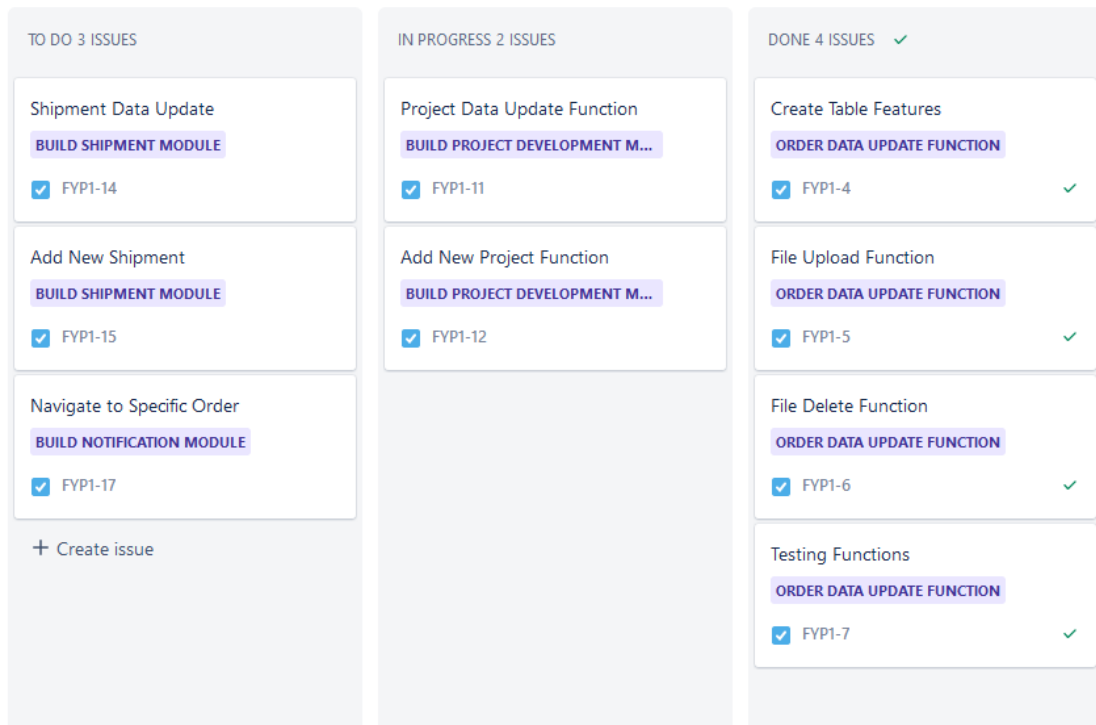


Figure 3.1.2 Kanban framework for project development

3.1.3 Technologies Involved

Visual Studio Code is a powerful source code editor that improves coding speed by highlighting incorrect syntax, installing extensions that support code completion. Tabnine is a plugin that helps predict and suggest the next line of code. It is an AI assistant that can be installed in Visual Studio Code as an extension. Visual Studio Code also supports multiple programming languages such as Java Script, C++, Python, and more. It meets the system requirements for setting up a React JS environment. React requires Node.js and npm to be installed to start the project. Node.js builds a full-stack JavaScript framework with front-end and back-end applications. The collaboration between npm and React.js is a key of success in developing the features. The reason is that npm provides thousands of packages that are convenient for web development, such as icon components, UI materials, table functions, etc.

Express JS is a Node.js web application framework for building server-side environments using APIs. It creates RESTful API that accept requests from the front end, and it can manage requests from mobile applications and web applications. Each API is used to connect to a specific database and store the data sent by the client. The

CHAPTER 3

client must retrieve the data by requesting the response from the API. Therefore, clients can share data while connecting to the same API.

Table 3.1.3 Tools and Software

No.	Type	Requirement
1	Development tool	Visual Studio Code
2	Server	Jenkins
3	Front-end library	React.js
4	Back-end framework	Express.js, Node.js
5	Database	MongoDB
6	Languages	HTML, Java Script, Type Script

3.2 System Design Diagram

3.2.1 System Architecture Diagram

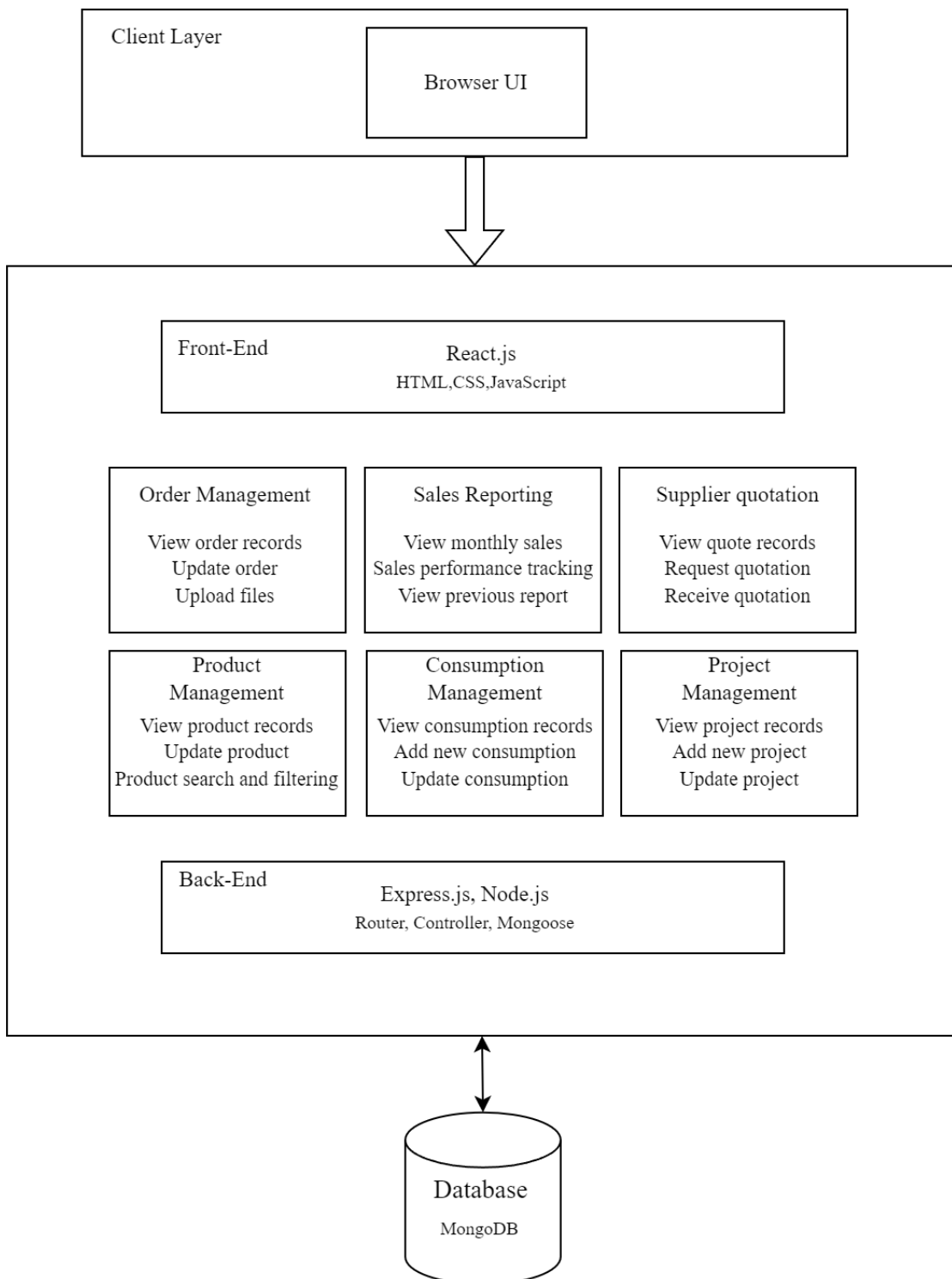


Figure 3.2.1 System Architecture

CHAPTER 3

This project adopts MERN stack as the main technology for building front-end website and back-end server. The Figure shows the overall system architecture model. The front-end web development applies React for creating dynamic interface. The system architecture comprises several components that perform distinct functions, including order management, sales reporting, supplier quotation, product management, consumption management, and customer management. These components work together seamlessly to provide a comprehensive solution that meets the system requirements. For example, the order management component allows users to view or update the order records, while the sales reporting component provides insights into sales data, such as revenue and profit margins. Similarly, the supplier quotation component facilitates the collaboration with suppliers and manages the quotation process. The product management component enables users to manage product information, such as prices, inventory, and status. The data existing in these components are required to be stored in a database through API Gateway.

Express.js plays a crucial role in receiving HTTP requests from the front-end, with the router forwarding accepted requests to the relevant controller. Controller is the bridge between router and Mongoose model. If controller receive a HTTP get method from Router, it contacts to Mongoose model to get the requested data and send back to browser. Mongoose model use Object Data Model to interact with MongoDB. Data can be converted from database format to JavaScript objects.

3.2.2 Entity Relationship Diagram

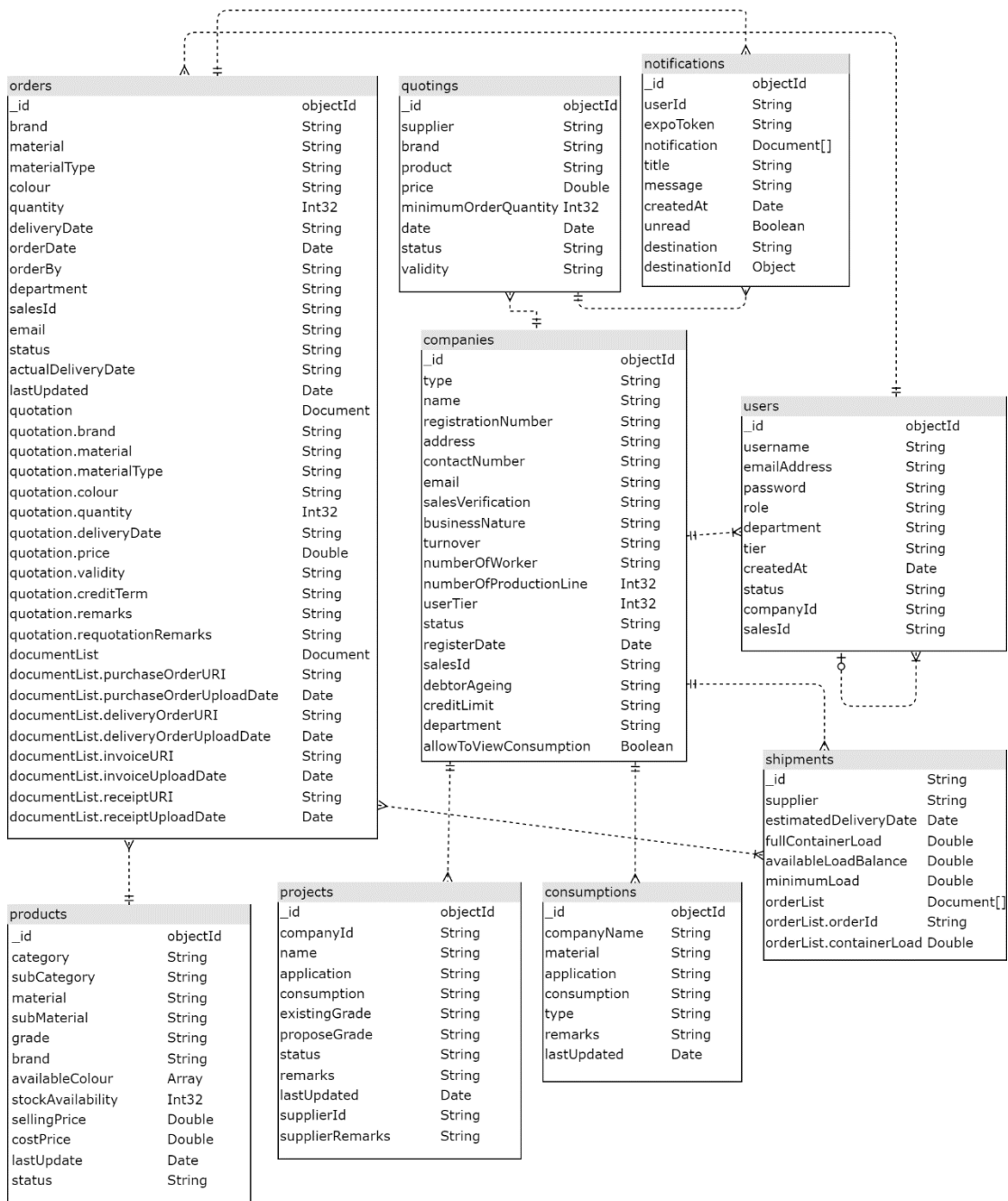


Figure 3.2.2 Entity Relationship Diagram for the project

3.2.3 Use Case Diagram

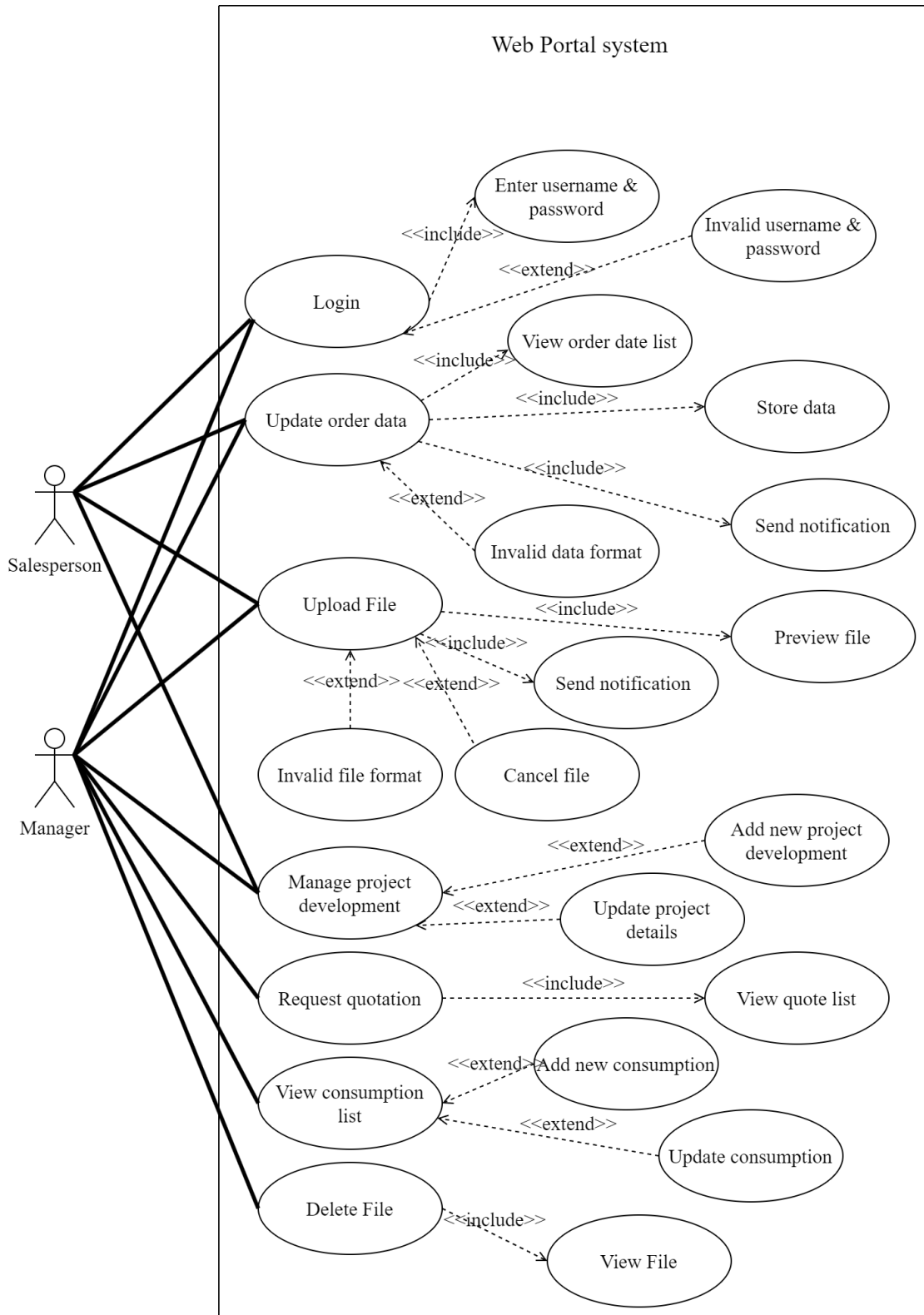


Figure 3.2.3 Use Case Diagram for the project

3.2.4 Use Case Description

Table 3.2.4.1 Login Use Case Description

Use Case ID	U3-2-4-1
Use Case Name	Login
Actor	Manager, Salesperson
Description	User login with an account to access the portal.
Preconditions	User must have an account.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User inserts username and password in the login form. 2. User clicks “Login” button. 3. System validates the credentials insert by user. 4. User can access the portal. 	
<p>Alternative Flow of Events:</p> <ol style="list-style-type: none"> 1.1 User inserts wrong username and password. <ol style="list-style-type: none"> 1.1.1 System indicates authentication failed and stay on login page. 	

Table 3.2.4.2 Update Order Date Use Case Description

Use Case ID	U3-2-4-2
Use Case Name	Update Order Data
Actor	Manager, Salesperson
Description	User update specific order data for further progress
Preconditions	User must login with the sales or management role account. User must receive an order.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User login with an account (Login Use Case). 2. User view all the order records. 3. User finds and clicks the icon on the specific order. 4. The order modal opens and show the order details. 5. User edit or insert the order data in the input form. 6. User clicks on action button. 7. System updates the data and send notification to customer. 	
<p>Alternative Flow of Events:</p> <ol style="list-style-type: none"> 1.1 User inserts wrong format for the data. <ol style="list-style-type: none"> 1.1.1 System indicates action invalid and stay on the order modal. 	

Table 3.2.4.3 Update File Use Case Description

Use Case ID	U3-2-4-3
Use Case Name	Upload File
Actor	Manager, Salesperson
Description	User upload file such as purchase order, delivery order, invoice, receipt to specific order.
Preconditions	User must login with the sales or management role account. Order must mark as completed.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User login with an account (Login Use Case). 2. User clicks the icon of the order that has completed. 3. The order modal opens and show the order details with file input form. 4. User clicks the input form and open the directory file. 5. User selects the file to upload and clicks the “open” button. 6. The file selected show in the form and user can clicks to preview the file. 7. User clicks on the “Upload” button. 8. System store the file and send notification to customer. 	
<p>Alternative Flow of Events:</p> <ol style="list-style-type: none"> 1.1 User inserts wrong format for the file. <ol style="list-style-type: none"> 1.1.1 System indicates wrong file format and reject the file. 1.2 User clicks on “Cancel” button. <ol style="list-style-type: none"> 1.2.1 The system clears the file data, and the user can select the file again. 	

Table 3.2.4.4 Manage Project Development Use Case Description

Use Case ID	U3-2-4-4
Use Case Name	Manage Project Development
Actor	Manager, Salesperson
Description	User manages project development listing
Preconditions	User must login with the sales or management role account.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User login with an account (Login Use Case). 2. User clicks on project development navigation button. 3. System redirects to project development page. 4. The system retrieves the data and displays the project development records associated with the account. 	
<p>Alternative Flow of Events:</p> <ol style="list-style-type: none"> 1.1 User adds new project development. <ol style="list-style-type: none"> 1.1.1 User clicks on “Add project” button. 1.1.2 User performs “Add New Project Development” Use Case. 1.2 User update project details. <ol style="list-style-type: none"> 1.2.1 User clicks on a specific row of project development record. 1.2.2 User performs “Update Project Details” Use Case. 	

Table 3.2.4.5 Add New Project Development Use Case Description

Use Case ID	U3-2-4-5
Use Case Name	Add New Project Development
Actor	Manager, Salesperson
Description	User adds new project development records
Preconditions	The user has access to the “Manage Project” use case. The user has the necessary permissions to add new projects.
Flow of Events:	
<ol style="list-style-type: none"> 1. The user selects the “Add New Project” option from the “Manage Project” use case. 2. A project development modal opens. 3. User can insert the data in the input form. 4. User clicks on the “Confirm” button. 5. System stores the new record and displays a confirmation message. 	

Table 3.2.4.6 Update Project Details Use Case Description

Use Case ID	U3-2-4-6
Use Case Name	Update Project Details
Actor	Manager, Salesperson
Description	User update the project development records
Preconditions	User must login with the sales or management role account. User must have existing project development record.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User finds and clicks on the icon for specific project development record from the “Manage Project” use case. 2. The project development modal opens and show the project development details. 3. User can change the data in the input form. 4. User clicks on the “Confirm” button. 5. System updates the record to replace previous record on the database. 	

Table 3.2.4.7 Request Quotation Use Case Description

Use Case ID	U3-2-4-7
Use Case Name	Request Quotation
Actor	Manager
Description	User request supplier to send quotation.
Preconditions	User must login with the management role account.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User login with a management account (Login Use Case). 2. User clicks on quote navigation button. 3. System redirects to quote page. 4. The system retrieves the data and displays the quote records with a list. 5. User clicks on “Request Quotation” button. 6. The quote modal opens. 7. User insert data in the input form. 8. User click confirm button. 9. Data stored in database and supplier can receive the data. 	
<p>Alternative Flow of Events:</p> <ol style="list-style-type: none"> 1.1 User inserts wrong format for the data. <ol style="list-style-type: none"> 1.1.1 System indicates action invalid and stay on the quote modal. 	

Table 3.2.4.8 View Consumption List Use Case Description

Use Case ID	U3-2-4-8
Use Case Name	View Consumption List
Actor	Manager
Description	User manages consumption listing
Preconditions	User must login with the management role account.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User login with an account (Login Use Case). 2. User clicks on consumption page navigation button. 3. System redirects to consumption page. 4. The system retrieves the data and displays the consumption records. 	
<p>Alternative Flow of Events:</p> <ol style="list-style-type: none"> 1.2 User adds new consumption data. <ol style="list-style-type: none"> 1.1.1 User clicks on “Add consumption” button. 1.1.2 User performs “Add New Consumption” Use Case. 1.2 User update project details. <ol style="list-style-type: none"> 1.2.1 User clicks on a specific row of consumption record. 1.2.2 User performs “Update Consumption” Use Case. 	

Table 3.2.4.9 Add New Consumption Use Case Description

Use Case ID	U3-2-4-9
Use Case Name	Add New Consumption
Actor	Manager
Description	User adds new consumption records
Preconditions	The user has access to the “View Consumption List” use case. The user has the necessary permissions to add new consumption.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. The user selects the “Add New Consumption” option from the “ View Consumption List “ use case. 2. A consumption modal opens. 3. User can insert the data in the input form. 4. User clicks on the “Confirm” button. 5. System stores the new record and displays a confirmation message. 	

Table 3.2.4.10 Update Consumption Use Case Description

Use Case ID	U3-2-4-10
Use Case Name	Update Consumption
Actor	Manager
Description	User updates the consumption records
Preconditions	User must login with the management role account. User must have existing consumption record.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User finds and clicks on the icon for specific consumption record from the “View Consumption” use case. 2. The consumption modal opens and show the consumption details. 3. User can change the data in the input form. 4. User clicks on the “Confirm” button. 5. System updates the record to replace previous record on the database. 	

Table 3.2.4.11 Delete File Use Case Description

Use Case ID	U3-2-4-9
Use Case Name	Delete File
Actor	Manager
Description	User deletes the file from order records.
Preconditions	User must login with the management role account. Order records include PDF file.
<p>Flow of Events:</p> <ol style="list-style-type: none"> 1. User login with a management account (Login Use Case). 2. User clicks an icon for a specific row of order. 3. The order modal opens and show the order details including PDF file. 4. User clicks the “Error” icon exists with the PDF file. 5. The confirmation modal open and displays confirmation message. 6. User clicks the “Yes” button. 7. System deletes the file from database and removes the file record from order modal. 	

3.2.5 Procurement and Shipment Module

This is the main module of web portal for sales and management proceed the order from customer. The diagram shown the overall activity started from customer enquire product. Salespersons need to quote the price based on customer enquiry material and quantity. Customer only can proceed to purchase after manager approve the quotation. The validity date entered together with price to ensure customer accept the quotation within the indicated period. If validity date expired, customer unable proceed to purchase unless they request the quotation again and wait for the quotation approved. After customer accept quotation, they should select the shipment arranged by the supplier and upload purchase order to salesperson. If they suddenly upload a wrong file, they should request manager to remove the file. They only can reupload the file after manager deletes the file. Once the customer receives the products, manager should prepare the documents such as delivery order, invoice and receipt and upload it through the web portal, so the customer can check and download the file anytime.

