INVENTORY TRACKER WITH ESTIMATION

BY

BEH WEI JUN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: __Inventory Tracker with Estimation_____

_____

_____

Academic Session: _June 2024_

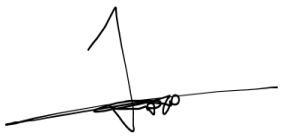
I __BEH WEI JUN_____

(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:
_3, Jalan Pulai 35,_____
_Taman Pulai Utama, _____
_81300 Skudai, Johor_____

Tan Chiang Kang
Supervisor's name

Date: _13 September 2024____

Date: 13/9/2024_____

FACULTY/INSTITUTE* OF _INFORMATION AND COMMUNICATION TECHNOLOGY___

UNIVERSITI TUNKU ABDUL RAHMAN

Date: __13 September 2024____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***Beh Wei Jun*** (ID No: ***20ACB05218***) has completed this final year project/ dissertation/ thesis* entitled "__Inventory Tracker with Estimation____" under the supervision of __Mr. Tan Chiang Kang__ (Supervisor) from the Department of ___Computer Science_____, Faculty/Institute* of __Information and Communication Technology_ , and _____ (Co-Supervisor)* from the Department of _____, Faculty/Institute* of _____.

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,

(*Beh Wei Jun*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**INVENTORY TRACKER WITH ESTIMATION**" 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 : Beh Wei Jun

Date : 13 September 2024

ACKNOWLEDGEMENTS

I sincerely thank my supervisor Mr. Tan Chiang Kang for giving me the valuable opportunity to embark on my journey of developing an estimated inventory tracker. With his guidance and mentorship, I took the first steps towards building a career in inventory management. His expertise, encouragement and unwavering support were instrumental in shaping this project.

Finally, I sincerely thank my parents and family for their unwavering love, support, and encouragement throughout my endeavours. Their belief in my abilities and constant encouragement have been my motivation to pursue excellence.

ABSTRACT

In today's rapidly evolving business landscape, small businesses must embrace digital transformation to remain competitive. Inventory management is a critical yet often overlooked aspect, and many small businesses still rely on traditional pen-and-paper inventory management. This challenges companies to maintain accurate and up-to-date inventory levels, reducing efficiency and accuracy. To address this challenge, the "Inventory Tracking and Estimating" project proposes the development of a user-friendly mobile application designed to simplify inventory management for small businesses. The app leverages the Flutter framework for cross-platform mobile app development, providing a solution that can be accessed through smartphones. By replacing traditional pen and paper methods, the app digitizes the inventory management process, increasing accuracy and efficiency. Key features include inventory tracking, estimation algorithms based on historical data, and integration with communication tools like WhatsApp for seamless order management. The project uses a structured approach that includes planning, analysis, design, implementation and maintenance phases to ensure the application meets the specific needs of the small business. Since its development, the app has been thoroughly tested and has successfully demonstrated its core functionalities, including user authentication, transaction management, and inventory operations. The testing confirmed that the system is both scalable and reliable, providing a robust solution for small businesses with limited resources. Overall, the project offers a comprehensive solution that modernizes inventory management for small businesses, empowering them to operate more efficiently and adapt to the competitive market through digital innovation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

In today's rapidly evolving business environment, transforming to digital is no longer just an option, but a necessity. It has become an important strategy for survival and development. Small businesses often constrained by limited resources and manpower, are actively seeking innovative tools to streamline operations and competitive. One critical aspect that modernization has mainly gone untouched in many small businesses is inventory management. Traditionally relying on manual tracking methods such as pen and paper, these businesses face many challenges in maintaining accurate and up-to-date inventory levels, such as time costs and inefficiencies. The inherent inefficiencies of manual tracking not only consume valuable time and resources, but also introduce the risk of errors and inaccuracies, reducing operational efficiency and hindering growth prospects. Therefore, recognizing the urgent need for this modernization, the "Inventory Tracking with Estimation" project aims to develop a user-friendly and efficient mobile application to address these challenges.

The application focuses on providing small businesses with powerful tools that allow them to digitize the inventory management process. By leveraging the capabilities of cross-platform mobile app development, specifically using the Flutter framework, create a versatile solution that can be easily accessed and updated from your phone. This approach is especially important for small business owners, as they are often closely involved in day-to-day operations, including inventory management. The application will replace traditional paper-and-pencil inventory management, improve inventory recording and tracking, and revolutionize the management of inventory levels. The app will apply algorithms [1] and historical data to estimate inventory levels to reduce the frequency of manual inventory checks and increase accuracy. Not only that, recognizing that many small businesses commonly rely on manual ordering processes, the app will leverage modern communication tools such as WhatsApp and email to facilitate seamless order management. Since many small businesses still rely on manual orders, integrating this functionality will bridge the gap between the digital and manual realms, making the transition smoother and easier. By digitizing and automating inventory tracking and estimating, we aim to provide these businesses with the tools they need to thrive in an increasingly competitive environment.

By digitizing and automating the inventory tracking and estimating process, the Inventory Tracking & Estimating project strives to provide small businesses with the essential tools they need to thrive in today's competitive environment. Beyond simply relieving the burden of manual inventory management, the initiative is designed to improve operational efficiency and unlock new growth opportunities for small businesses. By embracing digital transformation, we aim to not only increase the resilience and agility of small businesses, but also to foster a culture of innovation and adaptability that is critical to long-term success in the modern business environment.

1.1 Problem Statement and Motivation

In the modern business environment, many small businesses still track inventory manually, relying on pen and paper to manage the raw materials and goods purchased from suppliers. This leads to challenges for companies in maintaining accurate and up-to-date inventory levels, reducing efficiency and accuracy. This is because manual tracking methods often result in discrepancies between recorded and actual inventory levels, and inaccurate inventory levels can lead to stockouts, missed production schedules, and lost business opportunities. Not only that, but the lack of real-time visibility into inventory levels and the need for frequent manual inventory checks will hinder a business's productivity and growth potential. For example, periodic manual inventory checks take valuable time and effort from companies, diverting resources from core business activities. Although various inventory management applications exist in the market, they all have some limitations, such as high subscription costs, or are not suitable for all types of businesses. This is unaffordable for small businesses with tight financial constraints. Therefore, there is an urgent need for a free cross-platform mobile app that simplifies inventory tracking and integrates estimating capabilities. The application effectively reduces the need for manual inventory checks and increases the accuracy of the decision-making process.

The project was motivated by the desire to provide small businesses with a modern and efficient solution tailored to their specific inventory management needs. Due to various factors, many small businesses are still using outdated manual inventory tracking methods. Resource constraints, outdated inventory management practices, and a lack of customized digital solutions often constrain these businesses. Therefore, by providing a user-friendly interface and comprehensive functionality, our application is designed to facilitate digital transformation in the small business sector. The application is available cross-platform and combines

inventory tracking, estimating capabilities and streamlined order management, freeing up time and resources for business owners and employees to focus on core value-added activities. Our goal is to provide small businesses with the tools they need to overcome the limitations of outdated methods and improve their operational efficiency and competitive position in today's evolving business environment.

1.2 Objectives

The main objective of this project is to conceptualize, design, and implement a specifically customized cross-platform mobile application to streamline inventory management processes for small business. By providing a digital alternative to traditional paper-and pencil methods, the project aims to increase the efficiency and accuracy of inventory management operations, ultimately enabling small businesses to operate more efficiently in today's competitive market environment. The project focuses on integration of inventory estimation algorithms that leverage historical data and user-defined parameters to provide real-time inventory levels without the need for labor-intensive manual checks. By providing real-time inventory data and transaction history, the mobile app is designed to provide small business owners with the actionable insights they need to make informed decisions and optimize their inventory management strategies.

In addition to promoting efficient inventory tracking, the project is also designed to facilitate seamless communication and collaboration between business owners and suppliers. By integrating communications tools directly into the application, such as messaging capabilities for quick connections between business owners, employees and suppliers, the project aims to streamline the order placement process and enhancement overall operational agility. This integration not only facilitates timely discussion of inventory-related orders but also enables quick decision-making based on real-time inventory levels.

In summary, the project's primary objective is to develop a comprehensive mobile application that not only simplifies inventory management, but also enhances communication and collaboration within small businesses. By leveraging the power of technology to automate the inventory estimating process and facilitate real-time data access, the app helps assist small businesses complete their digital transformation.

1.3 Project Scope and Direction

The scope of the project included the development of a comprehensive cross-platform mobile application specifically tailored to meet the inventory management needs of small businesses. Upon completion of the project, a fully functional software solution will be delivered including a user-friendly interface, inventory entry functionality and automated inventory estimating algorithms. The solution is designed to streamline the inventory tracking process by providing accurate real-time synchronization across multiple devices, ensuring business owners and authorized personnel have access to the most up-to-date inventory information no matter where they are. Additionally, the app will be equipped with efficient communication tools such as integration with messaging apps like WhatsApp, facilitating seamless communication between business owners, workers and suppliers during order placement and inventory-related discussions.

Additionally, through a role-based access system, the application will protect private user information, inventory data and communication records, ensuring only authorized personnel can access and manage inventory levels and place orders. By adhering to cross-platform compatibility standards, the app will run seamlessly on both iOS and Android operating systems, ensuring widespread use by small businesses looking to digitize their inventory management processes. Upon successful completion, the project aims to provide a comprehensive solution that facilitates small businesses' digital transformation journey and improves the efficiency, accuracy, and operational effectiveness of inventory management.

1.4 Contributions

The project is expected to make several significant contributions to the fields of small business, inventory management, and mobile application development:

Real-time Estimating: The integration of estimating capabilities can reduce the need for frequent manual inventory checks [1]. This gives businesses real-time visibility into inventory levels, allowing them to make better decisions and optimize inventory management.

Efficient Inventory Tracking: This project introduces a streamlined inventory tracking method that more than replaces outdated inventory tracking methods. This contribution can improve the accuracy of inventory records, reduce human errors, and increase operational efficiency.

Enhanced decision-making: By providing businesses with up-to-date and accurate inventory information and estimates, this project helps improve the decision-making process.
Efficient order management: Integrating communication tools such as WhatsApp simplifies order placement, solving the manual ordering process still prevalent in small businesses and facilitating faster, smoother transactions.
Remote Accessibility: Workers can update inventory data through an app, enabling owners to stay informed remotely, facilitating collaboration, and reducing reliance on physical presence for decision-making.
Resource Efficiency: Reducing manual inventory checking helps save time and labor allocation resources, which can be repurposed into core business activities.

In conclusion, the contributions of the project range from championing digitalization to promoting growth in small businesses. It provides a holistic solution that drives these businesses to improve efficiency, make informed decisions, and achieve sustainable development in the modern business environment.

1.5 Report Organization

This report is organized into seven chapters, each addressing different aspects of the project. Chapter 1 introduces the project by outlining the problem statement, the motivation behind the development, the objectives, scope, and contributions of the inventory management system. This sets the context for the entire report. Chapter 2 presents a literature review, examining existing inventory management systems, including Sortly, BoxHero, Naye Inventory, the system used by Olahan Hasrat Sdn. Bhd., and a mobile-based system with QR code integration. This chapter also compares these systems with the proposed solution, highlighting the limitations of previous studies and justifying the need for a new approach.

Chapter 3 discusses the system methodology and approach, covering the system architecture, use case diagrams, and activity diagrams. These diagrams provide a comprehensive understanding of the system's functionality and flow. Chapter 4 focuses on system design, detailing the block diagrams for the admin and user modules, as well as flowcharts for each user type. The chapter also includes the entity relationship diagram (ERD) to describe the database structure and relationships between entities.

Chapter 5 provides an in-depth discussion of the system implementation, covering both hardware and software setup, configuration of the development environment, project settings, and mobile device configuration. It also addresses challenges encountered during the implementation phase. Chapter 6 evaluates the system through testing and performance metrics, presenting the results of the testing process, discussing challenges faced, and assessing how well the project met its objectives.

Finally, Chapter 7 concludes the report by summarizing the project's achievements and challenges. It also provides recommendations for future work, offering suggestions for improving the system and expanding its functionality. This organization ensures a logical flow of information from the initial problem identification to the final conclusions and recommendations.

Chapter 2

Literature Review

2.1 Previous Works on Inventory Management System

2.1.1 Study of the existing system: Sortly mobile application

Sortly is a multi-platform mobile program that is available for download from the Apple App Store and Google Play Store [2]. Sortly simplifies the process for organizations to organize and manage inventory, items, or assets and provides user-friendly solutions to track stocks effectively. The app allows users to create a digital catalogue of items with details, photos, total value and categories. Users can quickly add items by scanning barcodes or manually entering information, and users can also add pictures to each item to identify things quickly and visually in inventory or storage. The application also supports categories and tags. Users can classify and tag items to quickly obtain detailed stock information when searching.

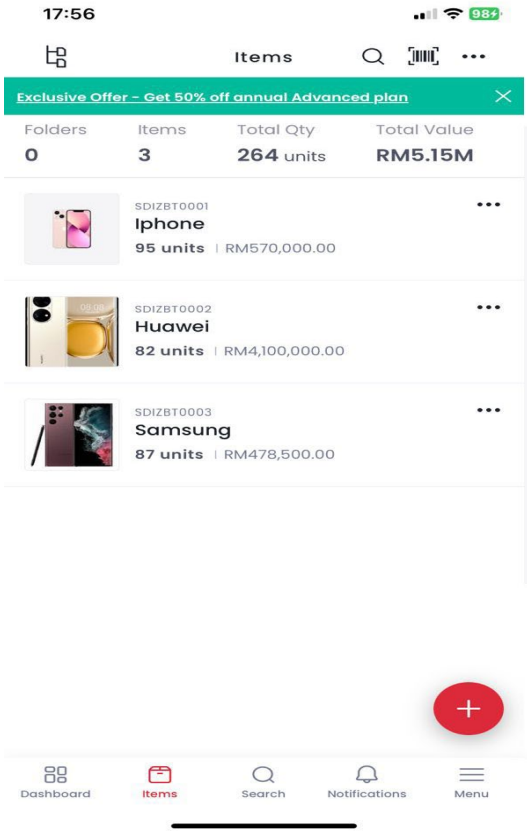


Figure 2.1 Item interface

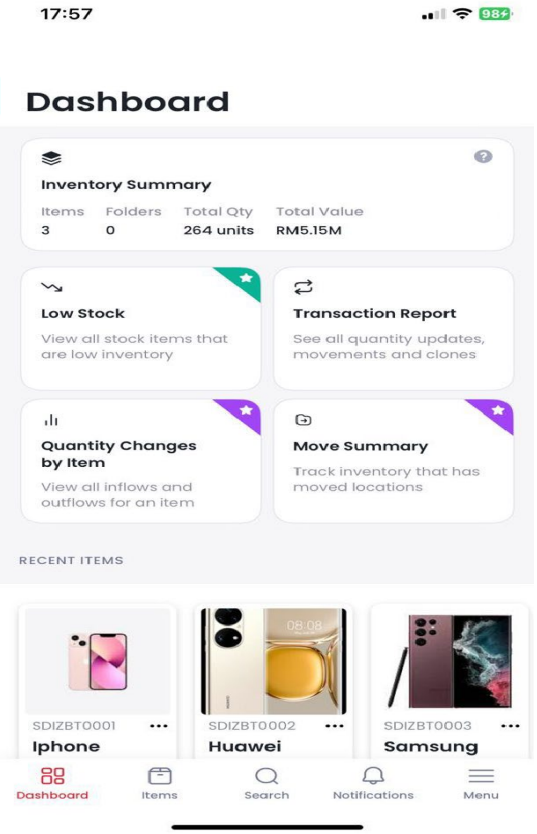


Figure 2.2 Dashboard interface

Also, Sortly supports QR code and barcode scanning, which can generate exclusive QR codes and barcodes for various items, allowing users to link to items in inventory quickly and

improve tracking efficiency. The application also provides cloud synchronization, so users can use it across multiple devices without worrying about information confusion. Users can access and track inventory levels anytime and anywhere if connected to the Internet. Sortly also supports a reminder function. Users can set a minimum quantity for various items. When the inventory level reaches the minimum quantity of the item, the user will be reminded through a message to replenish the stock. In addition, Sortly allows users to generate some basic reports, allowing users to understand inventory data better and help companies make decisions.

2.1.2 Study of the existing system: BoxHero mobile application

BoxHero is a mobile app for managing and tracking inventory tasks [3]. The app is available for Android and iPhone Operating System (IOS) and can monitor or track inventory levels from a desktop or laptop computer. It is an integrated inventory management software that provides the basic functions needed by enterprises, especially small and medium-sized enterprises (SMEs), to facilitate enterprises to manage inventory. As shown in Figure 2.4, the application supports manual input to add items, can generate exclusive barcodes for items, and can set the minimum quantity, cost price and selling price of items to facilitate enterprises to view the net price of items. BoxHero also supports the function of item check-in/check-out. Enterprises can know which supplier or customer provided or purchased the product, making inventory management and tracking more convenient and clearer.

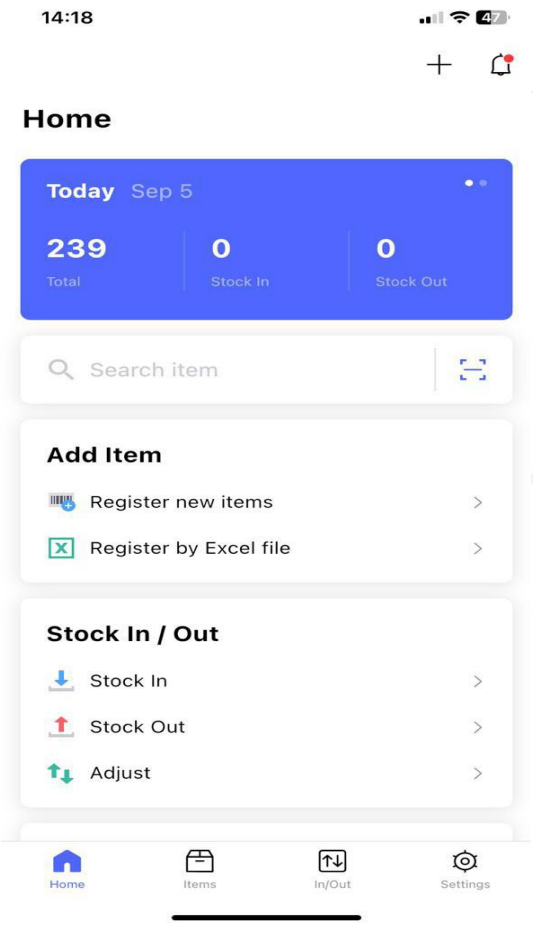


Figure 2.3 Home page

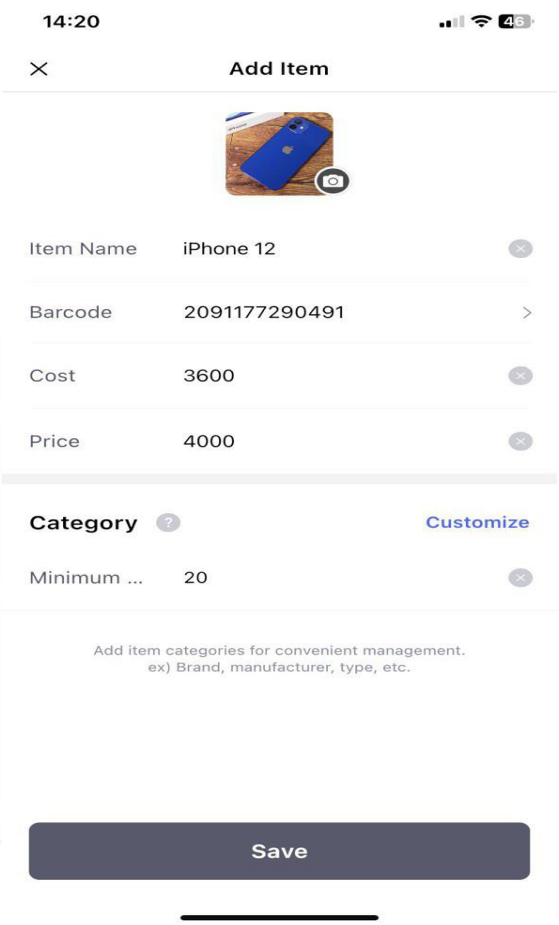


Figure 2.4 Add item

In addition, the application supports adding items through Excel files, which will be automatically organized and added to the application for the user, which will greatly reduce the time required to add items and improve efficiency. BoxHero also allows users to export items to Excel files, which is convenient for users to obtain item information. The application also supports analysis and reporting functions, which will generate a report and summary of the past month's transaction records and net income to facilitate business analysis.

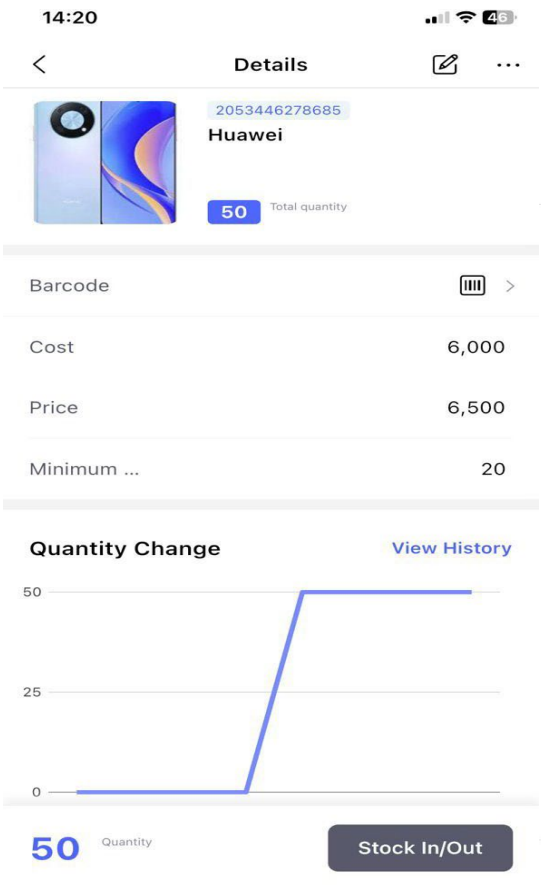


Figure 2.5 Product details

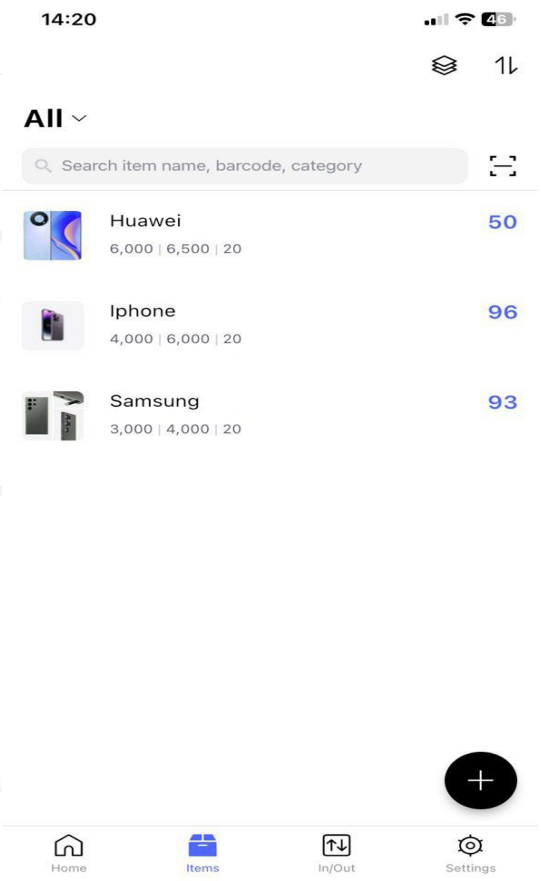


Figure 2.6 Items interface

2.1.3 Study of the existing system: Naye Inventory mobile application

Naye Inventory is a multi-platform mobile application designed for efficient inventory management. The application, available on Android and iOS, is designed to simplify tracking and managing inventory for businesses and organizations, reducing manual workload and improving accuracy [4]. Naye Inventory provides certain guarantees for corporate data security. The application provides user authentication for multiple roles, such as employees and administrators. Administrators can control user access to protect sensitive company information. Employees can log in only with their organization code, and their actions are logged to ensure the company is not vulnerable to malicious attacks. Employees will perform basic inventory management tasks such as adding, updating, and deleting items.

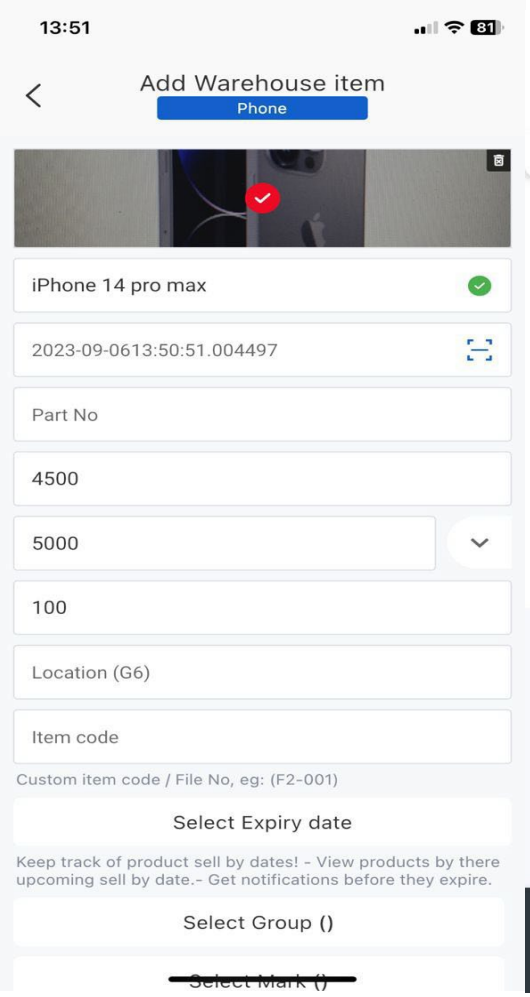


Figure 2.7 Add warehouse item

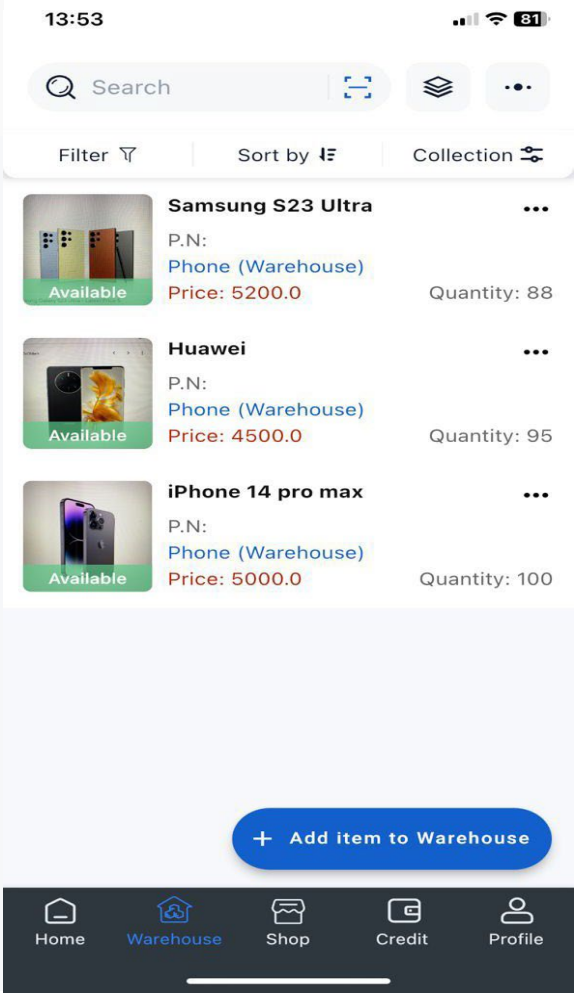


Figure 2.8 Warehouse interface.