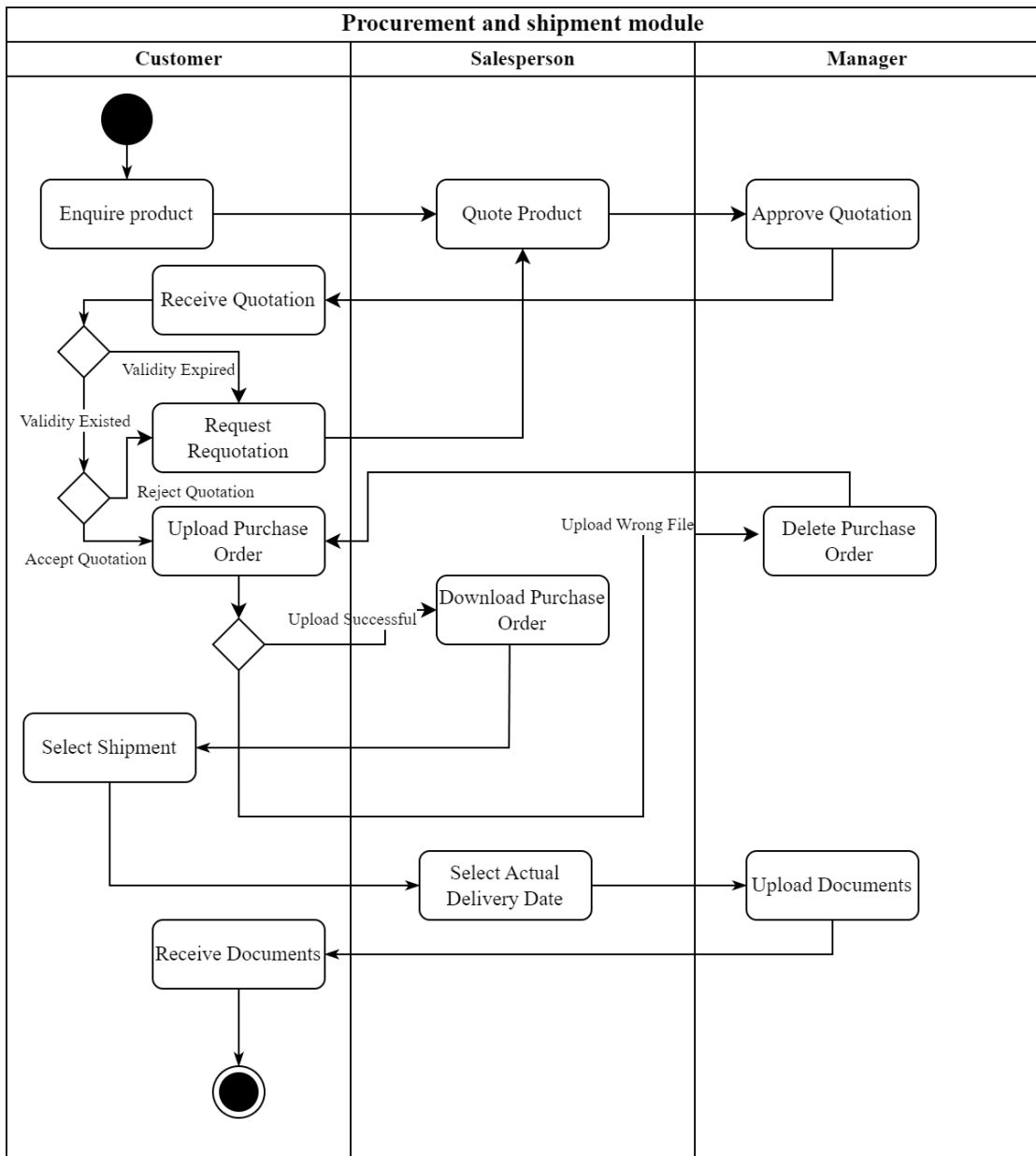


Figure 3.2.4 Activity Diagram for procurement and shipment module

CHAPTER 3

3.3 Timeline

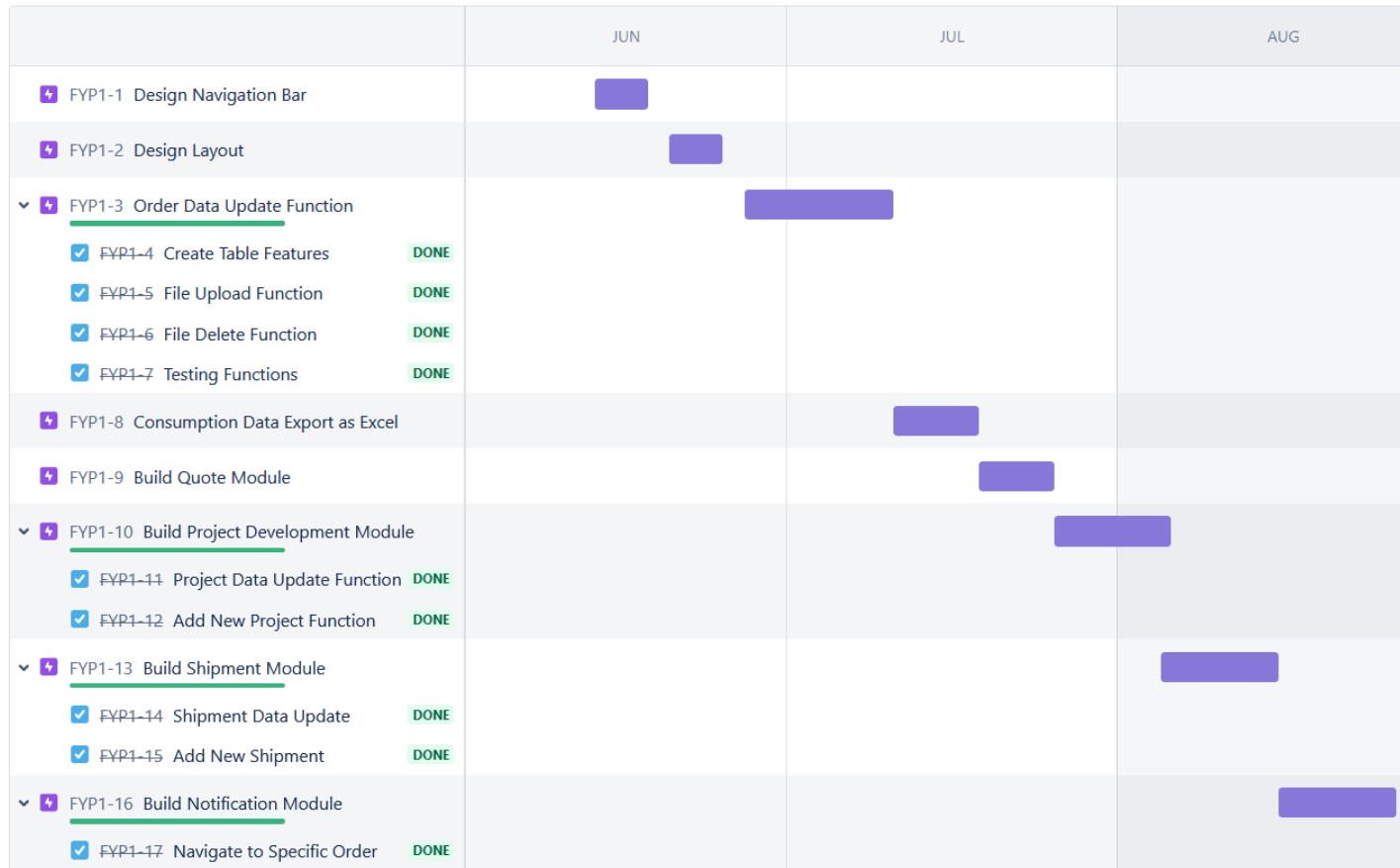


Figure 3.3.1 Timeline of the system development for FYP1

CHAPTER 3

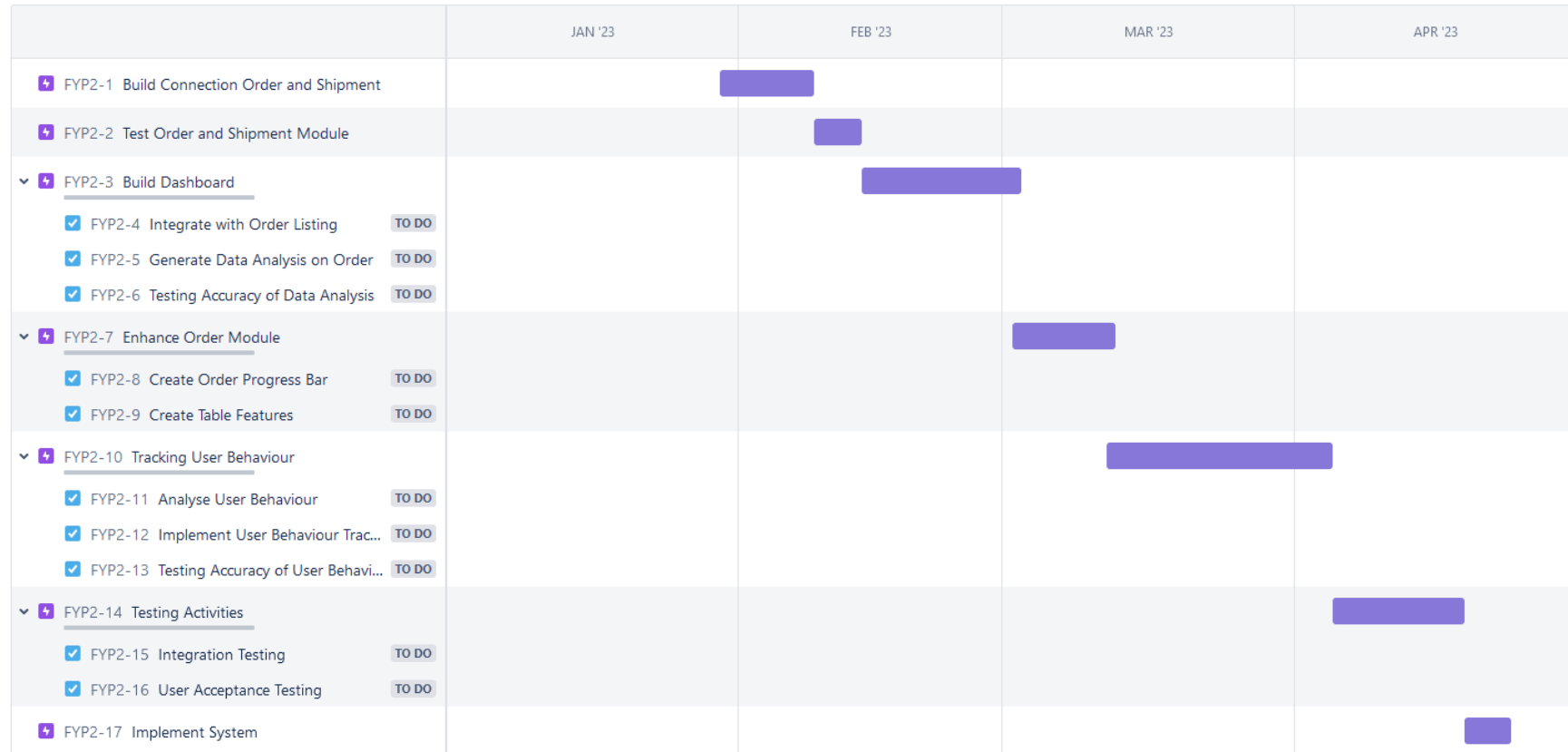


Figure 3.3.2 Timeline of the system development for FYP

Chapter 4

System Design

This chapter illustrates the system design through system block diagrams, flowcharts, and user interface design.

4.1 System Block Diagram

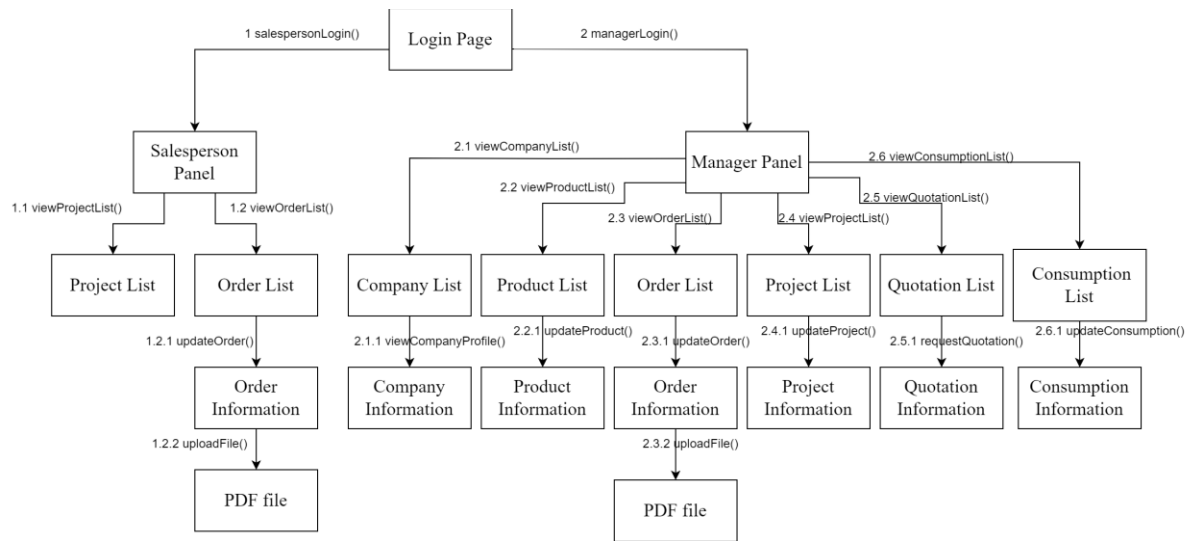


Figure 4.1 System Block Diagram

The proposed system's block diagram is depicted in Figure, illustrating the main blocks that represent the system's key functions. The blocks are interconnected with arrows to indicate their relationships. The system includes two user roles with distinct functional components. Users access the system using their username and password, which are stored in the database. The system redirects the user to the appropriate panel based on the role stored in the user's data. In the salesperson panel, the salesperson is only authorized to view the project list or order list. The salesperson's actions in the system are limited compared to those of the manager. Meanwhile, the manager panel includes the salesperson panel components. The manager has the right to view all order records within the department, while the salesperson only has access to the order records of their own customers. Both the manager and the salesperson can upload PDF files, such as delivery orders, invoices, and receipts when an order is completed. However, the salesperson cannot delete the PDF files.

The manager is authorized to view and update product information, such as product stock numbers and status. The product list is an essential component for customer enquiries about the product. The manager can manage projects that collaborate with customers and update them

Bachelor of Information Systems (Honours) Business Information Systems
Faculty of Information and Communication Technology (Kampar Campus), UTAR

as needed. The main component of the system is not only about customer orders but also about quotations with suppliers. The manager can request quotations from suppliers and await their approval. The consumption list is used to record customer consumption data in their operations, making it easier for the manager to support their needs.

4.2 System Flow Diagram

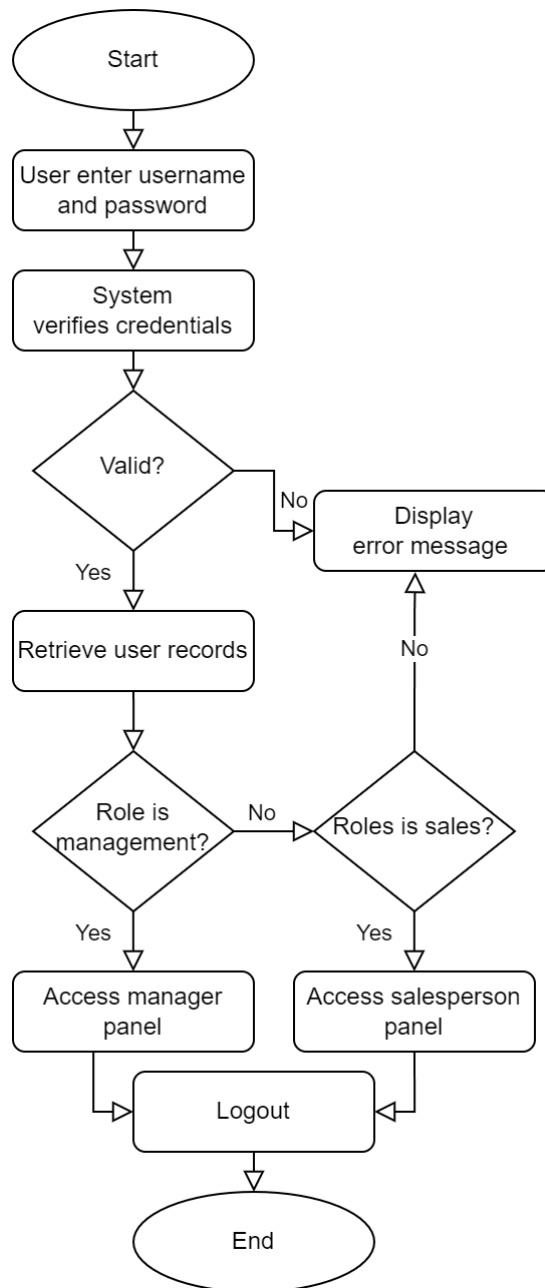


Figure 4.2.1 Login Flow Diagram

The sequence of steps for the login process is depicted in this flowchart. The user inputs their username and password, which are then validated by the system. In case of incorrect

credentials, an error message is displayed. On the other hand, if the credentials are correct, the system retrieves the user's information and determines their role. The system directs the user to the appropriate panel based on their role.

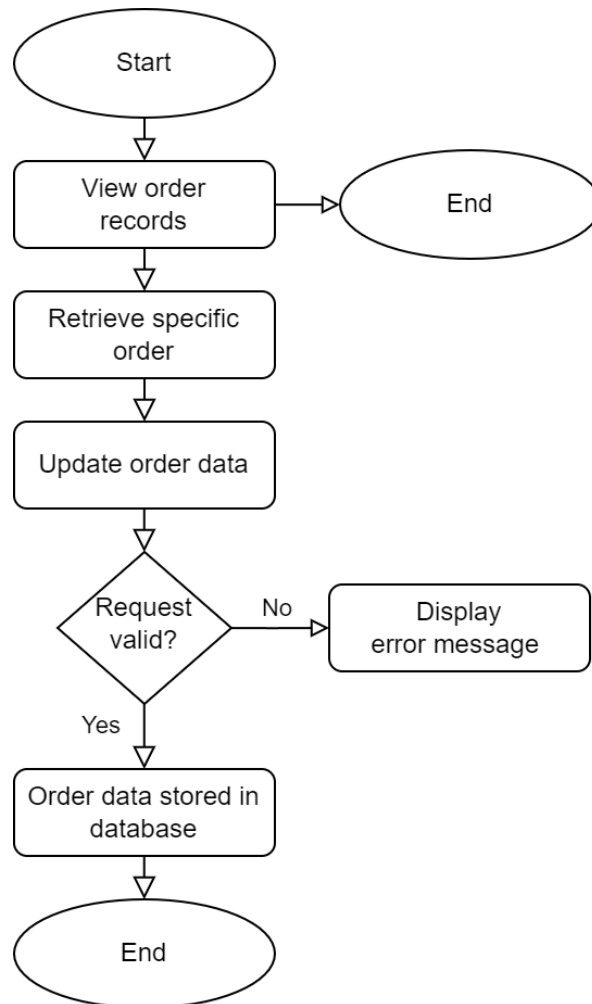


Figure 4.2.2 Update Order Flow Diagram

User is allowed to view the order records after access the panel. From the order records page, the user can find and click on the icon for a specific order. This action opens an order modal that displays the details of the selected order. Users can edit or insert order data according to order status. The action button is enabled, users can click and send the requests. If the request is valid, the order data will be stored in the database. If not, error messages will be displayed, and the user will need to correct any mistakes before trying again.

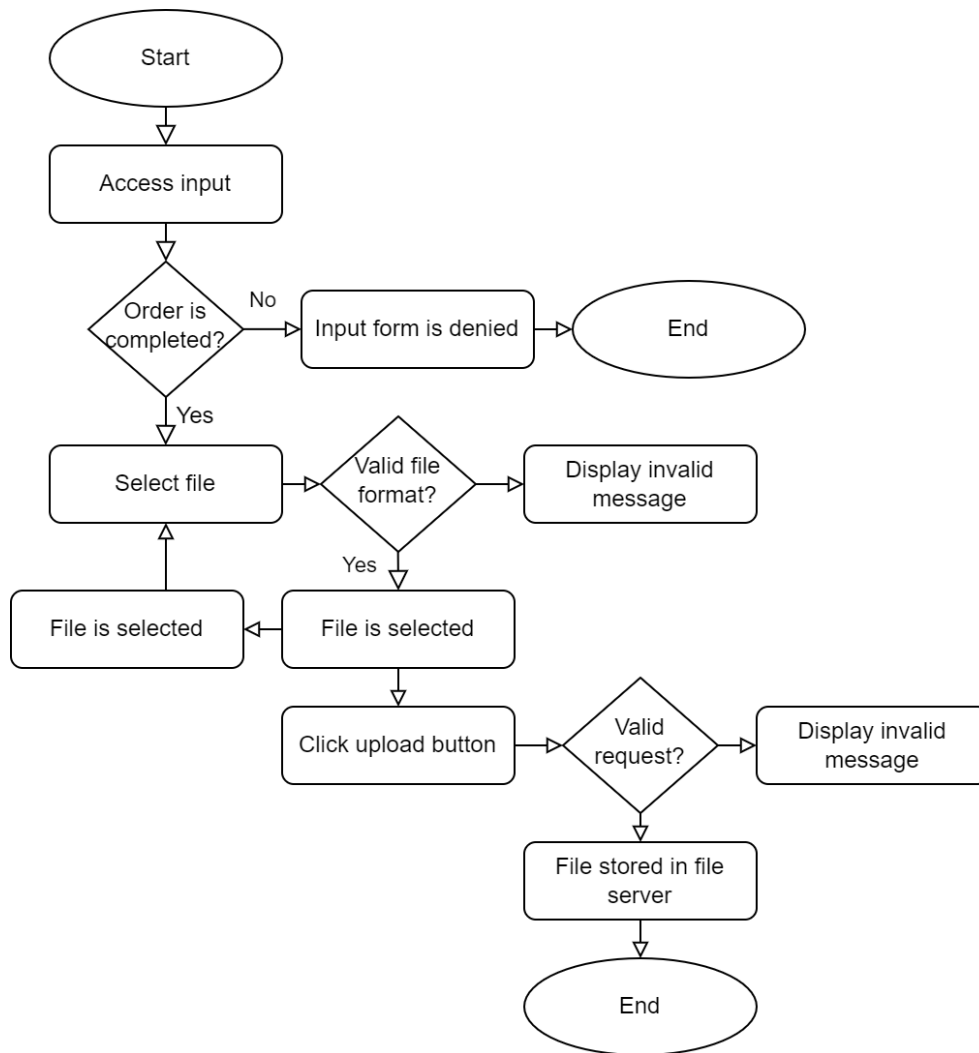


Figure 4.2.3 Upload File Flow Diagram

The user can view their completed orders and select the order they want to upload a file for by clicking on the corresponding icon. The order modal opens and displays the order details along with a file input form. The user clicks on the input form to open the directory file and selects the file they want to upload. If the selected file is PDF format, the file will be displayed in the form and the user can click to preview it. Otherwise, the system will indicate that the file format is invalid and reject the file. User can click on the “Cancel” button, the system will clear the file data and the user can select the file again. After reviewing the file, the user clicks on the “Upload” button. If the request is valid, and the file will be stored in the file server.

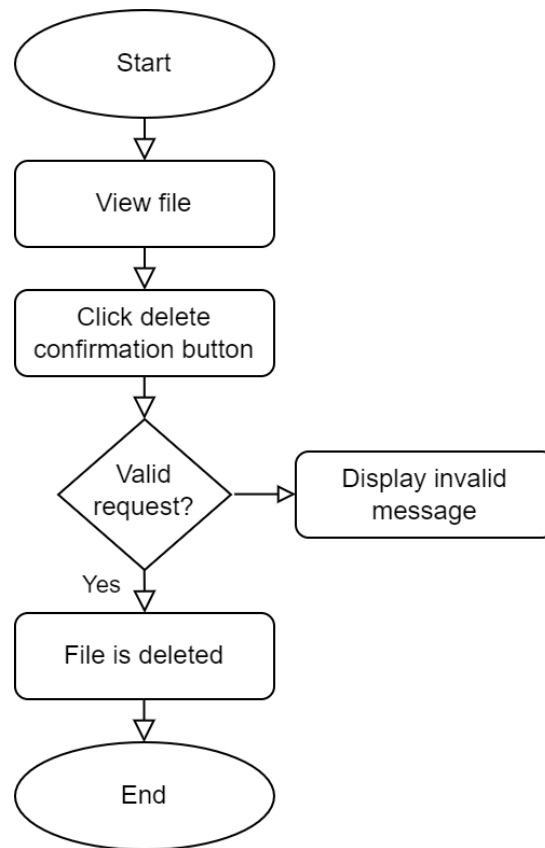


Figure 4.2.4 Delete File Flow Diagram

This flow diagram outlines the process for deleting a file that has been uploaded to a file server. User can view the uploaded file by clicking the document URI. When the user decides to delete the file, a confirmation button will be clicked. The system will check to ensure that the delete request is valid. If the request is valid, the file will be deleted from the file server. However, if the request is not valid, the system will display an error message to the user indicating that the request cannot be completed.

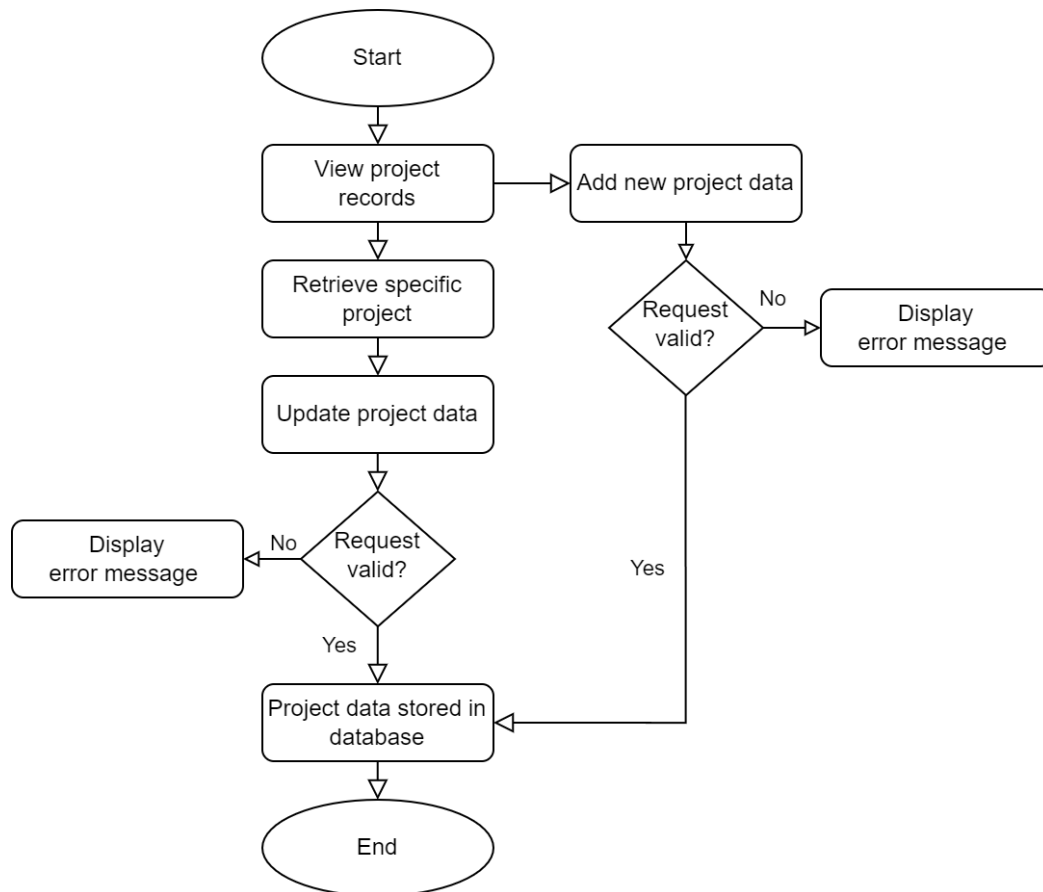


Figure 4.2.5 Project Management Flow Diagram

The flow diagram describes the process for updating and adding new project records in the system. The flow starts with user access the project page, where the user can select an existing project to update or add a new project to the system.

If the user selects an existing project, the system retrieves the project data and displays it on project modal. The user can update the project data and submit the changes. If the submitted data is valid, the system stores the updated project data in the database. If the data is invalid, such as missing required fields or incorrect format, the system displays an error message to the user.

If the user selects to add a new project, the system displays a new project modal. The user can input the new project data and submit the form. The system then checks if the submitted data is valid. If it is valid, the system stores the new project data in the database. If the data is invalid, the system displays an error message to the user.

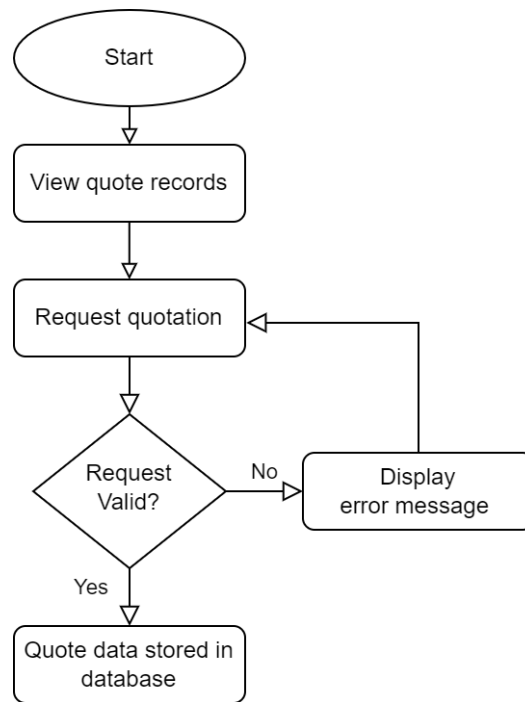


Figure 4.2.6 Request Quotation Flow Diagram

The flow diagram describes the process for requesting a quotation and storing the quotation data in the system. The flow starts with the quote page, where the user can view existing quotations or request a new quotation.

If the user selects to request a new quotation, the request quotation modal will be opened. The user can input the relevant data and submit the request. The system then checks if the submitted data is valid. If it is valid, the system stores the new quotation data in the database. However, if the data is invalid, the system will display an error message to the user and the user needs to enter the correct data format. The requested data will be sent to the corresponding supplier when the supplier request to view the records. Users can receive the quotation when the supplier submits the quotation successfully.

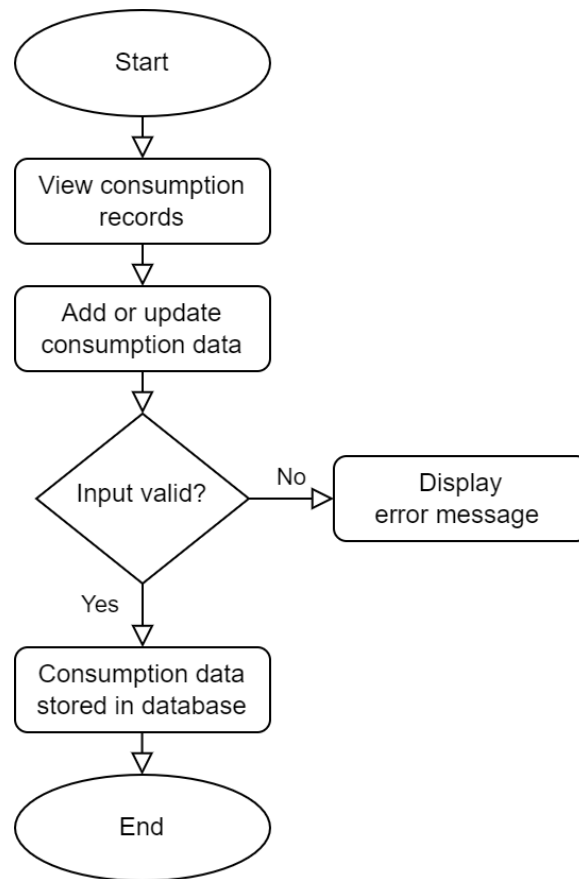


Figure 4.2.6 Consumption Management Flow Diagram

The flow diagram describes the process for adding or updating consumption data in the system. User can view existing consumption data after data retrieved from database. If the user selects to add consumption data, the modal will open with the input form. The user can input the relevant consumption data and submit the form. If the request is valid, the system stores the new consumption data in the database. Otherwise, an error message will be displayed. If the users select to update consumption data by clicking on the specific record, the modal will open and display the data. User can change the data and submit the form. If the request is valid, updated record will be stored in the database.

4.3 Database Design

Table 4.3.1 Data Dictionary for User Entity

Field Name	Description	Data Type	PK or FK	Null
_id	Unique identification for user	objectId	PK	No
username	Username	String		No
emailAddress	Email address	String		No
password	User password	String		No
role	User role	String		No
department	User department	String		Yes
tier	User tier	String		Yes
createdAt	Account creation date	Date		No
status	Account status	String		No
companyId	User associate company ID	String	FK	Yes
salesId	Customer associate salesperson ID	String	FK	Yes

Table 4.3.2 Data Dictionary for Company Entity

Field Name	Description	Data Type	PK or FK	Null
_id	Unique identification for company	objectId	PK	No
type	Company type	String		No
name	Company name	String		No
registrationNumber	Company registration number	String		No
address	Company address	String		No
contactNumber	Company's contact number	String		No
email	Company email	String		No
salesVerification	Company sales verification	String		No
businessNature	Company business nature	String		No
turnover	Company turnover rate	String		No
numberOfWorker	Number of workers in company	String		No

CHAPTER 4

numberOfProductionLine	Number of production line in company	Integer		No
userTier	Company user tier	Integer		No
status	Company status	String		No
registerDate	Company registration date	Date		No
salesId	Company associates salesperson ID	String	FK	No
debtorAgeing	Company debtor ageing	String		No
creditLimit	Company credit limit	String		No
department	Company department	String		No
allowToViewConsumption	Company allows to view consumption list	Boolean		Yes

Table 4.3.3 Data Dictionary for Product Entity

Field Name	Description	Data Type	PK or FK	Null
_id	Unique identification for product	objectId	PK	No
category	Product category	String		No
subCategory	Product subcategory	String		No
material	Product material	String		No
subMaterial	Product sub material	String		No
grade	Product grade	String		No
brand	Product brand	String		No
availableColour	Product available colour	Array		No
stockAvailability	Product stock availability	Integer		No
sellingPrice	Product selling price	Double		No
costPrice	Product cost price	Double		No
lastUpdate	The date the record was last updated	Date		No
status	Product status	String		No

Table 4.3.4 Data Dictionary for Order Entity

Field Name	Description	Data Type	PK or FK	Null
_id	Unique identification for order	objectId	PK	No
brand	Customer order product brand	String		No
material	Customer order product material	String		No
materialType	Customer order product material type	String		No
colour	Customer order product colour	String		No
quantity	Customer order product quantity	Integer		No
deliveryDate	Order delivery date with format “DD/MM/YYYY”	String		No
orderDate	Order date	Date		No
orderBy	Company name	String		No
department	Order associate department	String		No
salesId	Order associate salesperson ID	String		No
email	Order associate user email	String		No
status	Order status	String		No
actualDeliveryDate	Order actual delivery date	String		No
lastUpdated	The date the record was last updated	Date		No
price	Order price	Double		No
validity	Order validity date with format “DD/MM/YYYY”	String		No
remarks	Order remarks	String		Yes
requisitionRemarks	Order re-quotation remarks	String		Yes
purchaseOrderURI	Purchase order file URI	String		Yes
purchaseOrderUploadDate	Purchase order file upload date	Date		Yes
deliveryOrderURI	Delivery order file URI	String		Yes
deliveryOrderUploadDate	Delivery order file upload date	Date		Yes
invoiceURI	Invoice file URI	String		Yes
invoiceUploadDate	Invoice file upload date	Date		Yes

CHAPTER 4

receiptURI	Receipt file URI	String		Yes
receiptUploadDate	Receipt file upload date	Date		Yes

Table 4.3.5 Data Dictionary for Consumption Entity

Field Name	Description	Data Type	PK or FK	Null
_id	Unique identification for consumption	objectId	PK	No
companyName	Company name	String		No
material	Material used in the consumption	String		No
application	Description of the application in which the material was used	String		No
consumption	Quantity of material consumed	String		No
type	Material type	String		No
remarks	Remarks for the consumption	String		Yes
lastUpdated	The date the record was last updated	Date		No

Table 4.3.6 Data Dictionary for Project Entity

Field Name	Description	Data Type	PK or FK	Null
_id	Unique identification for project	objectId	PK	No
name	Project name	String		No
application	Description of the application for which the project is intended	String		No
consumption	Required material consumption	String		No
existingGrade	Current grade of materials for project	String		No
proposeGrade	Proposed grade of materials for project	String		No
status	Project status	String		No
remarks	Remarks for project	String		Yes

CHAPTER 4

lastUpdated	The date the record was last updated	Date		No
supplierId	Supplier ID for the project	String		No
supplierRemarks	Remarks by supplier	String		Yes

Table 4.3.7 Data Dictionary for Quotation Entity

Field Name	Description	Data Type	PK or FK	Null
_id	Unique identification for quotation	objectId	PK	No
supplier	Supplier name	String		No
brand	Product brand	String		No
product	Product name	String		No
price	Product price	Double		No
minimumOrderQuantity	Minimum order quantity	Integer		No
date	Quotation date	Date		No
status	Quotation status	String		No
validity	Quotation validity date with format "DD/MM/YYYY"	Date		No