In addition, users can enter detailed information about the product, such as name, barcode, quantity, description, and images, for easy search. Naye inventory also provides a quick search function. Figure 2.9 shows the search engine and the result of the search function. Users can quickly search through the built-in barcode and QR code scanner to obtain detailed information on inventory items, improving efficiency and accuracy. The application also maintains a comprehensive record of all inventory-related activities to provide transparency and a historical view of user interactions with the system. This includes tracking item additions, updates, deletions, and shipping status changes. Additionally, the application provides various reports to help businesses with analysis and decision-making. For example, warehouse report, shop report, business report, credit report, and product report. Naye Inventory is regularly maintained and updated to fix bugs, add new features, enhance security, and ensure compatibility with evolving mobile operating systems.

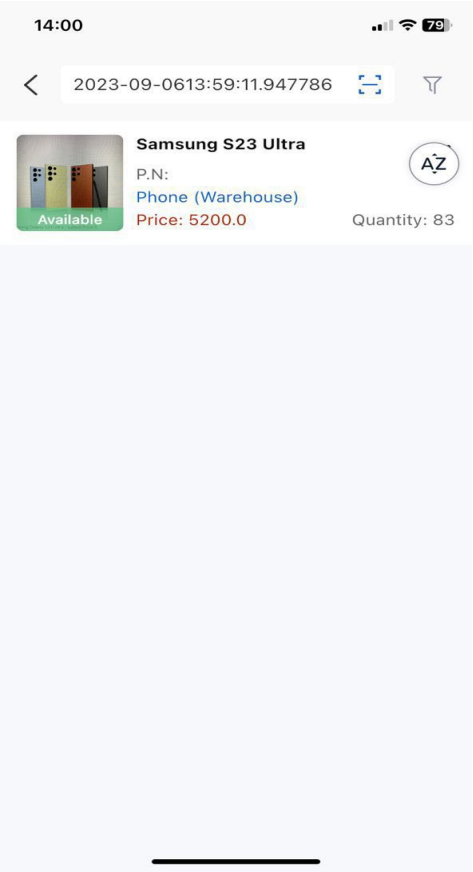


Figure 2.9 Search engine

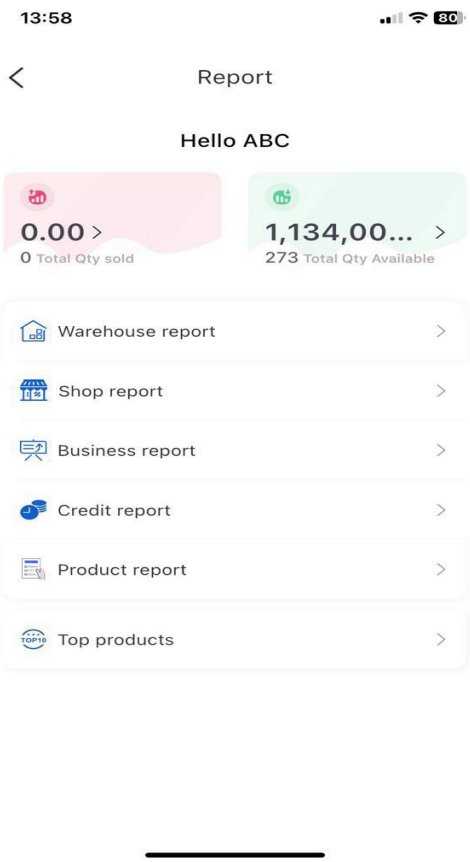
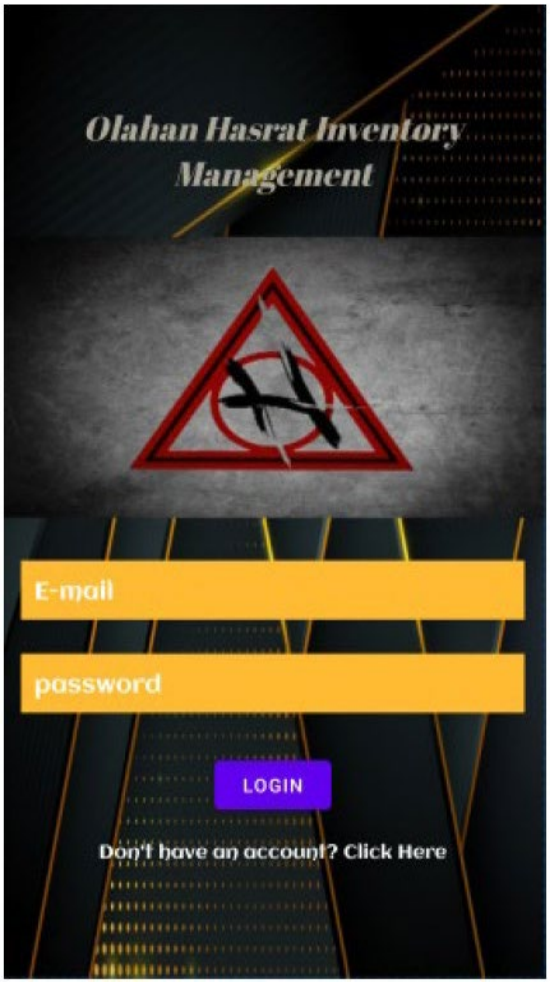


Figure 2.10 Report interface

2.1.4 Study of the existing system: Inventory management system of Olahan Hasrat Sdn. Bhd.

The mobile application is an inventory management system (IMS) specially developed for Olahan Hasrat Sdn. Bhd [5]. It is a software application developed exclusively for Android mobile devices, built using the Java programming language and Firebase as the database platform. The app is designed to help small and medium enterprises (SMEs) such as Olahan Hasrat Sdn. Bhd with efficient inventory management and financial tracking. First, the app offers a user-friendly interface and a range of features that allow users to easily manage inventory by quickly adding, removing and viewing items through barcode or quick response (QR) code scanning. The app also allows users to record various expenses related to their business operations, such as transportation, tools, and rent. This feature will help users monitor and control their spending to avoid overspending.



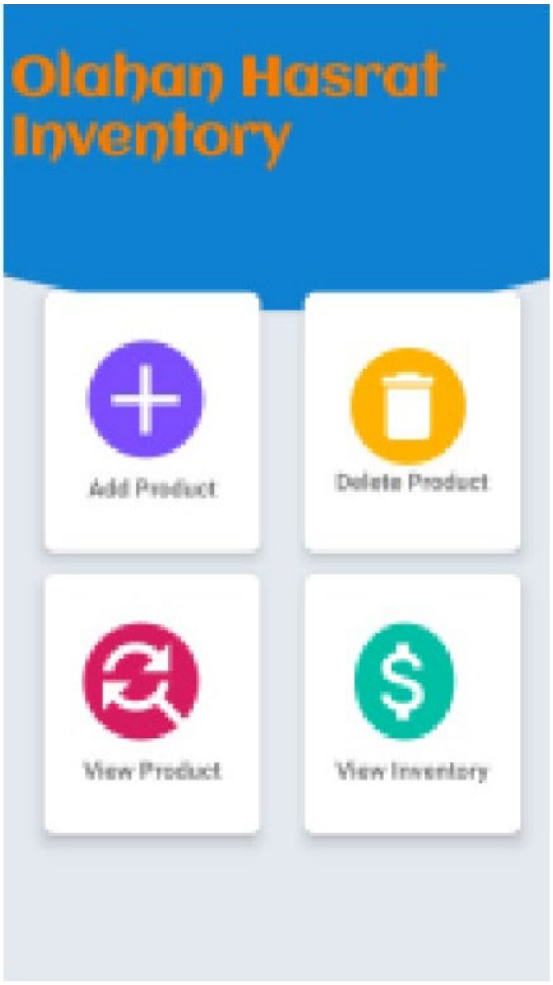


Figure 2.11 Login interface **Figure 2.12 Home page**

Not only that, but the most special feature of this app is the price comparison feature. Users can quickly compare the prices of specific products from different suppliers through online shopping platforms such as Shopee and Lazada. This feature helps companies select cost-effective suppliers and reduce unnecessary expenses. Additionally, the app can generate a report based on previous expenses, providing insights into a company's financial performance. Users can see a breakdown of expenses through pie charts, which helps them make informed decisions and predict future inventory needs. Overall, the inventory management system of Olahan Hasrat Sdn. Bhd significantly improves inventory management, expense tracking, and supplier price comparison for SMEs. It provides a user-friendly mobile application to streamline daily operations, improve inventory management efficiency, and support better decision-making.

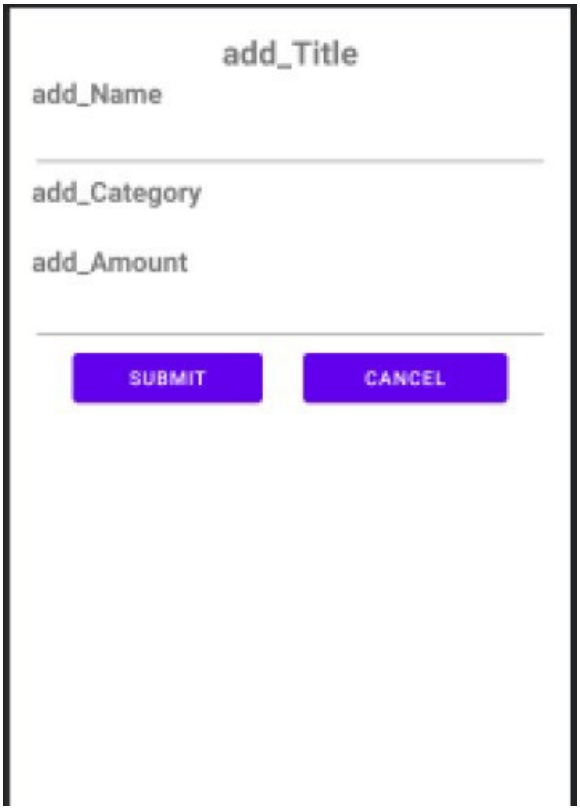


Figure 2.13 Add item

2.1.5 Study of the existing system: Mobile-based Inventory Management System with QR code

The paper [6] proposes an inventory management system suitable for organizations with warehouse and inventory management needs, such as retail businesses, manufacturing companies, or logistics providers. The system is developed using React-Native and Firebase for authentication, real-time data updates, real-time database functionality and cloud storage. The main purpose of this inventory management system is to manage inventory effectively, track shipments and maintain detailed records of inventory-related activities by scanning QR codes. It is designed to solve challenges faced in traditional inventory management, such as manual data entry errors and the need for real-time tracking. First, the system uses two types of login access, one for employees and another for administrators or managers. Figure 2.14 shows an example of the login interface. Employees use organization codes to log in and manage inventory items, such as adding, updating, and deleting items. Administrators can control user access, providing security and data isolation to prevent employees from gaining access to sensitive corporate information.

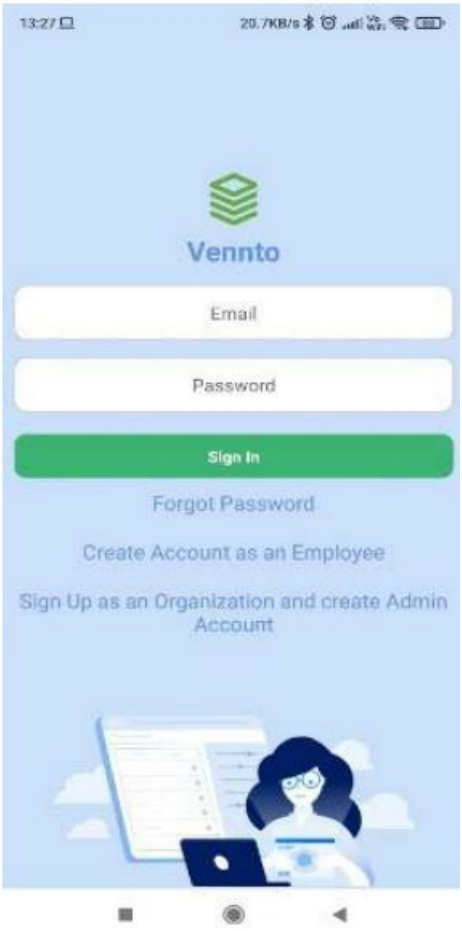


Figure 2.14 Login interface **Figure 2.15 Home page**

A distinctive feature of the system is using QR codes to manage, add, and track items quickly. QR codes simplify data entry, reduce the chance of errors, and are suitable for use by employees of all skill levels. Employees can scan QR codes for fast and accurate data entry to obtain item details. The system can also track shipment status from "Ready to Ship" to "Processed." This feature helps organizations monitor the flow of goods within the supply chain. Since the application is developed using Firebase, the system can handle many requests and data, and there is no need to consider scalability issues in the future. In addition to this, the system provides transparency and accountability. The system will keep records of all activities related to inventory management, such as tracking changes, additions, updates, and deletions made to inventory items for the convenience of businesses to manage and Track issues. In conclusion, the paper's proposed mobile device-based inventory management system with QR codes is a promising solution for organizations looking to modernize and automate their inventory management processes. It addresses key challenges in the field and leverages mobile technology and QR codes to increase efficiency and accuracy.

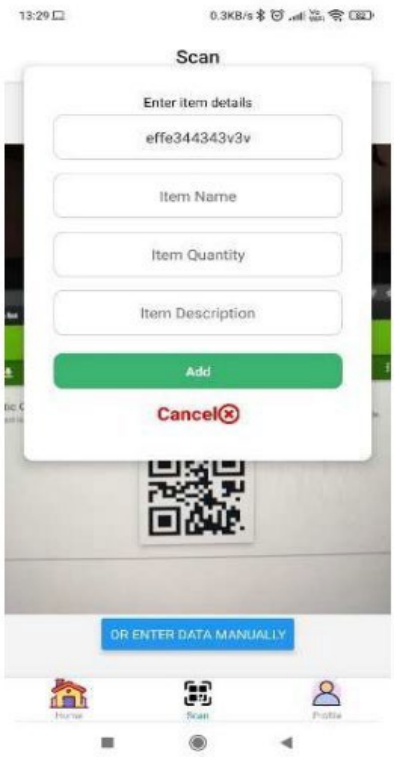


Figure 2.16 Scan QR code

2.2 Comparison with Existing System

Features	Sortly Mobile Application	BoxHero Mobile Application	Naye Inventory Mobile Application	Inventory Management System of Olahan Hasrat Sdn. Bhd.	Mobile-based Inventory Management System with QR code
Log in/Log out	√ (User ID and password)	√ (User ID and password)	√ (User ID and password)	√ (User ID and password)	√ (User ID and password)
Inventory tracking	√	√	√	√	√
Support various unit of measurement (kg, pcs, gram)	X	X	X	X	X
Expense tracker	X	√	√	√	√
Barcode and QR code	√ Need Subscription	√ Need Subscription	√	X	√

Search engine	√	√	√	√	X
Low stock alert	√ Need Subscription	√ Need Subscription	√ Need Subscription	X	X
Project report	√ Need Subscription	√ Need Subscription	√	√	X
Prediction Inventory level	X	X	X	X	X
Integration with Communication tool	√ Need Subscription	√ Need Subscription	√ Need Subscription	X	X

Table 2.1 Comparison of Existing System

2.3 Limitation of Previous Studies

While Sortly [2] is a great app, its limitations make it unsuitable for small businesses and help them efficiently organize and manage their inventory. First, Sortly has a very high subscription cost, and many functions require a subscription to be able to use, such as reminder functions, categories, and label functions. The free version of Sortly has a lot of limitations, such as only being able to add 100 items, allowing one employee to help users manage inventory, and being able to customize one unique information for the items. Most importantly, the free version does not support item check-in/check-out. This is fatal for managing inventory, making it impossible for users to know which supplier or customer provided or purchased the product, confusing managing inventory.

Beyond that, the free version of Sortly only offers very limited integrations, and integrating with other software or platforms can be difficult. For example, this proposal project may need to be linked to WhatsApp, which results in the enterprise needing to spend more than $59 a month to subscribe to the Enterprise plan package to be able to link. The above are the reasons and limitations why Sortly may not be suitable for proposal projects. The application has many functions but requires a monthly subscription to the application for normal use. The free version will not be able to use the functions required for inventory management fully. For example, linking WhatsApp or other communication tools to simplify ordering requires subscribing to the Enterprise plan package. This requires users to pay at least $59 monthly to manage inventory, which is unaffordable for small businesses strapped for cash. Figure 2.17 shows the subscription price of the Sortly application and the usage limit of each version.

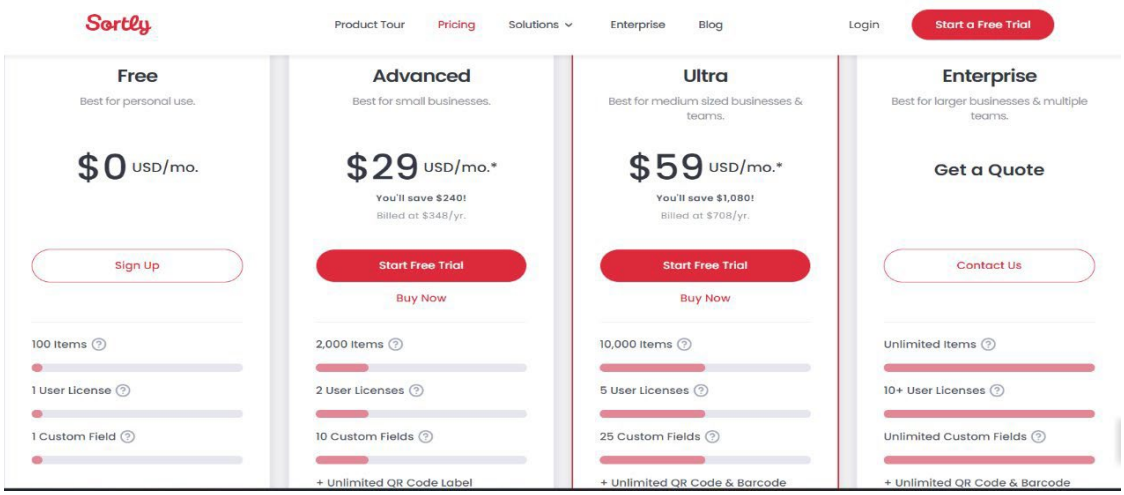
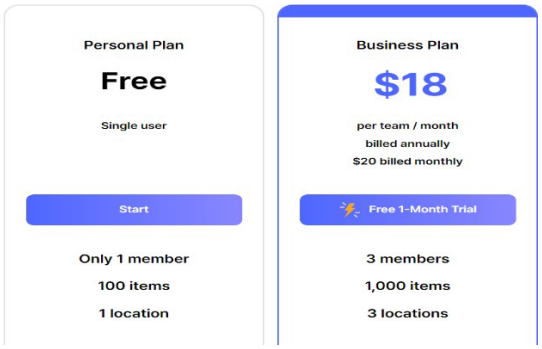


Figure 2.17 Subscription price and usage limits of Sortly

Furthermore, BoxHero [3] and Naye Inventory [4] mobile applications also have some limitations that prevent them from helping the proposal project manage inventory very well. They also require a subscription to unlock features needed to manage inventory, such as print barcode labels, analysis and reports, low stock notifications, purchase and sales orders, and more. Although the subscription costs of BoxHero and Naye Inventory mobile applications are only $18 and RM69.90 per month compared to the Sortly mobile application, it may become a limitation for some businesses, especially if the budget is tight. In addition, their paid version also has some limitations. For example, BoxHero can only allow 3 members or employees to manage inventory together, supporting up to 1,000 items and 3 locations [3]. If the companies need to add it, they can only use it for an additional fee. Most importantly, the free version does not support the generation of QR codes and barcodes, which causes enterprises to add items only by manually entering information, which will greatly reduce management efficiency. Also, while they may offer integrations with other software and platforms, they may not support all the specific integrations that the project requires, for example, WhatsApp and Skype, to name a few.



Stock In/Out/Adjust	✓	✓
Excel Import/Export	✓	✓
History Access	30 days	Unlimited
Mobile App	✓	✓
Print Barcode Labels	-	✓
Analysis & Reports	-	✓
Low Stock Notifications	-	✓
Purchase & Sales Orders	-	✓

Figure 2.18 & Figure 2.19 Subscription price and usage limits of BoxHero

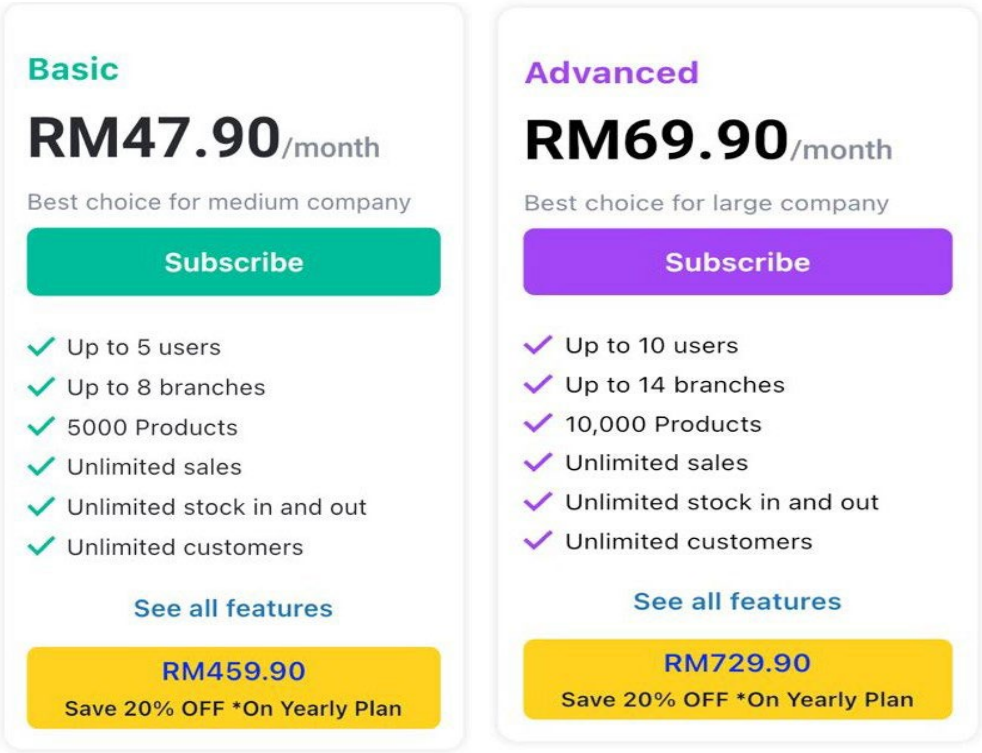


Figure 2.20 & Figure 2.21 Subscription price and usage limits of Naye Inventory

Although the Inventory Management System (IMS) in the paper [5] provides various benefits and features to help Olahan Hasrat Sdn. Bhd. in inventory management and financial tracking. However, the application was developed exclusively for Olahan Hasrat Sdn. Bhd. [5] has some limitations for other small and medium-sized businesses. First, the system's device compatibility will impose some limitations on other companies. The system is limited to Android devices and has no adaptations for iOS devices or other mobile platforms, forcing users to switch to Android devices and may incur additional costs for enterprises. Moreover, the application has a very high dependence on the Internet. For example, if the user is in an

area with an unreliable internet connection, the user will not be able to perform data synchronization or online shopping platform price comparison, preventing the user from using the application. On top of this, ongoing maintenance and updates are critical to keeping systems secure and up to date, which can incur additional costs.

Since the proposed mobile-based QR code inventory management system is still in the development and implementation stages [6], it also has some limitations and potential challenges. The QR code scanning function to add and manage items has certain requirements for the quality of the QR code. QR code quality issues may cause some limitations, and if the QR code is a little damaged or illegible, the scan may not work properly. Not only that, but QR codes can be vulnerable. Malicious actors may attempt to manipulate or replace QR codes, potentially resulting in inaccurate inventory data. The system focuses on inventory management and tracking, so it may lack some advanced analytics and reporting capabilities to aid business decision-making. Since this system is only proposed, it cannot be used on this project.

Also, the previous research has not covered the application of algorithms and historical data to accurately predict and estimate the latest inventory levels to reduce frequent inventory checks by employees. The previous research also did not support multi-unit measurement items. As a result, they may not be suitable for inventory management for all businesses. Furthermore, previous studies did not integrate the app with other communication tools for free to allow businesses to use the app to check inventory levels and place orders quickly through the communication tool. In conclusion, each mobile application has limitations preventing the project from using them for very effective inventory management and tracking, such as subscription costs, lack of integration, device compatibility issues, etc.

2.4 Proposed Solutions

The project aims to propose a cross-platform mobile application to solve problems related to system compatibility. The app is designed to simplify the inventory management process for small businesses. The app will provide all the features needed to manage inventory, such as barcode addition, low stock alerts, built-in barcode scanner, item stock-in and stock-out functionality, and much more, and the app will have no subscription fees and all functions will be open. Moreover, the application will incorporate additional functionalities to enhance usability and efficiency. It will support various units of measurement such as kilograms, grams, milliliters, liter, and pieces, accommodating diverse inventory needs. Additionally, the

inclusion of composed item functionality will allow users to manage complex inventory structures effortlessly. Additionally, the application will record item transactions, and use algorithms to accurately predict and estimate inventory levels based on historical data, allowing companies to better understand inventory trends and help guide decisions and strategies. In addition, the application will be integrated with communication tools such as WhatsApp and email, and users can quickly switch to WhatsApp to order through the application.

2.5 Comparison of existing applications with proposed solutions

Existing applications Sortly, BoxHero, and Naye Inventory all offered inventory capabilities, but were fraught with limitations that hindered their effectiveness for small businesses. These limitations include high subscription costs, limited integration with other software, and device compatibility issues. Additionally, these applications may lack basic functionality such as item stock-in/stock-out capabilities, multi-unit measurement support, and analytics capabilities.

In contrast, the proposed solution aims to overcome these limitations by providing a comprehensive cross-platform mobile application specifically tailored to meet the inventory management needs of small businesses. Unlike existing applications that require expensive subscriptions, the proposed solution will provide all functionality for free, ensuring that businesses on a tight budget can use it. Additionally, the app will offer advanced features such as adding barcodes for items, low-stock alerts, and automated inventory estimation algorithms based on historical data.

Furthermore, the proposed solution is unique in that it can be seamlessly integrated with communication tools such as WhatsApp, allowing for fast and efficient ordering and inventory-related discussions. Additionally, the application will support composed items, allowing businesses to manage complex inventory structures effortlessly. Overall, the proposed solution will address the limitations of existing applications, providing small businesses with comprehensive and efficient tools to streamline their operations and drive success.

Chapter 3
System Methodology/Approach

The project process is divided into SDLC phases: planning, analysis, design, implementation, and maintenance [7]. During the planning stages, the need for the project was identified, which was to develop a mobile app for small businesses to help them streamline the inventory management process. This will involve outlining the specific functionalities and features required for the application to meet the needs of small business. In the subsequent analysis phase, data was gathered by studying existing management inventory applications to form the project's specific requirements [8]. The goal of this process is to gather the strengths of existing systems, such as barcode scanning, low stock alerts, analytical reporting, etc., while addressing and improving existing application weaknesses, such as high subscription costs, lack of application algorithms, multiple units of measure for items and not integrated with communication tools. The process will gather the strengths and weaknesses of existing apps to develop an inventory management mobile app for small businesses.