4.4 User Interface Design

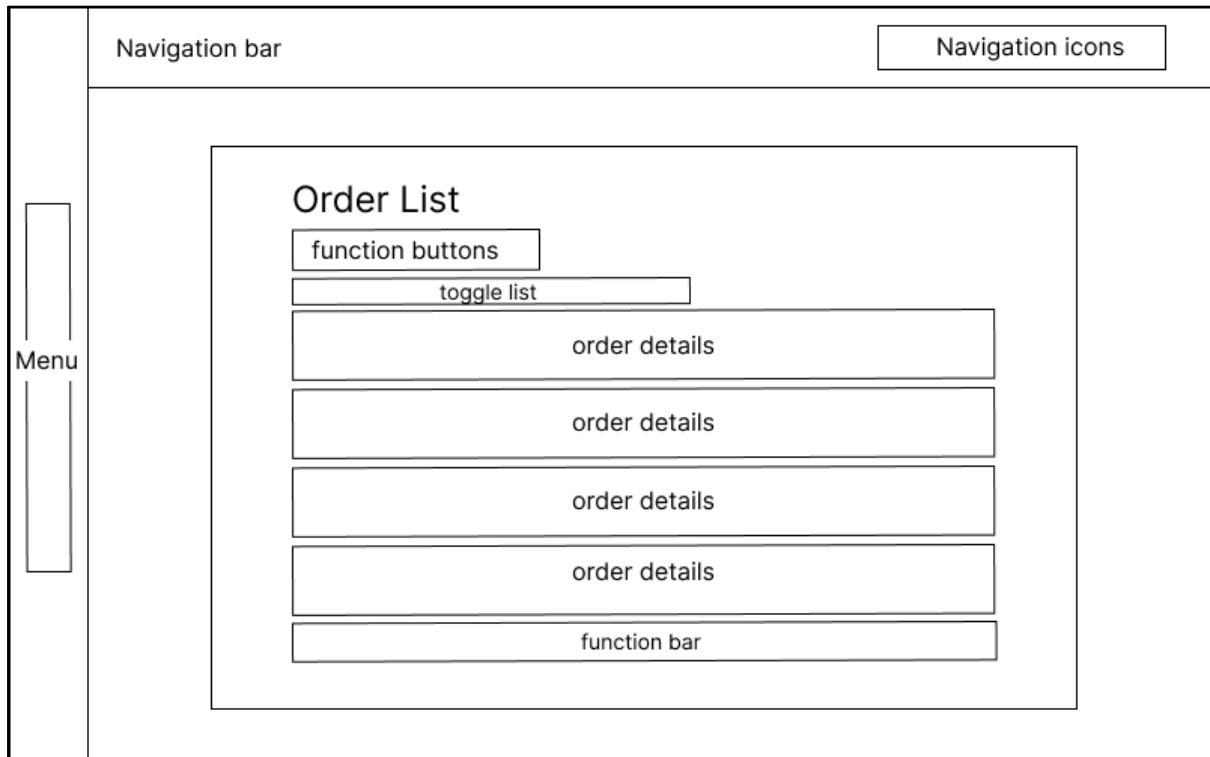


Figure 4.4.1 Order Management Wireframe

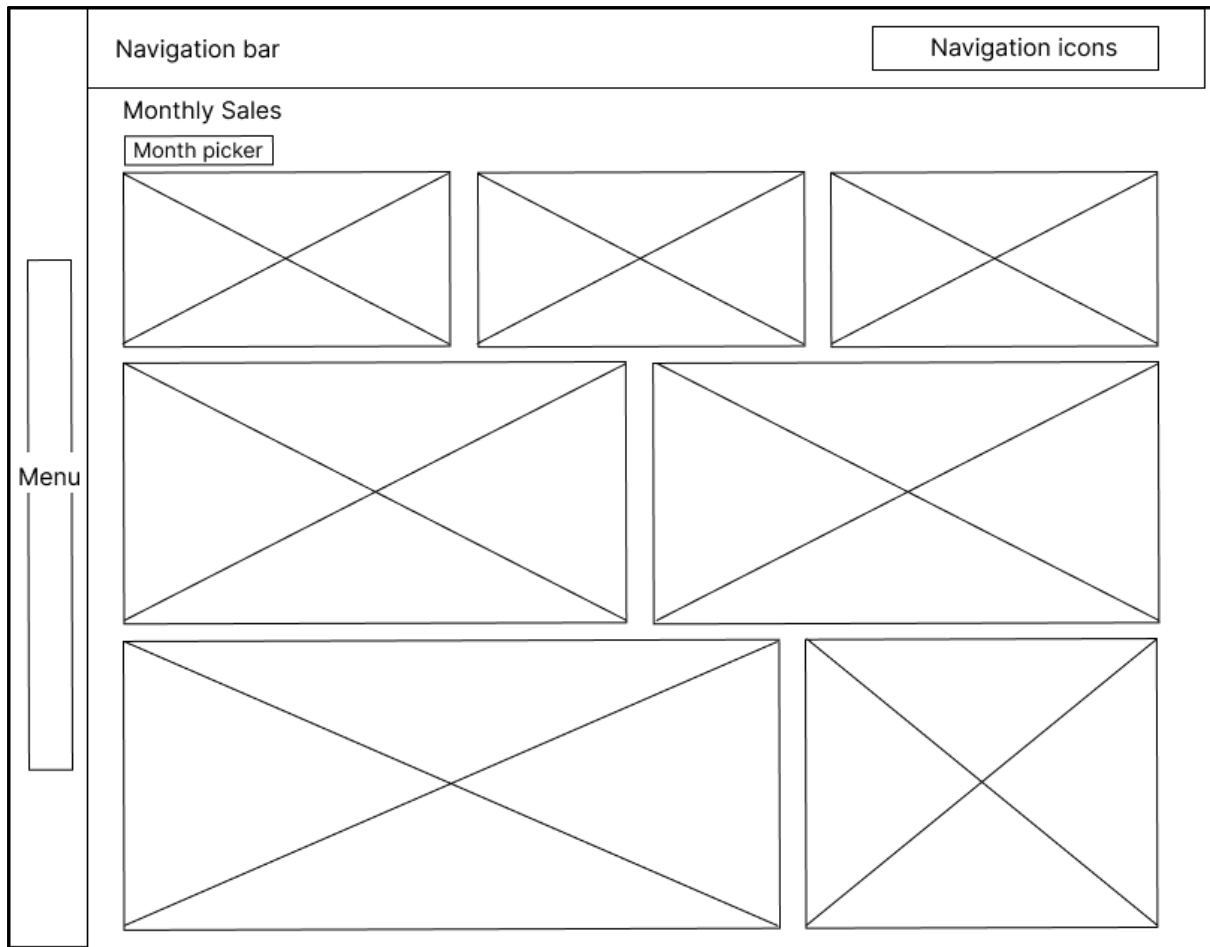


Figure 4.4.2 Sales Dashboard Wireframe

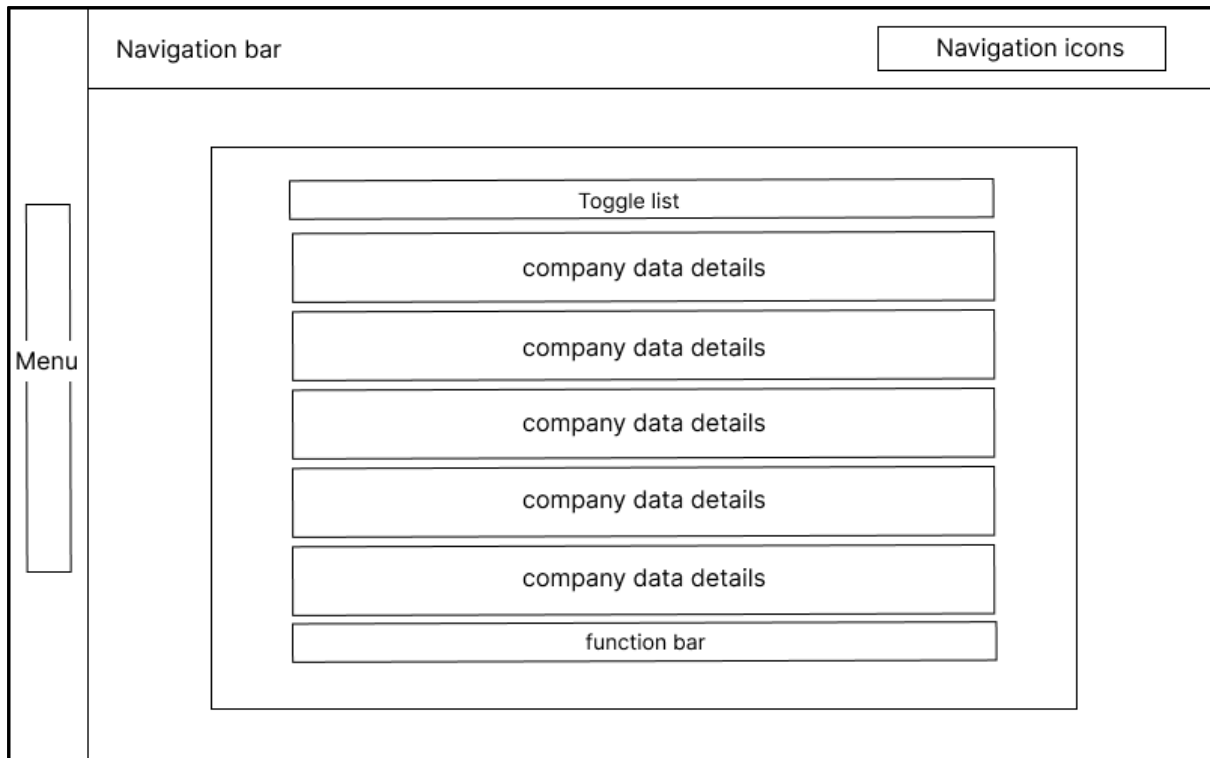


Figure 4.4.3 Company List Wireframe

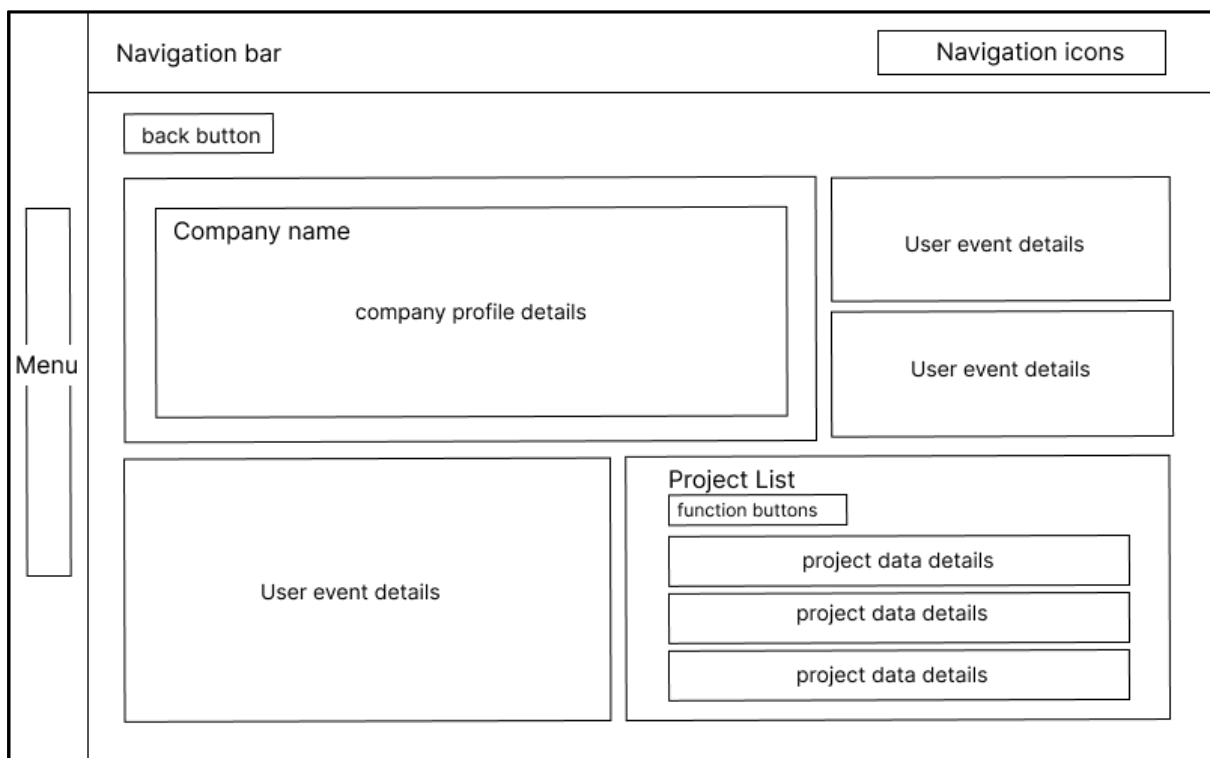


Figure 4.4.4 Company Profile Wireframe

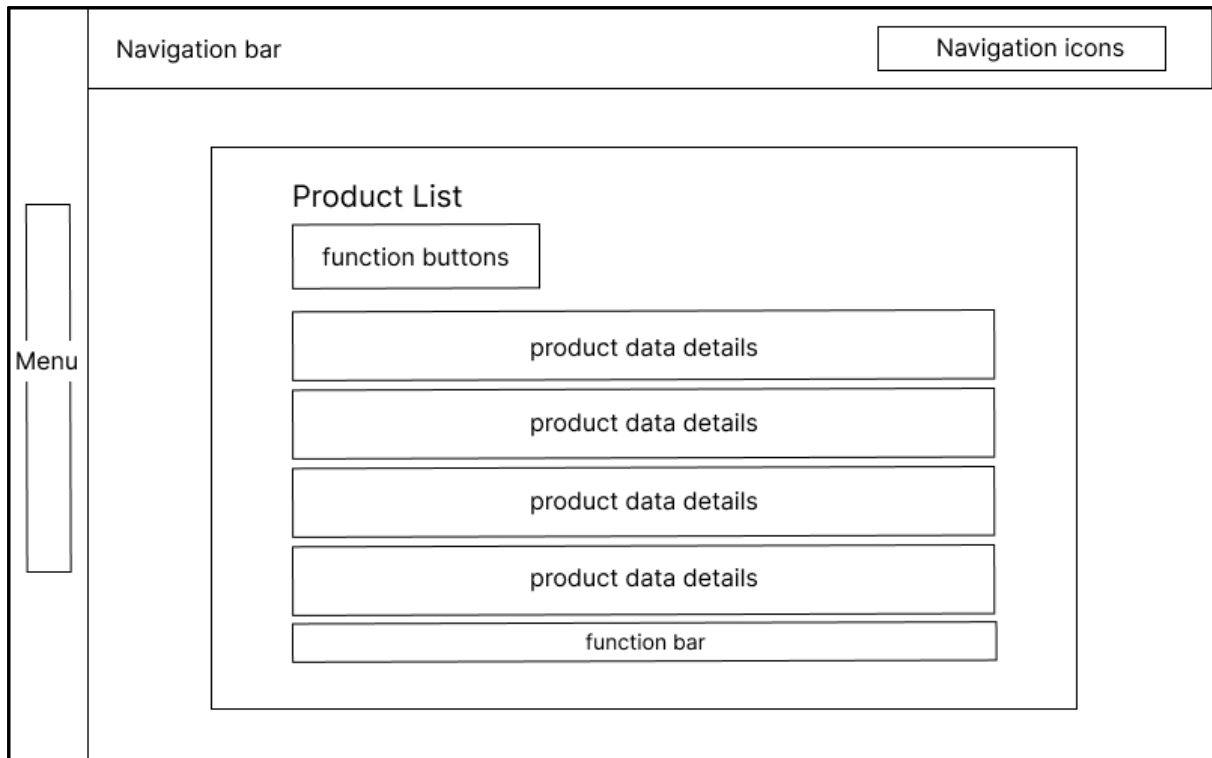


Figure 4.4.5 Product List Wireframe

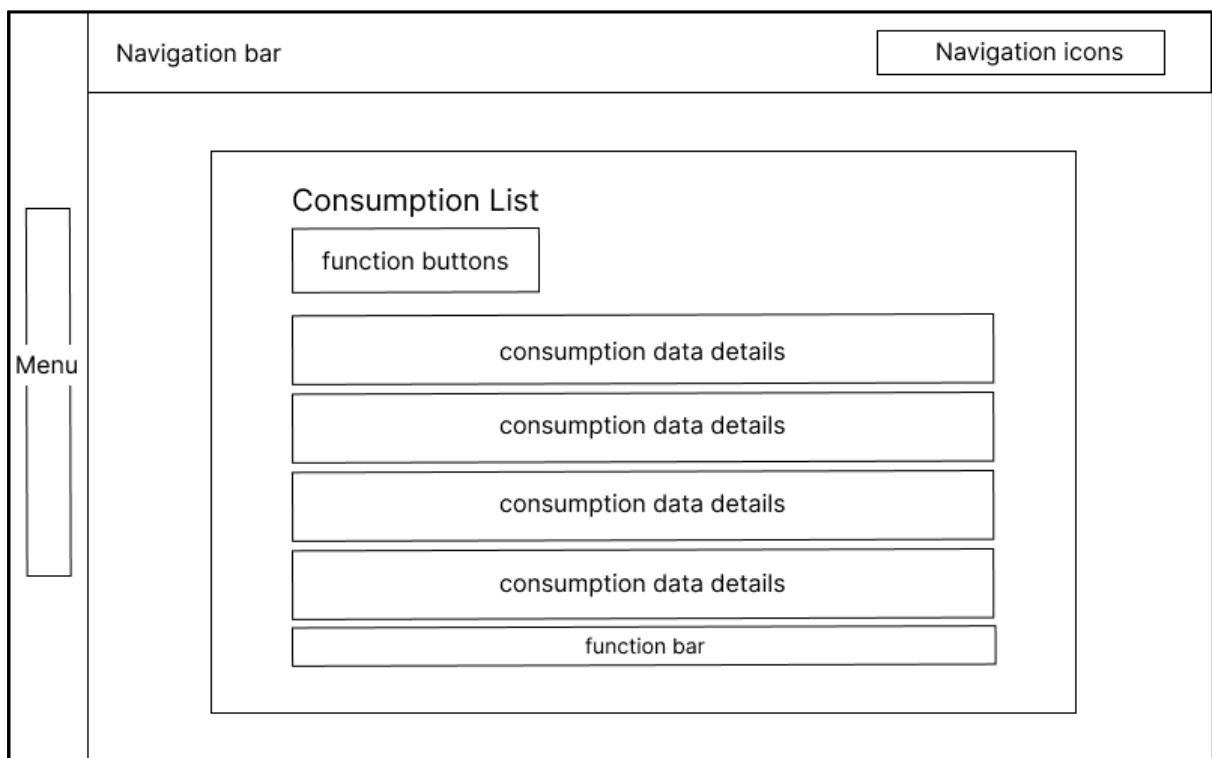


Figure 4.4.6 Consumption List Wireframe

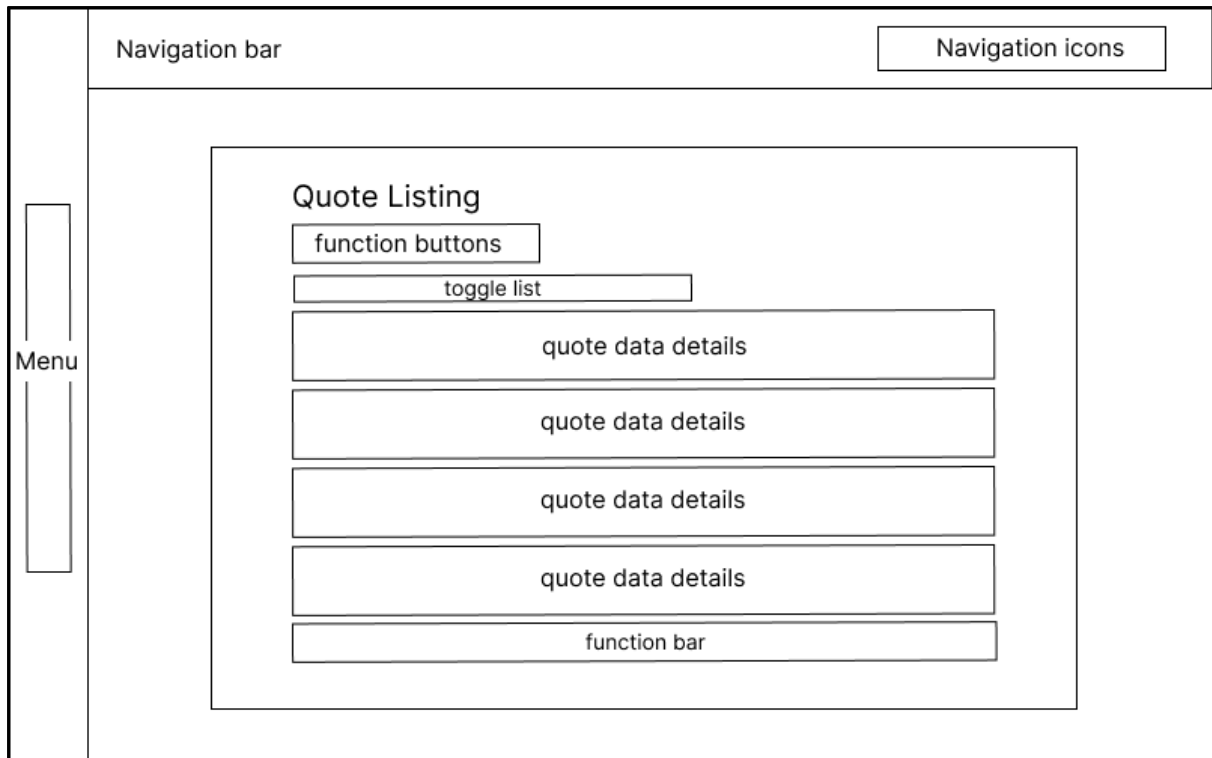


Figure 4.4.7 Quote List Wireframe

Chapter 5

System Implementation

This chapter discusses the implementation process of the system with the setting and configuration, system operations, and implementation challenges.

5.1 Hardware Setup

The equipment specifications used to develop this project are shown in the table below.

Table 5.1.1 Computer Specifications

Description	Specifications
Processor	AMD Ryzen™ 5 4500U
Operating System	Windows 11
Graphic	AMD Radeon™ Graphics
Memory	8GB DDR4 RAM
Storage	256GB PCIe NVMe SSD

Due to the need to run multiple APIs and frontends at the same time, memory recommends at least 8GB RAM to facilitate the smooth functioning of the project. Additionally, the project requires sufficient storage capacity to accommodate the web application's data and the system's logs. Insufficient storage space has been observed to cause device failures during project development. This issue can be resolved by ensuring adequate storage space is available.

5.2 Software Setup

The development of the web portal required the proper configuration and setup of various software tools and frameworks to ensure a smooth development process. React, NodeJS, and MongoDB were specifically required for this project. The process of installing Visual Studio Code and MongoDB, configuring the React and NodeJS frameworks, and integrating these tools for development purposes should be described.

5.2.1 Visual Studio Code

The Visual Studio Code desktop software can be downloaded from the official website, located at <https://code.visualstudio.com/download>. Upon accessing the website, the user can select the appropriate download option based on their operating system. According to the system used,

CHAPTER 5

the Visual Studio Code executable file with a version of 1.77.3 that supports 64-bit Windows system should be chosen. Selecting the appropriate executable file is crucial in ensuring system compatibility and optimal performance of the software. Upon clicking the executable file, a setup interface will appear, allowing the user to install Visual Studio Code. During the installation process, several necessary steps must be conducted, including accepting the license agreement and determining the destination location of the software.

5.2.2 Node.js

Node.js, an open-source JavaScript runtime environment, was set up on 64-bit Windows machines by following a few steps. The first step involved downloading the Node.js installer from the official website, located at “<https://nodejs.org/en/download>”. Once the download was completed, the installer was run, and the on-screen instructions were followed to complete the installation process. During the installation, users were prompted to select the features they wanted to install and could choose the default settings. To avoid additional effort in filling the gap, it was suggested to select for automatic installation of the necessary tools. After the installation was completed, the correct installation of Node.js was verified by opening a command prompt and typing “`node -v`” to check the version of Node.js installed. For this project, v18.14.2 was the version that was utilized. For this project, the version utilized was v18.14.2. It is worth noting that the latest version of React.js may not always be the most suitable option as certain packages may not be supported.

5.2.3 MongoDB

The utilization of MongoDB Community software is widely recognized as an effective solution for data storage requirements in modern web applications. The implementation process of MongoDB in a system is comprised of several essential steps. Initially, one must acquire the MongoDB Community installer for Windows from the official MongoDB website, which located at “<https://www.mongodb.com/try/download/community>”. Once the download is complete, the installer must be executed, and the on-screen instructions should be followed to complete the installation process. During the installation, users will be prompted to select the appropriate setup type and destination folder for installation. After the installation is complete, the accuracy of MongoDB's installation can be verified by launching a command prompt and typing “`mongo --version`” to check the installed version of MongoDB. For this specific web

development project, MongoDB version 4.2.24 was chosen as the compatible database management system.

5.3 Setting and Configuration

5.3.1 React.js

Setting up and configuring a React.js environment in Visual Studio Code is essential for developing modern web applications. First, ensure that Node.js and npm are installed on the system. Next, install the necessary libraries globally using the npm command. React.js library and npx library is important to create a React.js project. After the installation of React.js, proceed to open Visual Studio Code and enter the command “npx create-react-app web” to start the initialization process of a new React.js project. This command will automatically install any requisite libraries required and generate project initialization code. Upon the folder being prepared, navigate to the designated directory, and input the command “npm start” to start the execution of React.js.

In the event of working with an already existing project's source code, it is possible that the “npm install” command may fail to install a significant number of packages due to conflicts between dependencies. Consequently, it is advisable to utilize the “yarn install” command to guarantee the successful installation of all essential packages within the development environment. Yarn can be installed using an extension in Visual Studio Code. By following this approach, any potential conflicts between dependencies can be resolved, resulting in a successful installation process. Finally, configure the Visual Studio Code settings to enable auto-formatting, auto-linting, and code completion for a seamless development experience. With the React.js environment fully configured, developers can write and test modern web applications with ease, taking advantage of the powerful features of both React.js and Visual Studio Code.

5.3.2 Express.js

Setting up and configuring Express.js in a system implementation involves a series of steps. First, Node.js must be downloaded and installed. Next, a project directory should be created, and the “npm init” command used to initialize the project and create a “package.json” file. Express.js can then be installed using “npm install express”, which downloads and installs the latest version of the framework and its dependencies. Once this is done, an entry point file must be created, and the Express.js module required. The project is designed to store all source files

in the directory named “src.” The primary entry point file should be written in the “package.json” file. As the project uses ES6 modules, files for configuring routes and connecting to databases were written in the “mjs” file format. Finally, the application can be started using the command “node src/app.mjs”, and it will listen on a specified port. Following these steps will ensure the successful implementation of Express.js in the system.

5.3.3 Configuration between React.js, Express.js and MongoDB

React.js, Express.js, and MongoDB are all popular technologies that can be used together to create a full-stack web application. Express.js serves as the bridge connection between React.js and MongoDB. The configuration process involves several steps.

First, the database name and collection should be set up in MongoDB. This can be done using MongoDB Compass or the mongo command prompt. MongoDB Compass is recommended for its user-friendly interface. Once the databases are created, the connection between Express.js and MongoDB should be set up. Each Express.js server would connect to each database in MongoDB by using the MongoDB URI. A Mongoose driver should be installed in Express.js to facilitate this connection.

```
import mongoose from 'mongoose';

export const connectDb = async () => {
  mongoose.connect("mongodb://127.0.0.1:27017/orderdb", { useNewUrlParser: true, useUnifiedTopology: true }, (err) => {
    if (err) {
      console.error(`Database Error: Error Connecting to Order Database. Message: ${err}`);
      throw err;
    }else{
      console.log(`Success connect to orderDB`)
    }
  });
}

export const closeDbConnection = async () => {
  mongoose.connection.close();
}
```

Figure 5.3.3.1 Code for MongoDB Connection using Mongoose

To check if the connection to the MongoDB database is successful, Diagram shows the code for the connection of the order database in “order-api”.

```
PS D:\workspace\order-api> node .
(node:8696) ExperimentalWarning: Importing JSON modules is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
Promise { undefined }
Listening on port 5010
Success connect to orderDB
```

Figure 5.3.3.2 MongoDB Connection Success Message

CHAPTER 5

The diagram illustrates the verification process for the port number and connection status to the database, indicating that the order-api is successfully listening on port 5010 and the connection to the database has been established.

```
1 import mongoose from 'mongoose';
2
3 let orderSchema = new mongoose.Schema({
4   brand: String,
5   material: String,
6   materialType: String,
7   colour: String,
8   quantity: Number,
9   deliveryDate: String,
10  orderDate: Date,
11  orderBy: String,
12  department: String,
13  salesId: String,
14  email: String,
15  status: String,
16  pendingOn: String,
17  actualDeliveryDate: Date,
```

Figure 5.3.3.3 Mongoose Schema for orderdb Database

Once the connection is established, the schema for storing the documents should be written in Express.js. The diagram can show the schema for the order database, where the field name, data type, and the collection name should be defined in the schema.

```
export const getOrderById = async (token, orderId) => {
  try {
    const getOrderResponse = await axios.get("http://localhost:5010" + "/order/" + orderId, { headers: { 'Content-Type': 'application/json' } });
    return getOrderResponse.data;
  } catch (err) {
    console.log(`Error returned from GET order API. Message: ${err}. Response: ${JSON.stringify(err.response)}`);
    return false;
  }
}
```

Figure 5.3.3.4 Code for Requesting URI in React.js using Axios

To connect the front-end and back-end, the Axios library is used to make API calls from the React.js application to the Express.js server. Once the Axios library is installed in the React.js application, HTTP requests such as GET, POST, PUT, and DELETE requests can be implemented such as Diagram. The request URI is crucial and should be written to match the URI configured in the Express.js server.

Overall, the configuration of these technologies involves creating a React.js front-end, an Express.js back-end with RESTful APIs, a MongoDB database with Mongoose driver, and connecting them using the Axios library.

5.4 System Operation (with Screenshot)

5.4.1 Login and Order Module

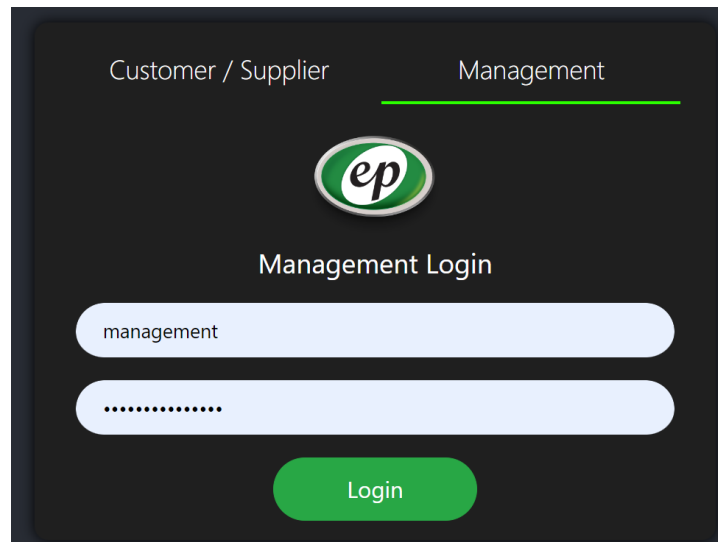


Figure 5.4.1.1 Role-Based Login Page

To authenticate and access the procurement and shipment portal, users are required to select the appropriate role between the “management” or “customer/supplier” options during login. The “management” was selected to login with the role of management and sales.

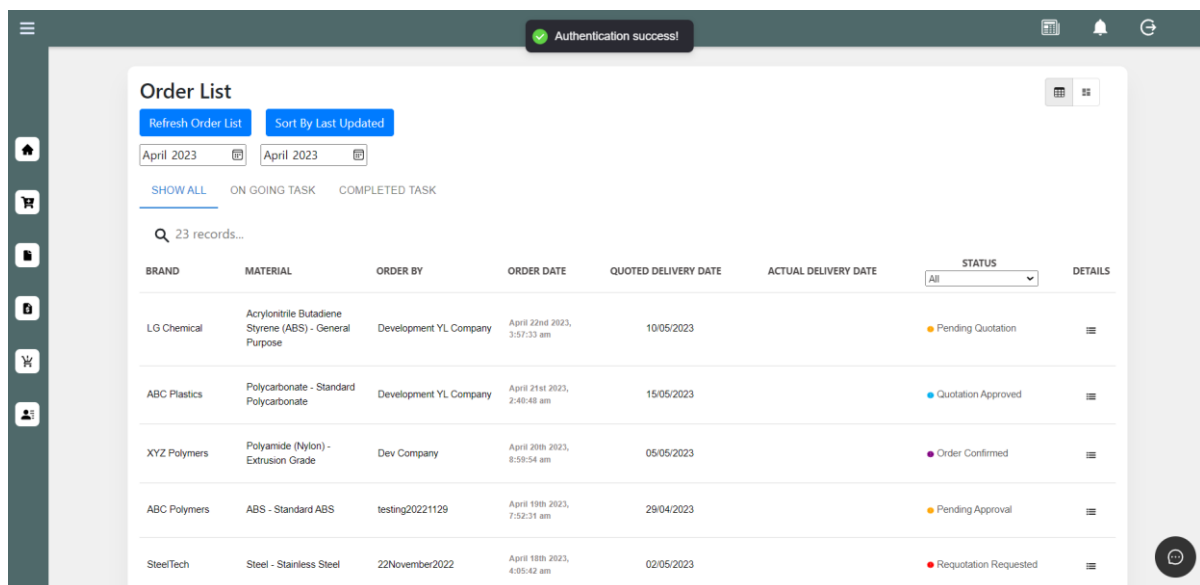
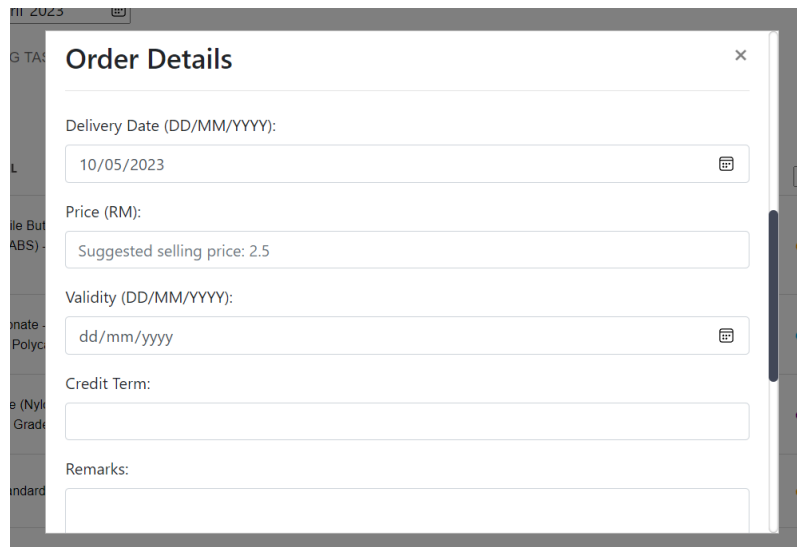


Figure 5.4.1.2 Order Page with Successful Authentication

If the authentication is successful, the system will grant the user access to the main page, and a toast notification appear on the screen indicating that the authentication was successful. The navigation bars help to navigate to other pages on the website. At the top right of the page, there are three icons for news, notifications, and logout. On the left side of the page, there is a sidebar with six icons for company, order, consumption, quote listing, product, and sales. The main section of the page is a table that displays the order list with details such as product,

CHAPTER 5

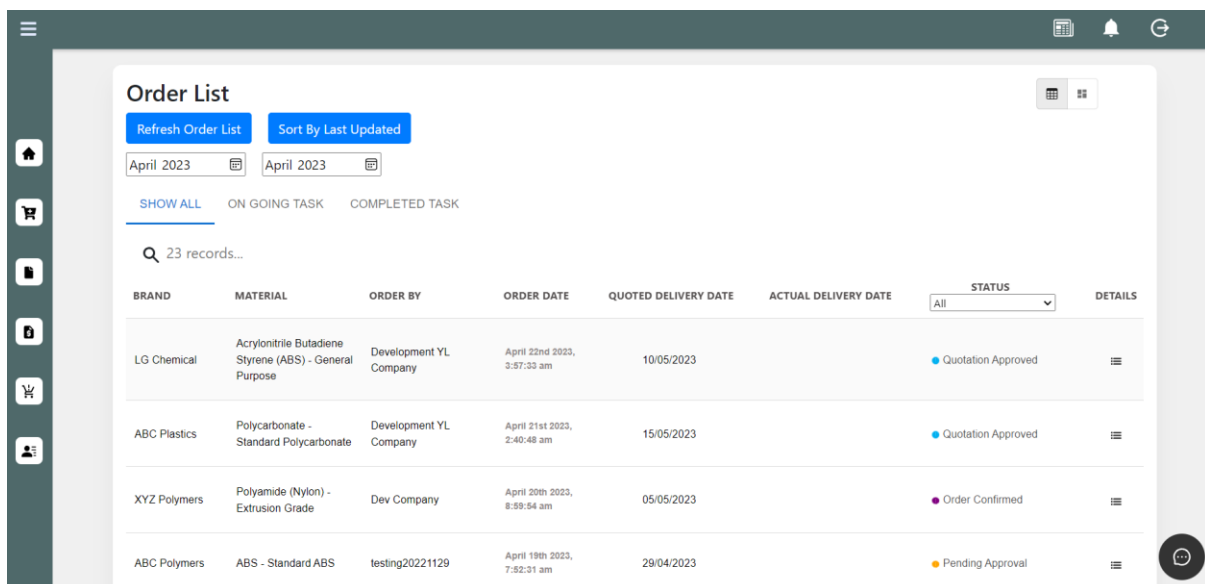
customer, order status, order date etc. Clicking the icon in each row opens a pop-up modal that displays the order details, which can be updated by the user through an input form.



The screenshot shows a modal window titled "Order Details" with a close button in the top right corner. The form contains several input fields: "Delivery Date (DD/MM/YYYY)" with the value "10/05/2023"; "Price (RM)" with the value "Suggested selling price: 2.5"; "Validity (DD/MM/YYYY)" with the value "dd/mm/yyyy"; "Credit Term" which is empty; and "Remarks" which is also empty. Each date field has a calendar icon to its right.

Figure 5.4.1.3 Order Modal for Order Quotation

Users enable to modify the required information at various stages of the order. For example, users can insert price and validity after receiving the order enquired by customer. User can scroll to the bottom of modal to click for the update button.



The screenshot shows the "Order List" page. At the top, there are buttons for "Refresh Order List" and "Sort By Last Updated". Below these are two date pickers, both set to "April 2023". There are tabs for "SHOW ALL", "ON GOING TASK", and "COMPLETED TASK". A search bar shows "23 records...". The main content is a table with columns: BRAND, MATERIAL, ORDER BY, ORDER DATE, QUOTED DELIVERY DATE, ACTUAL DELIVERY DATE, STATUS, and DETAILS. The table contains four rows of order data.

BRAND	MATERIAL	ORDER BY	ORDER DATE	QUOTED DELIVERY DATE	ACTUAL DELIVERY DATE	STATUS	DETAILS
LG Chemical	Acrylonitrile Butadiene Styrene (ABS) - General Purpose	Development YL Company	April 22nd 2023, 3:57:33 am	10/05/2023		Quotation Approved	
ABC Plastics	Polycarbonate - Standard Polycarbonate	Development YL Company	April 21st 2023, 2:40:48 am	15/05/2023		Quotation Approved	
XYZ Polymers	Polyamide (Nylon) - Extrusion Grade	Dev Company	April 20th 2023, 8:59:54 am	05/05/2023		Order Confirmed	
ABC Polymers	ABS - Standard ABS	testing20221129	April 19th 2023, 7:52:31 am	29/04/2023		Pending Approval	

Figure 5.4.1.4 Order Page

CHAPTER 5

Once the order is updated, the status is also updated, and the table can be refreshed by clicking the “Refresh Order List” button. The month picker and “Sort by Last Updated” button can be used to filter and sort the order records.