The design phases will focus on outlining the system architecture and user interface design using Flutter and Dart programming languages. Firebase will be utilized for backend services, including authentication and data storage. The application’s design will prioritize simplicity and responsiveness to ensure a seamless user experience across both IOS and Android platforms. Next, implementation will involve the actual coding and development of the application according to the design specifications [9]. This phase will be carried out independently, with a focus on adhering to best coding practices and ensuring the application’s functionality meets the defined requirements. Finally, the maintenance phase will involve ongoing support and updates to address any user feedback or issues to enhance user experience and functionality.

3.1 System Design Diagram/Equation

3.1.1 System Architecture Diagram

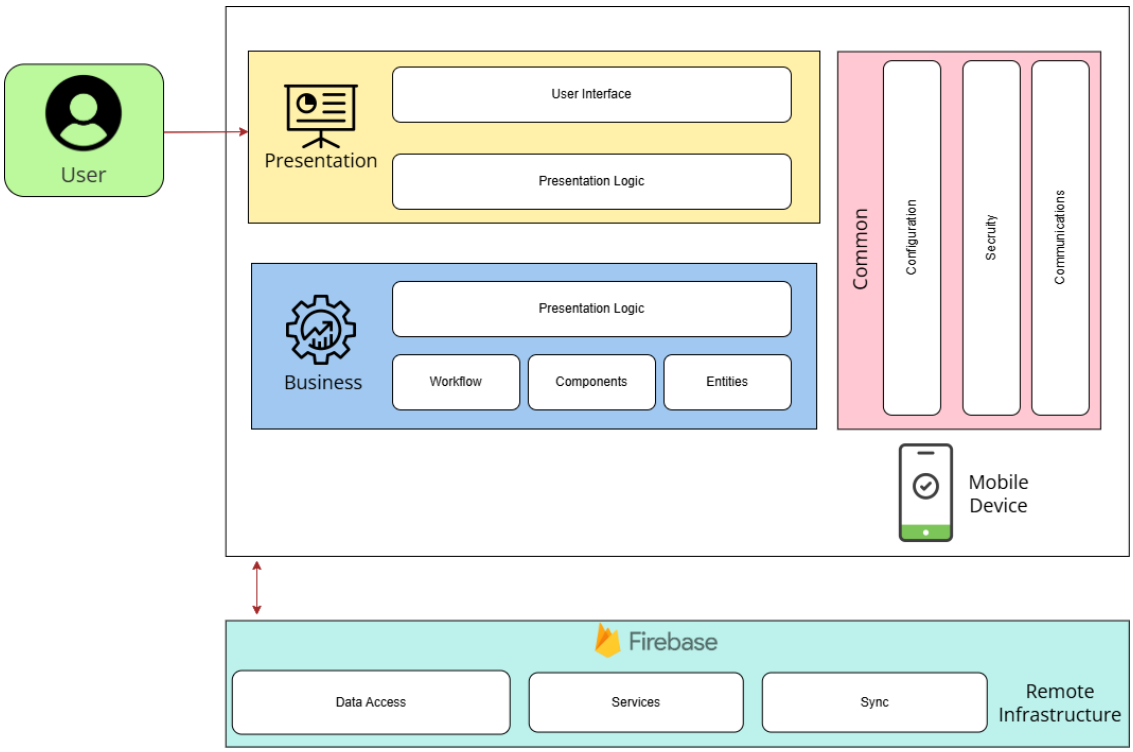


Figure 3.1 System Architecture Diagram

In this system architecture, mobile apps interact directly with Firebase as a backend service, without the need for local data storage. The architecture consists of two main components which are mobile and firebase remote infrastructure. The mobile application installed on the user's device acts as the interface through which the user interacts with the system. It includes a presentation logic, ensuring a seamless user experience. The business layer within the app manages the core application logic, including the workflows and components that interact with Firebase to handle business logic. Finally, common components for mobile devices, such as security, configuration, and communication tools, ensure that the app securely manages data and external communications such as WhatsApp integration, email, and notifications. On the other hand, Firebase acts as the sole backend service for the app, handling all data storage, synchronization, and backend processing. The data access components allow the mobile application to read and write data directly to Firebase's Firestore. Firebase's services include authentication and cloud storage, which manage user identity and authentication file storage, respectively. The synchronization components ensure real-time data synchronization between the app and Firebase, allowing the user's app to always have the latest information, such as inventory levels and transaction history. In conclusion, the system architecture is simplified, it is reducing complexity and maintenance requirements by relying on Firebase. Firebase

provides a scalable, real-time backend service that runs mobile apps without requiring a local database on the user's device.

3.1.2 Use Case Diagram and Description

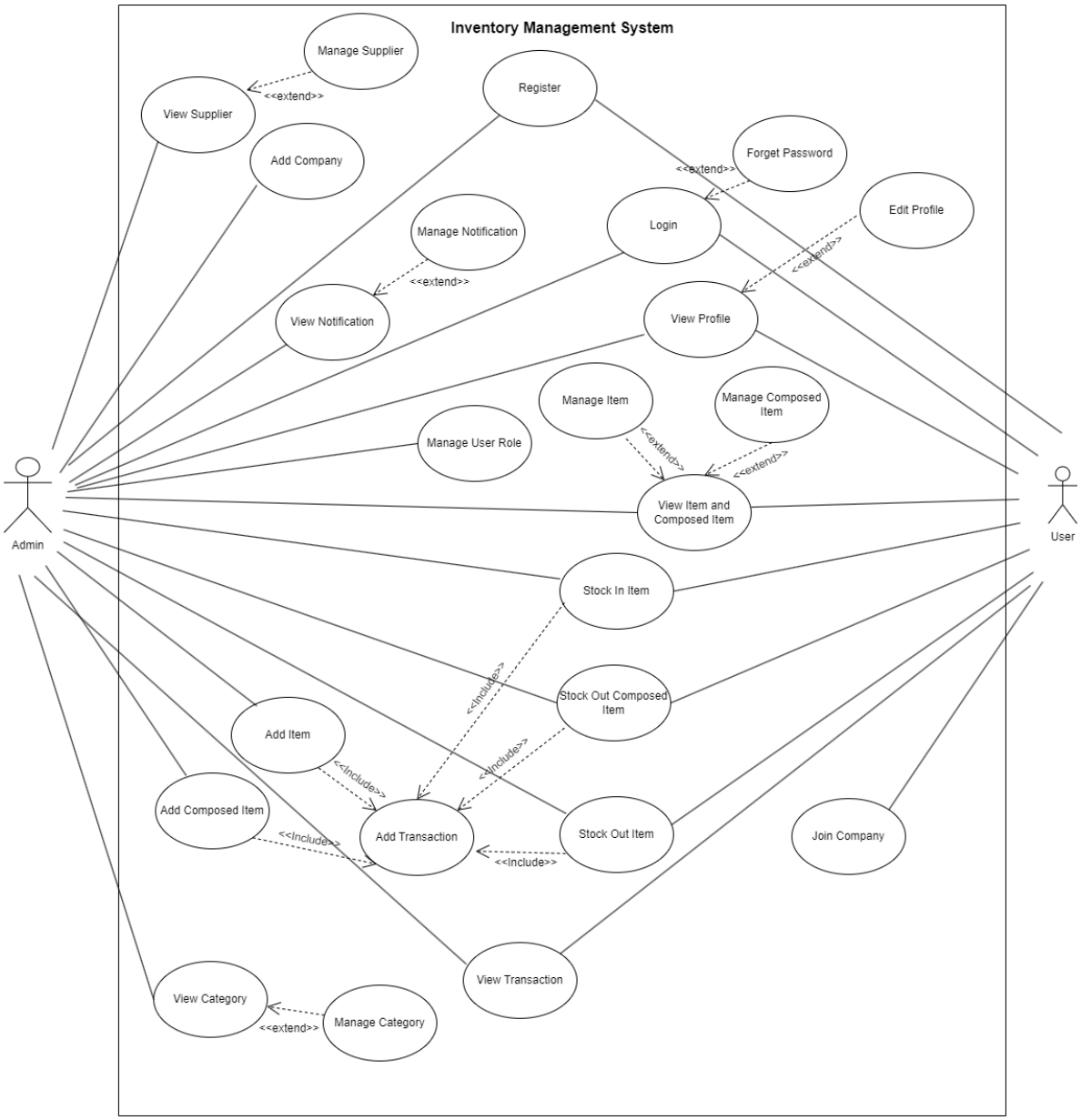


Figure 3.2 Use Case Diagram

Use Case ID	UC001	
Use Case	User Registration	
Purpose	To allow a new user to create an account in the application.	
Actor	The user (new or unregistered user) who wants to create an account in the system.	
Trigger	The user navigates to the registration page and initiates the registration process by clicking the “Sign up” button.	
Precondition	The user must access the registration page, which should be available and operational.	
Scenario Name	**Step**	**Action**
Main	1	Open the registration page.
	2	The system displays input fields for username, email, password, and confirm password.
	3	User key in required field (e.g. username, email, password, and confirm password).
	4	Click sign up button
	5	System validates the input data.
	6	System creates an account and stores it into database.
	7	System navigate user into selection company page.
Alternative Flow - Empty Input	3.1	User leaves empty for input fields.
	3.2	System display error message “Please fill in all fields!”
Alternative Flow - Invalid Email Format	3.1	User input invalid email format.
	3.2	System display error message “The email address is badly formatted.”
Alternative Flow - Different Password & Confirm Password	3.1	User input different password and confirm password.
	3.2	System display error message “Password don’t match!”
Alternative Flow - Registered Email	3.1	User input registered email.
	3.2	System display error message “Email is already registered.”
Postconditions	The user successfully creates an account and is redirected to the company selection page.	

	If an error occurs, the user is prompted to correct their input based on the alternative flows.
Rules	User does not have an existing account.

Table 3.1 User Registration

Use Case ID	UC002	
Use Case	Create a New Company	
Purpose	To allow a newly registered user to create a company	
Actor	User (registered user who has not yet joined or created a company)	
Trigger	The user selects the "Create a Company" option after completing registration.	
Precondition	The user must have successfully registered and be on the company selection page.	
Scenario Name	**Step**	**Action**
Main	1	Open the create company page.
	2	System requests for company name and notes (optional).
	3	User key in company name and any additional notes.
	4	Click create company button.
	5	The system checks the validation of company name (checks for duplicates) and assign Admin roles for the newly created company.
	6	System navigate user into Admin Home Page associated with their company.
Alternative Flow - Empty Input	3.1	User leaves empty for company name field.
	3.2	System display error message "Please input company name."
Alternative Flow - Registered Company Name	3.1	User input registered company name.
	3.2	System display error message "Company name already exists."
Postconditions	A new company is created, and the user is designated as the Admin. The user is directed to the Admin Home Page pf the newly created company.	
Rules	The user must not already belong to a company.	

Table 3.2 Create a New Company

Use Case ID	UC003	
Use Case	Join an Existing Company	
Purpose	To allow a newly registered user to join an existing company using a company join code.	
Actor	User (registered user who has not yet joined or created a company)	
Trigger	The user selects the "Join a Company" option after registration.	
Precondition	The user must have received a valid join code from a company administrator.	
Scenario Name	**Step**	**Action**
Main	1	Open the join company page.
	2	System prompts the user to enter a company join code.
	3	User key in the join code providing by admin.
	4	Click join company button.
	5	System checks the validation of join code and assign User role to the user.
	6	System navigate user into user home page based on the company.
Alternative Flow - Empty Input	3.1	User leaves empty for join code field.
	3.2	System displays error message "Invalid join code. Please check and try again"
Alternative Flow - Wrong Join Code	3.1	User input wrong join code.
	3.2	System displays error message "Invalid join code. Please check and try again"
Postconditions	The user is successfully added to the company and navigated to the User Home Page. If the join code is invalid or missing, the user is prompted to enter a correct code.	
Rules	The user must enter a valid join code provided by the company administrator.	

Table 3.3 Join an Existing Company

Use Case ID	UC004	
Use Case	User Login	
Purpose	To authenticate the user by verifying their credentials and granting access to the application.	
Actor	The user (registered user) who wants to log into the system.	
Trigger	The user navigates to the login page and initiates the login process by clicking the “Sign In” button after entering their credentials.	
Precondition	The user must already have registered an account in the system.	
Scenario Name	**Step**	**Action**
Main	1	Open the login page.
	2	System requests for input email and password
	3	User key in a registered email and correct password.
	4	Click sign in button
	5	System navigate user into home page based on their role.
Alternative Flow - Invalid Email Format	3.1	User input invalid email format.
	3.2	System display error message “An unexpected error occurred. Please try again.”
Alternative Flow - Wrong Email or Password	3.1	User input wrong email or wrong password.
	3.2	System display error message “Incorrect email or password”
Rules	User has an account.	

Table 3.4 User Login

Use Case ID	UC005	
Use Case	Add New Item to Inventory	
Purpose	To ensure the admin can successfully add new items to the inventory system.	
Actor	Admin	
Trigger	Admin clicks the Add New Item button.	
Precondition	Admin must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.

	2	Click the “Plus” button in the bottom right page.
	3	User input all required details (item name, quantity, unit, cost, price, min stock level, barcode, category, and supplier).
	4	Click add item button to add the item to the inventory.
	5	System generates a transaction record with details of the action.
	6	Verify that the item appears in the Item Tab.
	7	Click the list tile of item to verify the item details is correct.
Alternative Flow - Image	3.1	User don’t input image of item.
	3.2	System display error dialog and message “Image is required.”
Alternative Flow - Empty Input	3.1	User leaves any required field blank.
	3.2	System display error dialog and message “All fields are required.”
Alternative Flow - Cost & Price	3.1	User input cost lower than price.
	3.2	System display error dialog and message “Price cannot be less than cost.”
Alternative Flow - Registered Item Name	3.1	User input a registered item name.
	3.2	System display error dialog and message “An item with the same name already exist. Please choose a different name.”
Rules	User must be an Admin Role of the company.	

Table 3.5 Add New Item to Inventory

Use Case ID	UC006	
Use Case	Edit Existing Item	
Purpose	To ensure the admin can successfully edit the details of an existing item in the inventory.	
Actor	Admin	
Trigger	Admin selects an item to edit.	
Precondition	Admin must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.

	2	Select an existing item from the Item Tab and navigate to Item Details Page.
	3	Click Edit and update necessary field.
	4	Click Save Changes button to update the item details.
	5	Verify that the updated details are reflected in the Item Details Page.
Alternative Flow - Image	3.1	User leaves image fields of the item blank.
	3.2	System display error dialog and message "Image is required."
Alternative Flow - Empty Input	3.1	User leaves any required field blank.
	3.2	System display error dialog and message "All fields are required."
Rules	User must be an Admin Role of the company.	

Table 3.6 Edit Existing Item

Use Case ID	UC007	
Use Case	Delete Item from Inventory	
Purpose	To ensure the admin can delete an item from the inventory, unless the item is part of a composed item.	
Actor	Admin	
Trigger	Admin selects an item to delete.	
Precondition	Admin must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.
	2	Select an existing item from the Item Tab and navigate to Item Details Page.
	3	Click Delete button.
	4	System prompts the admin to confirm the deletion.
	5	Confirm deletion of the item.
	6	System checks if the item is part of a composed item.
	7	System deletes the item and updates the Item Tab.
Alternative Flow –	6.1	System checks if the item is part of a composed item.

Part of Composed Item	6.2	System displays an error message “Item cannot be deleted because it is a part of a composed item.”
Rules	User must be an Admin Role of the company.	

Table 3.7 Delete Item from Inventory

Use Case ID	UC008	
Use Case	Stock In/Stock Out Item	
Purpose	To ensure the admin can update inventory stock levels by stocking in or out items.	
Actor	Admin	
Trigger	Admin clicks the Stock In/Stock Out button on the Item Details Page.	
Precondition	Admin must be logged in and have access to the Stock In/Out feature.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.
	2	Select an existing item from the Item Tab and navigate to Item Details Page.
	3	Admin clicks Delete Stock In/Out button.
	4	Admin selects either Stock in or Stock out.
	5	User inputs the quantity.
	6	User chooses methods for stock in item if Stock In.
	7	The system updates the inventory stock accordingly.
	8	System generates a transaction record with details of the action.
Alternative Flow – Insufficient stock available	5.1	User inputs a quantity larger than the available stock for Stock Out.
	5.2	System displays an error dialog and message “Insufficient quantity for stock out.”
Rules	User must be an Admin Role of the company.	

Table 3.8 Stock In/Out Item

Use Case ID	UC009	
Use Case	User Stock Out Item	
Purpose	To ensure the user can update inventory stock levels by stocking out items.	
Actor	User	
Trigger	User clicks the Stock In/Stock Out button on the Item Details Page.	
Precondition	User must be logged in and have access to the Stock In/Out feature.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.
	2	Select an existing item from the Item Tab and navigate to Item Details Page.
	3	User clicks Delete Stock In/Out button.
	4	Admin click the Stock out button.
	5	User inputs the quantity.
	6	The system updates the inventory stock accordingly.
	7	System generates a transaction record with details of the action.
Alternative Flow – Insufficient stock available	5.1	User inputs a quantity larger than the available stock for Stock Out.
	5.2	System displays an error dialog and message “Insufficient quantity for stock out.”
Rules	User is a user role of the company.	

Table 3.9 User Stock Out Item

Use Case ID	UC010
Use Case	Add Composed Item
Purpose	To verify that an admin can successfully add a composed item to the inventory.
Actor	Admin
Trigger	Admin clicks the “Plus” button in the Composed Item Tab.

Precondition	Admin must be logged in and have access to the Composed Item Tab.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Composed Item Tab.
	2	Admin clicks the "Plus" button in bottom right of page.
	3	Admin inputs required details (composed item name, category, barcode, and description).
	4	Admin selects individual items and specifies quantities from the Choose Item List Page.
	5	Admin clicks the Done button and verifies the composed items.
	6	Admin clicks the "Add Composed Item" button to save the composed item.
	7	System generates a transaction record with details of the action.
	8	Verify that the composed item appears in the Composed Item Tab.
Alternative Flow - Empty Image	3.1	Admin leaves the image field blank.
	3.2	System displays an error dialog and message "Image is required."
Alternative Flow – Empty Input	3.1	User fails to input all required fields.
	3.2	System displays an error dialog and message "Please fill all the fields except description."
Alternative Flow – No Item Selection	4.1	Admin don't select more than one item.
	4.2	System displays an error dialog and message "Please select at least two items."
Rules	User must be an Admin role of the company.	

Table 3.10 Add Composed Item

Use Case ID	UC011
Use Case	Edit Composed Item

Purpose	To verify that only admins can edit the details of a composed item.	
Actor	Admin	
Trigger	Admin clicks the “Edit” button on the Composed Item Details Page.	
Precondition	Admin must be logged in and have access to the Composed Item Tab.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Composed Item Tab.
	2	Select an existing composed item from the Composed Item Tab and navigate to Composed Item Details Page.
	3	Admin clicks the “Edit” button.
	4	Admin modifies the composed item’s details (name, items, or quantities).
	5	Admin clicks “Save” to apply the changes.
	6	System updates the composed item and reflects the changes in the Composed Item Details Page.
Alternative Flow - Empty Image	4.1	Admin leaves the image field blank.
	4.2	System displays an error dialog and message “Image is required.”
Alternative Flow – Empty Input	4.1	User fails to input all required fields.
	4.2	System displays an error dialog and message “Please fill all the fields except description.”
Alternative Flow –	4.1	Admin don’t select more than one item.
No Item Selection	4.2	System displays an error dialog and message “Please select at least two items.”
Rules	Only admins should see and use the “Edit” Button	

Table 3.11 Edit Composed Item

Use Case ID	UC012
Use Case	Delete Composed Item from Inventory
Purpose	To verify that only admins can delete a composed item from the inventory.

Actor	Admin	
Trigger	Admin clicks the “Delete” button on the Composed Item Details Page.	
Precondition	Admin must be logged in and have access to the Composed Item Tab.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Composed Item Tab.
	2	Select an existing item from the Composed Item Tab and navigate to Item Details Page.
	3	Admin clicks the “Delete” button.
	4	System prompts the admin to confirm the deletion.
	5	Admin confirms the deletion.
	6	System deletes the composed item and updates the Composed Item Tab.
Rules	User must be an Admin Role of the company.	

Table 3.12 Delete Composed Item from Inventory

Use Case ID	UC013	
Use Case	Stock Out Composed Item	
Purpose	To ensure that users can stock out a composed item, and system appropriately deducts the correct quantities from the individual items.	
Actor	Admin/User	
Trigger	User clicks the “Stock Out” button for a composed item.	
Precondition	Users must be logged in and have access to the Composed Item Details Page.	
Scenario Name	**Step**	**Action**
Main	1	User clicks the “Stock Out” button on the Composed Item Details Page.
	2	User inputs the quantity of the composed item to stock out.
	3	The system checks whether sufficient quantities of individual items are available.

	4	System deducts the specified quantities from the individual items.
	5	The system generates a transaction record with details of the composed item stock-out action.
Alternative Flow - Insufficient Quantity	2.1	The user inputs a quantity that exceeds the available quantities of individual items.
	2.2	System displays an error dialog and message "One or more items do not have enough quantity to perform this stock out."
Rules	The system must check the availability of each individual item before allowing the stock-out action.	

Table 3.13 Stock Out Composed Item

Use Case ID	UC014	
Use Case	Manage Categories	
Purpose	To verify that the admin can add, update, or delete categories.	
Actor	Admin	
Trigger	Admin clicks the "Add New Category" button.	
Precondition	User must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Categories Management Page.
	2	Admin clicks on the "Add New Category" button.
	3	Admin inputs the required fields (category name and description).
	4	Admin clicks "Add" button to add the new category.
	5	Verify that the category appears in the category list.
Alternative Flow – Empty Input	3.1	Admin leaves the required fields blank.
	3.2	System displays an error message "Both fields are required."
Alternative Flow – Registered Category Name	3.1	Admin input a registered category name.
	3.2	System displays an error message "Category name already exists."
Rules	System should only allow valid inputs for the category name.	

Table 3.14 Manage Categories

Use Case ID	UC015	
Use Case	Manage Suppliers	
Purpose	To verify that the admin can add, update, or delete suppliers.	
Actor	Admin	
Trigger	Admin click the “Add New Supplier” button.	
Precondition	Admin must be logged in and have access to the Suppliers Management Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Suppliers Management Page.
	2	Admin clicks on the “Add New Supplier” button.
	3	Admin inputs the supplier details. (supplier name, phone number, and email)
	4	Admin clicks “Add” button to add the new supplier.
	5	Verify that the supplier appears in the supplier list.
Alternative Flow – Empty Input	3.1	Admin leaves the required fields blank.
	3.2	System displays an error message “All fields are required.”
Alternative Flow – Registered Supplier Name	3.1	Admin input a registered supplier name.
	3.2	System displays an error message “Supplier name already exists.”
Alternative Flow – Invalid Email Format	3.1	Admin input invalid email format.
	3.2	System displays an error message “Please enter a valid name address.”
Rules	All required fields must be filled out before saving.	

Table 3.15 Manage Suppliers

Use Case ID	UC016
Use Case	User Role Management
Purpose	To verify that the admin can manage user roles effectively by editing or deleting roles.
Actor	Admin
Trigger	Admin selects a user and attempts to change or delete their role.

Precondition	Admin must be logged in and have access to the User Role Management Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the User Role Management Page.
	2	Admin selects a user to edit their role.
	3	Admin modifies the user's role and clicks save button.
	4	System updates the user's role.
Rules	All updates to roles must be accurately reflected in the system.	

Table 3.16 User Role Management

3.1.3 Activity Diagram

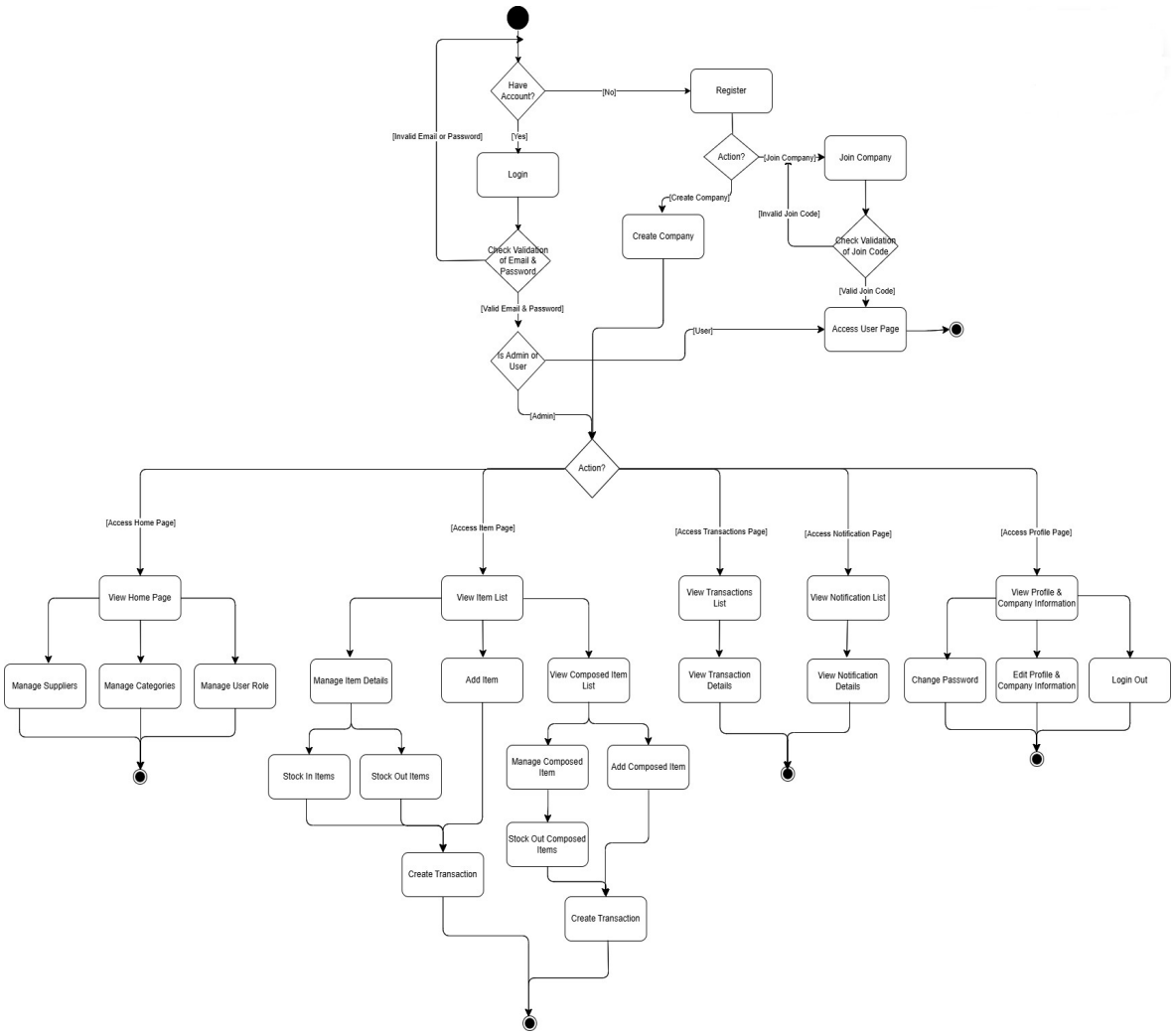


Figure 3.3 Activity Diagram for Admin

The activity diagram illustrates the comprehensive flow of actions available to an Admin can perform in the inventory management system. It begins with the initial decision of whether the admin has an account. If not, they must register; if they already have an account, they proceed to the login process. After logging in, the system will validate the admin's credentials, and depending on whether the admin belongs to a company, they can either create a new company or join an existing one using a valid company's join code.

Once logged in and associated with a company, the admin has access to several key features. These features include managing suppliers and categories, allowing the admin to add, update and delete information in their areas. The admin can also control the roles of users to better manage the privacy issues and confidentiality of the company. Not only that, but admin can also manage inventory by adding items, stocking in or out of items, and managing composed items. Each item's operation generates a transaction record that the admin can review later to better track records.

In addition to inventory management, admin can view system-related notifications, such as low stock level alert, and other important messages. They can also manage profiles, change passwords, and update company information. The admin's workflow ends with logging out of the system or completing a specific task, such as viewing transactions or managing inventory. This diagram highlights the admin's role in the operation and configuration of the management system, showing how the various tasks relate to each other and reflecting the admin's overall control over the system's functionality, ensuring the smooth operation of inventory management.

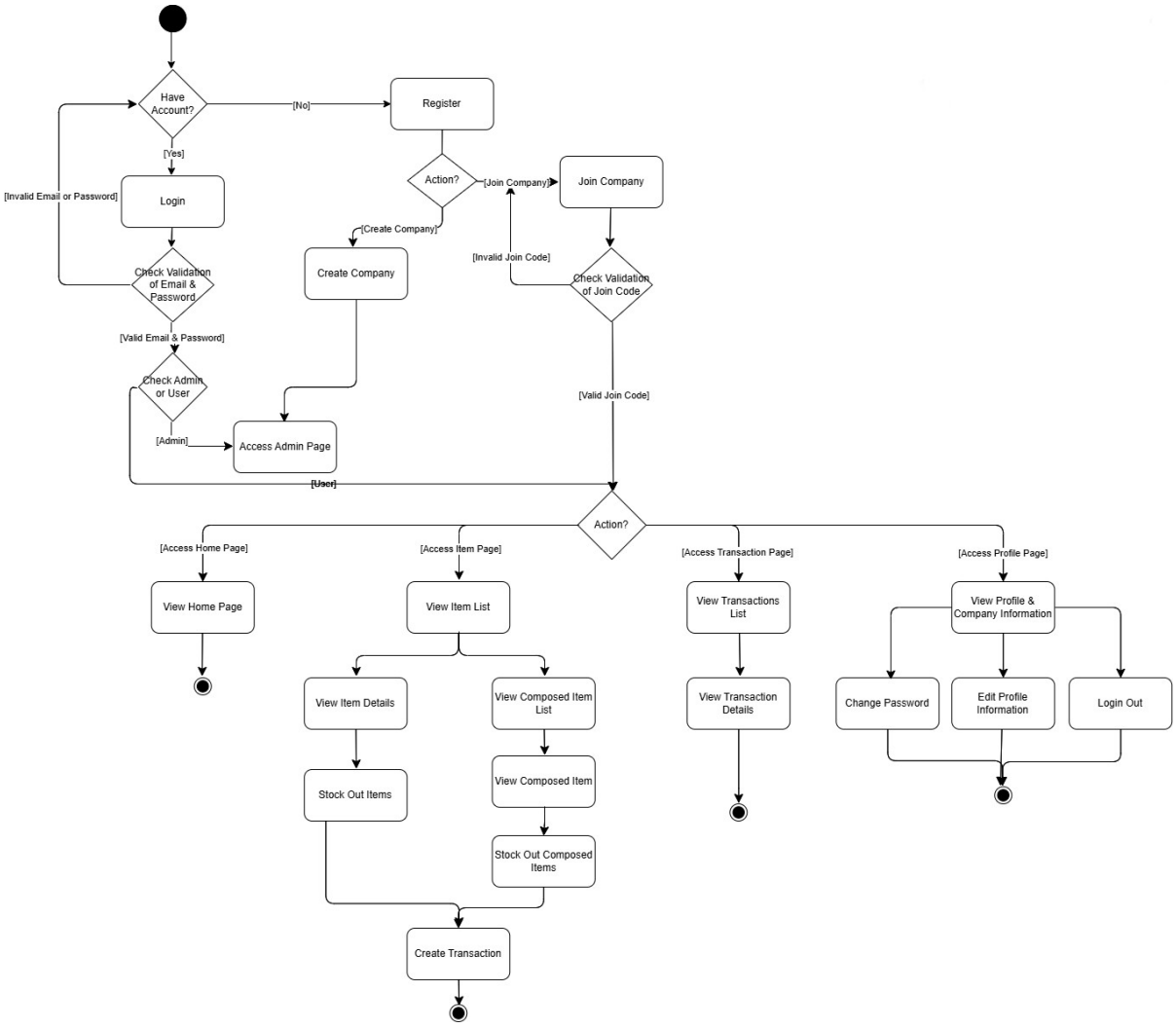


Figure 3.4 Activity Diagram for User

The activity diagram for the User in the inventory management system outlines the sequence of actions and the available functionalities tailored to their role. The process begins with the User determining whether they have an account, if they do not, they must go through the registration process. If the User already has an account, they can log in, where the system will validate their email and password. Upon successful login, the system identifies the User's role (Admin or User) and, if the User is a non-Admin, they are redirected to the User-specific interface.

Once logged in, the User is presented with the Home Page which serves as a summary dashboard. The User Home Page displays a summary of the total number of items and composed items available in the inventory. Additionally, it shows a list of recent transactions performed by the User, providing an overview of their recent activity within the system. In the item management section, the User can view the list of items and composed items in the

inventory. However, their permissions are limited to stock out operations. They can remove stock from items or composed items, which automatically generates transaction records in the system. The User does not have the authority to edit items details or delete items; these actions are reserved for Admin.

In the transaction section, the system only displays transactions that the User has personally created, ensuring that they do not have visibility into other users' transactions. This maintains privacy and ensures that each User only manages their own activities. The User also has options to manage their account settings. They can change their password, edit their profile, and log out of the system when finished. The diagram provides a clear flow of how User interact with the system, focusing on their key function. The restrictions on editing and viewing others' transactions ensure that the User's role is appropriately limited to their designated responsibilities within the system.

3.2 Algorithm

$$\text{Days to Minimum Stock Level} = \frac{\text{Current Quantity} - \text{Minimum Stock Level}}{\left(\frac{\text{Total Consumption}}{\text{Number of Days in Period}}\right)}$$

The algorithm of this application is a straightforward inventory prediction system designed to help businesses manage stock levels effectively by using historical data to forecast when items will need to be reordered. It starts by fetching all inventory items from the Firestore database using the fetchAllItemsFromInventory() method and retrieving recent transaction data through the fetchRecentTransactions() method. The transaction data, which spans the past 30 days, is used to calculate the average daily consumption rate for each item over the last 7 days.

The calculateAverageConsumptionForItem() method estimates how quickly an item is being consumed based on recent transactions. With this information, the system predicts when an item's stock will drop below its predefined minimum level by using the predictReorderDateForItem() method. This method calculates how many days remain before the item reaches the minimum stock level, considering the current quantity and average consumption rate.

If the system determines that an item is likely to reach its minimum stock level within the next 3 days, it generates a notification for the admin. This notification is stored in the Firestore

database, and if the user is an admin, a local notification is also triggered on their device. This proactive approach ensures that businesses can maintain optimal stock levels and avoid stockouts by giving timely warnings when action is needed.

Overall, this method provides a practical, consumption-based prediction model that allows small businesses to monitor their inventory in real time and receive alerts when it's time to restock. While simple and effective, the system could be further improved with more advanced forecasting algorithms to enhance accuracy.

Chapter 4

System Design

4.1 System Block Diagram

4.1.1 Admin Module

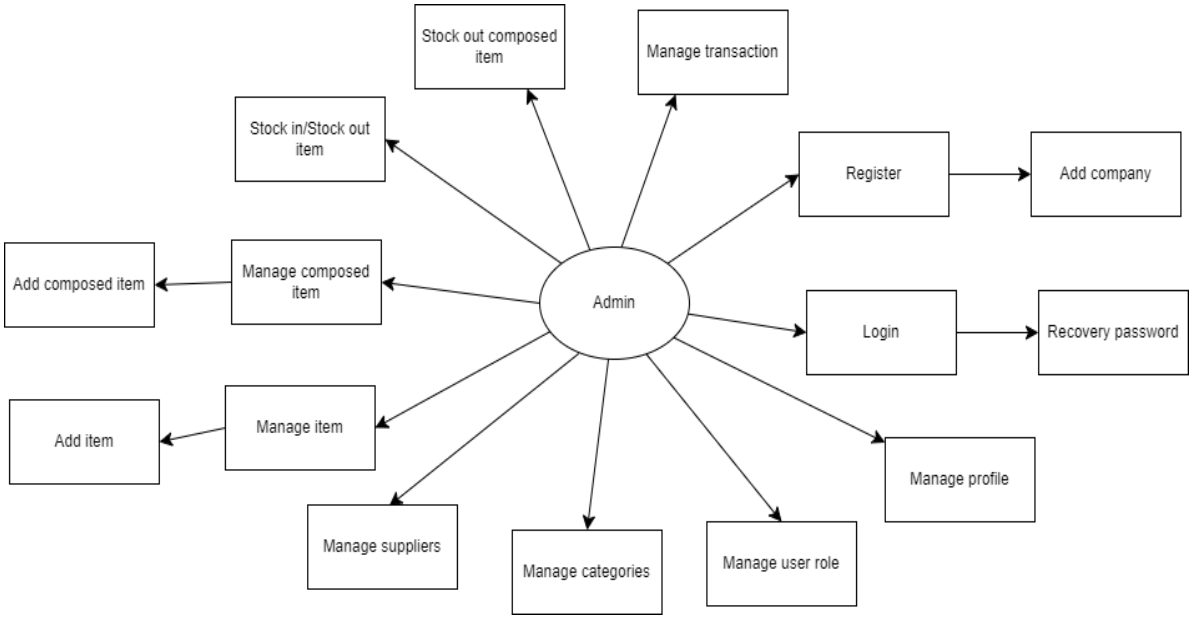


Figure 4.1 Admin Module

The admin module in inventory management app is a comprehensive and powerful tool designed to facilitate the smooth operation of inventory processes and system administration. Admin has powerful capabilities, starting with the ability to register new companies, laying the foundation for organizational participation within the system. A central feature of the admin module is user management, where admins can assign and control roles to users, ensuring proper access control within the system. The admin is granted exclusive rights to manage inventory items. They can add, edit, and delete items, as well as manage composed items, which are combinations of multiple items with specified quantities. This capability is particularly useful when stocking out composed items, as the system automatically reduces the set quantity for each item in the composed item, streamlining the process of selling multiple items simultaneously.

In addition to item management, the admin module includes advanced inventory functions, such as multi-item stock in and stock out, allowing admins to manage multiple items at once. The stock in process offers two convenient ordering methods: WhatsApp and email, providing flexibility in communication with suppliers.

Admins are also responsible for overseeing inventory transactions, with the ability to stock in and stock out items, create transactions. This insight helps in optimizing resource utilization and operational efficiency. In addition, the admin module also enables efficient supplier management, ensuring that all aspects of inventory organization are streamlined and easily accessible for optimal operational control.

In contrast to users, who can only manage stock out processes and view their own transactions, admins have full control over inventory operations and supplier communication, ensuring that all inventory activities remain within their authority. Overall, the admin module offers a robust suite of tools for managing inventory, suppliers, categories and user role, empowering admins to maintain efficient, organized operations.

4.1.2 User Module

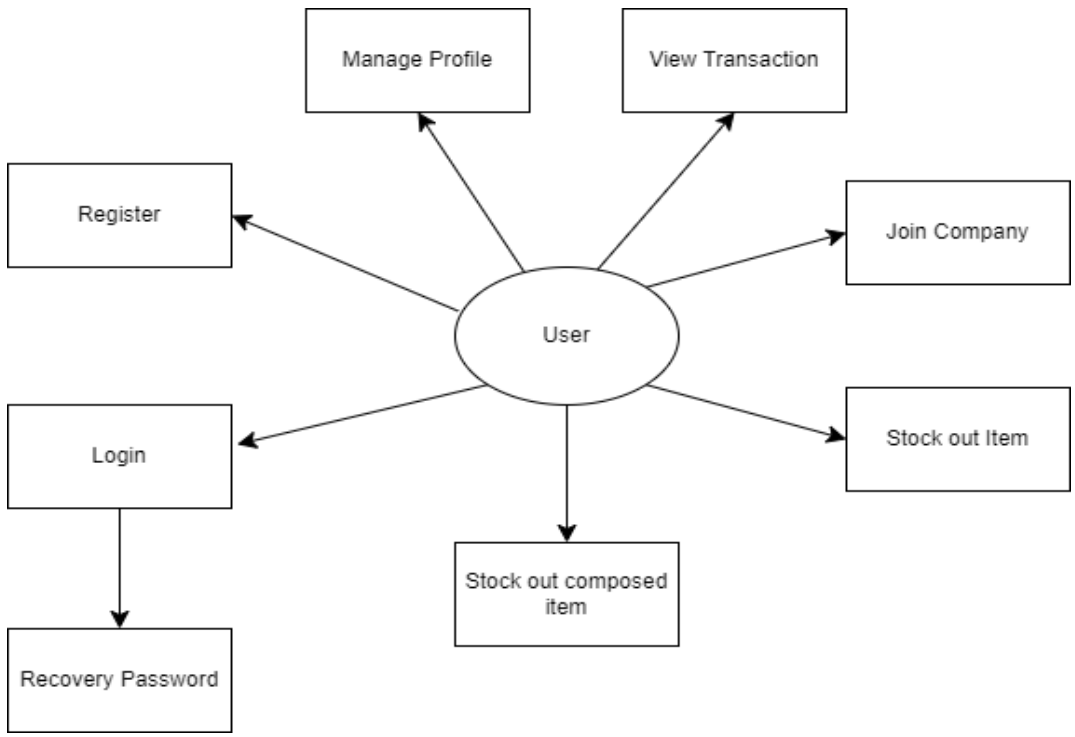


Figure 4.2 User Module

The user module provides a simplified interface for users to effectively interact with the inventory management system. Users can register to create a personalized account and log in securely to access the system's features. If users encounter any issues with their login credentials, the module provides password recovery options to ensure continued access. After registration, users are prompted to create a new company or join an existing company. If they choose to join a company, they can do so by entering a valid join code, which links them to that company's inventory system. Once connected, users can perform several actions on the company.

The user module also allows users to effectively manage inventory operations. Users can perform stock out operations on items and composed items, which are combinations of multiple items with predefined quantities. This feature enables users to conveniently manage sales involving multiples items. Additionally, users can view transactions that they have created, allowing them to monitor their inventory activity. However, users cannot view transactions created by others, edit item details, or delete items. This set of permissions helps to maintain the security and integrity of the inventory system while providing users with the tools they need to participate in inventory management tasks.

4.2 Flowchart

4.2.1 Flowchart for Admin

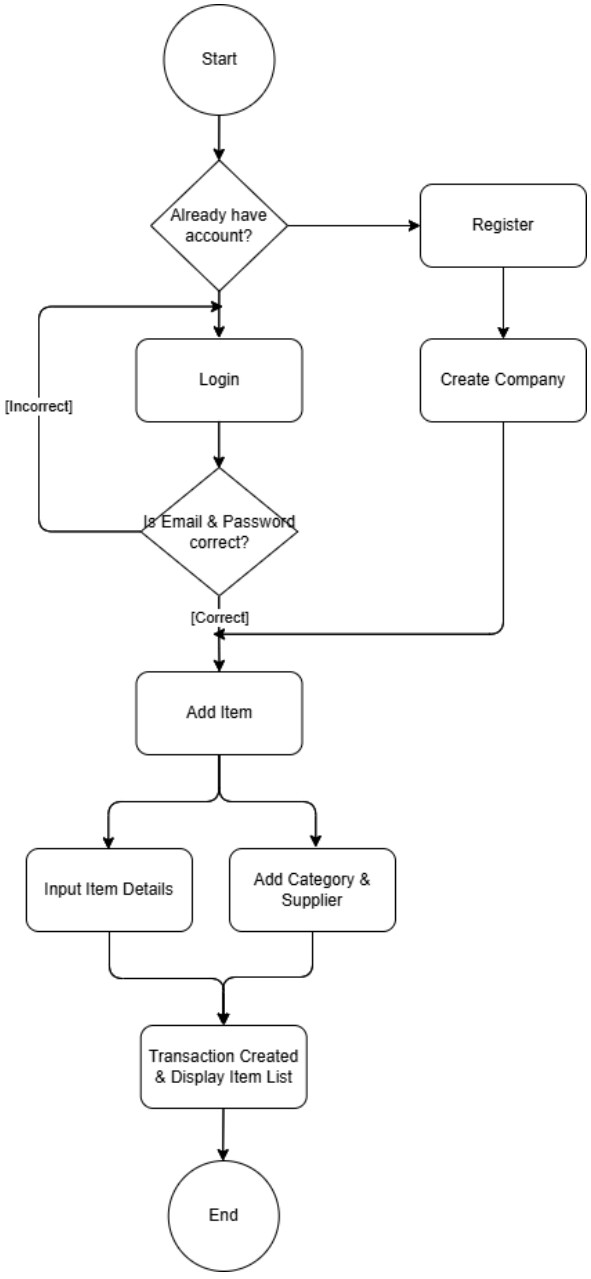


Figure 4.3 Flowchart

The flowchart of figure 4.3 outlines the process by which an admin registers an account and adds items to the inventory database. First, the user determines if they already have an account. If so, they proceed to the login step; otherwise, they are directed to register a new

account. After registration, the system prompts the user to create a company or join an existing company. By selecting create a company, the user becomes an administrator, and a unique join code is generated for the company. After company created, the admin logs in to access the system. After a successful login, the administrator can add items to the inventory by entering item details and add category and supplier for the item. Once the details are submitted, a transaction is created and the system updates and displays the list of items in the inventory. This structured process ensures that the admin can effectively manage and update inventory within the system.

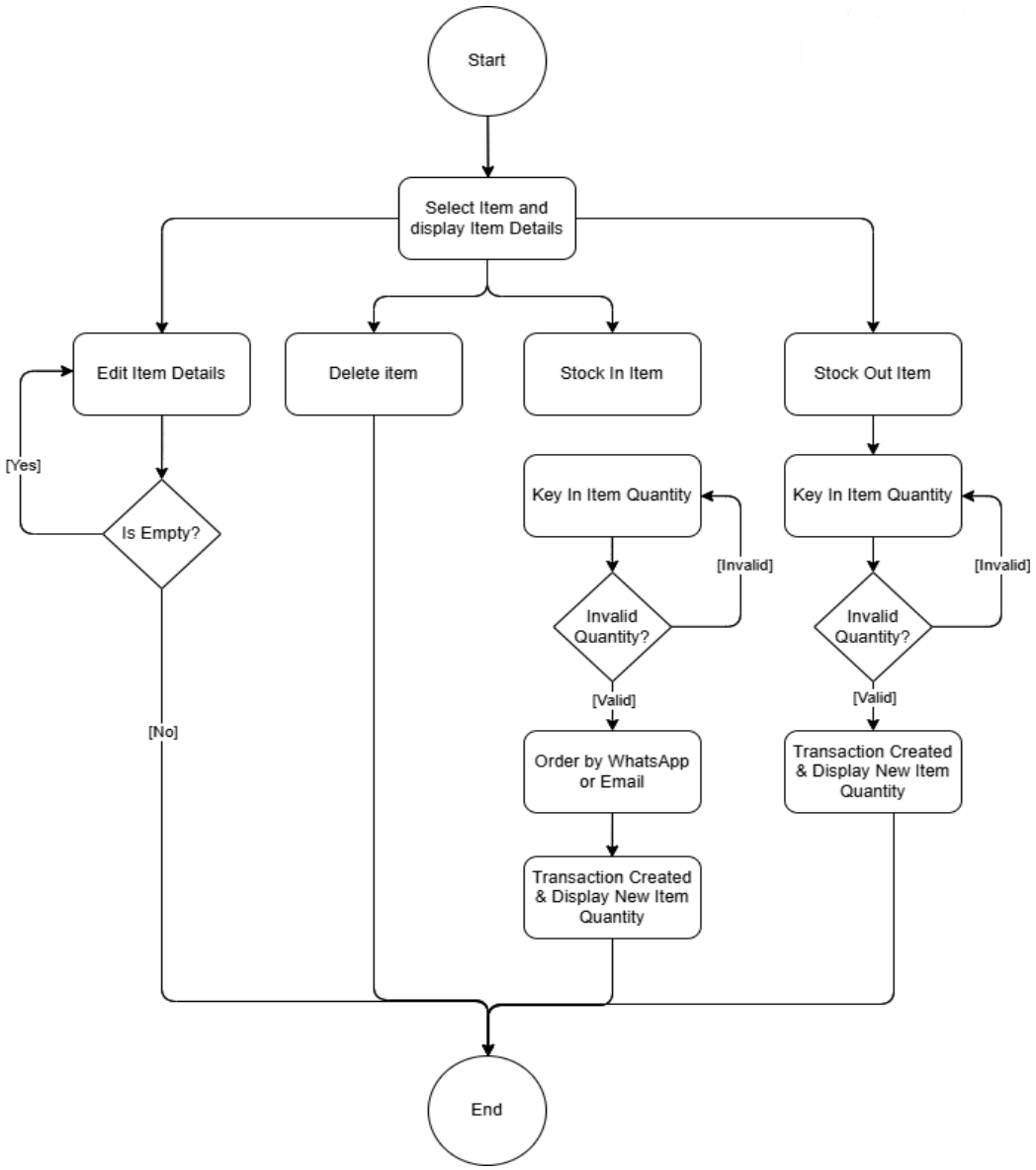


Figure 4.4 Flowchart for Manage Items

Next, the flowchart of figure 4.4 illustrates the continuation of the inventory management process for an admin after adding items to the system. After an admin successfully adds an item to the inventory system, they can manage the item through various actions such as editing, deleting, or adjusting the stock levels. When the admin selects an item, its details, including quantity and other relevant attributes, are displayed. The admin has several options: they can edit the item details, delete the item, stock in or stock out item. If the admin chooses to edit the item, they can modify details like the name, description, price, or category. The system checks for any missing data, ensuring valid input before saving the changes.

For stock in items, the admin inputs a quantity, which will be added to the current inventory level. The system validates the input to ensure it is correct. Once the quantity is confirmed, the system updates the stock, records a transaction, and displays the new quantity. Additionally, the stock in process offers integrated methods for ordering items from suppliers, either via WhatsApp or email. These methods facilitate communication with suppliers, enabling the admin to reorder items directly, streamlining the restocking process.

If the admin chooses to stock out items, they can reduce the item's quantity by entering a value that will be subtracted from the current stock level. Similar to the stock in process, the system validates the entry and updates the stock accordingly. All actions are logged in to the system, ensuring that inventory is accurately maintained and tracked.

This comprehensive process allows the admin to efficiently manage the inventory, ensuring that item details are accurate, stock levels are properly maintained, and any necessary orders are placed promptly.

4.2.1 Flowchart for User

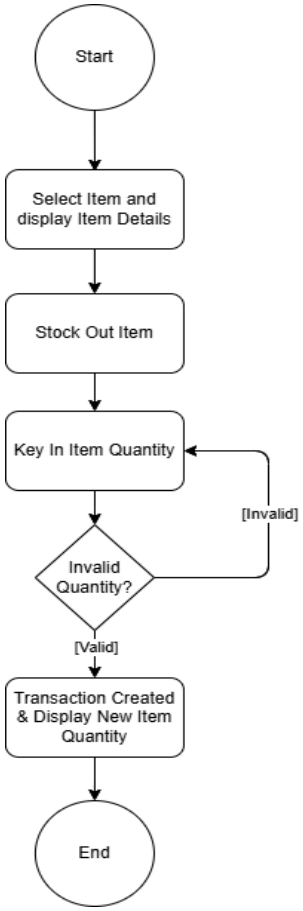


Figure 4.5 Flowchart for User

This flowchart of figure 4.5 illustrates the stock out operation flow for a user role in the inventory management system. The user is responsible for selling the stock by reducing the quantity of the item when necessary. It begins when a user selects an item from the inventory list. Upon selection, the system displays the relevant details, such as the item's name and current stock quantity. The process then clicks the stock out button, where the user is prompted to enter the quantity of the item to be removed from stock. The system validates the entered quantity to ensure it is correct, checking whether it is a positive number and does not exceed the available stock. If the quantity is invalid, the user is prompted to re-enter a valid value. Once the quantity is validated as correct, a transaction is created in the system to reflect the adjustment, and the new item quantity is updated and displayed. The process concludes after the successful creation of the transaction, ensuring that the inventory remains accurate and properly tracked.

4.3 Entity Relationship Diagram

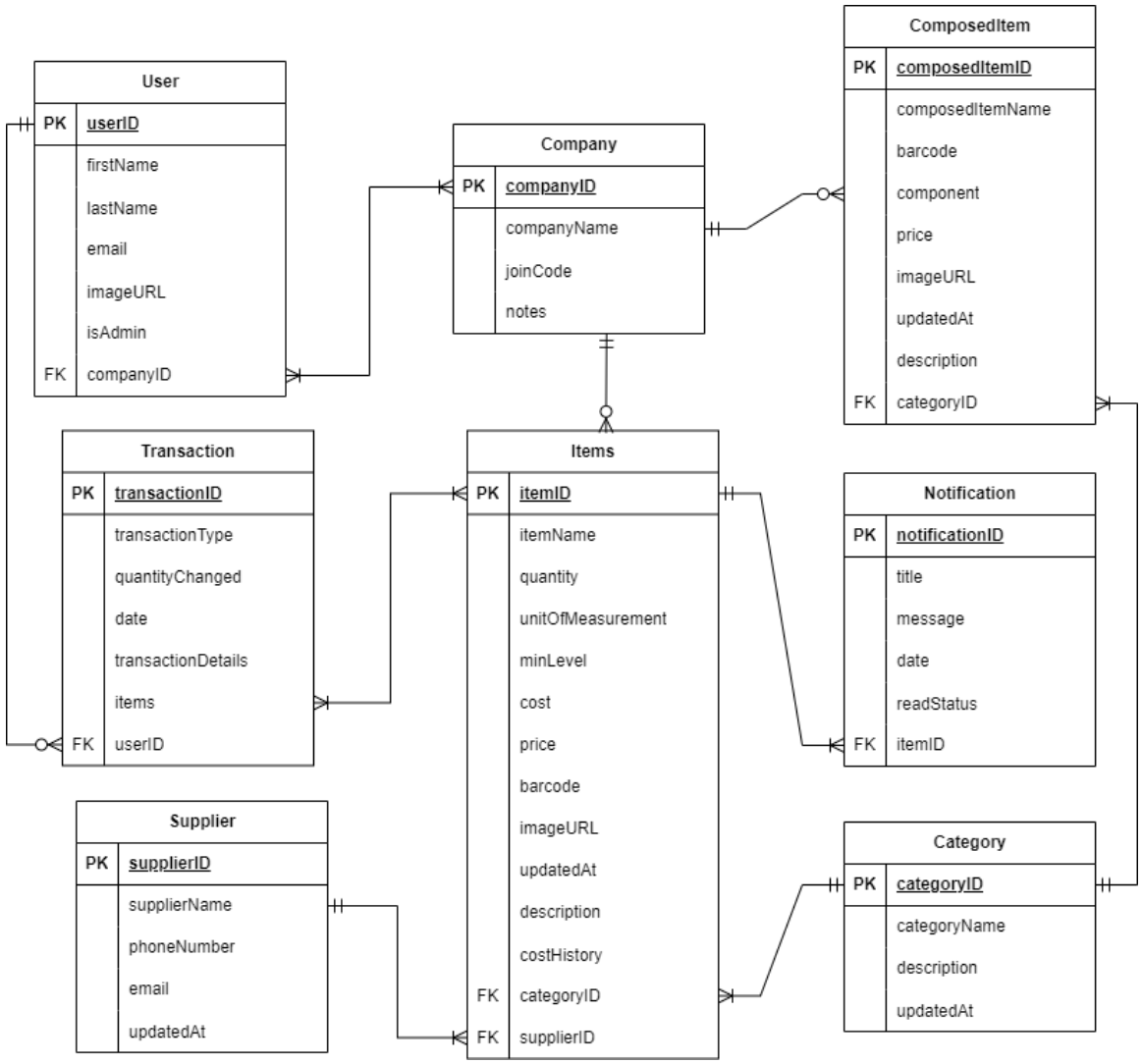


Figure 4.6 Entity Relationship Diagram

The inventory management system employs a NoSQL database model to organize and manage data across multiple collections effectively. Each collection is tailored to store specific types of inventory-related information, ensuring efficient data handling.