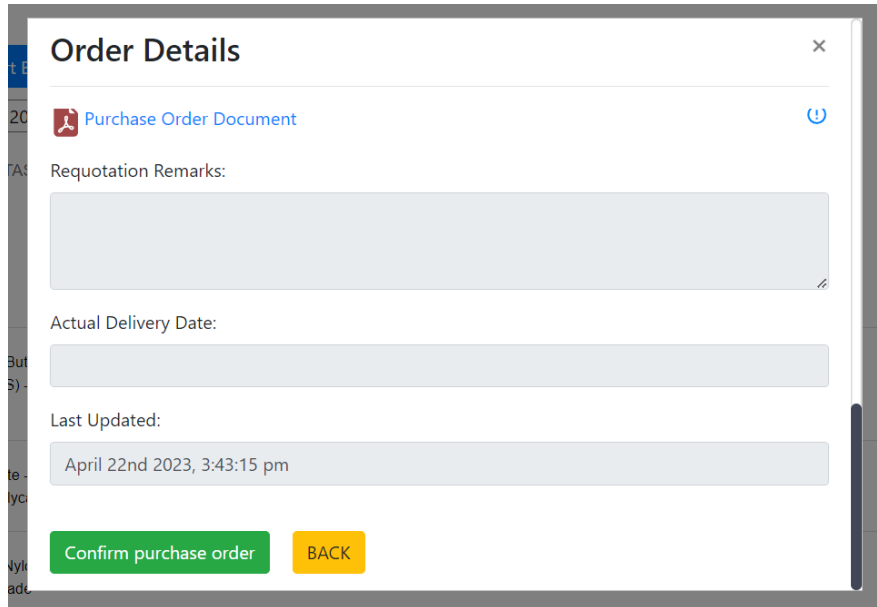
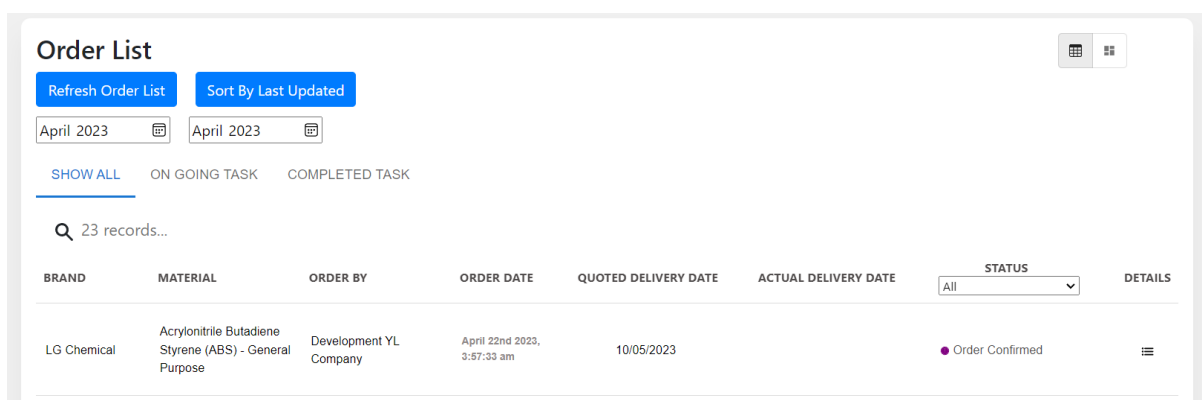


Figure 5.4.1.5 Order Modal for Purchase Order Confirmation

The status changed to “quotation confirmed” after customer accepted quotation and uploaded a purchase order document. Management can success to check the purchase order document before clicking the “confirm purchase order” button.

The image shows a screenshot of the "Order List" interface. At the top, there are buttons for "Refresh Order List" and "Sort By Last Updated". Below these are two month pickers, both set to "April 2023". There are also tabs for "SHOW ALL", "ON GOING TASK", and "COMPLETED TASK". A search bar shows "23 records...". The main part of the image is a table with the following columns: BRAND, MATERIAL, ORDER BY, ORDER DATE, QUOTED DELIVERY DATE, ACTUAL DELIVERY DATE, STATUS, and DETAILS. The table contains one row of data:

BRAND	MATERIAL	ORDER BY	ORDER DATE	QUOTED DELIVERY DATE	ACTUAL DELIVERY DATE	STATUS	DETAILS
LG Chemical	Acrylonitrile Butadiene Styrene (ABS) - General Purpose	Development YL Company	April 22nd 2023, 3:57:33 am	10/05/2023		Order Confirmed	

Figure 5.4.1.6 Updated Order Record after Confirm Purchase Order

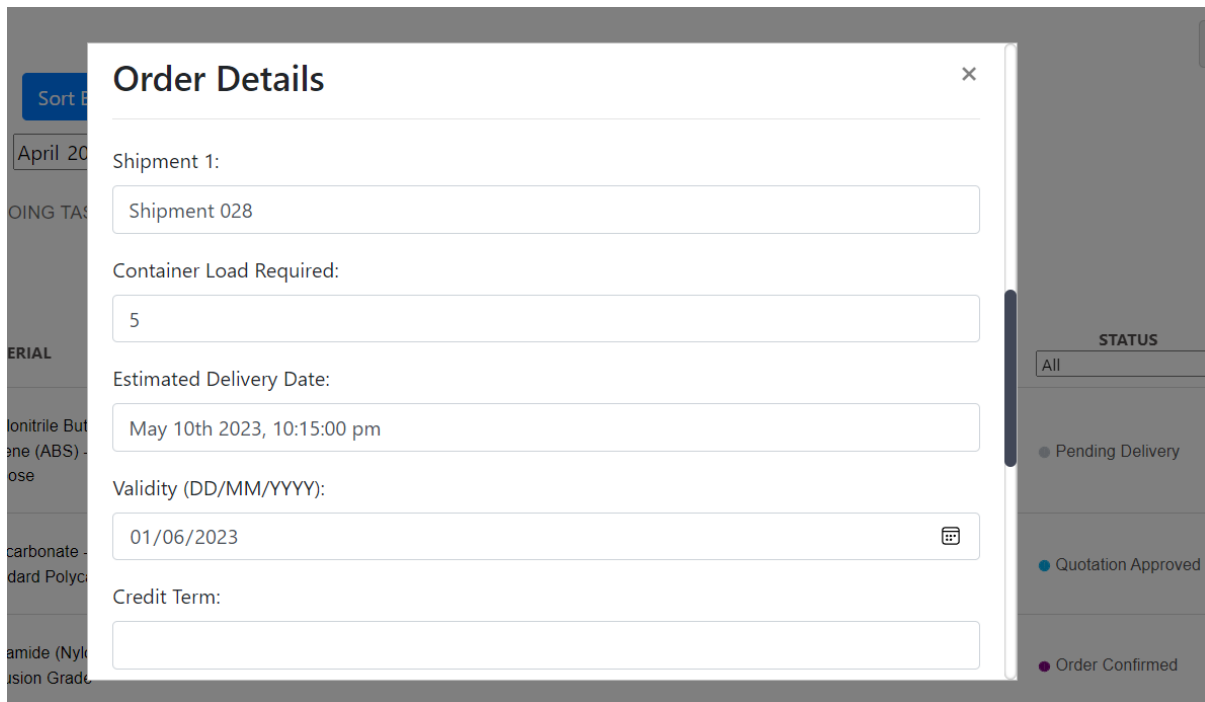


Figure 5.4.1.7 Order Modal with Selected Shipment Details

Once the customer selected a shipment for the order, the order status was updated to “pending delivery,” and the management was able to access the details of the shipment.

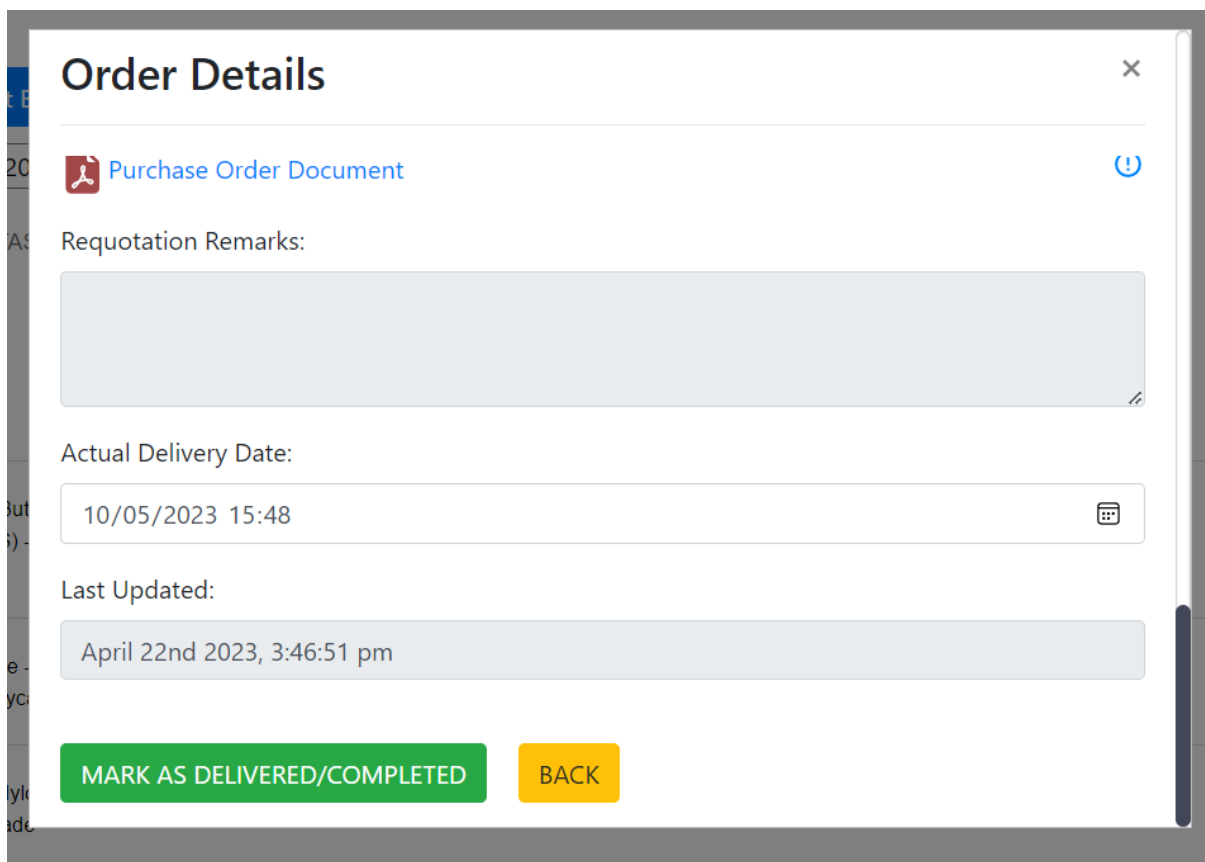
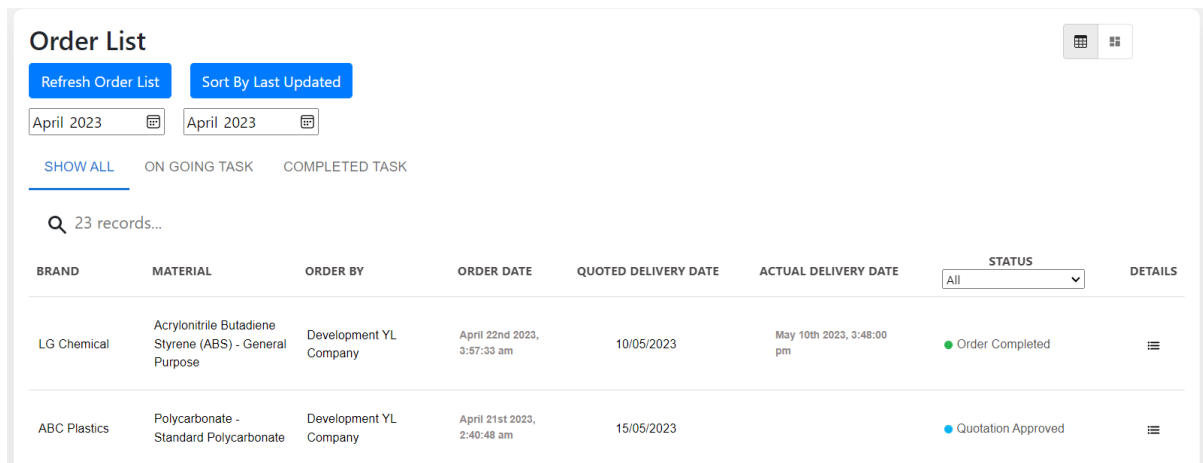


Figure 5.4.1.8 Order Modal for Mark Order as Completed

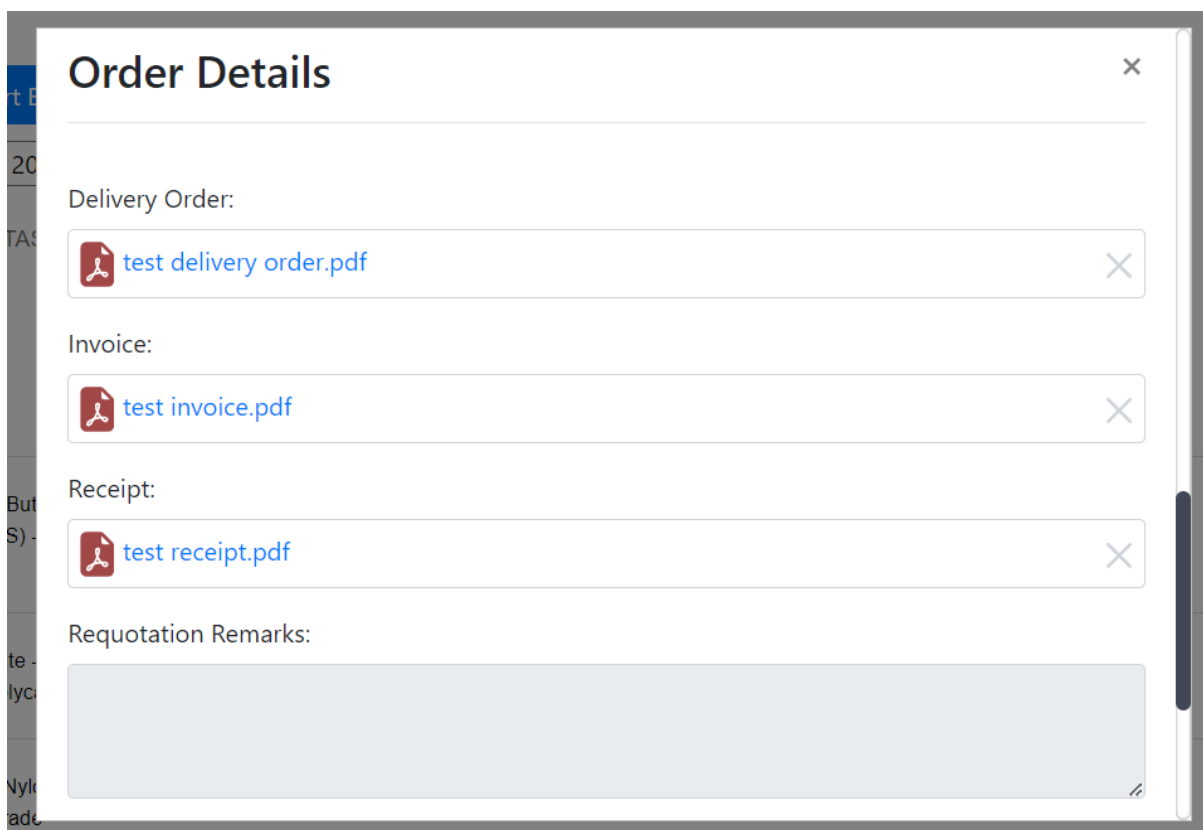
CHAPTER 5

The management was asked to indicate the completion of the order by adding the “actual delivery date” once the shipment has been delivered to the customer.



BRAND	MATERIAL	ORDER BY	ORDER DATE	QUOTED DELIVERY DATE	ACTUAL DELIVERY DATE	STATUS	DETAILS
LG Chemical	Acrylonitrile Butadiene Styrene (ABS) - General Purpose	Development YL Company	April 22nd 2023, 3:57:33 am	10/05/2023	May 10th 2023, 3:48:00 pm	● Order Completed	☰
ABC Plastics	Polycarbonate - Standard Polycarbonate	Development YL Company	April 21st 2023, 2:40:48 am	15/05/2023		● Quotation Approved	☰

Figure 5.4.1.9 Updated Order Record with Order Completed Status



Order Details

Delivery Order:

test delivery order.pdf

Invoice:

test invoice.pdf

Receipt:

test receipt.pdf

Re quotation Remarks:

Figure 5.4.1.10 Order Modal for Upload Files

When the order is finished, the user has the option to upload the delivery order, invoice, and receipt documents by either opening the file picker or dragging the file. The user can preview the file or choose a different file before clicking on the “upload” button.

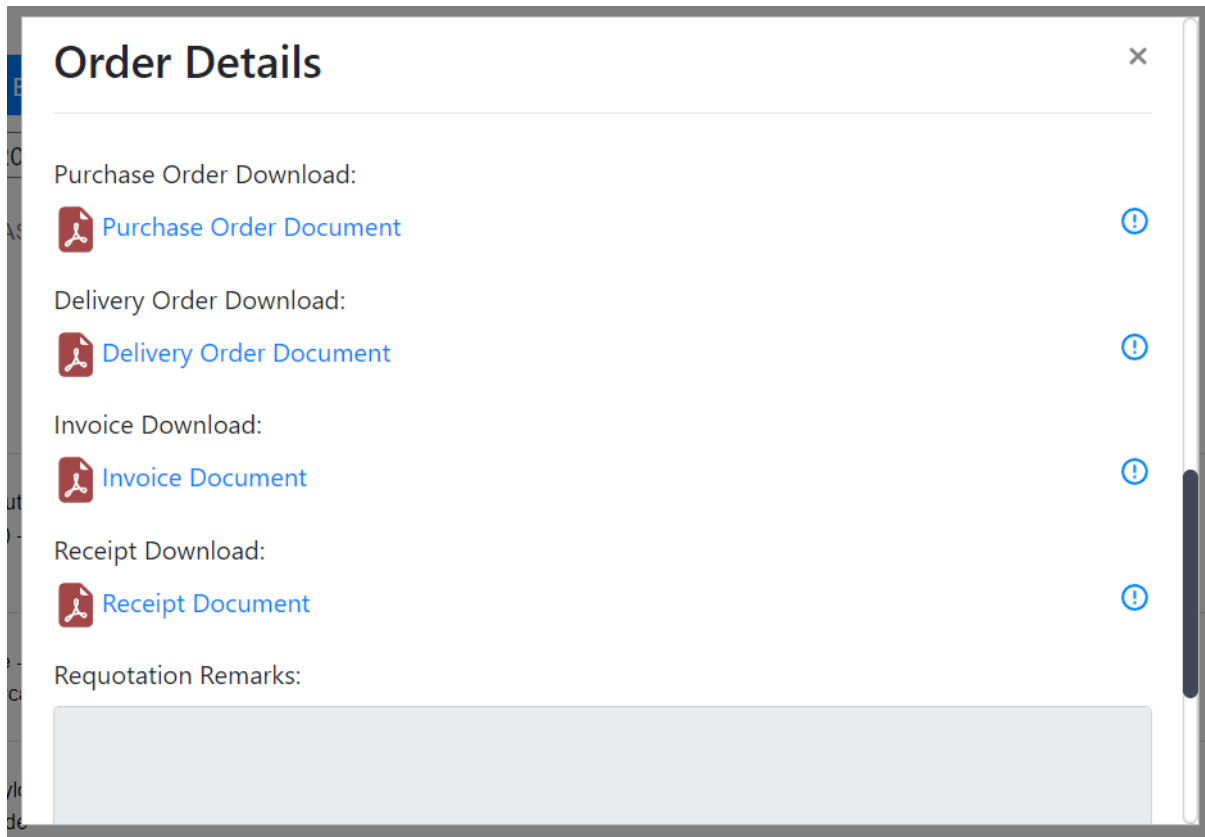


Figure 5.4.1.11 Order Modal with Uploaded Files

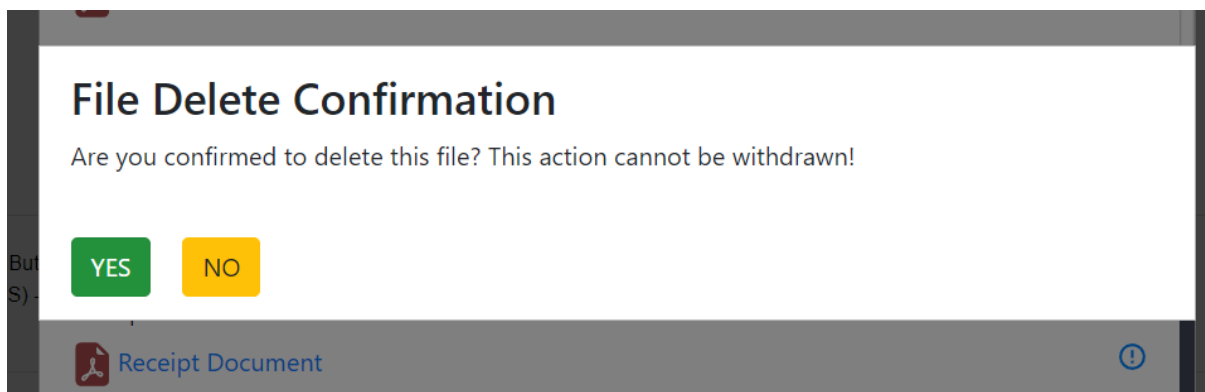


Figure 5.4.1.12 File Delete Confirmation Pop Up Modal

To delete a document, the user can click on the icon located on the right side of the document link, a message modal will pop up to ask for confirmation before deletion.

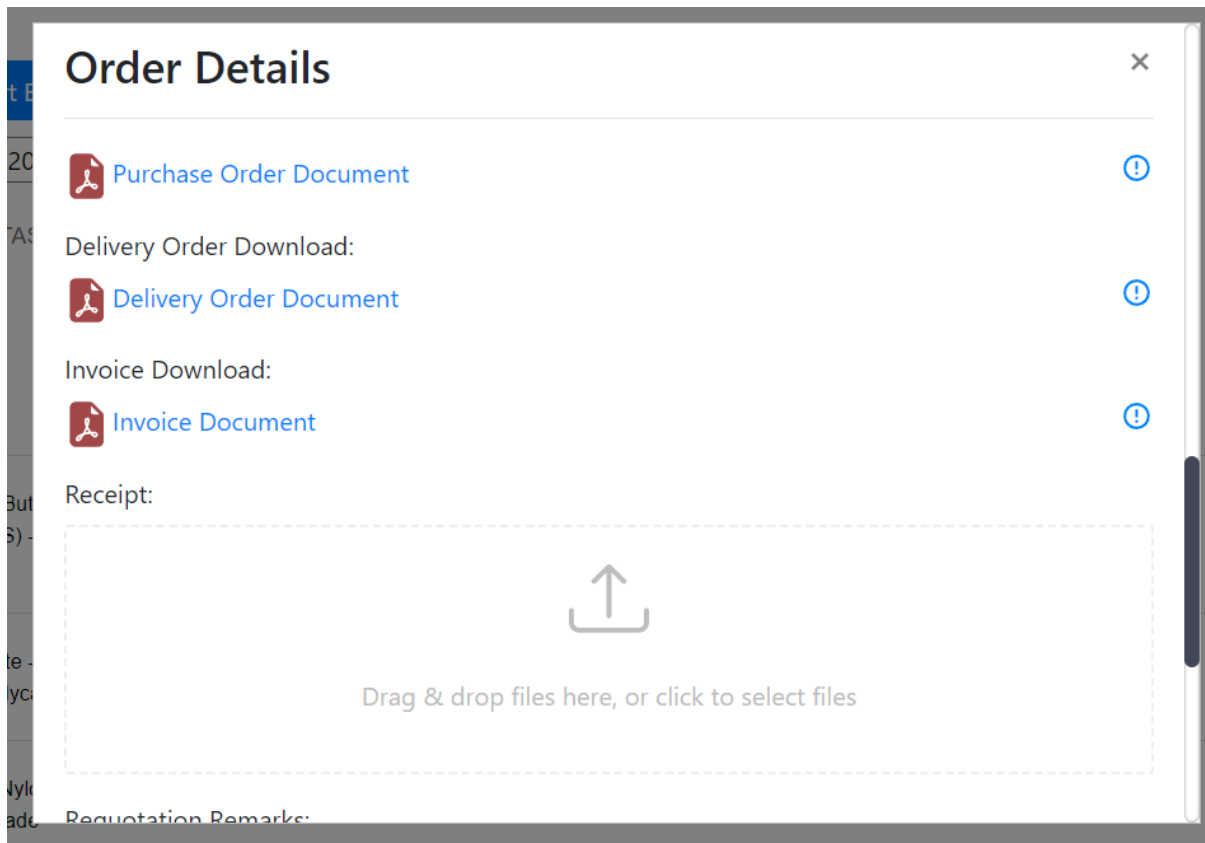


Figure 5.4.1.13 Order Modal Display after File Deleted

The receipt document has been removed, and the user can upload it again.

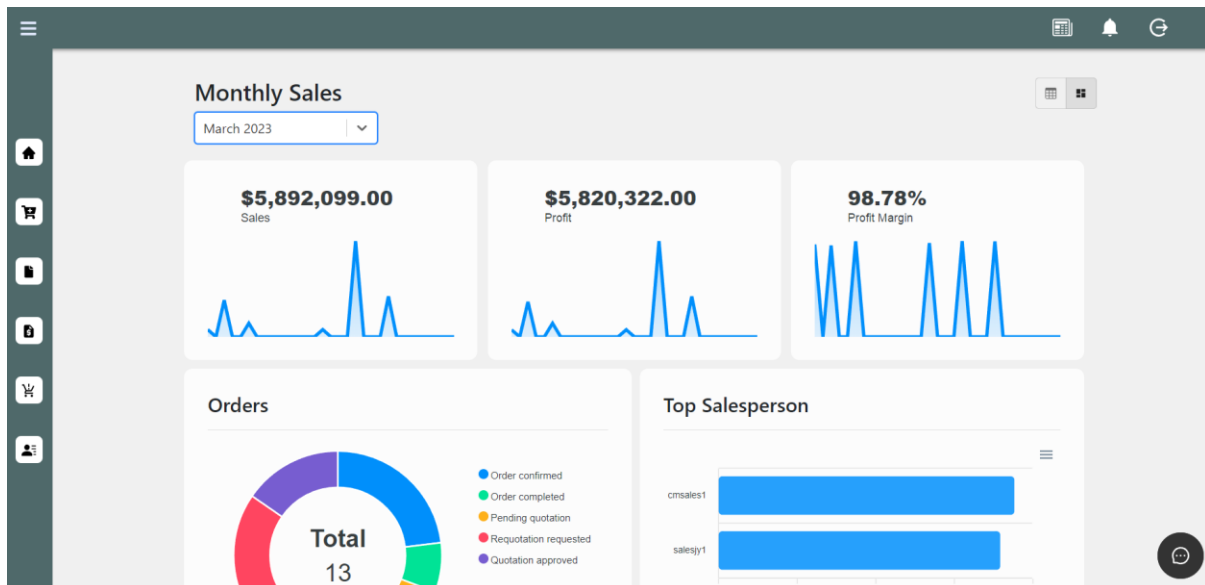


Figure 5.4.1.14 Monthly Sales Dashboard

CHAPTER 5

The toggle in the top right corner of the container can switch between the table and sales dashboard, which displays several key sales indicators such as revenue, profit, orders of the month, top salesperson, products sold performance, and top customers.