The User collection captures essential details about the app users. Each user is assigned a unique userID and is linked to their username, email, and companyID, which associates the user with their respective company. Then isAdmin field indicates whether a user has administrative privileges, distinguishing admins from regular users.

The Company collection stores data specific to each company. Each company is uniquely identified by companyID and includes details such as the company's name (companyName), notes for additional information, and a unique joinCode used for onboarding new users into the company.

The Item Collection, nested under the company, stores detailed information about items in the company's inventory. Items are uniquely identified by itemID and include various attributes such as itemName, supplierID (indicating the supplier), categoryID, current quantity, minLevel (minimum acceptable stock level), cost, price, barcode, imageURL, updatedAt timestamp, description, and unitOfMeasurement. Additionally, a costHistory array is maintained for tracking past costs, aiding in decision-making for future orders.

The ComposedItem Collection, also under the company, handles composed items, which are combinations of multiple individual items. Each composed item is identified by composedItemID and includes attributes such as composedItemName, categoryID, an array of component itemIDs (representing the items that make up the composed item), barcode, imageURL, updatedAt timestamp, and description.

The Transaction Collection logs all inventory-related transactions. Each transaction is uniquely identified by transactionID and contains an items array, storing multiple items involved in the transaction. This includes details like itemID, itemName, quantityChanged, and unitOfMeasurement. The collection also tracks the date of the transaction, transactionType (e.g., purchase, sale, adjustment), and transactionDetails, such as the reason for the adjustment. The userID field identifies the user who executed the transaction.

The Supplier Collection stores information about the company's suppliers. Each supplier is identified by supplierID and includes fields such as supplierName, phoneNumber, email, and updatedAt timestamp.

The Category Collection categorizes items within the company. Each category is represented by a unique categoryID and includes attributes like categoryName, description, and an updatedAt timestamp.

The Notification Collection within the company is used to manage notifications sent to admins. Each notification is uniquely identified by notificationID and includes the associated itemID, title, message, date, and readStatus, indicating whether the notification has been read.

The database structure organizes data efficiently across collections to handle various aspects of inventory management, from user roles and company details to item tracking, supplier management, and transaction logging. This modular design ensures that each component of the system is well-structured and easily accessible, enabling smooth inventory operations.

Chapter 5

System Implementation

5.1 Hardware Setup

The hardware involved in this project is a computer, an Android mobile device and an Android emulator [10]. A computer is issued for the development of mobile application purposes, and the mobile devices are used to test and deploy this application to identify issues and refine it.

Description	Specifications
Model	ILLEGEAR ONYX V
Processor	Intel Core i7-10750H
Operating System	Windows 11
Graphic	NVIDIA GeForce GTX 1650
Memory	16GB DDR4 RAM
Storage	512GB NVME M.2 SSD

Table 5.1 Specifications of Laptop

Description	Specifications
Model	Xiaomi Mi 8
Chip	Octa-core Max 2.8GHz
Operating System	Android 10.0
Memory	6GB RAM
Capacity	128GB

Table 5.2 Specifications of Android Mobile Device

Description	Specifications
Model	Pixel 7a API 14
Operating System	Android 14.0
Memory	8GB RAM
Capacity	128GB

Table 5.3 Specifications of Android Emulator

5.2 Software Setup

Before starting the application development process, the necessary software tools must be installed and downloaded to my laptop.

1. Visual Studio Enterprise 2019 16.11.27
2. Android Studio (version 2023.2)
3. Flutter SDK 3.19.5
4. Dart SDK 3.2.4
5. Firebase CLI

Installing these software tools will provide you with the necessary infrastructure and resources to commence the development process effectively.

The development software utilized in this project includes Flutter for the front-end development and Firebase for backend services such as authentication. Dart is the primary programming language employed for both front-end and back-end development. The Firebase Authentication library will be integrated into the application to authenticate users securely.

Description	**Specifications**
Front-end	Flutter, Dart
Back-end	Firebase Authentication
Database	Firebase Firestore, Firebase Storage

Table 5.4 Specifications of Software

5.3 Setting and Configuration

In this section, it outlines the setup and configuration of the development environment, project settings, and devices used for the development and testing of the inventory management mobile application. This ensures a smooth development process and facilitates testing across different platforms.

5.3.1 Development Environment Configuration

The development environment was configured on a Windows-based laptop. Below are the steps taken to setup the environment for mobile application development:

1. Download and install the latest versions of visual studio, Android Studio, Flutter and Dart SDK.
2. Configure the Android SDK within Android Studio for emulation and debugging.
3. Install and configure Flutter plugins in Visual Studio and Android Studio.
4. Use Firebase CLI to authenticate and link the app to Firebase services.

5.3.2 Project Configuration

The configuration of the Flutter project was critical for ensuring smooth development and deployment.

1. Initialization a flutter project using the “flutter create” command generating the necessary project structure.
2. Next, go to the Firebase Console and create a new Firebase project, which acts as the central hub for all services associated with the app.
3. Once the Firebase project is created, the next step is to link the Android version of the app to Firebase. Enter the Android package name (found in the “android/app/src/main/AndroidManifest.xml” file of the flutter project) and the app nickname.

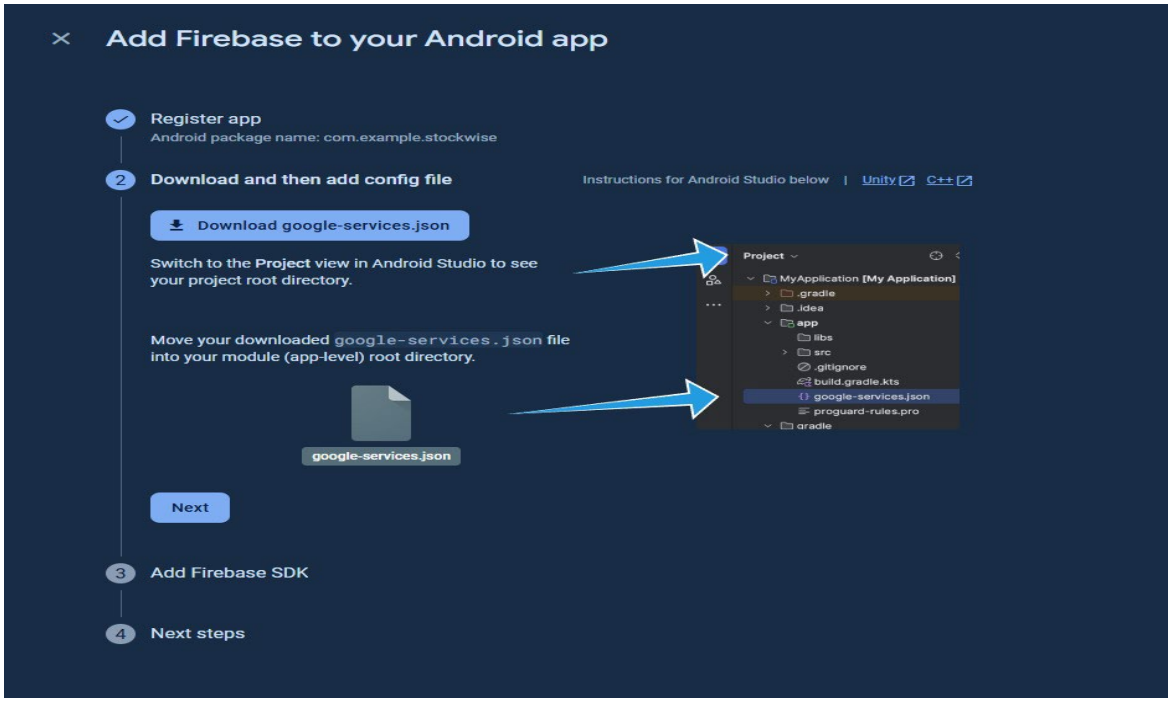


Figure 5.1 Download Firebase JSON File

4. Download the “google-services.json” file provided by Firebase.
5. Move the “google-services.json” file into the Flutter project directory: android/app/.

Figure 5.2 Configuration Command of Firebase Services

6. Open the “android/build.gradle” file and “android/app/build.gradle” file to add the command to configure the Flutter project to use Firebase services.

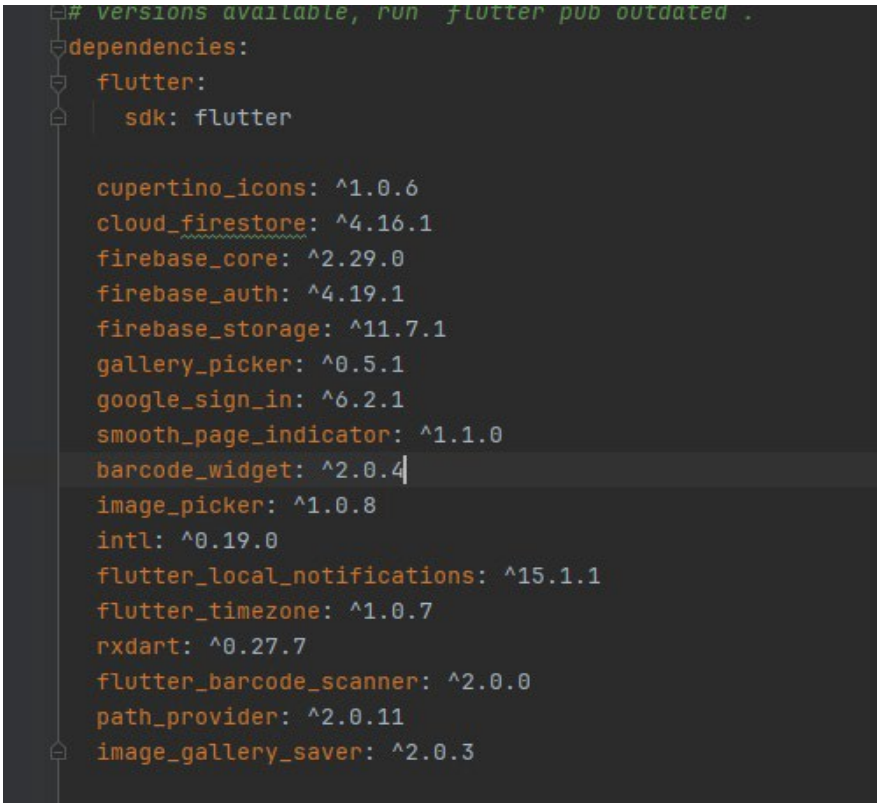


Figure 5.3 Dependencies for Firebase & Flutter

7. Configure the "pubspec.yaml" file to include all required dependencies for Firebase and Flutter.
8. Lastly, initialize Firebase (Firebase.initializeApp()) in the main application file.

5.3.3 Mobile Device Configuration

1. The mobile device used for testing (Xiaomi Mi 8) was configured to enable testing and deployment of the application.
2. Enabled developer model and USB debugging on the device to allow for direct installation of the app via Android Studio.
3. The app was deployed on Xiaomi Mi 8 for real-world testing to ensure the application's performance on physical devices.

5.3.4 Testing Setup

1. Create an Android Virtual Device (AVD) using Android Studio for initial testing.

2. Deploy the app to the physical device using “flutter run” command.
3. Simulate different network conditions to test the app’s offline functionality and performance under poor network coverage.

5.4 System Operation

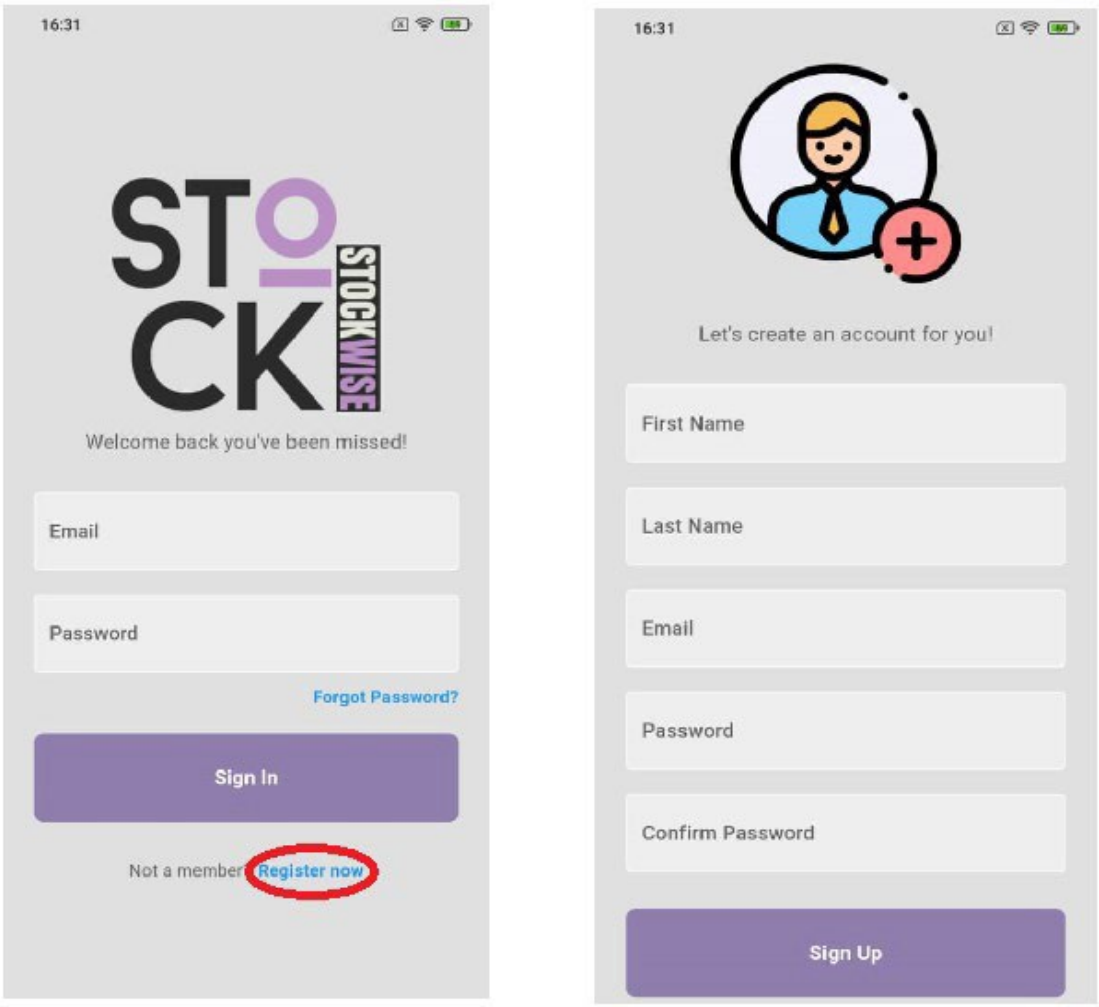


Figure 5.4 Login Page & Register Page

To starting using the app, if the user does not already have an account, they will need to register one. On the login page, you will be asked if you have an account. If you do, simply enter your correct email and password to log in immediately. If you do not have an account, you will need to click Register Now in the lower right corner to be taken to the registration page where you

can fill in details such as first name, last name, email, and password. After registration, you will need to decide whether to create a new company or join an existing company.

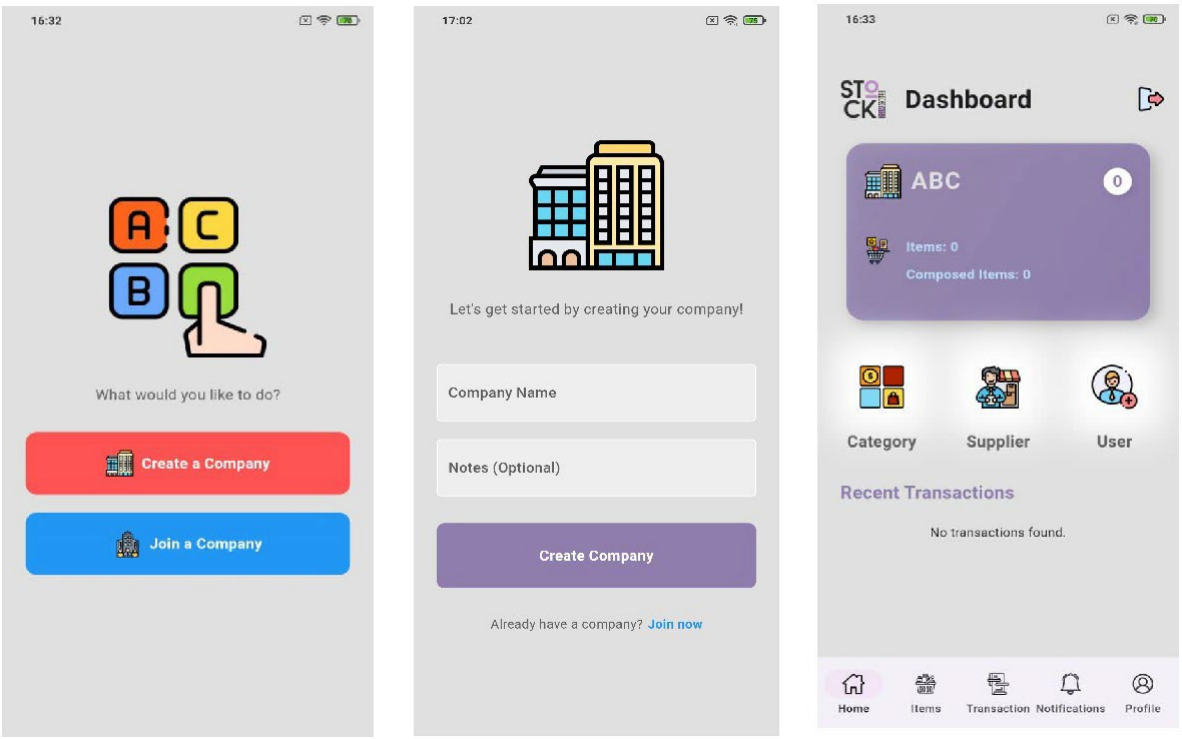


Figure 5.5 Create Company Page & Admin Home Page

If you choose to create a company, you will be prompted to provide details of the company, such as the company name. after successful creating a company, a unique join code will be generated. This join code allows other users to join your company. As a company creator, you will be assigned administrator permissions, enabling you to manage the company's settings, items, transactions, and other important features. Once the company is set up, you will be directed to the admin home page where you can manage and view various aspects of the company.

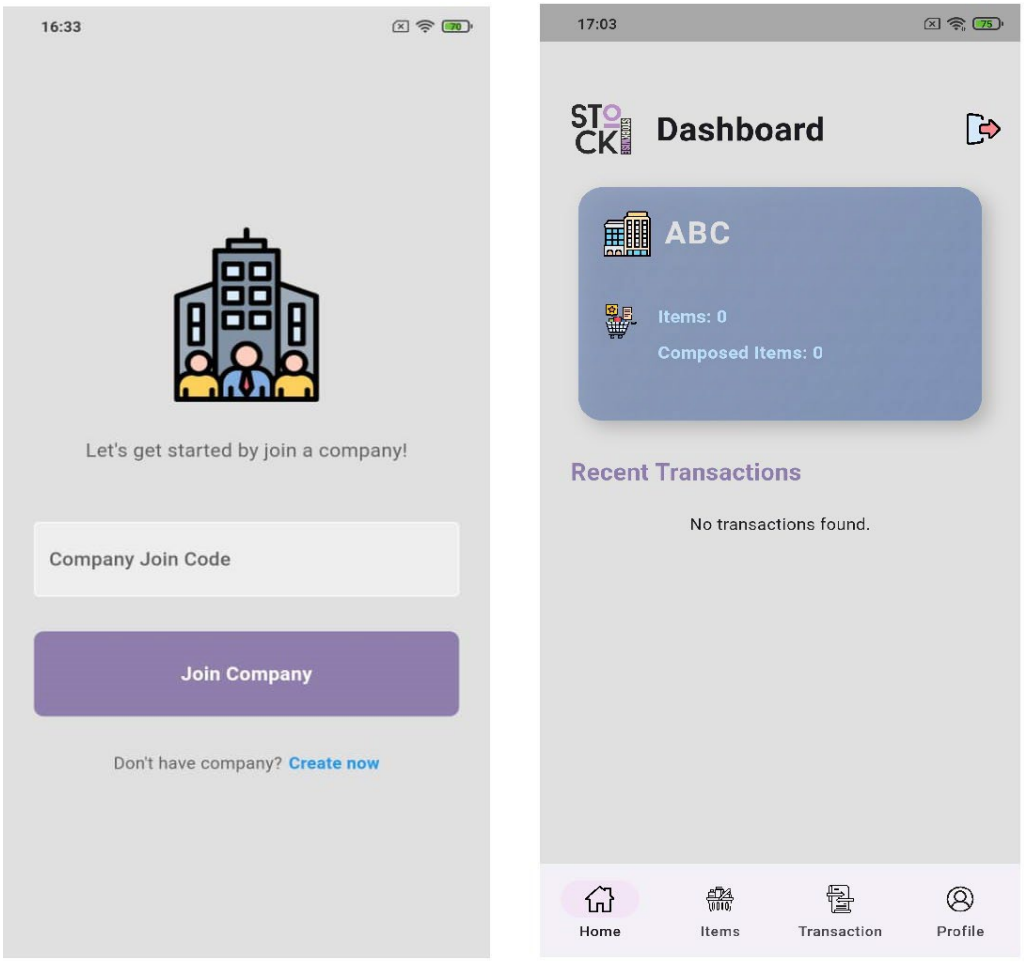


Figure 5.6 Join Company Page & User Home Page

Alternatively, if you wish to join an existing company, select the Join Company option. You will need to enter the unique join code provided by the company administrator. The system will verify the code, and if correct, you will become a member of the company. After successfully joining, you will be given user-level access unless designated as an administrator. This will allow you to view inventory items, transactions, and access other relevant company information. Depending on your role, you will be directed to either the Admin Home Page or the User Home Page where you can perform tasks based on your access level.

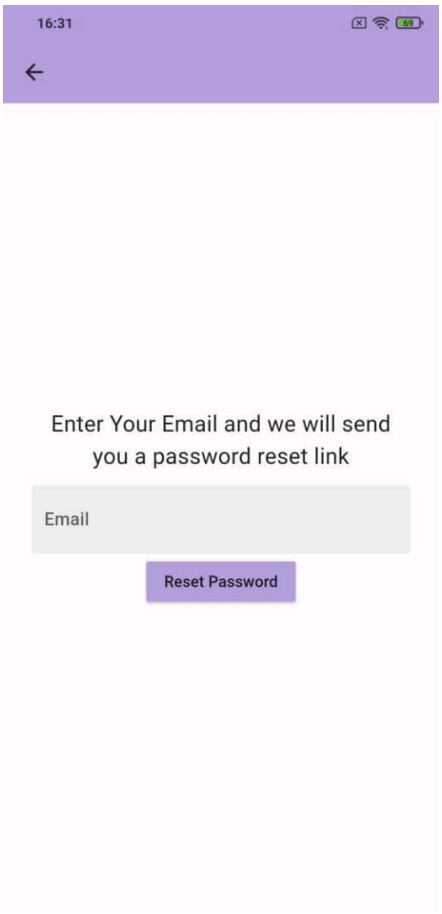


Figure 5.7 Reset Password Page

If there are any issues logging in, such as an invalid email or password, double-check your credentials or reset your password if necessary. If you encounter issues with your join code, make sure it is correct and request a new code from your company administrator if necessary.

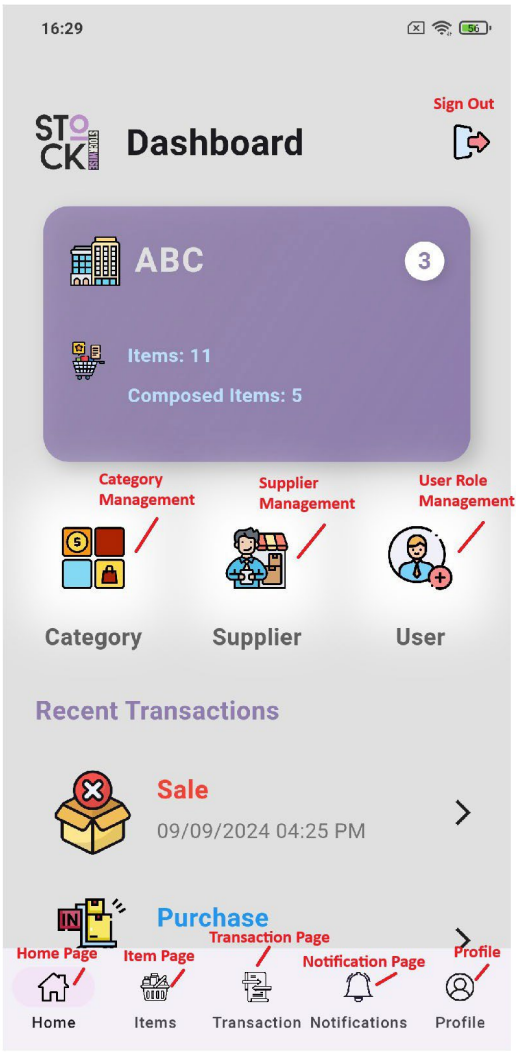


Figure 5.8 Admin Home Page

The Admin Home Page in the inventory management system serves as the central hub for managing key aspects of the company. At the top right of the page, there is a Sign Out button that allows the admin to securely log out of the application. Just below the sign-out button, a Company Summary is displayed, showing the company name, the total number of items, and the number of composed items currently in the inventory.

Beneath the summary, there is a row with three management buttons. The first button is for Category Management, where the admin can manage item categories within the system. The middle button leads to Supplier Management, enabling the admin to manage the suppliers linked to the items. The third button on the right is for User Role Management, allowing the admin to control user roles and permissions within the company.

Below this management row, the page displays the five most recent transactions, providing a quick snapshot of recent sales or purchases made within the system. This section helps the admin monitor recent inventory activities immediately.

At the bottom of the page is the navigation bar, which enables easy access to other key pages in the application. The bar is arranged with the following tabs: Home Page (for the dashboard), Item Page (to manage items), Transaction Page (for transaction details), Notification Page (for system alerts), and Profile Page (to view or edit admin profile and company information). The admin can easily switch between these pages by clicking the respective icons in the navigation bar.

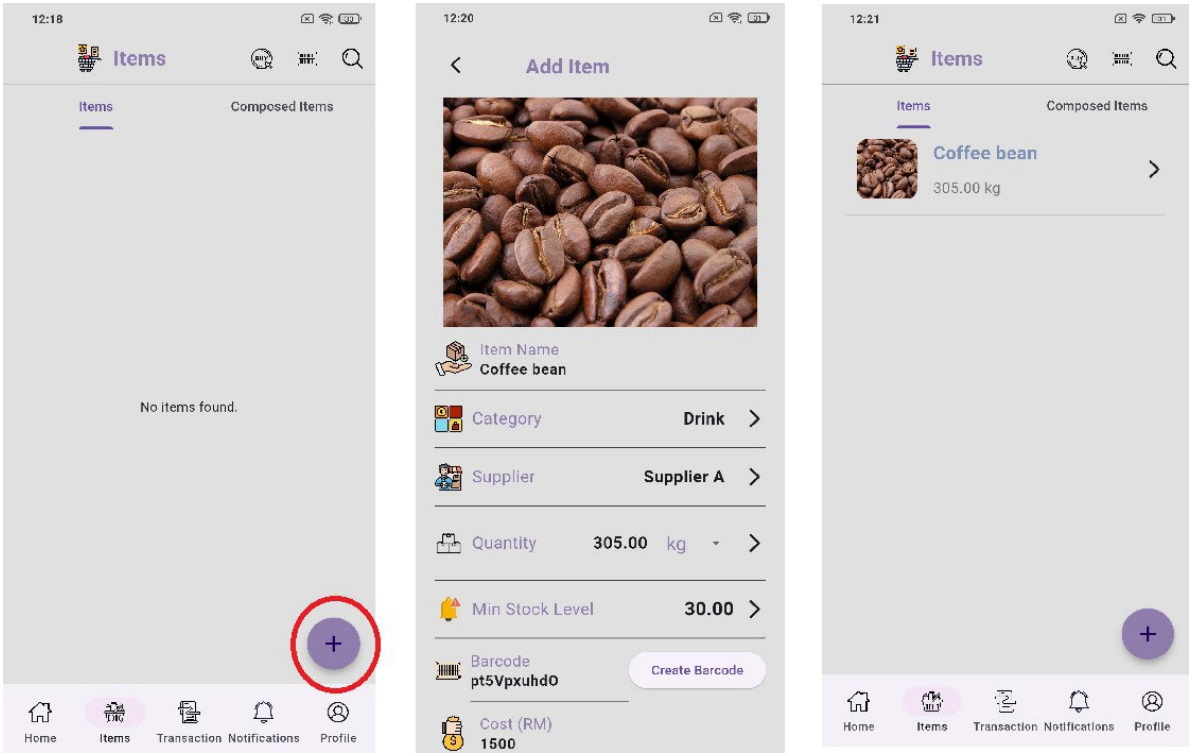


Figure 5.9 Item Page & Add Item Page

To add an item to the inventory in the system, begin by navigating to the Item Page after logging in. The Item Page is divided into two tabs: the Item Tab on the left side, which displays all individual items currently in the inventory, and the Composed Item Tab on the right side, which shows items composed of two or more individual components.

In the Item Tab, you can view details such as item names, quantities, and other relevant information for each individual item. To add a new item, click on the button located at the

bottom left corner of the Item Page. This will take you to the Add Item Page, where you can input the necessary details for the item, including the item name, quantity, unit of measurement, cost, price, barcode, image, supplier, and category. Once all the details are filled in, the item will be added to the inventory and displayed in the Item Tab.

On the Add Item Page, you will need to provide the item's name, quantity, unit of measurement (e.g., units, kilograms, Liters), and set a minimum stock level to receive alerts when stock falls below this threshold. You will also enter the cost of the item and its selling price, as well as assign a barcode for easy identification and tracking. A description field allows for additional information or notes about the item.

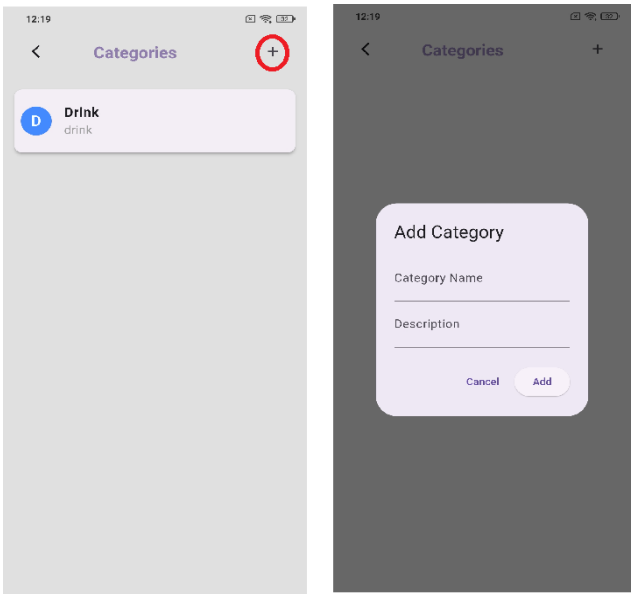


Figure 5.10 Choose Category Page

Additionally, you can upload an image of the item to make it easier for users to visually identify the product within the system. It is also necessary to select the item's Supplier from a pre-existing list; if the supplier is not yet added, it must first be created through the supplier management function. Similarly, you will need to assign the item to a Category (such as electronics or office supplies), which can be managed via the category management feature.

Once all the details are filled in, click the Save button to add the item to the inventory. After saving, the new item will appear in the item list on the Item Page with all its details, including the image, supplier, and category information. Depending on your user role, you can manage

these items by viewing, editing, or deleting them. Admins have full control over inventory management, while regular users might have restricted access based on their assigned permissions.

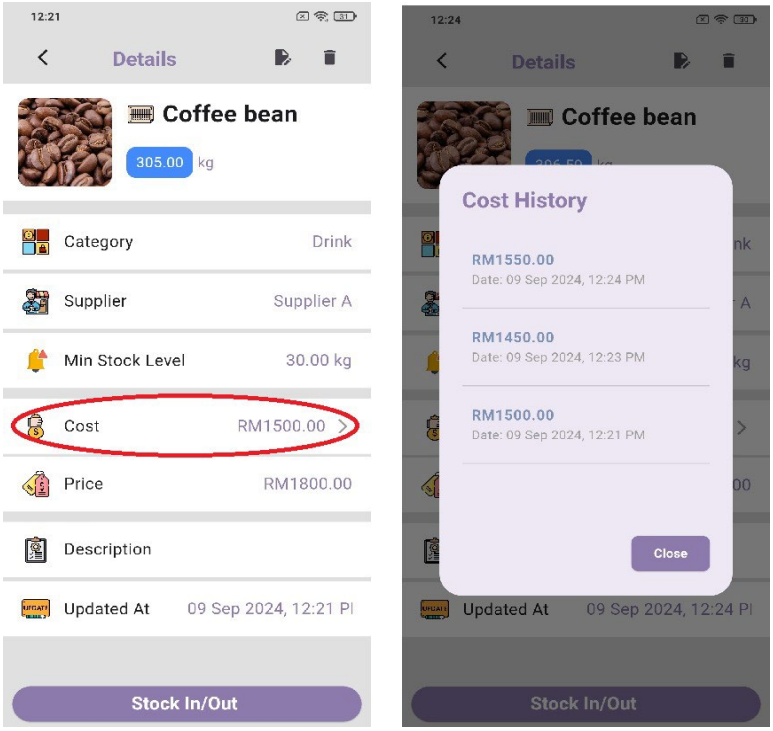


Figure 5.11 Item Details Page

The Item Details Page provides a comprehensive overview of a specific item within the inventory. Here, users can see all relevant details, such as the item's name, quantity, unit of measurement, minimum stock level, cost, price, barcode, and a description. Additionally, next to the item name, there is a barcode icon. When a user clicks on this icon, the barcode image for the item will be displayed, making it easy for scanning or identifying the item quickly.

One important feature on the page is the Cost row. By clicking on this row, the user can access the Cost History of the item. The cost history shows a log of every time the item's cost has been updated, providing a timeline of cost changes. This feature is especially helpful for owners or admins as it allows them to review previous cost adjustments and make more informed decisions about future pricing strategies or purchasing.

At the top right corner of the Item Details Page, two icons are displayed for users with admin privileges: the Edit button and the Delete button. Admins can use the Edit button to modify any

item details and the Delete button to remove the item from the inventory entirely. These options are only available to users with admin rights. Regular users will not see these icons, ensuring the security and integrity of the item data by preventing unauthorized changes or deletions.

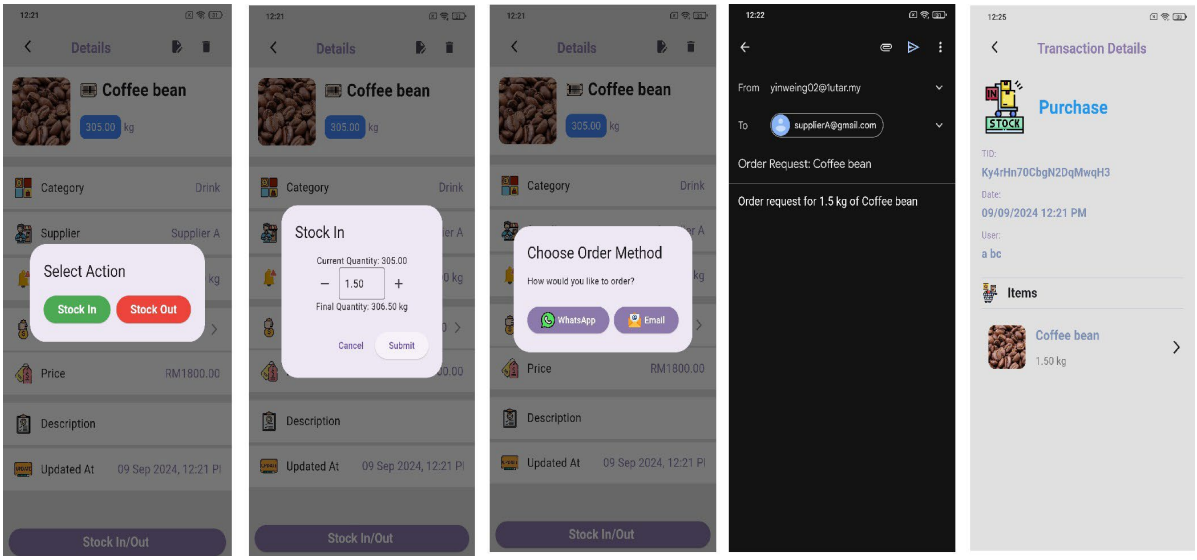


Figure 5.12 Flow of Stock-in Action

At the bottom of the Item Details Page, there is a Stock In/Stock Out button, allowing users to update the inventory for the selected item. Users can choose either to stock in or stock out the item and then enter the quantity accordingly. There are two methods available for stocking in items: via WhatsApp or Email, offering flexibility in how stock updates are managed. The detailed flow of the stock-in process is illustrated in Figure 5.12, which demonstrates the step-by-step process users follow when updating stock levels. Once the stock-in or stock-out action is completed, the system will automatically generate a Transaction Record. This record includes details such as the item name, the type of action (stock in or stock out), the quantity involved, and information about the user who performed the action. This automatic logging of transactions ensures accountability and transparency in the inventory management process.

For Stock Out, the system will verify whether there is sufficient quantity available in the inventory before completing the action. If the requested quantity exceeds the available stock, an error message will alert the user to the insufficient quantity, preventing the stock-out action from proceeding. Similarly, users are required to input valid quantities for both stock-in and stock-out actions. The system will validate these entries to ensure that invalid or erroneous quantities are not submitted, helping maintain the accuracy and integrity of the inventory.

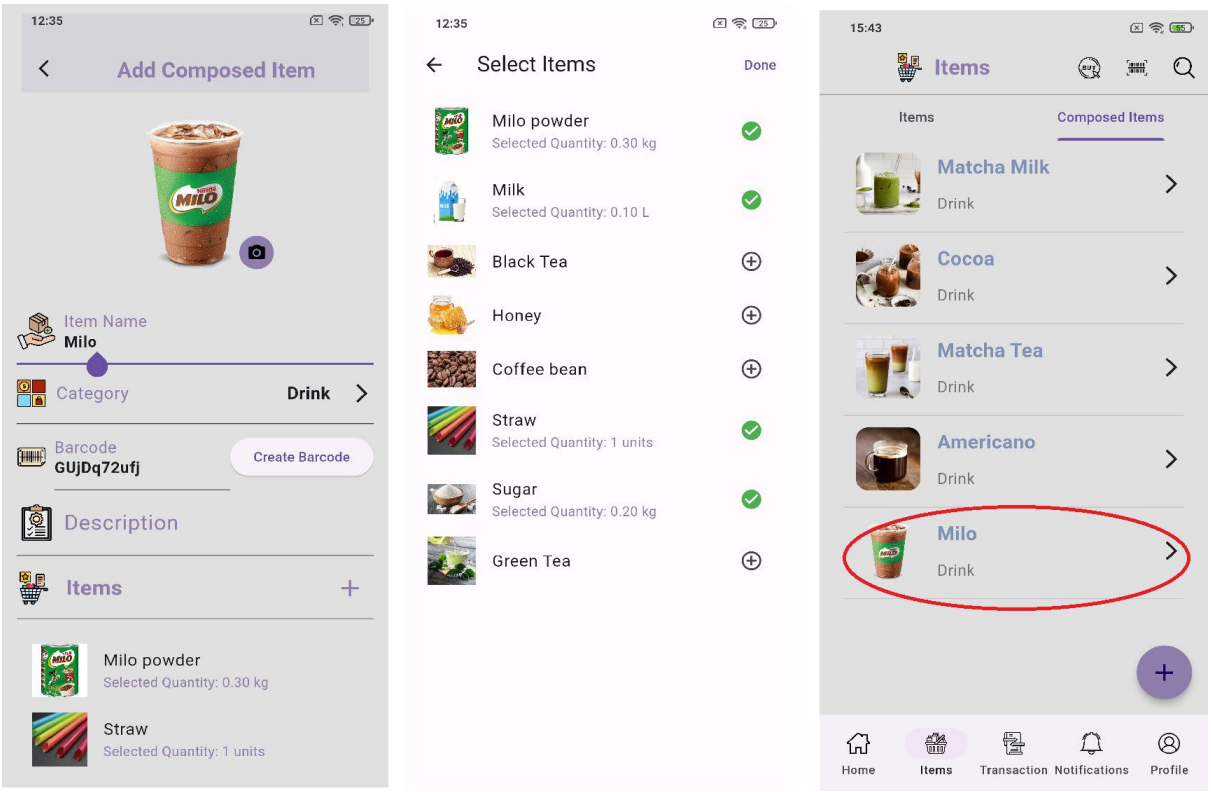


Figure 5.13 Flow of Add Composed Item

The Composed Item Page is designed for managing items that are composed of two or more individual items in the inventory. This feature provides convenience by allowing users to stock out a group of items as a single unit, rather than stocking out each item one by one. To Add a Composed Item, start by navigating to the Composed Item Tab (located on the right side of the Item Page). Here, you will see a list of all composed items in the inventory. To add a new composed item, click the Add button at the bottom of the Composed Item Tab. The process of adding a composed item is similar to adding a regular item. First, the user inputs the composed item's name, selects a category, and assigns a barcode. After this, the user clicks the Add Item button, which redirects them to the Choose Item List Page.

On the Choose Item List Page, the user can select the individual items that will be part of the composed item and specify the required quantity for each. A composed item must consist of at least two individual items. Once the user has selected the necessary items and their quantities, they can click the Done button, which will take them back to the Add Composed Item Page. The selected items will now be displayed, and the user can proceed by clicking the Add Composed Item button to finalize and store the composed item in the inventory.

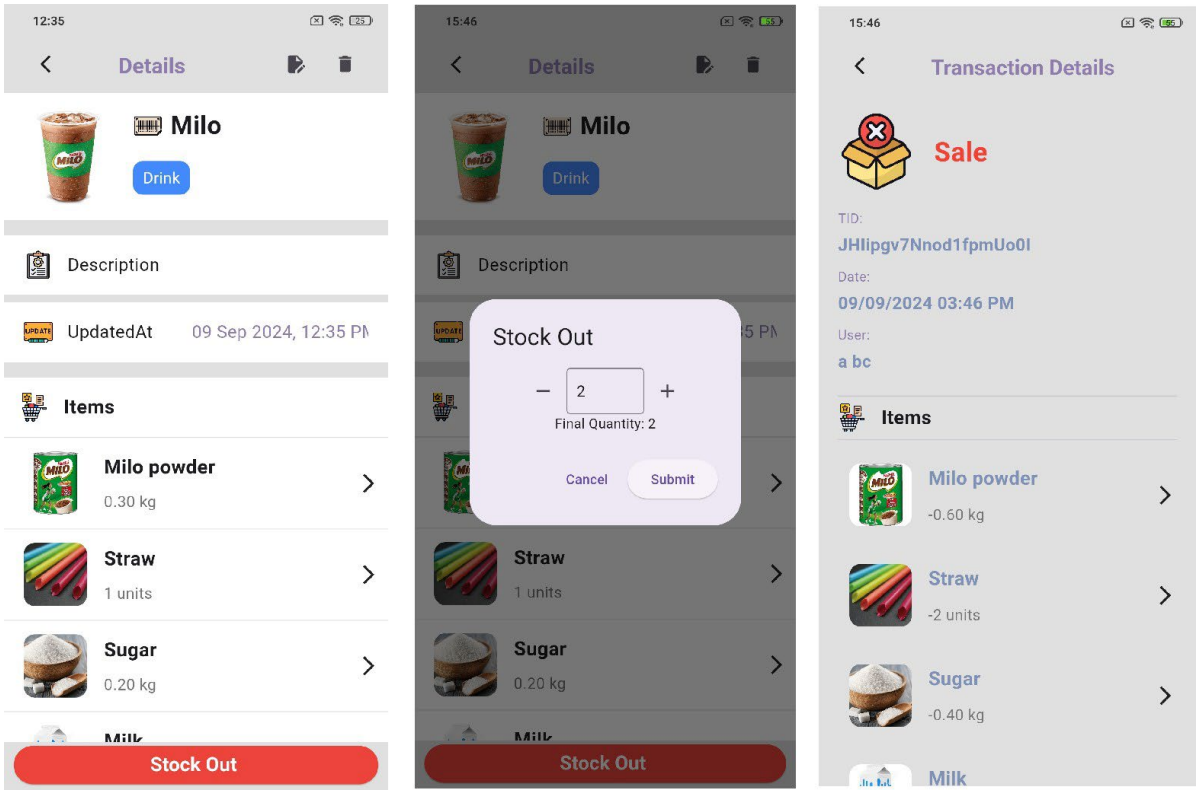


Figure 5.14 Flow of Stock Out Composed Item

On the Composed Item Details Page, all information related to the composed item is displayed, including the list of items it is composed of and their respective quantities. Similar to the regular Item Details Page, users with admin privileges will see the Edit and Delete buttons, allowing them to modify or remove the composed item if necessary. Regular users, on the other hand, do not have access to these options, ensuring that only authorized personnel can alter or delete composed items.

At the bottom of the Composed Item Details Page, there is a Stock Out button. When clicked, the user can specify the quantity of the composed item to stock out. The detailed flow of the stock-out process of composed item is illustrated in Figure 5.14, which demonstrates the step-by-step process users follow when updating stock levels. Once the stock-out action is completed successfully, the system automatically generates a Transaction Record, logging the details of the composed item, the quantity stocked out, and the user who performed the action. The system also checks for insufficient quantity before allowing a stock-out, ensuring that the composed item cannot be stocked out if there are not enough individual items in the inventory.

This process ensures that the integrity of the inventory is maintained, and all transactions are properly recorded for accountability and future reference.

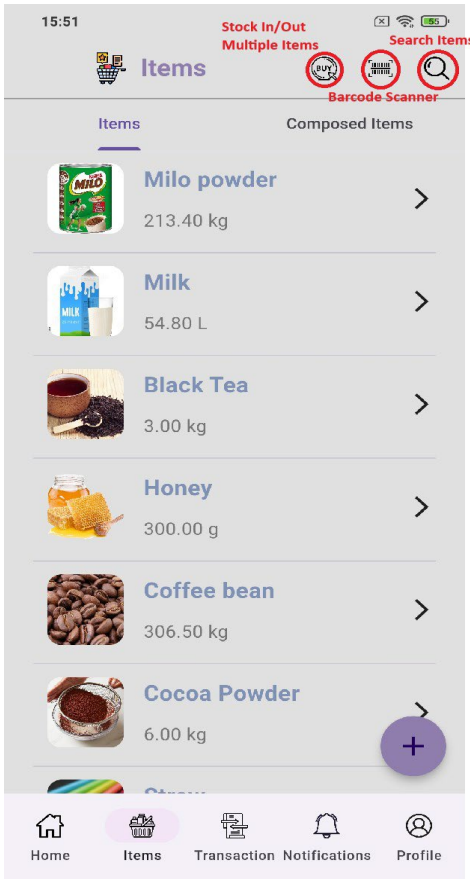


Figure 5.15 Features of Item Page

The Item Page provides several features to help users manage and interact with the inventory effectively. As shown in Figure 5.15, in the upper right corner of the page, there are three icon buttons that enhance the functionality and usability of the page. The first feature is Stock in/out of Multiple Items. It allows users to stock in or stock out multiple items at once. By clicking this button, users can select several items from the inventory and perform stock in or stock out actions on them simultaneously. This is particularly useful when updating stock levels for multiple items in bulk, streamlining inventory management. The middle button opens the Barcode Scanner feature, enabling users to quickly find items by scanning their barcodes. This is a fast and efficient way to locate items in the inventory, as the system will immediately display the corresponding item based on the barcode scanned. This feature simplifies the process of finding items, especially when dealing with a large inventory. The right icon

activates the Search Function. The application provides a live search feature, allowing users to input various details to find specific items. Users can search by item name, category name, barcode, or description. As the user types, the results are dynamically updated, showing matching items in real-time. This feature ensures that users can easily and quickly find the items they are looking for, improving efficiency when managing the inventory.

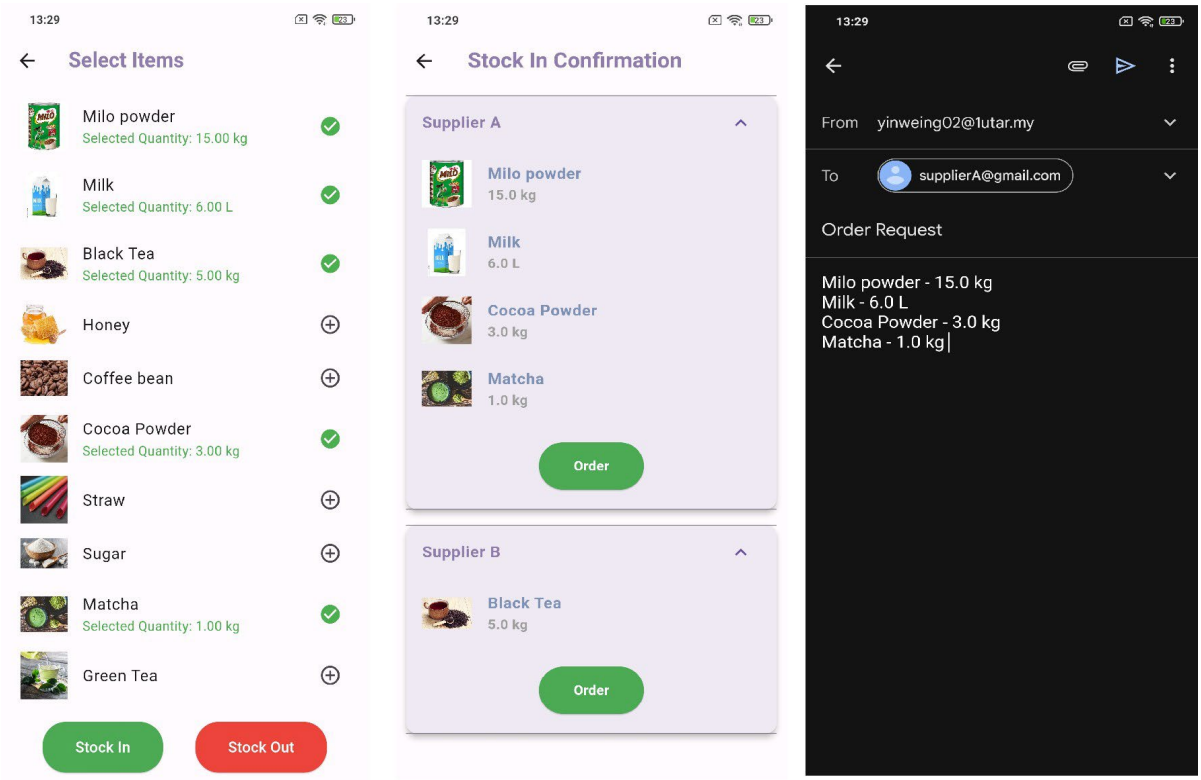


Figure 5.16 Flow of Stock in Process of Multiple Items

To stock in or out multiple items in the system, users can begin by clicking the Stock In/Out of Multiple Items button located at the top right of the Item Page. This action will display a list of all the items in the inventory, allowing the user to select the items that need to be stocked in or out and input the desired quantity for each item. Once the selection is complete, the user can either proceed with stocking in or stocking out the selected items. In the system, users with the user role can stock out items but cannot stock in.

For stocking in items, after confirming the item selection and quantity, the system will redirect the user to a page that organizes the selected items by their respective suppliers. This grouping helps streamline the ordering process. As shown in Figure 5.16, the user will need to click the Order button and choose between ordering via WhatsApp or Email. The system will

automatically generate a message with the details of the items and quantities that need to be ordered. However, due to limitations with Flutter (which doesn't support sending multiple emails or WhatsApp messages to different suppliers simultaneously), users must place orders one supplier at a time. The system ensures that the messages are formatted correctly and ready to send for each supplier, making the process straightforward.

For stocking out items, after selecting the items and quantities, the user will be directed to a page to review the selection. Once the items and quantities are confirmed, the user clicks the Confirm button to proceed with the stock out. The system will also automatically check for any insufficient quantity issues, ensuring that users cannot stock out more than what is available in the inventory.

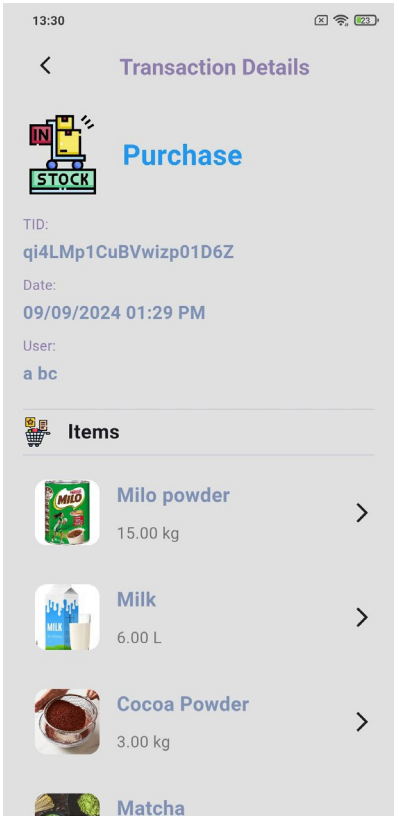


Figure 5.17 Example of Multiple Items Stock-in Transaction

Both stock in and stock out actions will generate a Transaction Record automatically, logging the details of the items, quantities, the type of action, and the user responsible for the transaction. This ensures transparency and accountability in the inventory management process.