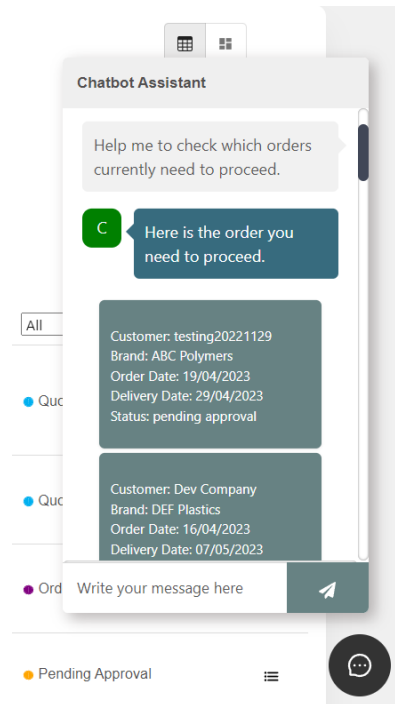


Figure 5.4.1.15 Order Records Display in the Chatbot

The Chatbot feature can be accessed by clicking the float button on the left bottom corner of the page, which allows users to ask for assistance and retrieve information as requested.

5.4.2 Company Module

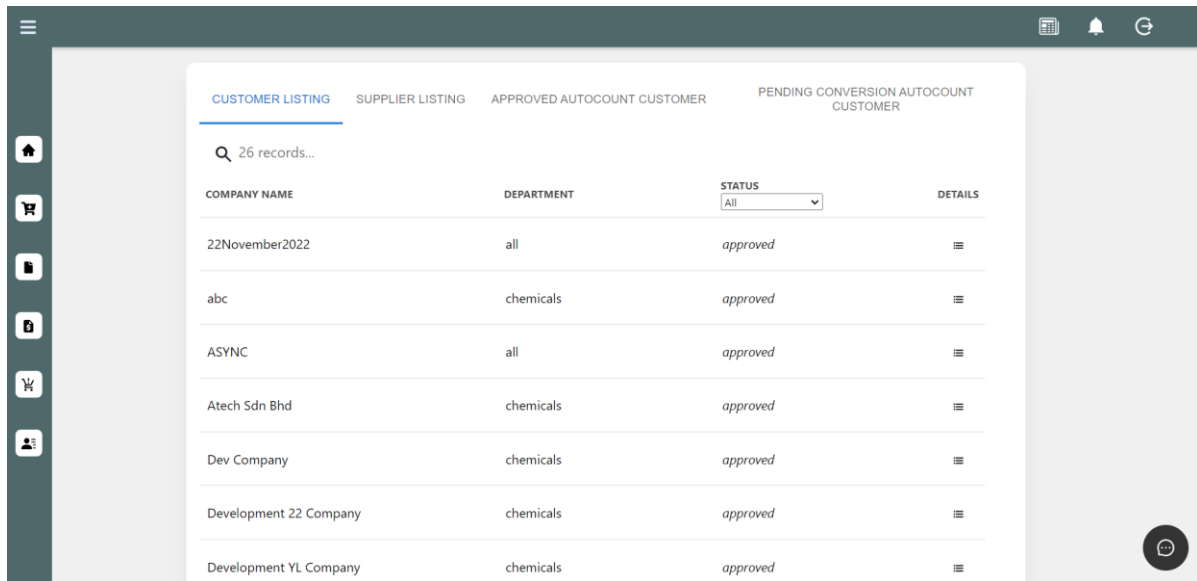


Figure 5.4.2.1 Company List Page

The customer listing and supplier listing are displayed on the company page, using the same display method as the order table. Users can click on the icons of each row to access the details of the respective customer or supplier.

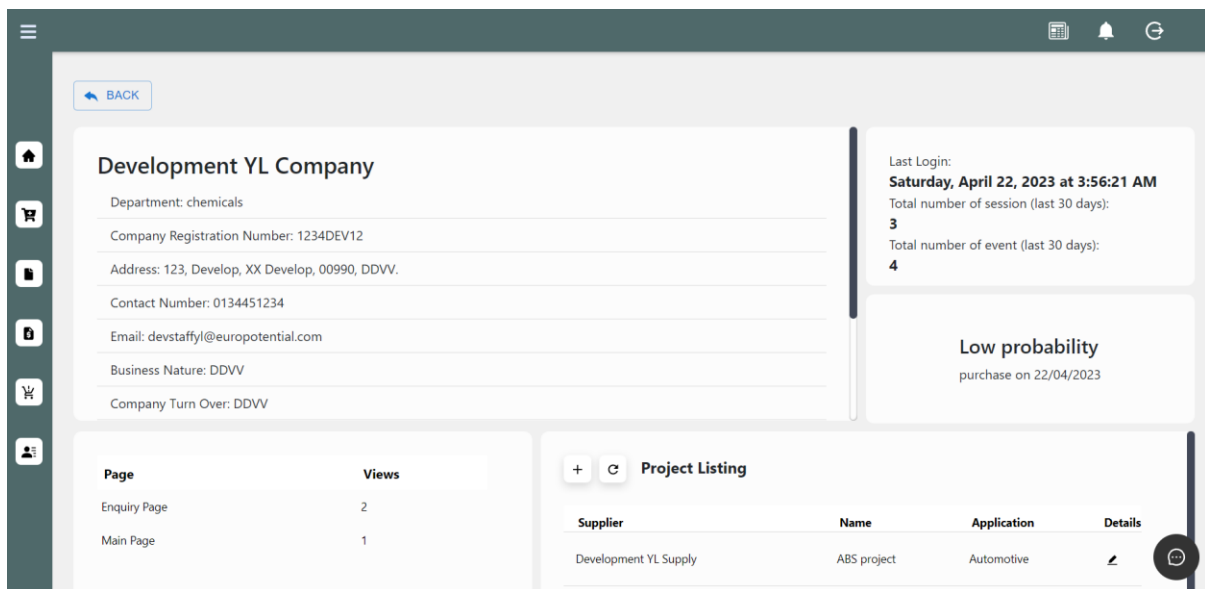
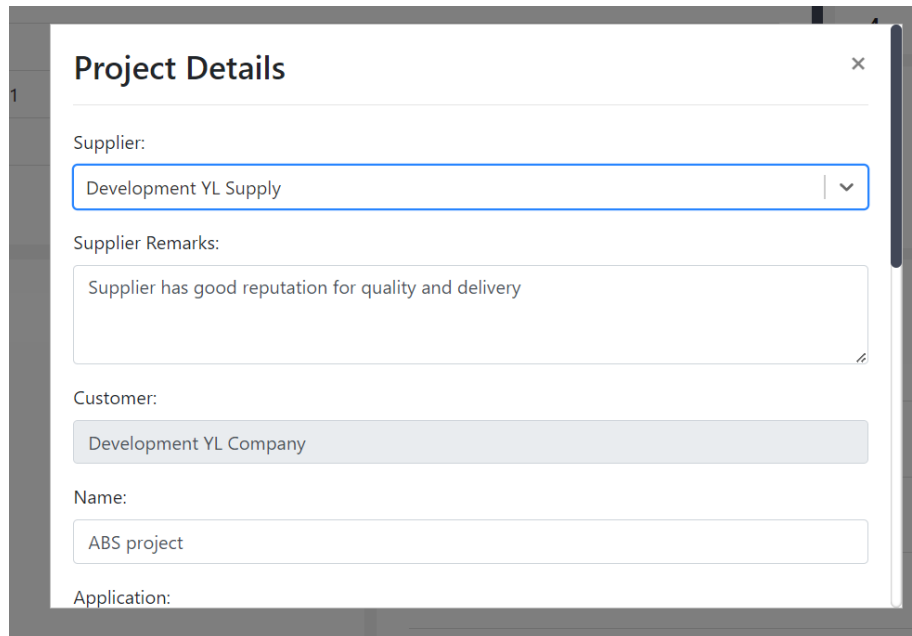


Figure 5.4.2.2 Customer Profile Dashboard

The diagram illustrates the page that appears after clicking on a specific customer row. It displays the customer's company details and their activities on the website, including the date of their last login, total number of sessions, events they have performed, and project details.

CHAPTER 5

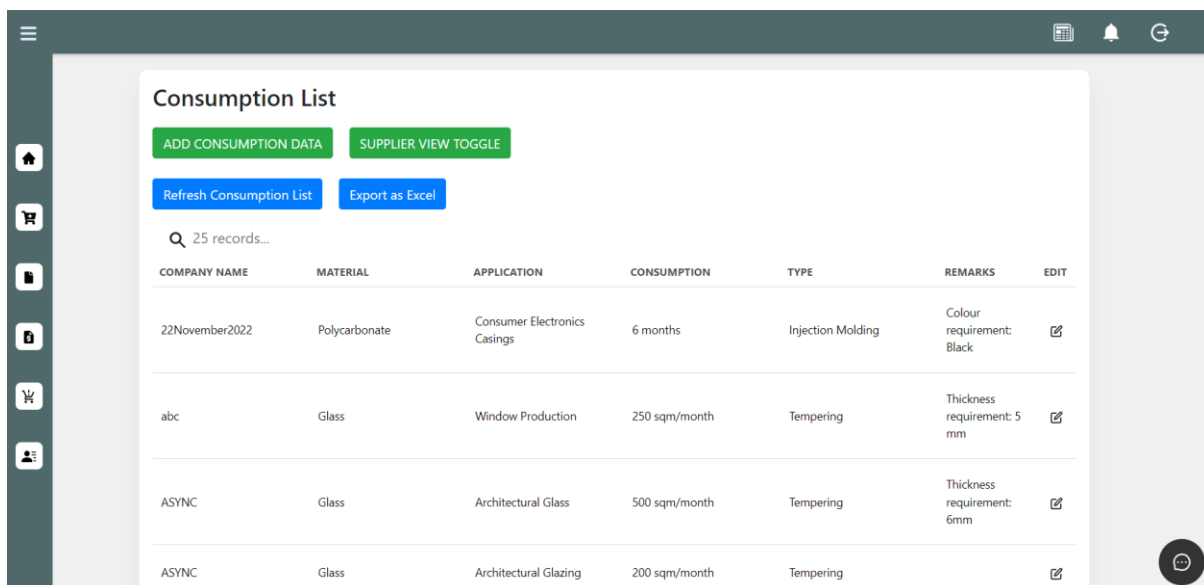
Users can add a new project and update project details with the customer. Additionally, the system performs a prediction of the next purchase date based on the customer's purchase history, to gain insights into their behaviour.



The screenshot shows a 'Project Details' modal window. It contains several input fields: a dropdown menu for 'Supplier' with 'Development YL Supply' selected, a text area for 'Supplier Remarks' containing 'Supplier has good reputation for quality and delivery', a dropdown menu for 'Customer' with 'Development YL Company' selected, a text input for 'Name' with 'ABS project', and an empty text input for 'Application'.

Figure 5.4.2.3 Project Modal with Record

5.4.3 Consumption Module



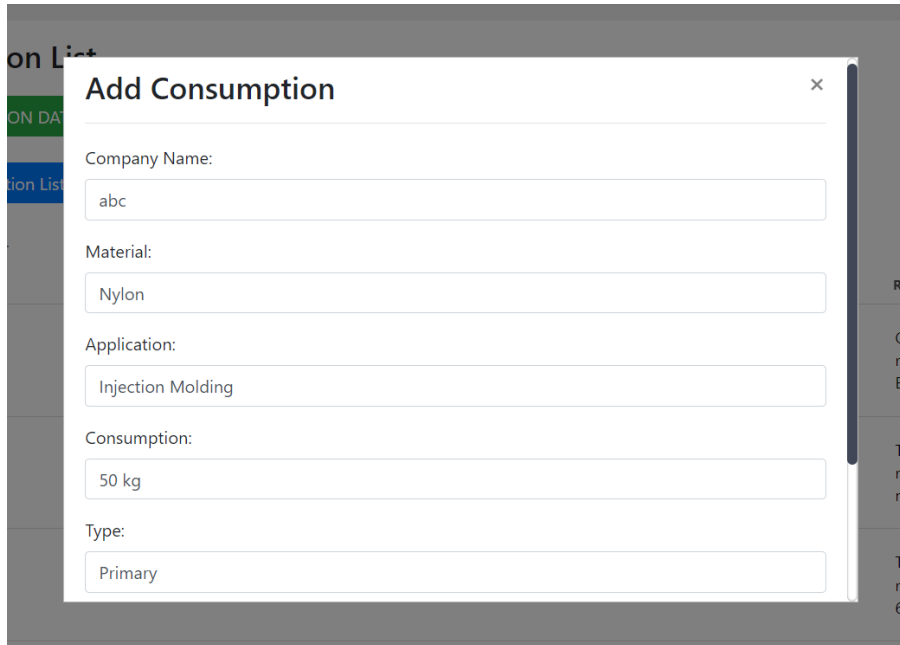
The screenshot shows the 'Consumption List' page. It features a search bar with '25 records...' and several action buttons: 'ADD CONSUMPTION DATA', 'SUPPLIER VIEW TOGGLE', 'Refresh Consumption List', and 'Export as Excel'. Below the search bar is a table with the following data:

COMPANY NAME	MATERIAL	APPLICATION	CONSUMPTION	TYPE	REMARKS	EDIT
22November2022	Polycarbonate	Consumer Electronics Casings	6 months	Injection Molding	Colour requirement: Black	
abc	Glass	Window Production	250 sqm/month	Tempering	Thickness requirement: 5 mm	
ASVNC	Glass	Architectural Glass	500 sqm/month	Tempering	Thickness requirement: 6mm	
ASVNC	Glass	Architectural Glazing	200 sqm/month	Tempering		

Figure 5.4.3.1 Consumption List Page

CHAPTER 5

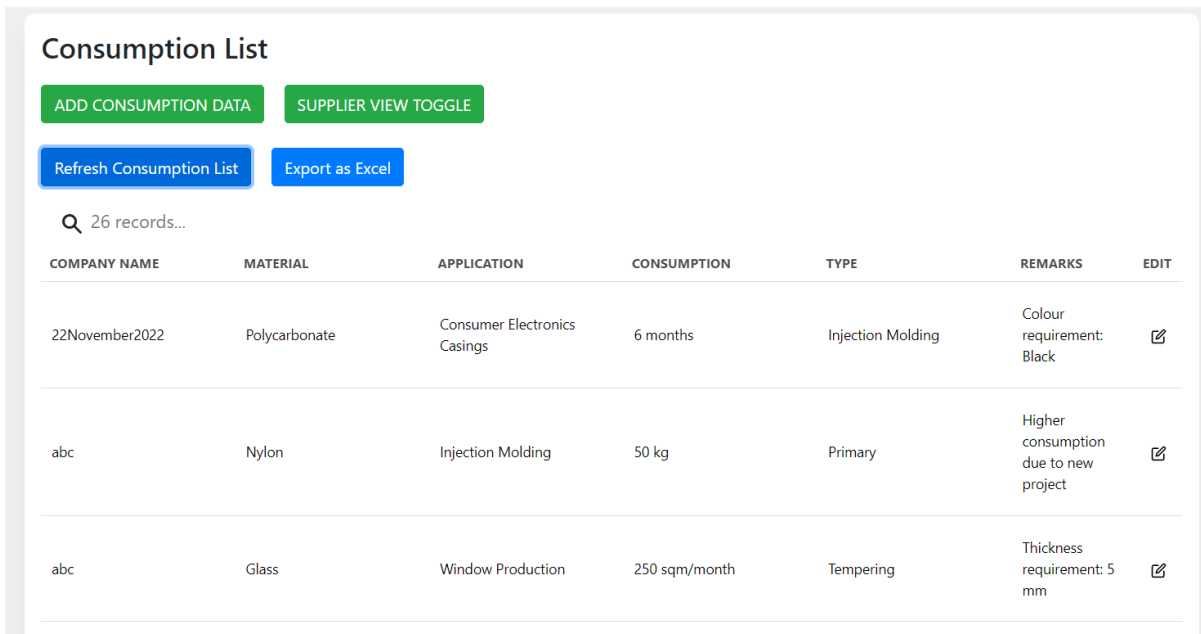
The consumption table presents a detailed list of consumption information, including company name, material, and application. To input new consumption data, use the “Add consumption data” button.



The screenshot shows a modal window titled "Add Consumption" with a close button (X) in the top right corner. The form contains the following fields:

- Company Name: abc
- Material: Nylon
- Application: Injection Molding
- Consumption: 50 kg
- Type: Primary

Figure 5.4.3.2 Add Consumption Modal



The screenshot shows a "Consumption List" interface with the following buttons: "ADD CONSUMPTION DATA", "SUPPLIER VIEW TOGGLE", "Refresh Consumption List", and "Export as Excel". A search bar shows "26 records...". The table below contains the following data:




COMPANY NAME	MATERIAL	APPLICATION	CONSUMPTION	TYPE	REMARKS	EDIT
22November2022	Polycarbonate	Consumer Electronics Casings	6 months	Injection Molding	Colour requirement: Black	
abc	Nylon	Injection Molding	50 kg	Primary	Higher consumption due to new project	
abc	Glass	Window Production	250 sqm/month	Tempering	Thickness requirement: 5 mm	

Figure 5.4.3.3 Consumption List with Updated Record

After adding new consumption data, it is important to click the “Refresh Consumption List” button to ensure that the new data is successfully added to the table list.

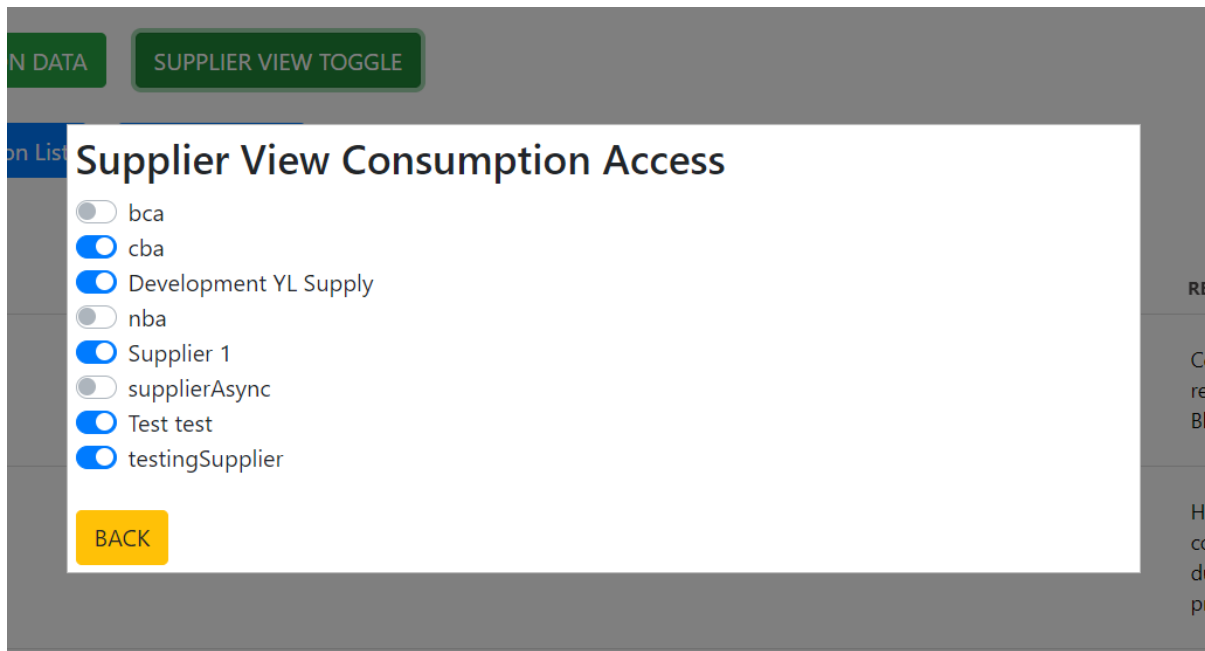


Figure 5.4.3.4 Supplier View Consumption Access Modal

The “Supplier Toggle View” button allows access to a pop-up modal that facilitates changing the supplier's view access. Once the toggle is turned on, suppliers can access consumption data from their page.

	A	B	C	D	E	F	G	H	I	J	K
1	Consumption List										
2	Company Name		Material		Consumption		Application		Type		
3	22November2022		Polycarbonate		6 months		Consumer Electronics Casings		Injection Molding		
4	abc		Glass		250 sqm/month		Window Production		Tempering		
5	ASYNC		Glass		500 sqm/month		Architectural Glass		Tempering		
6			Glass		200 sqm/month		Architectural Glazing		Tempering		
7			Glass		500 sqm/year		Display Panels		Tempering		
8			Glass		2 months		Smartphone Screen		Cutting		
9	Dev Company		ABS		150 MT/month		Automotive Interior Trim		Injection Molding		
10			Polycarbonate		50 kg/month		Optical Lenses		Injection Molding		
11			Nylon		25 MT/month		Textile Fibers		Extrusion		
12	Development 22 Company		Polycarbonate		10 MT/year		Optical Lenses		Injection Molding		
13	Development YL Company		ABS		100 MT/year		Automotive Interior Trim		Injection Molding		
14			ABS		8 months		Automotive Interior Trim		Injection Molding		
15			Polycarbonate		10 MT/month		Electrical Enclosures		Injection Molding		

Figure 5.4.3.5 Excel File with Exported Consumption List

To export consumption data as an Excel file on the desktop, use the “Export as Excel” button. The consumption data in the Excel file is organized by company data, making it easy to read.

5.4.4 Quote Module

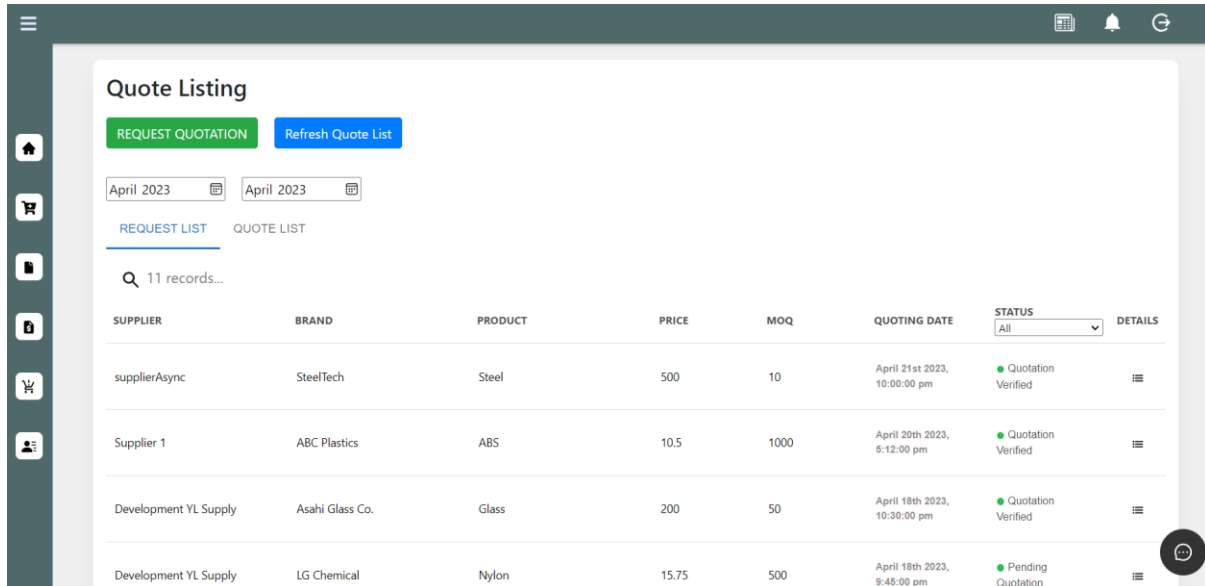


Figure 5.4.4.1 Request List Page

The quote page contains two distinct tables: the request list and the quote list. The request list displays the quote data that has been requested by management, whereas the quote list displays the data that has been directly quoted by suppliers.

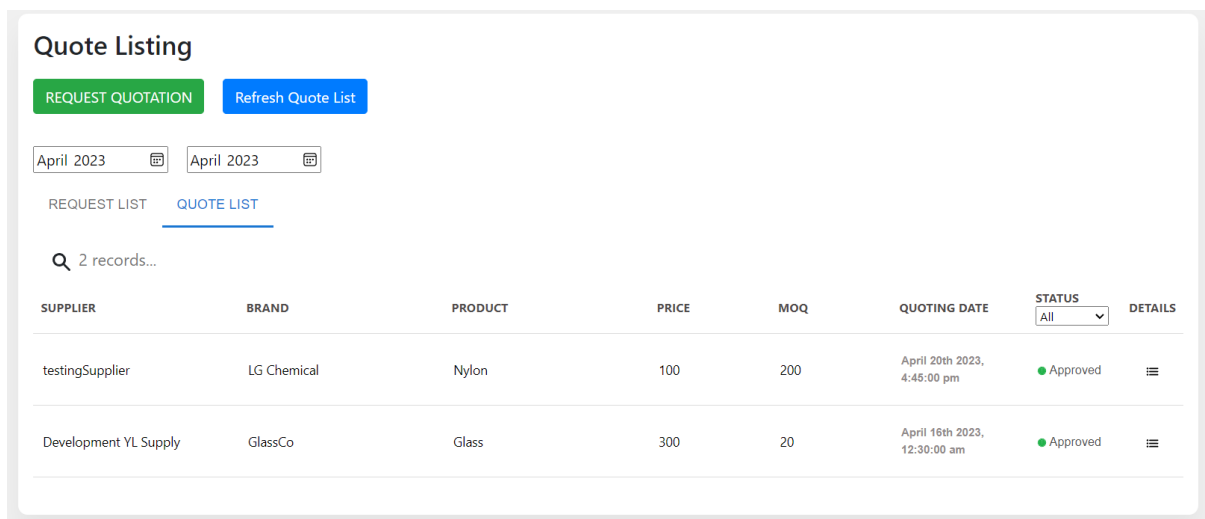
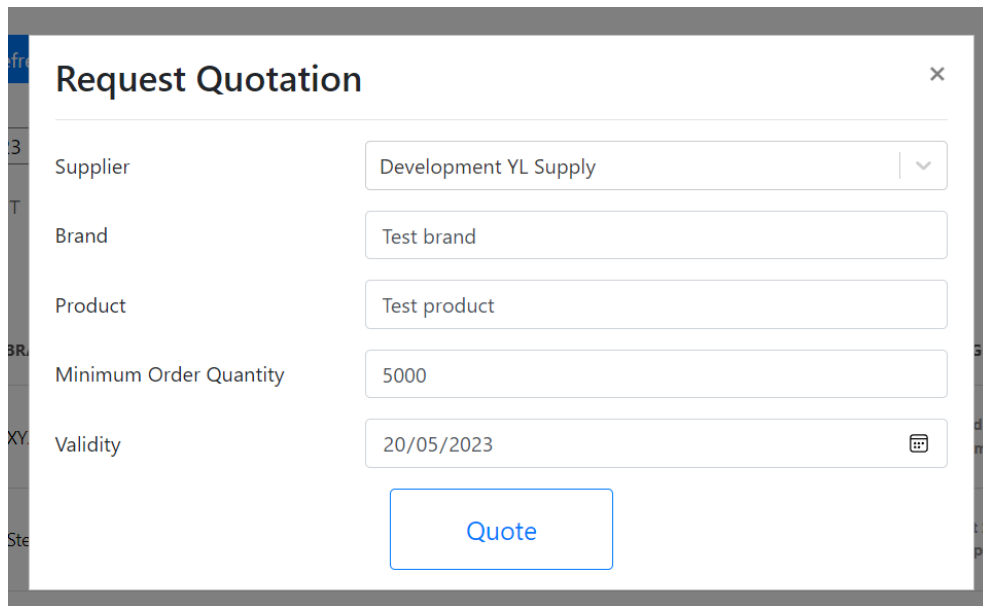
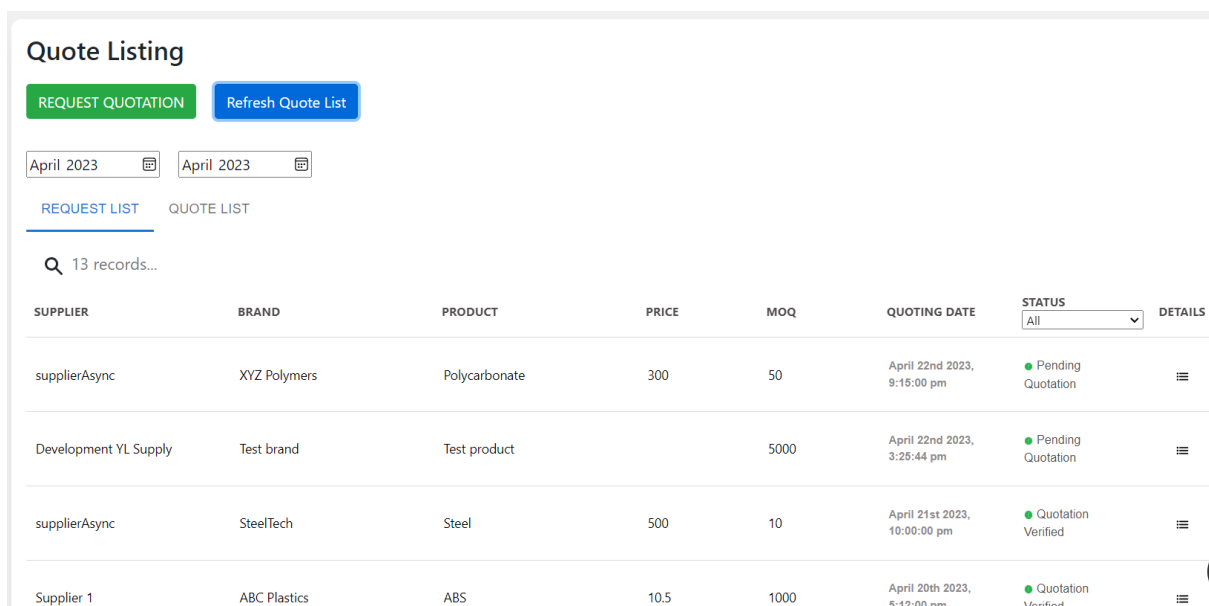


Figure 5.4.4.2 Quote List Page



The image shows a 'Request Quotation' modal window. It contains several input fields: 'Supplier' (Development YL Supply), 'Brand' (Test brand), 'Product' (Test product), 'Minimum Order Quantity' (5000), and 'Validity' (20/05/2023). A 'Quote' button is located at the bottom.

Figure 5.4.4.3 Request Quotation Modal



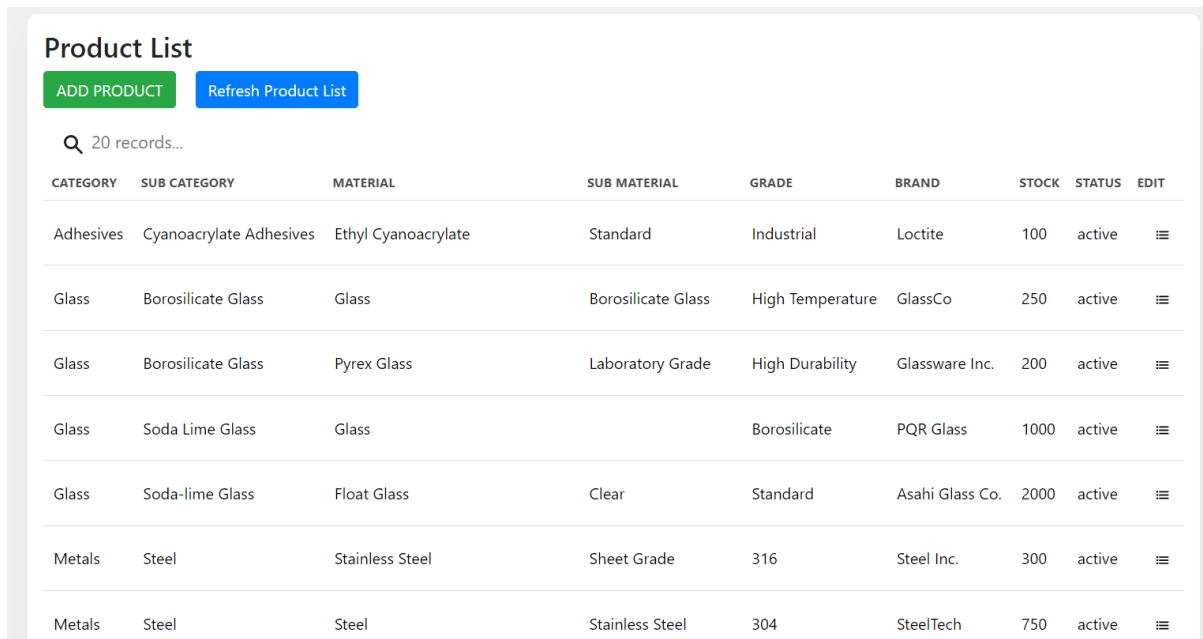
The image shows a 'Quote Listing' interface. It includes a 'REQUEST QUOTATION' button, a 'Refresh Quote List' button, and two date pickers set to 'April 2023'. Below the buttons are tabs for 'REQUEST LIST' and 'QUOTE LIST'. A search bar shows '13 records...'. The main content is a table with columns: SUPPLIER, BRAND, PRODUCT, PRICE, MOQ, QUOTING DATE, STATUS, and DETAILS. The table contains four rows of data.

SUPPLIER	BRAND	PRODUCT	PRICE	MOQ	QUOTING DATE	STATUS	DETAILS
supplierAsync	XYZ Polymers	Polycarbonate	300	50	April 22nd 2023, 9:15:00 pm	Pending Quotation	⋮
Development YL Supply	Test brand	Test product		5000	April 22nd 2023, 3:25:44 pm	Pending Quotation	⋮
supplierAsync	SteelTech	Steel	500	10	April 21st 2023, 10:00:00 pm	Quotation Verified	⋮
Supplier 1	ABC Plastics	ABS	10.5	1000	April 20th 2023, 5:12:00 pm	Quotation Verified	⋮

Figure 5.4.4.4 Request List with Newly Added Record

After clicking the “request quotation” button and providing the necessary data, the user can request that a quotation be sent to a specific supplier. To verify that new requests have been added to the request list, the user can click the “Refresh the quote list” button. Upon approval of the request by the supplier from their page, the status shall change to “quotation verified”.

5.4.5 Product Module



Product List

[ADD PRODUCT](#) [Refresh Product List](#)

🔍 20 records...

CATEGORY	SUB CATEGORY	MATERIAL	SUB MATERIAL	GRADE	BRAND	STOCK	STATUS	EDIT
Adhesives	Cyanoacrylate Adhesives	Ethyl Cyanoacrylate	Standard	Industrial	Loctite	100	active	☰
Glass	Borosilicate Glass	Glass	Borosilicate Glass	High Temperature	GlassCo	250	active	☰
Glass	Borosilicate Glass	Pyrex Glass	Laboratory Grade	High Durability	Glassware Inc.	200	active	☰
Glass	Soda Lime Glass	Glass		Borosilicate	PQR Glass	1000	active	☰
Glass	Soda-lime Glass	Float Glass	Clear	Standard	Asahi Glass Co.	2000	active	☰
Metals	Steel	Stainless Steel	Sheet Grade	316	Steel Inc.	300	active	☰
Metals	Steel	Steel	Stainless Steel	304	SteelTech	750	active	☰

Figure 5.4.5.1 Product List Page

The product module used to manage products in the system. The product list displays details of the project, including category, material, grade, status etc. Users are permitted to add new products or modify existing product details, such as updating the stock number of a particular product.

5.4.6 News Module

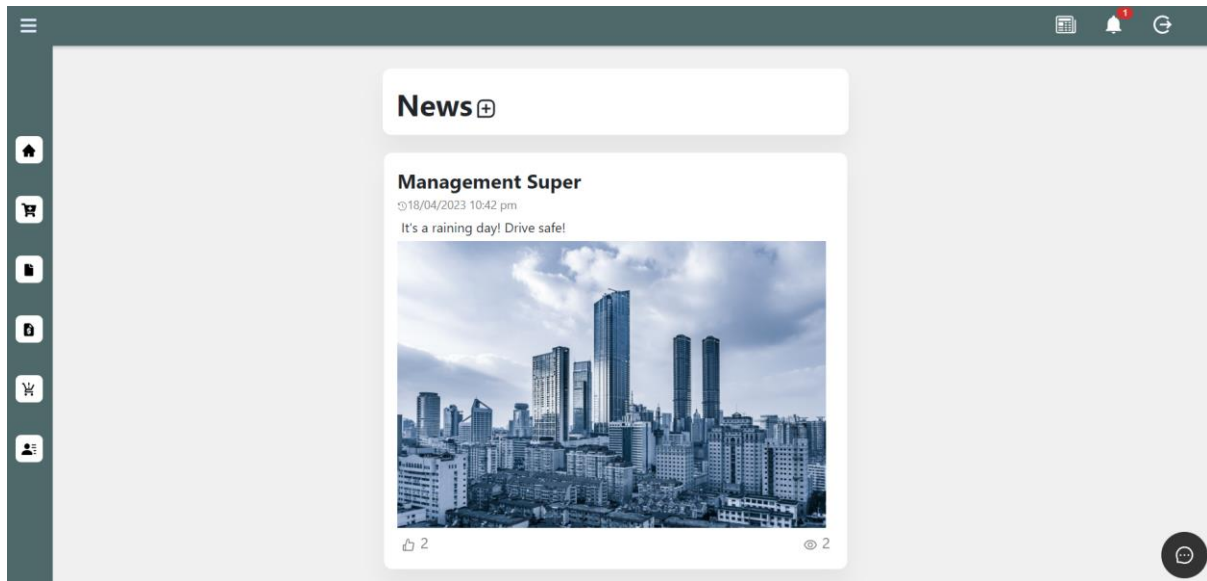


Figure 5.4.6.1 News Page

The news module is intended for the management to share updates to customers. To add a news item, the management can click on the icon next to the “News” header text. The news details include the username of the poster, the posting date, the captions, and image. Additionally, customers can “like” the news, and the management can track customer likes and views by clicking on the icons located at the bottom of each news item.

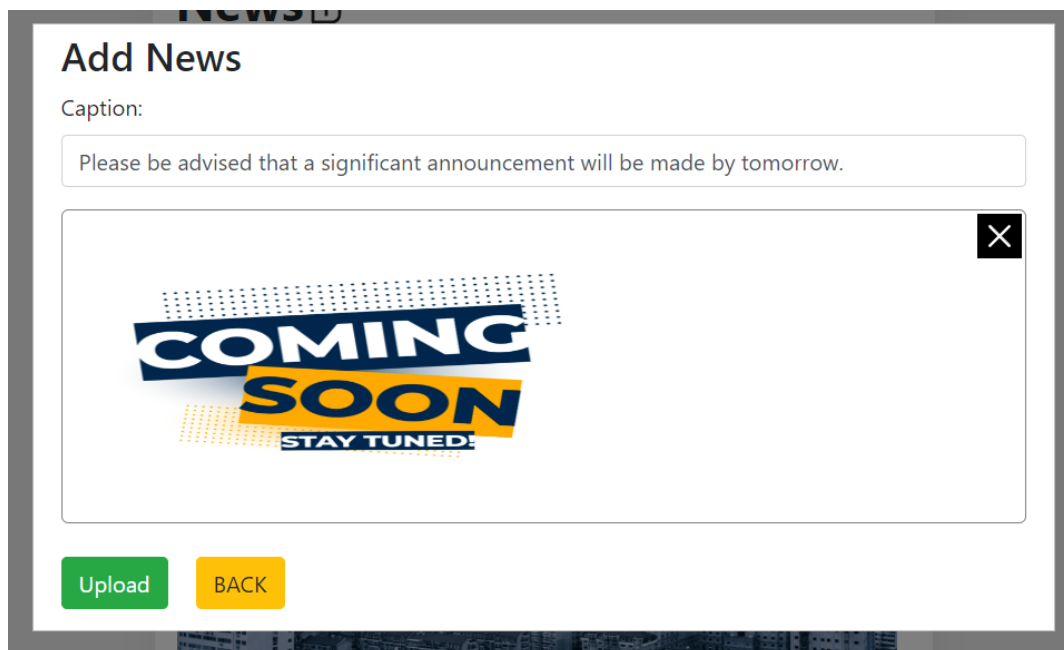


Figure 5.4.6.2 News Modal with New Input

Management Super

🕒 22/04/2023 4:09 pm

Please be advised that a significant announcement will be made by tomorrow.



👍 0

👁️ 0

Figure 5.4.6.3 Newly Added News

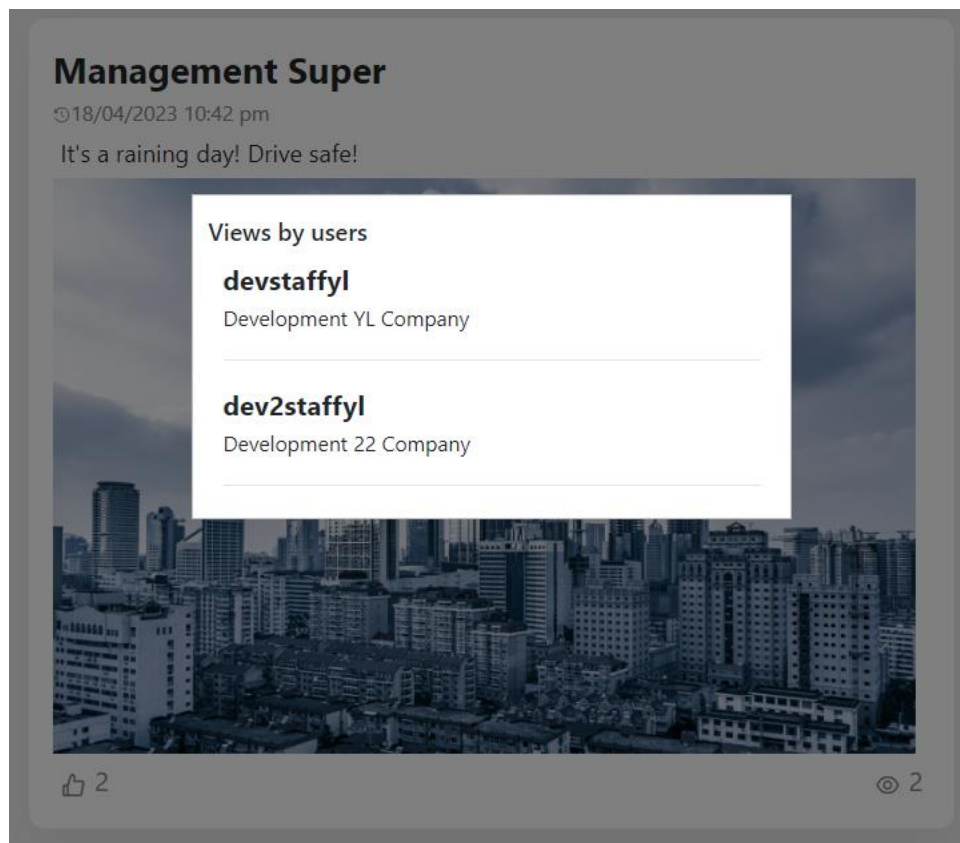


Figure 5.4.6.4 News Viewed by Users Record

5.4.7 Notification Module

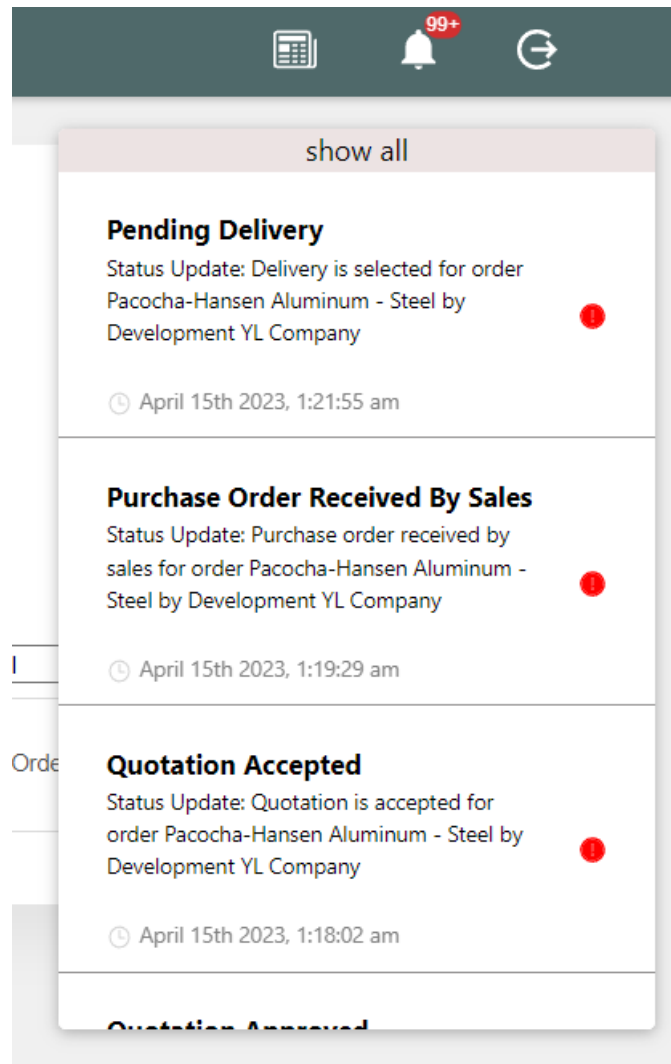


Figure 5.4.7.1 Hover-Over Icon Notification Modal Diagram

The notification modal will be displayed when the user hovers over the icon.

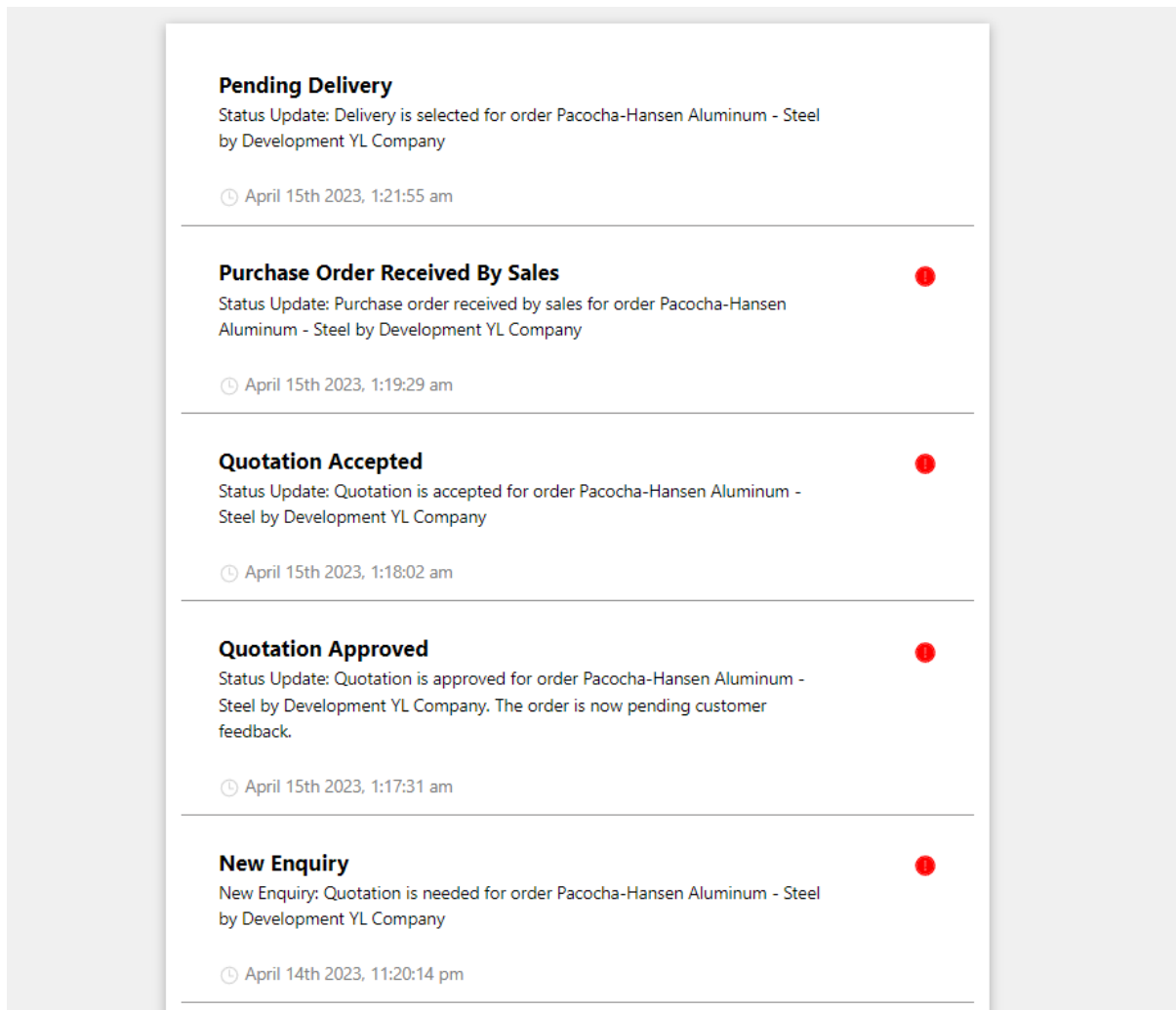


Figure 5.4.7.2 Notification List Page

By clicking the “show all” link, the notification can be displayed on a full page. The user can navigate to specific data by clicking on the notification row.

5.5 Implementation Issues and Challenges

When installing package files using npm command prompt, implementation issues arose due to the large number of packages required for the development of the project. These implementation issues were caused by compatibility issues between dependencies, which made it difficult to install packages properly. To address this problem, the Yarn package was installed as a replacement for npm. Yarn is a package manager that offers many benefits over npm, including faster and more reliable installation of dependencies. It is designed to be more efficient than npm, especially in cases where the number of dependencies is large. Ultimately, to solve the installation problem, the project should use the Yarn command to install the

dependencies. By doing so, the installation process will be smoother and more reliable, reducing the likelihood of errors or compatibility issues. This will help ensure that the project can be developed and deployed with minimal disruption or delay.

While implementing the Express.js back-end web framework, a compatibility issue arose concerning file formats. Express.js supports several file formats, such as TypeScript files (.ts), JavaScript files (.js), and ECMAScript modules (.mjs). Initially, a TypeScript configuration file was used to connect React.js and MongoDB. However, it was discovered that the TypeScript file format was not compatible with the system environment of the desktop being used. Moreover, JavaScript files with the “.js” extension were also found to be incompatible with the server. Ultimately, to resolve the issue, the server was configured to support the “.mjs” file extension, which allowed the Express.js project to run successfully.

5.6 Concluding Remark

In conclusion, this chapter details the process of integrating MongoDB, Express.js, React.js, and Node.js to create a full-stack web application. This phase involves the installation and configuration of multiple components, including database management, server configuration, and front-end development. It also involves the resolution of compatibility issues between different technologies, as well as the development of custom solutions for specific project requirements. Successful implementation ensures that the MERN stack web application functions seamlessly, providing an optimal user experience. It is the foundation for further development, testing, and deployment of the system, ultimately leading to a high-quality and reliable product.

Chapter 6

System Evaluation and Discussion

6.1 System Testing and Performance Metrics

The system testing will be conducted using the black box testing approach. This is a widely accepted method of software testing that focuses on evaluating the system's external behavior without the tester having access to the system's internal workings. In this approach, the tester will focus on testing the system's inputs, outputs, and user interfaces to assess its functionality and usability. The test cases have identified with the web portal's functionality and performed various types of inputs and outputs to generate responses. These responses have been verified against the expected outputs. Any test cases that produced incorrect results were retested, and the expected outputs were validated. Black box testing can help to ensure that the web portal meets the expected standards of functionality and usability.