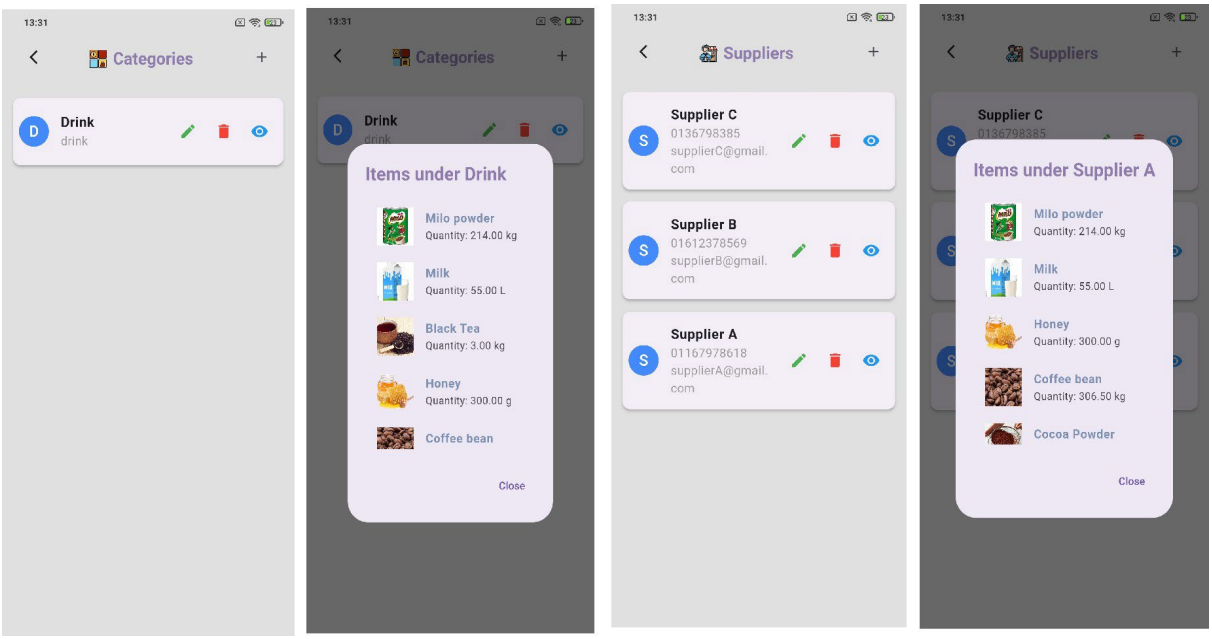


Figure 5.18 Categories Management & Suppliers Management Page

The Categories Management and Suppliers Management pages are crucial features that are accessible only by the admin. These pages allow the admin to maintain and manage the information related to item categories and suppliers within the inventory system. In both the Categories Management Page and the Suppliers Management Page, the admin can edit and delete existing categories or suppliers as well as add new ones. When adding a new category or supplier, the admin needs to input the relevant details to store in the system.

However, there are important restrictions regarding deletion. If a category or supplier is currently associated with any items in the inventory, the system will prevent the admin from deleting it. This ensures that the integrity of the inventory data is maintained, as deleting a category or supplier with associated items would create gaps or inconsistencies in the records. Additionally, these pages feature an eye icon next to each category or supplier. By clicking on this icon, the admin can view a list of all items that are linked to that specific category or supplier. This functionality helps the admin verify the relationships between categories, suppliers, and items before making any changes or deletions. This management capability allows for effective organization and ensures that the inventory system remains accurate and reliable while providing flexibility for the admin to make necessary adjustments.

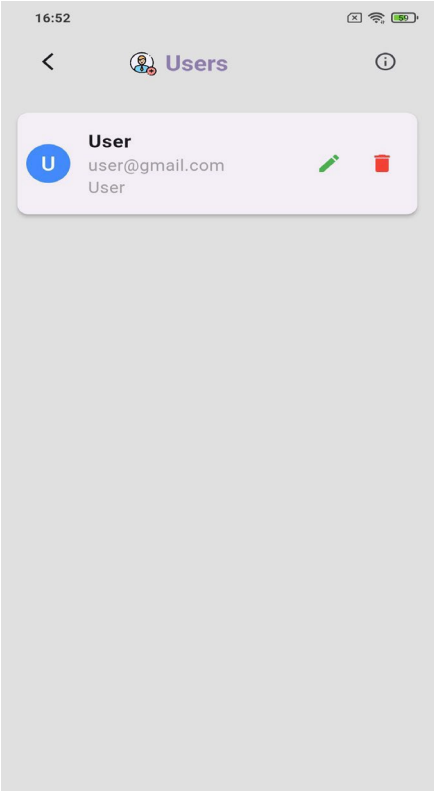


Figure 5.19 User Role Management Page

User role management is a critical component of organizational systems, ensuring that everyone has the appropriate level of access and responsibility. Admins play a critical role in this process, as they are responsible for overseeing and managing user roles within the company. They can view all users associated with the organization, allowing them to maintain an accurate overview of personnel and their respective roles. Admins also have the authority to edit user roles, adjusting permissions and responsibilities as needed based on organizational requirements. Additionally, they can delete roles when necessary, ensuring that users' access remains aligned with their current duties.

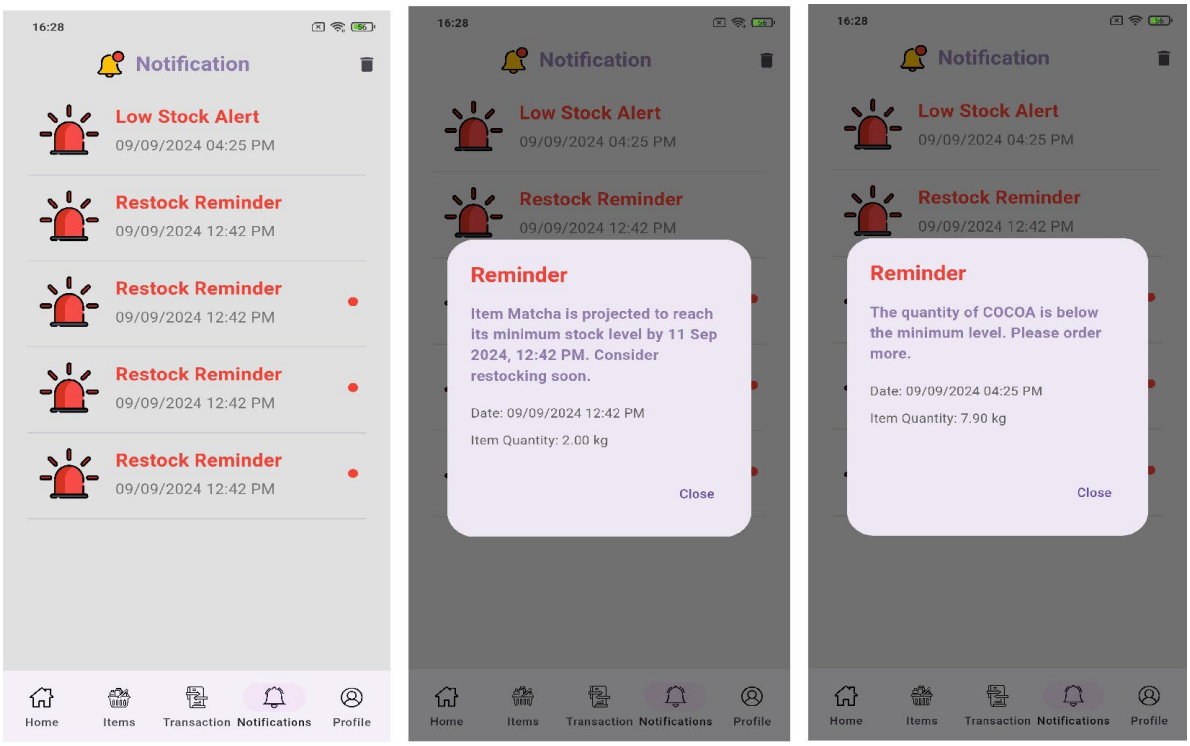


Figure 5.20 Notification Page

The notification page as shown at Figure 5.20 is designed to help admins efficiently manage inventory by providing timely alerts about stock levels. There are two primary types of notifications that admins will receive. The first is the Low Stock Notification, which activates when the quantity of an item falls below the minimum stock level set during the item's addition. This alert ensures that admins are aware of items that need replenishing to prevent stockouts. The second type is the Stock Level Warning, which is generated by the app's algorithm. This notification predicts when the quantity of an item will fall below the minimum stock level within the next three days, allowing admins to take proactive measures to avoid running out of stock. These notifications are exclusively accessible to admins, providing them with the critical information needed to maintain optimal inventory levels and streamline stock management.

Figure 5.21 Profile Page

The profile page provides a centralized view of essential information, displaying the company name, company join code, and user details. This page allows users to manage both their personal and company information efficiently. At the top right of the profile page, there are three functional buttons: the first, Edit Profile and Company Information, enables users to update their personal details and modify company-related settings. The second button, Change Account Password, allows users to securely update their password to maintain account security. The third button, Sign Out, provides an option for users to log out of their account, ensuring their session is closed securely. Together, these features offer a streamlined approach to managing user profiles and account security.

5.5 Implementation Issues and Challenges

One of the main challenges when developing this project was database performance and scalability. As the app grows, the amount of data related to users, inventory, and notifications will increase significantly. Managing this data effectively, especially when handling concurrent

operations from multiple users, can put pressure on the database. Firebase, while powerful, can face performance bottlenecks under heavy use, especially when retrieving large data sets or performing complex queries. Optimizing database structure, indexing, and query management are critical to ensuring fast and reliable performance as the user base and data volume expand.

Another challenge was ensuring a smooth user experience in offline scenarios. Because the app relies heavily on Firebase for real-time data access, users may experience difficulties in areas with poor or unstable network connectivity. If the app does not handle offline scenarios correctly, it may result in incomplete data, failed operations, or inaccessible key features such as inventory management or role assignments. Implementing offline functionality, where the app can temporarily store changes and synchronize data when connectivity is restored, is critical to maintaining a positive user experience when connectivity is unreliable.

Finally**,** ensuring seamless integration between different system components can present challenges. The app's various modules, such as user role management, inventory, and notifications, must work together cohesively to provide a seamless experience. Any inconsistency between modules can lead to functional breakdowns, security loopholes, or confusion in user roles and permissions. Thorough testing and quality assurance across all modules will be critical to ensure these systems work harmoniously.

5.6 Concluding Remark

In conclusion, the successful implementation of this inventory management system involved various stages of development, including hardware and software setup, as well as the integration of key functionalities such as user management, stock tracking, and notification systems. Leveraging Flutter for the front-end and Firebase for the back-end services, the application offers a seamless and efficient experience for managing inventory in real-time. The implementation effectively supports user roles, item categories, supplier management, and stock operations, providing a comprehensive solution to address the challenges of inventory management.

Despite facing issues related to database performance, scalability, and offline access, strategies were employed to optimize the system's architecture and functionality. The integration of real-

time notifications and predictive alerts further enhances the system's ability to prevent stock shortages, ensuring smooth business operations.

Overall, the development of this inventory management system represents a significant step forward in streamlining inventory operations and enhancing user management. Despite the challenges faced, including database performance and offline functionality, the implemented solutions ensure a robust and reliable system. The integration of real-time notifications and predictive alerts helps users stay informed and proactive, effectively addressing potential stock issues before they arise. As the system continues to evolve, ongoing improvements and adaptations will further refine its capabilities, supporting businesses in maintaining efficient and accurate inventory management while accommodating future growth and technological advancements.

Chapter 6

System Evaluation and Discussion

6.1 System Testing and Performance Metrics

The testing was performed using black box testing, which focuses on validating the functionality of application based on its requirements, without accessing the internal code structure. The goal of this testing was to ensure that the system behaves as expected from a user's perspective, handling both normal flows and error scenarios.

To implement this, use case testing was employed as the primary technique. Each use case describes a sequence of action or steps that users might perform in the application. By defining these use cases, we were able to identify and test both the primary workflows and alternative scenarios (error cases). This allowed for a comprehensive evaluation of the system's response to different inputs, ensuring that all possible interactions with the system were covered.

Use Case ID	UC001	
Use Case	User Registration	
Purpose	To allow a new user to create an account in the application.	
Actor	The user (new or unregistered user) who wants to create an account in the system.	
Trigger	The user navigates to the registration page and initiates the registration process by clicking the "Sign up" button.	
Precondition	The registration page should be available and accessible.	
Scenario Name	**Step**	**Action**
Main	1	Open the registration page.
	2	System requests input username, email, password, and confirm password.
	3	User key in required field (e.g. username, email, password, and confirm password).
	4	Click sign up button
	5	System validates the input data.
	6	System creates an account and stores it into database.

	7	System navigate user into selection company page.
Alternative Flow - Empty Input	3.1	User leaves empty for input fields.
	3.2	System display error message “Please fill in all fields!”
Alternative Flow - Invalid Email Format	3.1	User input invalid email format.
	3.2	System display error message “The email address is badly formatted.”
Alternative Flow - Different Password & Confirm Password	3.1	User input different password and confirm password.
	3.2	System display error message “Password don’t match!”
Alternative Flow - Registered Email	3.1	User input registered email.
	3.2	System display error message “Email is already registered.”
Rules	User does not have an account.	

Table 6.1 User Registration

Use Case ID	UC002	
Use Case	Create a New Company	
Purpose	To allow a newly registered user to create a company	
Actor	The user (new or unregistered user) who wants to create a company in the system	
Trigger	The user selects the “Create a Company” option after completing registration.	
Precondition	The user must have successfully registered and be on the company selection page.	
Scenario Name	**Step**	**Action**
Main	1	Open the create company page.
	2	System requests for company name and notes (optional).
	3	User key in company name and notes.
	4	Click create company button.
	5	System checks the validation of name and assign Admin roles to the user.

	6	System navigate user into admin home page based on the company.
Alternative Flow - Empty Input	3.1	User leaves empty for company name field.
	3.2	System display error message “Please input company name.”
Alternative Flow - Registered Company Name	3.1	User input registered company name.
	3.2	System display error message “Company name already exists.”
Rules	User successfully register an account.	

Table 6.2 Create a New Company

Use Case ID	UC003	
Use Case	Join an Existing Company	
Purpose	To allow a newly registered user to join an existing company using a join code.	
Actor	The user (new or unregistered user) who wants to join an existing company.	
Trigger	The user selects the “Join a Company” option after registration.	
Precondition	The user must have received a valid join code from a company administrator.	
Scenario Name	**Step**	**Action**
Main	1	Open the join company page.
	2	System requests for company join code.
	3	User key in the join code providing by admin.
	4	Click join company button.
	5	System checks the validation of join code and assign User role to the user.
	6	System navigate user into user home page based on the company.
Alternative Flow - Empty Input	3.1	User leaves empty for join code field.
	3.2	System displays error message “Invalid join code. Please check and try again”

Alternative Flow - Wrong Join Code	3.1	User input wrong join code.
	3.2	System displays error message “Invalid join code. Please check and try again”
Rules	User successfully register an account.	

Table 6.3 Join an Existing Company

Use Case ID	UC004	
Use Case	User Login	
Purpose	To authenticate the user by verifying their credentials and granting access to the application.	
Actor	The user (registered user) who wants to log into the system.	
Trigger	The user navigates to the login page and initiates the login process by clicking the “Sign In” button after entering their credentials.	
Precondition	The user must already have registered an account in the system.	
Scenario Name	**Step**	**Action**
Main	1	Open the login page.
	2	System requests for input email and password
	3	User key in a registered email and correct password.
	4	Click sign in button
	5	System navigate user into home page based on their role.
Alternative Flow - Invalid Email Format	3.1	User input invalid email format.
	3.2	System display error message “An unexpected error occurred. Please try again.”
Alternative Flow - Wrong Email or Password	3.1	User input wrong email or wrong password.
	3.2	System display error message “Incorrect email or password”
Rules	User has an account.	

Table 6.4 User Login

Use Case ID	UC005
Use Case	Admin Home Page Navigation

Purpose	To verify that the admin can access and navigate all sections of the home page.	
Actor	Admin	
Trigger	The admin logs in and is directed to the Admin Home Page.	
Precondition	The admin is logged in with appropriate permissions.	
Scenario Name	**Step**	**Action**
Main	1	Admin logs in and is directed to the Admin Home Page.
	2	Verify that the Sign Out is present and functional.
	3	Verify that the Company Summary is displayed correctly.
	4	Verify that the Category Management, Supplier Management, and User Role Management buttons are functional and lead to the correct pages
	5	Verify that the Recent Transactions section displays the five most recent transactions.
	6	Verify that the navigation bar has functional tabs: Home, Item, Transaction, Notification, and Profile
Alternative Flow – Sign Out	2.1	Admin clicks Sign Out, and the system logs the user out successfully.
Rules	Admin successfully login company account.	

Table 6.5 Admin Home Page Navigation

Use Case ID	UC006	
Use Case	Add New Item to Inventory	
Purpose	To ensure the admin can successfully add new items to the inventory system.	
Actor	Admin	
Trigger	Admin clicks the Add New Item button.	
Precondition	Admin must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.
	2	Click the “Plus” button in the bottom right page.

	3	User input all required details (item name, quantity, unit, cost, price, min stock level, barcode, category, and supplier).
	4	Click add item button to add the item to the inventory.
	5	System generates a transaction record with details of the action.
	6	Verify that the item appears in the Item Tab.
	7	Click the list tile of item to verify the item details is correct.
Alternative Flow - Image	3.1	User don't input image of item.
	3.2	System display error dialog and message "Image is required."
Alternative Flow - Empty Input	3.1	User leaves any required field blank.
	3.2	System display error dialog and message "All fields are required."
Alternative Flow - Cost & Price	3.1	User input cost lower than price.
	3.2	System display error dialog and message "Price cannot be less than cost."
Alternative Flow - Registered Item Name	3.1	User input a registered item name.
	3.2	System display error dialog and message "An item with the same name already exist. Please choose a different name."
Rules	User must be an Admin Role of the company.	

Table 6.6 Add New Item to Inventory

Use Case ID	UC007	
Use Case	View Item Details	
Purpose	To verify that the user can view all relevant details of a specific item in the inventory.	
Actor	Admin/User	
Trigger	User selects an Item from the inventory list.	
Precondition	User must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Details Page.

	2	System displays item details (name, quantity, unit of measurement, minimum stock level, cost, price, barcode, description, etc.).
	3	User clicks the barcode icon.
	4	System displays the barcode image.
Rules	All fields must display correct item details.	

Table 6.7 View Item Details

Use Case ID	UC008	
Use Case	View Cost History	
Purpose	To ensure the user can view the cost history of the item.	
Actor	Admin/User	
Trigger	User clicks the Cost row.	
Precondition	User must have access to the Item Details Page	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Details Page.
	2	System displays item details (name, quantity, unit of measurement, minimum stock level, cost, price, barcode, description, etc.).
	3	User clicks on the “Cost” row.
	4	System displays the item’s cost history (previous and current costs with timestamps).
Rules	Cost history must be displayed correctly with each cost update. The system should log every cost change with a timestamp.	

Table 6.8 View Cost History

Use Case ID	UC009
Use Case	Edit Existing Item
Purpose	To ensure the admin can successfully edit the details of an existing item in the inventory.
Actor	Admin
Trigger	Admin selects an item to edit.

Precondition	Admin must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.
	2	Select an existing item from the Item Tab and navigate to Item Details Page.
	3	Click Edit and update necessary field.
	4	Click Save Changes button to update the item details.
	5	Verify that the updated details are reflected in the Item Details Page.
Alternative Flow - Image	3.1	User leaves image fields of the item blank.
	3.2	System display error dialog and message “Image is required.”
Alternative Flow - Empty Input	3.1	User leaves any required field blank.
	3.2	System display error dialog and message “All fields are required.”
Rules	User must be an Admin Role of the company.	

Table 6.9 Edit Existing Item

Use Case ID	UC010	
Use Case	Delete Item from Inventory	
Purpose	To ensure the admin can delete an item from the inventory, unless the item is part of a composed item.	
Actor	Admin	
Trigger	Admin selects an item to delete.	
Precondition	Admin must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.
	2	Select an existing item from the Item Tab and navigate to Item Details Page.
	3	Click Delete button.
	4	System prompts the admin to confirm the deletion.
	5	Confirm deletion of the item.
	6	System checks if the item is part of a composed item.

	7	System deletes the item and updates the Item Tab.
Alternative Flow – Part of Composed Item	6.1	System checks if the item is part of a composed item.
	6.2	System displays an error message “Item cannot be deleted because it is a part of a composed item.”
Rules	User must be an Admin Role of the company.	

Table 6.10 Delete Item from Inventory

Use Case ID	UC011	
Use Case	Stock In/Stock Out Item	
Purpose	To ensure the admin can update inventory stock levels by stocking in or out items.	
Actor	Admin	
Trigger	Admin clicks the Stock In/Stock Out button on the Item Details Page.	
Precondition	Admin must be logged in and have access to the Stock In/Out feature.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.
	2	Select an existing item from the Item Tab and navigate to Item Details Page.
	3	Admin clicks Delete Stock In/Out button.
	4	Admin selects either Stock in or Stock out.
	5	User inputs the quantity.
	6	User chooses methods for stock in item if Stock In.
	7	The system updates the inventory stock accordingly.
	8	System generates a transaction record with details of the action.
Alternative Flow – Insufficient stock available	5.1	User inputs a quantity larger than the available stock for Stock Out.
	5.2	System displays an error dialog and message “Insufficient quantity for stock out.”

Rules	User must be an Admin Role of the company.

Table 6.11 Stock In/Out Item

Use Case ID	UC012	
Use Case	User Stock Out Item	
Purpose	To ensure the user can update inventory stock levels by stocking out items.	
Actor	User	
Trigger	User clicks the Stock In/Stock Out button on the Item Details Page.	
Precondition	User must be logged in and have access to the Stock In/Out feature.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Item Page.
	2	Select an existing item from the Item Tab and navigate to Item Details Page.
	3	User clicks Delete Stock In/Out button.
	4	Admin click the Stock out button.
	5	User inputs the quantity.
	6	The system updates the inventory stock accordingly.
	7	System generates a transaction record with details of the action.
Alternative Flow – Insufficient stock available	5.1	User inputs a quantity larger than the available stock for Stock Out.
	5.2	System displays an error dialog and message “Insufficient quantity for stock out.”
Rules	User is a user role of the company.	

Table 6.12 User Stock Out Item

Use Case ID	UC013
Use Case	Add Composed Item
Purpose	To verify that an admin can successfully add a composed item to the inventory.

Actor	Admin	
Trigger	Admin clicks the “Plus” button in the Composed Item Tab.	
Precondition	Admin must be logged in and have access to the Composed Item Tab.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Composed Item Tab.
	2	Admin clicks the “Plus” button in bottom right of page.
	3	Admin inputs required details (composed item name, category, barcode, and description).
	4	Admin selects individual items and specifies quantities from the Choose Item List Page.
	5	Admin clicks the Done button and verifies the composed items.
	6	Admin clicks the “Add Composed Item” button to save the composed item.
	7	System generates a transaction record with details of the action.
	8	Verify that the composed item appears in the Composed Item Tab.
Alternative Flow - Empty Image	3.1	Admin leaves the image field blank.
	3.2	System displays an error dialog and message “Image is required.”
Alternative Flow – Empty Input	3.1	User fails to input all required fields.
	3.2	System displays an error dialog and message “Please fill all the fields except description.”
Alternative Flow – No Item Selection	4.1	Admin don’t select more than one item.
	4.2	System displays an error dialog and message “Please select at least two items.”
Rules	User must be an Admin role of the company.	

Table 6.13 Add Composed Item

Use Case ID	UC014	
Use Case	View Composed Item Details	
Purpose	To ensure the user can view detailed information about a composed item, including its components.	
Actor	Admin/User	
Trigger	User selects a composed item from the Composed Item Tab.	
Precondition	User must be logged in and have access to the Composed Item Tab.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Composed Item Tab.
	2	Select an existing composed item from the Composed Item Tab and navigate to Composed Item Details Page.
	3	The system displays all the details and showing all components and their respective quantities.
Rules	The system must display the correct individual items and their quantities in the composed item.	

Table 6.14 View Composed Item Details

Use Case ID	UC015	
Use Case	Edit Composed Item	
Purpose	To verify that only admins can edit the details of a composed item.	
Actor	Admin	
Trigger	Admin clicks the “Edit” button on the Composed Item Details Page.	
Precondition	Admin must be logged in and have access to the Composed Item Tab.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Composed Item Tab.
	2	Select an existing composed item from the Composed Item Tab and navigate to Composed Item Details Page.
	3	Admin clicks the “Edit” button.
	4	Admin modifies the composed item’s details (name, items, or quantities).
	5	Admin clicks “Save” to apply the changes.

	6	System updates the composed item and reflects the changes in the Composed Item Details Page.
Alternative Flow - Empty Image	4.1	Admin leaves the image field blank.
	4.2	System displays an error dialog and message “Image is required.”
Alternative Flow – Empty Input	4.1	User fails to input all required fields.
	4.2	System displays an error dialog and message “Please fill all the fields except description.”
Alternative Flow – No Item Selection	4.1	Admin don’t select more than one item.
	4.2	System displays an error dialog and message “Please select at least two items.”
Rules	Only admins should see and use the “Edit” Button	

Table 6.15 Edit Composed Item

Use Case ID	UC016	
Use Case	Delete Composed Item from Inventory	
Purpose	To verify that only admins can delete a composed item from the inventory.	
Actor	Admin	
Trigger	Admin clicks the “Delete” button on the Composed Item Details Page.	
Precondition	Admin must be logged in and have access to the Composed Item Tab.	
Scenario Name	**Step**	**Action**
Main	1	Navigate to the Composed Item Tab.
	2	Select an existing item from the Composed Item Tab and navigate to Item Details Page.
	3	Admin clicks the “Delete” button.
	4	System prompts the admin to confirm the deletion.
	5	Admin confirms the deletion.

	6	System deletes the composed item and updates the Composed Item Tab.
Rules	User must be an Admin Role of the company.	

Table 6.16 Delete Composed Item from Inventory

Use Case ID	UC017	
Use Case	Stock Out Composed Item	
Purpose	To ensure that users can stock out a composed item, and system appropriately deducts the correct quantities from the individual items.	
Actor	Admin/User	
Trigger	User clicks the “Stock Out” button for a composed item.	
Precondition	Users must be logged in and have access to the Composed Item Details Page.	
Scenario Name	**Step**	**Action**
Main	1	User clicks the “Stock Out” button on the Composed Item Details Page.
	2	User inputs the quantity of the composed item to stock out.
	3	The system checks whether sufficient quantities of individual items are available.
	4	System deducts the specified quantities from the individual items.
	5	The system generates a transaction record with details of the composed item stock-out action.
Alternative Flow - Insufficient Quantity	2.1	The user inputs a quantity that exceeds the available quantities of individual items.
	2.2	System displays an error dialog and message “One or more items do not have enough quantity to perform this stock out.”
Rules	The system must check the availability of each individual item before allowing the stock-out action.	