6.2 Testing Setup and Result

Table 6.2.1 Login Test Case

Test Case	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
User login with valid credentials at “management” tab	<ol style="list-style-type: none"> 1. Enter Username 2. Enter Password 	Username = management Password = management@123	User should navigate to “order” page.	Same with expected result.	Pass
User login with invalid credentials at “management” tab	<ol style="list-style-type: none"> 1. Enter Username 2. Enter Password 	Username = management Password = Management@12	User should receive an authentication failure message.	Same with expected result.	Pass
User login with valid credentials at “customer/supplier” tab	<ol style="list-style-type: none"> 1. Enter Username 2. Enter Password 	Username = management Password = management@123	User should receive an authentication failure message.	Same with expected result.	Pass
User pastes the “order page” URL when login	<ol style="list-style-type: none"> 1. Copy the “order page” URL. 2. Paste the “order page” URL at the address bar and click the “Enter” keypad. 	URL: https://fyp-project-348109.firebaseio.com/#/OrderListing	User will access the “order page”	Same with expected result.	Pass
User pastes the “order page” URL when logout	<ol style="list-style-type: none"> 1. Copy the “order page” URL. 2. Paste the “order page” URL at the address bar and click the “Enter” keypad. 	URL: https://fyp-project-348109.firebaseio.com/#/OrderListing	The website will redirect to login page.	Same with expected result.	Pass
User pastes the “login page” URL when login	<ol style="list-style-type: none"> 1. Copy the “order page” URL. 2. Paste the “order page” URL at the address bar and click the “Enter” keypad. 	URL: https://fyp-project-348109.firebaseio.com/#/	The website will redirect to “order page”.	Same with expected result.	Pass
User stay inactive after login	<ol style="list-style-type: none"> 1. Login with valid authentication 2. Stop interact with web page 		The session will be closed after user inactive for 5 minutes.	Same with expected result.	Pass

CHAPTER 6

User stay active after login	<ol style="list-style-type: none"> 1. Login with valid authentication 2. Keep interact with web page 		The session will not be closed after user active for 5 minutes.	Same with expected result.	Pass
------------------------------	--	--	---	----------------------------	------

Table 6.2.2 Order Module Test Case

Test Case	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
User retrieves order data	<ol style="list-style-type: none"> 1. Access the “order page”. 2. Click on a row in the order list. 	Brand = LG Chemical Customer = Development YL Company Order Date = April 22nd 2023, 3:57:33 am	The specific order details will show in the Pop-up modal that same with test date.	Same with expected result.	Pass
User edits input with invalid data	<ol style="list-style-type: none"> 1. Access the “order page”. 2. Click on a row in the order list. 3. Click the Validity date input. 4. Type invalid date 	Validity = “21/02/23123 ” Validity = “Test Validity”	User should not be able to insert the data.	Same with expected result.	Pass
User clicks button without insert necessary data	<ol style="list-style-type: none"> 1. Access the “order page”. 2. Click on a row with status “pending quotation” in order list. 3. Click the update button 	Price is empty. Validity date is empty.	User should receive an invalid message. User should not be able to update the status.	Same with expected result.	Pass
User clicks button with insert necessary data	<ol style="list-style-type: none"> 1. Access the “order page”. 2. Click on specific row of order with status “pending quotation”. 3. Insert data. 4. Click the update button. 	Price = 12.50. Validity date = “20/05/2023.	The order data should be updated. User can receive the new order data.	Same with expected result.	Pass

CHAPTER 6

Customer receives the notification after the order data updated by management	<ol style="list-style-type: none"> 5. Access the “purchase page” with customer authentication. 6. Open the notification modal. 7. Check the notification. 	Customer = Development YL Company Username = “devstaffyl” Password = “staff”	Customer should receive the notification.	Same with expected result.	Pass
User select PDF file using file picker and preview the PDF file	<ol style="list-style-type: none"> 1. Open the order modal with status “order completed”. 2. Select the PDF file using file picker. 3. PDF file shows in the input form 4. User clicks the link to preview file. 		The preview file is same with the upload file.	Same with expected result.	Pass
User select Word file using file picker	<ol style="list-style-type: none"> 1. Open the order modal with status “order completed”. 2. Select the Word file using file picker. 		System indicates wrong file format and reject the file.	Same with expected result.	Pass
User select PDF file using drag and drop method	<ol style="list-style-type: none"> 1. Open the order modal with status “order completed”. 2. Open file explorer and find a PDF file. 3. Drag the PDF file and drop at the file drop zone. 		The PDF will show in the input form and able to preview the file.	Same with expected result.	Pass
User selects PDF file and upload a document to server	<ol style="list-style-type: none"> 1. Select a PDF file for “delivery order”. 2. Click the upload button 		The “successful uploaded” message will be received. User can check the PDF file with URL.	Same with expected result.	Pass

CHAPTER 6

User selects three PDF files and upload it to server	<ol style="list-style-type: none"> 1. Select a PDF file for “delivery order”. 2. Select a PDF file for “invoice”. 3. Select a PDF file for “receipt”. 		The “successful uploaded” message will be received at once. Three PDF files can be uploaded. User can check the PDF files with URL.	Same with expected result.	Pass
User deletes PDF file from file server	<ol style="list-style-type: none"> 1. Open order modal with upload file. 2. Select the icon beside the file 3. Confirmation modal pop up 4. Click confirm button. 		The “successful deleted” message will be received. Three PDF files can be uploaded. User can check the PDF files with URL.		

Table 6.2.3 Project Management Test Case

Test Case	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
User retrieves project data	<ol style="list-style-type: none"> 1. Access the “project list”. 2. Click on a row in the project list. 	Project name = ABS Application = Automotive	The specific project details will show in the Pop-up modal that same with test date.	Same with expected result.	Pass
User edits project data	<ol style="list-style-type: none"> 3. Edit the data. 4. Click on upload button 	Status = Inactive	Project status data should change from “Active” to “Inactive”.	Same with expected result.	Pass
User adds a new project data	<ol style="list-style-type: none"> 1. Click the add button. 2. Add new project modal pop up. 3. Enter the project data. 4. Click confirm button. 	Customer = Development YL Company Supplier = Development YL Supply Project name = ABS Project	User should receive a success message. User can retrieve the data to show on the table list.	Same with expected result.	Pass

CHAPTER 6

	5. Retrieve the new data by clicking the refresh button.	Application = Automotive parts			
User adds a new project without insert data	<ol style="list-style-type: none"> 1. Click the add button. 2. Add new project modal pop up. 3. Click confirm button. 		<p>User should receive a message stated no empty is allowed. The input form will mention which data should be added. The new project data can't be added.</p>	Same with expected result.	Pass

Table 6.2.4 Consumption Management Test Case

Test Case	Test Steps	Test Data	Expected Results	Actual Results	Pass /Fail
User retrieves consumption data	<ol style="list-style-type: none"> 1. Access the “consumption page”. 2. Click on a row in the consumption list. 	Company = abc Material = Nylon Application = Injection Molding	The specific consumption details will show in the Pop-up modal that same with test date.	Same with expected result.	Pass
User edits consumption data	<ol style="list-style-type: none"> 3. Edit the data. 4. Click on confirm button 	Consumption = 10 MT/month	Project status data should change from “6 months” to “10 MT/month”.	Same with expected result.	Pass
User adds new consumption data	<ol style="list-style-type: none"> 1. Access the “consumption page”. 2. Click “add consumption data” button. 3. Modal open. 4. Insert consumption data. 5. Click confirm button. 	Company = ASYNC Material = Glass Application = Architecture Consumption = 300 sqm/month Type = Tempering	The new consumption should be saved in DB. User can retrieve the data by clicking the refresh button.	Same with expected result.	Pass

CHAPTER 6

User adds new consumption without insert data	<ol style="list-style-type: none"> 1. Access the “consumption page”. 2. Click “add consumption data” button. 3. Modal open. 4. Click confirm button. 	All the consumption data.	User should receive a message stated no empty is allowed. The new consumption data should not be saved in DB. Input form specifies required data.	Same with expected result.	Pass
User exports the consumption data as Excel file	<ol style="list-style-type: none"> 1. Access the “consumption page”. 2. Click “export as Excel” button 		The user receives an Excel file on their desktop that contains all the data in a spreadsheet format. The data is organized by company name and is accurate.	Same with expected result.	Pass

Table 6.2.5 Quote Management Test Case

Test Case	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
User retrieves quote data	<ol style="list-style-type: none"> 1. Access the “quote page”. 2. Click on a row in the quote list. 	Supplier = supplierAsync Brand = XYZ Polymers Product = Polycarbonate	Project details shown in modal that same with test data.	Same with expected result.	Pass
User request new quotation	<ol style="list-style-type: none"> 1. Access the “quote page”. 2. Click “request quotation” button. 3. Request quotation modal open. 4. Insert quote data. 5. Click quote button. 	Supplier = Development YL Supply Brand = LG Chemical Product = Nylon Price = 15 Quantity = 4000 Validity = 30/04/2023	New quotation saved to DB. User retrieves data via refresh button.	Same with expected result.	Pass

CHAPTER 6

User request new quotation without insert data	<ol style="list-style-type: none"> 1. Access the “quote page”. 2. Click “request quotation” button. 3. Request quotation modal open. 4. Click quote button. 		Input form specifies required data. Empty values not allowed (prevents DB save).	Same with expected result.	Pass
User filters the request list by changing the first month picker	<ol style="list-style-type: none"> 1. Access the “quote page”. 2. Click the first month picker. 	First month picker = January 2023 Last month picker = April 2023	User receives records between January 2023 and April 2023.	The first results show that records in April 2023 are filtered out. After modifying the code, the records can be retrieved.	Pass
User filters the request list by changing the last month picker	<ol style="list-style-type: none"> 1. Access the “quote page”. 2. Click the last month picker. 	First month picker = April 2023 Last month picker = March 2023	System auto-updates first month picker to March 2023. User receives records within same month.	First attempt at changing the first month picker failed. Code fix resulted in expected results on fifth test.	Pass

6.3 Project Challenges

First, dealing with unfamiliar domains and ambiguous requirements is a challenge at the beginning of project development. Web portal development is designed for a company in the welding and safety industry. Some business terms may be incomprehensible to novices. This is a critical problem if the requirements are misunderstood, and the results are not as expected. In addition, communication was also difficult because some technical terms were difficult to explain verbally. Fortunately, these problems can be resolved after the newbie period.

Low integration between customer or supplier modules and mobile applications can be a significant challenge. Since the modules are handled by different developers, the function completion time may not be synchronized, which will cause these modules not to be well connected or coordinated, resulting in inefficient and difficult data exchange or communication, which may eventually affect the overall function of the system. Meeting these challenges requires a proactive approach, effective communication, and a willingness to adapt and learn.

Finally, data validation is another challenge that can arise, especially during system testing. Ensuring that the data being processed by the system is accurate, reliable, and conforms to the expected format can be time-consuming and resource intensive. Since the data used for the test is randomly generated, the situation may differ from the actual scenario, such as the number of orders the company processes every day.

6.4 Objectives Evaluation

The first objective is achieved to develop a user-friendly web portal with customizable role-based access controls, designed to streamline the procurement and shipment process, and provide an efficient alternative to traditional business operations. The developed portal is designed for role-based access control, and salespeople have limited access to orders compared to managers. Specifically, salespeople can solely access order quantities for their respective customers, whereas managers can access orders for different salespeople. Conventional business operations are both time-consuming and expensive, as the company is obliged to cover transportation costs for each salesperson to meet customers in person. The portal is considered more efficient as it enables users to handle numerous orders in a shorter duration. By this means, the initial objective has been accomplished.

The second objective is to develop an integrated order and shipment module that allows users to manage the entire order progress from a single platform, including order progress tracking, updating orders, and uploading files. Currently, the portal has fully developed the

CHAPTER 6

entire order process involving order shipping options based on the requirements of the client. Managers and salespeople can easily track order progress, especially when a customer updates an order, notifications are sent to the associated account. Managers and salespeople can update orders and upload files during certain order statuses. Thus, the web portal is achieved to manage the entire order progress.

The last goal of data consistency and accuracy between mobile applications and web portals on the same module has been achieved by implementing real-time synchronization and data validation mechanisms, minimizing data discrepancies, and ensuring data integrity. The backend server can validate the request before continuing to process the request. If validation fails, the request will be rejected.

6.5 Concluding Remark

Based on the successful completion of the system testing, declaration of project challenges, and achievement of project objectives, it can be concluded that the project has met all requirements and has been executed efficiently.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

In conclusion, the procurement and shipment portal using MERN stack has been successfully developed and is now ready to be launched. The web portal produced with user-friendly interfaces and meets all the requirements specified in the project scope. The functionality of system has been designed and delivered. The use of MERN stack has allowed for a seamless integration of the frontend and backend, resulting in a smooth and efficient experience for users. With the implementation of features such as order progress tracking, updating orders and uploading files, the portal is sure to streamline the procurement and shipment processes for businesses. We are confident that this portal will be an asset to any organisation that chooses to utilize it.

7.2 Recommendations

To enhance the reliability of the web portal in future development, it is recommended that the client's requirements are clearly identified. Since the client is the primary user of the product, the developed functions should meet their expectations, and the user interface design should be simple yet efficient. It is essential to ensure that the design is not based solely on personal opinions but is also acceptable to others. In addition, it is crucial to consider developing usable functionalities for the web portal. While some features may not be initially requested by the client, conducting studies may reveal their usefulness in enhancing business operations effectiveness. Developing advanced features is encouraged to build a robust system that can keep up with evolving needs.

REFERENCES

- [1] K. Vyas, "Mobile App Vs Website – Which to build first for your startup or Business?," CustomerThink, 27 July 2018. [Online]. Available: <https://customerthink.com/mobile-app-vs-website-which-to-build-first-for-your-startup-or-business/>. [Accessed 29 July 2022].
- [2] "React - The library for web and native user interfaces," reactjs.org, [Online]. Available: <https://react.dev/>. [Accessed 15 March 2023].
- [3] B. Venkat, S. Indla, Y. Puranik, P. G. Student, and P. E. S. M. College, "Review on React JS," vol. 5, no. 4, pp. 1137–1139, 2021. [Accessed 17 March 2023].
- [4] "Virtual dom and Internals," React. [Online]. Available: <https://legacy.reactjs.org/docs/faq-internals.html>. [Accessed 23 March 2023].
- [5] S. Aggarwal, "Modern Web-Development using ReactJS," International Journal of Recent Research Aspects, vol. 5, pp. 133–137, 2018, Available: <http://ijrra.net/Vol5issue1/IJRRRA-05-01-27.pdf>. [Accessed 23 March 2023].
- [6] N. Raicic, "Introduction to React Ecosystem," Medium, Jun. 13, 2020. <https://medium.com/@nikola.raicic/introduction-to-react-ecosystem-a2fd67525530> [Accessed 23 March 2023].
- [7] "Introduction | Vue.js," vuejs.org. <https://vuejs.org/guide/introduction.html#single-file-components>. [Accessed 1 April 2023].
- [8] "The Good and the Bad of Vue.js Framework Programming," AltexSoft, Sep. 11, 2019. <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/>. [Accessed 1 April 2023].
- [9] "Vue Partners," vuejs.org. <https://vuejs.org/partners/>. [Accessed 1 April 2023].
- [10] "Angular," angular.io. <https://angular.io/guide/what-is-angular>. [Accessed 2 April 2023].
- [11] C. Deshpande, "What is Angular?: Architecture, Features, and Advantages," Simplilearn.com, Feb. 10, 2022. <https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>. [Accessed 2 April 2023].
- [12] Altexsoft, "The Good and the Bad of Angular Development," AltexSoft, Mar. 25, 2020. <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/>. [Accessed 2 April 2023].

REFERENCES

- [13] K. Vyas, "8 Big Advantages of Using MySQL," Datamation, Feb. 03, 2023. <https://www.datamation.com/storage/8-major-advantages-of-using-mysql/#:~:text=Scalability%20on%20Demand>. [Accessed 2 April 2023].
- [14] "MySQL Security - Common Threats And 8 Best Practices," Satori. <https://satoricyber.com/mysql-security/mysql-security-common-threats-and-8-best-practices/>. [Accessed 15 April 2023].
- [15] D. Munasingha, "Advantages and Disadvantages of using MySQL," Medium, May 21, 2021. <https://diliru.medium.com/advantages-and-disadvantages-of-using-mysql-36f6ffce3fa3>. [Accessed 15 April 2023].
- [16] "Introduction to MongoDB — MongoDB Manual," <https://www.mongodb.com/docs/manual/introduction/>. [Accessed 15 April 2023].
- [17] D. Bruce, "Understanding the pros and cons of mongodb," knowledgenile. [Online]. Available: <https://www.knowledgenile.com/blogs/pros-and-cons-of-mongodb/#MongoDBScalability>. [Accessed 15 April 2023].
- [18] "Firebase Advantages and Disadvantages," Sep. 08, 2022. https://blog.back4app.com/firebase-advantages-and-disadvantages/#Firebase_Advantages [Accessed 20 April 2023].
- [19] A. Elliott, "Drawbacks of Firestore," Medium, Oct. 10, 2018. https://medium.com/@aelliott_68874/drawbacks-of-firestore-9f46b46adb05 [Accessed 20 April 2023].
- [20] "Easy quotation software," Odoo, [Online]. Available: <https://www.odoo.com/app/sales>. [Accessed 12 July 2022].
- [21] "Enquiry & Quote Details," WisdmLabs, [Online]. Available: <https://quoteup.wisdmlabs.com/wp-admin/admin.php?page=quoteup-details-new>. [Accessed 27 July 2022].
- [22] "Edit Enquiry," WisdmLabs, [Online]. Available: <https://quoteup.wisdmlabs.com/wp-admin/admin.php?page=quoteup-details-edit&id=1527>. [Accessed 26 July 2022].
- [23] Site Impact, [Online]. Available: <https://www.siteimpact.com/>. [Accessed 28 July 2022].
- [24] K. Beck, J. Grenning, R. C. Martin, M. Beedle, J. Highsmith, S. Mellor, A. v. Bennekum, A. Hunt, K. Schwaber, A. Cockburn, R. Jeffries, J. Sutherland, W. Cunningham, J. Kern, D. Thomas, M. Fowler and B. Marick, "Manifesto for Agile Software

REFERENCES

Development," Agile Alliance, 2001. [Online]. Available: <http://agilemanifesto.org/>. [Accessed 15 August 2022].

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 2
Student Name & ID: Chin Pey Mian, 20ACB04071	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Development of Non-ICT Company Procurement and Shipment Portal using Referenced Mobile Application	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

The order and shipment module development are done with able to complete the whole order progress including selecting shipments and checking the shipment.

2. WORK TO BE DONE

Change the method of uploading and retrieving files.
Reorganise the order progress flow.

3. PROBLEMS ENCOUNTERED

Some request performance is low.

4. SELF EVALUATION OF THE PROGRESS

The progress was delayed.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 4
Student Name & ID: Chin Pey Mian, 20ACB04071	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Development of Non-ICT Company Procurement and Shipment Portal using Referenced Mobile Application	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

The method of uploading and retrieving files have been changed.
The new action was added for the confirmation of the purchase order upload by the customer.

2. WORK TO BE DONE

Improves system performance, allowing faster processing of requests.

3. PROBLEMS ENCOUNTERED

No problems encountered.

4. SELF EVALUATION OF THE PROGRESS

So far so good.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 6
Student Name & ID: Chin Pey Mian, 20ACB04071	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Development of Non-ICT Company Procurement and Shipment Portal using Referenced Mobile Application	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

A new request was added to handle calls for orders with the date restriction. The month picker was added to filter the orders by month durations.

2. WORK TO BE DONE

Create a dashboard for monthly sales.

3. PROBLEMS ENCOUNTERED

Some bugs with the data retrieval and some data were filtered out which needs to be displayed in the test case.

4. SELF EVALUATION OF THE PROGRESS

Spending time on improving performance leads to delays in function development.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 8
Student Name & ID: Chin Pey Mian, 20ACB04071	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Development of Non-ICT Company Procurement and Shipment Portal using Referenced Mobile Application	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Sales data can be visualised through charts, including revenue, profit, and total order index for the current month.

2. WORK TO BE DONE

Develop the monthly sales dashboard that can view the previous month's report.
Develop functionality to track user activity.

3. PROBLEMS ENCOUNTERED

The order data used for the visualization is randomly generated with low accuracy and it is not well represented.

4. SELF EVALUATION OF THE PROGRESS

The development of the monthly sales dashboard turned out slightly differently than expected.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 10
Student Name & ID: Chin Pey Mian, 20ACB04071	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Development of Non-ICT Company Procurement and Shipment Portal using Referenced Mobile Application	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

A monthly sales dashboard can display previous records using a monthly picker.
Tracking of user activity on the portal is developed but not fully functional.

2. WORK TO BE DONE

Finish the extra functionalities in the system.
Fix some minor bugs found.

3. PROBLEMS ENCOUNTERED

Redundant code in the module.

4. SELF EVALUATION OF THE PROGRESS

Need to speed up the development progress.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 12
Student Name & ID: Chin Pey Mian, 20ACB04071	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Development of Non-ICT Company Procurement and Shipment Portal using Referenced Mobile Application	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Finalised the visual diagram in monthly sales dashboard.
Developed extra functions such as News module for create news, Chatbot for request some queries.

2. WORK TO BE DONE

Test the functionalities and fix the bugs.

3. PROBLEMS ENCOUNTERED

Some extra functions tried to develop but fail to meet the requirements.

4. SELF EVALUATION OF THE PROGRESS

Focus on time management.



Supervisor's signature



Student's signature

POSTER



Chin Pey Mian
Supervisor: Ts Dr Goh Hock Guan

Development of Non-ICT Company Procurement and Shipment Portal using Referenced Mobile Application

INTRODUCTION

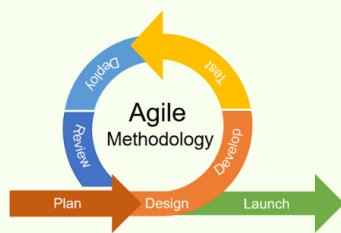
The purpose of project is to develop a web portal for salesperson and manager to support their business operations. The data on the portal should be consistent with the referenced mobile application. The company can adopt the procurement and shipment module to perform the order workflow and reduce the dependency of traditional business model.

OBJECTIVES

- Develop a sales and management role-based portal.
- Develop an integrated order and shipment module.
- Ensure data consistency and accuracy between the mobile application and web portal.

METHODS

Agile Methodology with Kanban Framework



Technologies Involved



- Developed by React.js on Visual Studio Code
- Express.js and Node.js as a backend server connect to MongoDB
- Data stored in MongoDB

RESULTS

The order and shipment module provides a seamless process that creates a linkage with customers to perform the tasks. The data is consistency between web portal and mobile application.

CONCLUSION

The procurement and shipment module implemented in the system to overcome the situation of the company. The other modules developed for the user increase the effectiveness of the system.

PLAGIARISM CHECK RESULT

DEVELOPMENT OF NON-ICT COMPANY PROCUREMENT AND SHIPMENT PORTAL USING REFERENCED MOBILE APPLICATION

ORIGINALITY REPORT

9%	7%	1%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
3	Submitted to Heriot-Watt University Student Paper	1%
4	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
5	www.toylabs.eu Internet Source	<1%
6	Submitted to University of Greenwich Student Paper	<1%
7	Saumyamani Bhardwaz, Rohan Godha. "Svelte.js: The Most Loved Framework Today", 2023 2nd International Conference for Innovation in Technology (INOCON), 2023 Publication	<1%

PLAGIARISM CHECK RESULT

8	dspace.lib.ntua.gr Internet Source	<1 %
9	Submitted to HELP UNIVERSITY Student Paper	<1 %
10	appdividend.com Internet Source	<1 %
11	cit.mak.ac.ug Internet Source	<1 %
12	Submitted to University of Hull Student Paper	<1 %
13	Submitted to Washington University of Science and Technology Student Paper	<1 %
14	flatlogic.com Internet Source	<1 %
15	Submitted to i-CATS University College Student Paper	<1 %
16	Submitted to Taylor's Education Group Student Paper	<1 %
17	docplayer.net Internet Source	<1 %
18	Submitted to Excelsior University Student Paper	<1 %
19	Submitted to University of Westminster	

PLAGIARISM CHECK RESULT

	Student Paper	<1 %
20	fict.utar.edu.my Internet Source	<1 %
21	Submitted to Manchester Metropolitan University Student Paper	<1 %
22	geografi.unima.ac.id Internet Source	<1 %
23	Submitted to University of Keele Student Paper	<1 %
24	human-ist.unifr.ch Internet Source	<1 %
25	tuttoaster.com Internet Source	<1 %
26	docs9.chomikuj.pl Internet Source	<1 %
27	Submitted to Informatics Education Limited Student Paper	<1 %
28	Submitted to Ngee Ann Polytechnic Student Paper	<1 %
29	dissertations.umi.com Internet Source	<1 %
30	eprints.ucm.es Internet Source	<1 %

PLAGIARISM CHECK RESULT

		<1 %
31	opensource.com Internet Source	<1 %
32	Pro Node js for Developers, 2013. Publication	<1 %
33	Submitted to South Gloucestershire and Stroud College Student Paper	<1 %
34	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
35	www.vingle.net Internet Source	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 8 words



Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1
Full Name(s) of Candidate(s)	CHIN PEY MIAN		
ID Number(s)	20ACB04071		
Programme / Course	Bachelor of Information Systems (Honours) Business Information Systems		
Title of Final Year Project	Development of Non-ICT Company Procurement and Shipment Portal using Referenced Mobile Application		

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u>9</u> % Similarity by source Internet Sources: <u>7</u> % Publications: <u>1</u> % Student Papers: <u>6</u> %	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Goh Hock Guan

Date: 24/4/2023

Signature of Co-Supervisor

Name: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB04071
Student Name	CHIN PEY MIAN
Supervisor Name	TS DR GOH HOCK GUAN

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 24 APRIL 2023