Table 6.17 Stock Out Composed Item

Use Case ID	UC018	
Use Case	Stock In Multiple Items	
Purpose	To verify that an admin can stock in multiple items simultaneously.	
Actor	Admin	
Trigger	Admin clicks the “Stock In/Out of Multiple Items” button and chooses to stock in.	
Precondition	Admin must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Item Page
	2	Admin clicks on the “Stock In/Out of Multiple Items” button.
	3	Admin selects multiple items from the inventory and specifies quantities for stock-in.
	4	Admin clicks the “Stock In” button to initiate the stock-in process.
	5	The system groups the selected items by supplier and redirects them to the order page.
	6	Admin clicks “Order” and chooses to order via WhatsApp or Email.
	7	System generates a message with order details.
	8	Admin sends the order for each supplier.
	9	The system generates the transaction record and stock level of items are updated.
Alternative Flow – Empty Item Selection	2.1	Admin attempts to stock in without selecting any items.
	2.2	System displays an error dialog and message “Please select at least one item before proceeding.”
Rules	The system should allow users to stock in only valid quantities.	

Table 6.18 Stock In Multiple Item

Use Case ID	UC019
Use Case	Stock Out Multiple Items
Purpose	To verify that users can stock out multiple items simultaneously.
Actor	Admin/User

Trigger	User clicks the "Stock In/Out of Multiple Items" button and chooses to stock out.	
Precondition	Users must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	User navigates to the Item Page
	2	User clicks on the "Stock In/Out of Multiple Items" button.
	3	User selects multiple items from the inventory and specifies quantities for stock out.
	4	User clicks the "Stock Out" button to review the selection.
	5	Use confirms the stock-out quantities and clicks "Confirm" button.
	6	System checks for sufficient stock of each item.
	7	System deducts the quantities from the inventory.
	8	The system generates the transaction record and stock level of items are updated.
Alternative Flow – Insufficient Quantity	3.1	Admin attempts to stock out more than the available quantity.
	3.2	System displays an error dialog and message "Selected quantity cannot exceed the available quantity."
Alternative Flow – Empty Item Selection	4.1	Admin attempts to stock out without selecting any items.
	4.2	System displays an error dialog and message "Please select at least one item before proceeding."
Rules	The system must prevent users from stocking out more than what is available in the inventory.	

Table 6.19 Stock Out Multiple Items

Use Case ID	UC020
Use Case	Barcode Scanner
Purpose	To verify that users can search for items or composed items using the barcode scanner and view the corresponding item details.
Actor	Admin/User
Trigger	User clicks the "Barcode Scanner" button and scans a barcode.

Precondition	Users must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	User navigates to the Item Page
	2	User clicks on the “Barcode Scanner” button.
	3	User scans a barcode.
	4	The system identifies the corresponding item and displays its details.
Alternative Flow – Invalid or Unrecognized Barcode	3.1	User scans an invalid or unrecognized barcode.
	3.2	System displays an error message “No Items found.”
Rules	The system must match the scanned barcode with the correct item in the inventory.	

Table 6.20 Barcode Scanner

Use Case ID	UC021	
Use Case	Search Function	
Purpose	To verify that users can search for items or composed items by name, category, barcode, or description using the live search feature.	
Actor	Admin/User	
Trigger	User inputs search criteria into the search bar on the Item Page.	
Precondition	User must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	User navigates to the Item Page
	2	User inputs a search term in the search bar.
	3	System dynamically updates the search results as the user types.
	4	User clicks on an item from the search results to view its details.
Alternative Flow – Invalid Search Term	3.1	User enters a search term that does not match any items.
	3.2	System displays an error message “No Items found.”

Rules	The system must match the scanned barcode with the correct item in the inventory.

Table 6.21 Search Function

Use Case ID	UC022	
Use Case	Add New Category	
Purpose	To verify that the admin can successfully add a new category to the inventory system.	
Actor	Admin	
Trigger	Admin clicks the “Add New Category” button.	
Precondition	User must be logged in and have access to the Item Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Categories Management Page.
	2	Admin clicks on the “Add New Category” button.
	3	Admin inputs the required fields (category name and description).
	4	Admin clicks “Add” button to add the new category.
	5	Verify that the category appears in the category list.
Alternative Flow – Empty Input	3.1	Admin leaves the required fields blank.
	3.2	System displays an error message “Both fields are required.”
Alternative Flow – Registered Category Name	3.1	Admin input a registered category name.
	3.2	System displays an error message “Category name already exists.”
Rules	System should only allow valid inputs for the category name.	

Table 6.22 Add New Category

Use Case ID	UC023
Use Case	Edit Category Details
Purpose	To verify that the admin can successfully edit an existing category to the inventory system.
Actor	Admin
Trigger	Admin clicks the “Edit Category” button.

Precondition	User must be logged in and have access to the Categories Management Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Categories Management Page.
	2	Admin clicks on the “Edit Category” button.
	3	Admin inputs update details in the required fields (category name and description).
	4	Admin clicks “Update” button to update the category details.
	5	Verify that the update category details appear in the category list.
Alternative Flow – Empty Input	3.1	Admin leaves the required fields blank.
	3.2	System displays an error message “Both fields are required.”
Alternative Flow – Registered Category Name	3.1	Admin input a registered category name.
	3.2	System displays an error message “Category name already exists.”
Rules	System should only allow valid inputs for the category name.	

Table 6.23 Edit Category Details

Use Case ID	UC024	
Use Case	Delete Category with Associated Items	
Purpose	To ensure the system prevents the deletion of a category that is associated with any items.	
Actor	Admin	
Trigger	Admin attempts to delete a category	
Precondition	The category is associated with one or more items in the inventory.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Categories Management Page.
	2	Admin clicks on the “Delete Category” button for a category that is associated with items.
	3	The system displays an error message “Cannot delete this category because there are items associated with it”.
Rules	System should prevent deletion if the category is linked to any items.	

Table 6.24 Delete Category with Associated Items

Use Case ID	UC025	
Use Case	View Items Linked to a Category	
Purpose	To verify that the admin can view all items associated with a specific category.	
Actor	Admin	
Trigger	Admin click the “Eye” icon next to a category.	
Precondition	Admin must be logged in and have access to the Categories Management Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Categories Management Page.
	2	Admin clicks on the “Eye” button.
	3	The system displays a list of all items linked to the selected category.
Rules	The system must correctly display all items associated with the selected category.	

Table 6.25 View Items Linked to a Category

Use Case ID	UC026	
Use Case	Add New Supplier	
Purpose	To verify that the admin can add a new supplier to the system.	
Actor	Admin	
Trigger	Admin click the “Add New Supplier” button.	
Precondition	Admin must be logged in and have access to the Suppliers Management Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Suppliers Management Page.
	2	Admin clicks on the “Add New Supplier” button.
	3	Admin inputs the supplier details. (supplier name, phone number, and email)
	4	Admin clicks “Add” button to add the new supplier.
	5	Verify that the supplier appears in the supplier list.

Alternative Flow – Empty Input	3.1	Admin leaves the required fields blank.
	3.2	System displays an error message “All fields are required.”
Alternative Flow – Registered Supplier Name	3.1	Admin input a registered supplier name.
	3.2	System displays an error message “Supplier name already exists.”
Alternative Flow – Invalid Email Format	3.1	Admin input invalid email format.
	3.2	System displays an error message “Please enter a valid name address.”
Rules	All required fields must be filled out before saving.	

Table 6.26 Add New Supplier

Use Case ID	UC027	
Use Case	Edit Supplier Details	
Purpose	To verify that the admin can successfully edit an existing supplier to the inventory system.	
Actor	Admin	
Trigger	Admin clicks the “Edit Supplier” button.	
Precondition	User must be logged in and have access to the Suppliers Management Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Suppliers Management Page.
	2	Admin clicks on the “Edit Supplier” button.
	3	Admin inputs update details in the required fields (supplier name, phone number and email).
	4	Admin clicks “Update” button to update the category details.
	5	Verify that the update category details appear in the category list.
Alternative Flow – Empty Input	3.1	Admin leaves the required fields blank.
	3.2	System displays an error message “All fields are required.”
Alternative Flow – Registered Supplier Name	3.1	Admin input a registered supplier name.
	3.2	System displays an error message “Supplier name already exists.”

Alternative Flow – Invalid Email Format	3.1	Admin input invalid email format.
	3.2	System displays an error message “Please enter a valid name address.”
Rules	System should only allow valid inputs for the category name.	

Table 6.27 Edit Supplier Details

Use Case ID	UC028	
Use Case	Delete Supplier with Associated Items	
Purpose	To ensure the system prevents the deletion of a supplier that is associated with any items.	
Actor	Admin	
Trigger	Admin attempts to delete a supplier	
Precondition	The supplier is associated with one or more items in the inventory.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Suppliers Management Page.
	2	Admin clicks on the “Delete Supplier” button for a supplier that is associated with items.
	3	The system displays an error message “Cannot delete this supplier because there are items associated with it”.
Rules	System should prevent deletion if the supplier is linked to any items.	

Table 6.28 Delete Supplier with Associated Items

Use Case ID	UC029	
Use Case	View Items Linked to a Supplier	
Purpose	To verify that the admin can view all items associated with a specific supplier.	
Actor	Admin	
Trigger	Admin click the “Eye” icon next to a supplier.	
Precondition	Admin must be logged in and have access to the Suppliers Management Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Suppliers Management Page.

	2	Admin clicks on the “Eye” button.
	3	The system displays a list of all items linked to the selected supplier.
Rules	The system must correctly display all items associated with the selected supplier.	

Table 6.29 View Items Linked to a Supplier

Use Case ID	UC030	
Use Case	User Role Management	
Purpose	To verify that the admin can manage user roles effectively by editing or deleting roles.	
Actor	Admin	
Trigger	Admin selects a user and attempts to change or delete their role.	
Precondition	Admin must be logged in and have access to the User Role Management Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the User Role Management Page.
	2	Admin selects a user to edit their role.
	3	Admin modifies the user’s role and clicks save button.
	4	System updates the user’s role.
Rules	All updates to roles must be accurately reflected in the system.	

Table 6.30 User Role Management

Use Case ID	UC031	
Use Case	Low Stock Notification	
Purpose	To ensure the system correctly notifies the admin when an item’s stock level falls below the minimum threshold.	
Actor	Admin	
Trigger	The system detects that an item’s stock level is below the minimum stock level.	
Precondition	Admin must be logged in and have access to the Notification Page.	
Scenario Name	**Step**	**Action**

Main	1	Admin navigates to the Notification Page.
	2	Admin views a Low Stock Notification for an item.
	3	Admin verifies that the notification accurately reflects the item's low stock status.
Rules	The system must trigger a Low Stock Notification when the stock falls below the set minimum level.	

Table 6.31 Low Stock Notification

Use Case ID	UC032	
Use Case	Stock Level Predict Notification	
Purpose	To verify that the system generates a Stock level Warning notification when it predicts an item's stock will fall below the minimum level within three days.	
Actor	Admin	
Trigger	The system's algorithm predicts a stock level issue.	
Precondition	Admin must be logged in and have access to the Notification Page.	
Scenario Name	**Step**	**Action**
Main	1	Admin navigates to the Notification Page.
	2	Admin views a Stock Level Notification for an item.
	3	Admin verifies that the warning is accurate and reflects the item's predicted low stock.
Rules	The system must generate a notification if it predicts that stock levels will drop below the minimum threshold within three days.	

Table 6.32 Stock Level Predict Notification

Use Case ID	UC033
Use Case	Edit Profile and Company Information
Purpose	To ensure that users can edits their profile and company information successfully.
Actor	Admin/User
Trigger	User clicks the "Edit Profile and Company Information" button on the Profile Page.

Precondition	Users must be logged in and have access to the Profile Page.	
Scenario Name	**Step**	**Action**
Main	1	User navigates to the Profile Page.
	2	User clicks on the “Edit Profile and Company Information” button.
	3	User updates their personal or company details.
	4	User clicks “Save” to store the changes.
Alternative Flow – Empty Input	3.1	User leaves the required fields blank.
	3.2	System displays an error message “All fields are required.”
Rules	Any changes must be reflected on immediately.	

Table 6.33 Edit Profile and Company Information

6.2 Testing Setup and Result

To ensure the functionality and reliability of the app, a comprehensive testing setup was implemented. The testing was conducted on physical Android devices, Xiaomi Mi 8, and Android emulators with various configurations to simulate different device environments. The development environment included Android Studio and Visual Studio Code, with Flutter as the primary framework and Firebase as the backend for real-time database interactions and authentication. Black box testing, use case testing, and functional testing were employed to validate key features, including user registration, login, inventory management, and notifications. Performance tests were conducted under varying network conditions, and UI/UX tests ensured consistency across different screen sizes. The testing setup successfully verified the core functionality of the app, although iOS testing was delayed due to hardware constraints.

Test Case	**Objective**	**Test Scenarios**	**Expected Result**	**Actual Result**	**Status**
User Registration and Login	Verify that new users can successfully create accounts and log in to	1. New user registration with valid and invalid inputs.	Users should be able to register and log in without	As expected, the system handled valid and invalid inputs	Passed

	the system, and that appropriate error messages are displayed for invalid input.	2. User login with valid credentials. 3. User login with incorrect password or unregistered email.	issues. Errors should be displayed for invalid inputs.	correctly. Error messages were shown for incorrect login attempts.	
Item Management	Test the admin's ability to add new items to the inventory, edit item details, and delete items.	1. Adding a new item with all required fields filled. 2. Editing an existing item's details. 3. Attempting to delete currently linked to a composed item.	Items should be added, edited, and deleted correctly, with the system preventing deletion of items linked to composed item.	All test passed. Admin could manage items effectively, and the system blocked deletion of linked items.	Passed
Composed Items Management	Test the functionality of managing composed items that consist of multiple individual items.	1. Adding a new composed item by selecting two or more existing items. 2. Editing the details of a composed item. 3. Deleting a composed item and ensuring it removes the	Composed items should be created, edited, and deleted without affecting the inventory of individual items.	All tests passed. The system handled composed item management as expected.	Passed

		composed structure but keeps individual item intact.			
Stock In/Out	Test the functionality of stock-in and stock-out actions for individual and multiple items are once.	1. Stocking in/out multiple items and confirming quantities. 2. Stocking out an item with insufficient quantity. 3. Stocking in via WhatsApp or email and generating transaction logs.	Stock in and stock out actions should update inventory level correctly, prevent stock-out of insufficient quantities, and generate accurate transaction logs.	As expected, the system updated stock levels correctly and prevented invalid stock-out actions.	Passed
Category and Supplier Management	Test the admin's ability to add, edit, and delete categories and suppliers.	1. Adding new categories and suppliers. 2. Editing existing category and supplier details. 3. Attempting to delete a category or supplier linked	Admin should be able top manage categories and suppliers, but deletion should be blocked for those linked to items.	All tests passed. The system prevented the deletion of categories and suppliers linked to inventory items.	Passed

		to existing items. 4. View the items that linked to suppliers and categories.			
User Role Management	Verify that the admin can manage user roles effectively.	1. Assigned roles to users. 2. Editing and removing user roles. 3. Logging out from the profile page.	User roles should be updated, and access control should function as defined.	The system allowed the admin to manage user roles correctly, with appropriate restrictions for non-admin users.	Passed
Profile Management	Test the user's ability to manage their profile, including editing personal details and logging out.	1. Editing profile information. 2. Changing account profile. 3. Logging out from the profile.	Users should be able to update their profile and logout without issues.	All tests passed, and the system handled profile management as expected.	Passed

Table 6.34 Testing Result

6.3 Project Challenges

During the development and testing phase of the project, several challenges were encountered that impacted workflow and progress. One significant challenge was simulating real-world usage conditions, such as weak network connectivity and varying device performance. These

scenarios could affect how users experience the app, particularly when interacting with the database or using offline features. While simulated environments were helpful, they didn't fully replicate actual user conditions, making it difficult to guarantee a flawless user experience.

Another major challenge was the inability to test the iOS version of the application. Since I primarily work on a Windows laptop, I couldn't build, test, or run the iOS version, which requires a macOS environment. This limitation forced me to focus extensively on testing the Android version, delaying cross-platform functionality validation. Although the underlying code for both platforms is similar, subtle differences during deployment could affect the app's behavior on iOS devices. I plan to address this issue once I gain access to the necessary hardware or environment, but for now, the Android version serves as the primary reference for stability and functionality.

Additionally, a further challenge encountered during development was with the inventory prediction algorithm. The system is designed to notify the admin when it predicts that an item will reach its minimum stock level within the next 3 days. While functional, the current prediction model uses basic estimation techniques, which may not always provide the most accurate forecasts, especially when inventory demand fluctuates. Since the algorithm relies heavily on historical data, sudden changes in demand or stock movement may not be adequately accounted for. This presents an opportunity for future iterations to incorporate more advanced machine learning models or demand forecasting techniques, allowing for more precise and adaptive predictions.

6.4 Objective Evaluation

The main objectives of this project were centered around creating a cross-platform mobile application tailored for small businesses to streamline inventory management and enhance operational efficiency. The project successfully achieved several key goals outlined in the objectives, particularly in terms of digitizing inventory processes and integrating real-time data tracking.

The application effectively replaced traditional manual inventory methods with a digital solution, increasing both accuracy and efficiency in managing inventory. By implementing features such as real-time stock tracking, item categorization, supplier management, and user role control, the app empowers small business owners to oversee their inventory more

effectively. Furthermore, the algorithm-driven inventory estimation tool provides accurate predictions on future stock levels based on historical data, reducing the need for manual stock checks.

In terms of communication and collaboration, the integration of messaging tools for quick connections between business owners and suppliers met its intended goal of simplifying the order placement process. This feature, alongside real-time inventory notifications, improved the overall operational agility, allowing for faster decision-making and more efficient communication.

However, some objectives, such as cross-platform testing, remain partially unfulfilled due to the limitations in testing the iOS version during the initial phase. While the Android version has been thoroughly tested and functions as expected, the iOS version is slated for future testing.

Overall, the project has successfully met most of its core objectives, laying a solid foundation for future enhancements, particularly about cross-platform functionality and additional feature integration.

6.5 Concluding Remark

In conclusion, this project has successfully developed a cross-platform mobile application that significantly enhances the inventory management process for small businesses. By transitioning from manual, paper-based methods to a digital system, the app provides real-time inventory tracking, automated notifications, and streamlined communication between business owners and suppliers. The use of inventory estimation algorithms adds further value by predicting future stock needs, reducing manual labor, and preventing stockouts.

Although there were some challenges, particularly with testing the iOS version due to hardware limitations, the project has laid a strong foundation for further development. The Android version of the app has been thoroughly tested and functions according to the intended design, meeting the project's core objectives of improving efficiency and accuracy in inventory management.

Looking ahead, the focus will be on completing cross-platform testing, optimizing user experience across devices, and incorporating any additional feedback from end-users. The app is poised to be a valuable tool in helping small businesses improve their inventory operations and adapt to the demands of a competitive market.

Chapter 7
Conclusion and Recommendation

7.1 Conclusion

This project has successfully developed a cross-platform mobile application that addresses the critical needs of small businesses for efficient inventory management. By transitioning from traditional, manual methods to a digital, automated system, the application provides significant improvements in real-time inventory tracking, minimizing stockouts, and streamlining communication between business owners, staff, and suppliers. These advances not only help reduce human errors and manual labour, but also improve the accuracy and efficiency of managing inventory levels and order processes. Key features such as automatic notifications when inventory reaches critical stock levels and a user-friendly interface help make the operational process smoother for small businesses. The inventory estimation algorithm, while basic in its current form, adds further value by predicting future inventory needs, reducing guesswork, and helping to prevent out-of-stocks that can disrupt business operations.

During the testing and development phases, the Android version of the app was extensively tested and found to meet all the intended functional requirements. The project has demonstrated the feasibility of transitioning from manual, paper-based inventory tracking methods to a more efficient and scalable digital system. However, challenges were encountered, particularly with the inability to test the iOS version due to hardware limitations. Despite these challenges, the project successfully met its core objectives, and the Android version of the app serves as a solid foundation for future improvements and cross-platform deployment.

7.2 Recommendation

While the project achieved its primary goals, there are several areas where further development and enhancements can be made. The first recommendation is to prioritize the testing and optimization of the iOS version. Since testing was primarily focused on the Android platform due to hardware constraints, ensuring the app functions smoothly on iOS devices will be essential for a fully cross-platform experience. This will help address any platform-specific issues that may arise and ensure a consistent user experience across all devices.

Secondly, the inventory prediction algorithm, while functional, currently relies on simple estimation techniques. To improve its accuracy, especially in environments with fluctuating inventory demand, future iterations of the application should consider integrating more advanced machine learning models or demand forecasting algorithms. These techniques would allow the app to better account for changes in buying trends and stock movement, providing more reliable and adaptive stock predictions for users.

Another important recommendation is to continue gathering user feedback during the deployment phase. Real-world usage by business owners and staff will provide invaluable insights into potential areas for improvement, such as enhancing user interface design, adding more customization options, or improving performance under varying network conditions. Additionally, expanding the app's functionality by incorporating advanced features, such as detailed analytics and reporting tools could further increase its value to small businesses.

Performance optimization, particularly for offline functionality, should also be a focus in the future. Considering that the app is likely to be used in environments with unstable network connections, improving the app's ability to operate seamlessly in offline mode will enhance the overall user experience. This may involve better data synchronization when connectivity is restored, ensuring that the system is kept up to date even if the user is temporarily offline.

Lastly, exploring additional features such as advanced reporting tools, comprehensive supplier management, and the ability to integrate with other business software systems could make the app more versatile and customizable. These enhancements would not only increase the app's appeal to a wider range of businesses but also offer more comprehensive inventory management capabilities, solidifying its position as an indispensable tool for small business operations. By addressing these recommendations, the application can continue to evolve and provide greater value to its users in the future.

REFERENCES

[1] F. Hamidy and I. Yasin, “Implementation of moving average for forecasting inventory data using ...,” ResearchGate, https://www.researchgate.net/publication/368591497_Implementation_of_Moving_Average_for_Forecasting_Inventory_Data_Using_CodeIgniter (accessed Aug. 25, 20239).

[2] Sortly Inc, “Inventory simplified,” Sortly, https://www.sortly.com/ (accessed Aug. 25, 2023).

[3] BGPworks, “Inventory management reinvented,” BoxHero, https://www.boxhero-app.com/en (accessed Aug. 25, 2023).

[4] Zeynu Hamid, “Simple & Powerful Inventory Management app,” Naye Inventory - Inventory management app, https://nayeinventory.com/ (accessed Aug. 26, 2023).

[5] F. H. Musa and R. Ibrahim, “Inventory management system olahan hasrat Sdn. Bhd.. using mobile application,” Applied Information Technology And Computer Science, https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/2319 (accessed Aug. 26, 2023).

[6] S. Kar, S. Bhimrajka, A. Kumar and S. Mukherjee, "Mobile based Inventory Management System with QR code," 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2022, pp. 1-6, doi: 10.1109/CONECCT55679.2022.9865739 (accessed Aug. 26, 2023).

[7] A. Ashishdeep, J. Bhatia, and K. Varma, “Software process models for mobile application development: A Review,” IJCSC, http://www.csjournals.com/IJCSC/PDF7-1/20.%20Anitha.pdf (accessed Aug. 27, 2023).

[8] R. Abdullah, K. Zi Xiang, and M. Ilman Hakimi Chua Abdullah, “E inventory management system using Android mobile application at ...,” Universiti Teknikal Malaysia Melaka, https://www3.utem.edu.my/care/proceedings/merd18/pdf/p180_190.pdf (accessed Aug. 25, 2023).

[9] L. Le, “Inventory management application,” Theseus, https://www.theseus.fi/handle/10024/752452?show=full (accessed Aug. 25, 2023).

[10] F. Darnis, “Mobile application for inventory control in a minimart - researchgate,” ResearchGate, https://www.researchgate.net/publication/320202392_Mobile_Application_for_Inventory_Control_in_a_Minimart (accessed Aug. 25, 2023).

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 2**
Student Name & ID: Beh Wei Jun 20ACB05218	
Supervisor: Mr. Tan Chiang Kang	
Project Title: Inventory Tracker with Estimation	

1. WORK DONE

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

I reviewed the core features implemented during FYP 1, focusing on user authentication, inventory management, and transaction handling. I brainstormed ways to improve the UI for a more user-friendly experience and started planning how to enhance the data structure to support new features like notifications, supplier and categories management, and user roles.

2. WORK TO BE DONE

1. Done the management of composed items.
2. Refine the UI design for better usability.
3. Implement an updated data structure to support new features.

3. PROBLEMS ENCOUNTERED

The main challenge was designing a data structure that supports the new features without compromising app performance.

4. SELF EVALUATION OF THE PROGRESS

Week 2 was productive in setting the groundwork for upcoming features, with progress in both UI improvements and data structure planning.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 4**
Student Name & ID: Beh Wei Jun 20ACB05218	
Supervisor: Mr. Tan Chiang Kang	
Project Title: Inventory Tracker with Estimation	

1. WORK DONE

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

I successfully implemented the management features for composed items and updated the data structure to accommodate these changes. I also started working on the management functionalities for categories and suppliers.

2. WORK TO BE DONE

1. Complete the implementation of category and supplier management features.
2. Working on management of user role
3. Test the new features for functionality and integration with existing components.

3. PROBLEMS ENCOUNTERED

Integrating new management features into the existing system presented some challenges with data consistency and user interface adjustments.

4. SELF EVALUATION OF THE PROGRESS

Week 4 saw significant progress with the successful addition of composed item management and initial steps towards category and supplier management.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 6**
Student Name & ID: Beh Wei Jun 20ACB05218	
Supervisor: Mr. Tan Chiang Kang	
Project Title: Inventory Tracker with Estimation	

1. WORK DONE

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

I completed the implementation of category, supplier, and user role management features, ensuring they are fully integrated with the application. Additionally, I conducted initial testing to verify that these features are functioning as intended and are seamlessly integrated with the existing system.

2. WORK TO BE DONE

1. Continue testing the new features to identify and resolve any issues.
2. Begin developing the user interface of profile page, let the user edit profile information and change password.

3. PROBLEMS ENCOUNTERED

Encountered challenges related to integrating new management features with the existing user interface, including maintaining consistency and ensuring smooth navigation.

4. SELF EVALUATION OF THE PROGRESS

Week 6 involved significant advancements with the completion of category, supplier, and user role management functionalities.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 8**
Student Name & ID: Beh Wei Jun 20ACB05218	
Supervisor: Mr. Tan Chiang Kang	
Project Title: Inventory Tracker with Estimation	

1. WORK DONE

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

I successfully developed and implemented the user interface for the profile page, which allows users to edit their profile information and change their password. Additionally, I performed thorough testing of these new features to ensure their functionality and integration with the existing system.

2. WORK TO BE DONE

1. Begin working on integrating notification management features to enhance user engagement and alerting capabilities.

3. PROBLEMS ENCOUNTERED

Faced challenges with ensuring data synchronization between the profile updates and the backend database, which required additional debugging and validation.

4. SELF EVALUATION OF THE PROGRESS

Week 8 saw significant progress with the successful implementation of the profile page functionality.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 10**
Student Name & ID: Beh Wei Jun 20ACB05218	
Supervisor: Mr. Tan Chiang Kang	
Project Title: Inventory Tracker with Estimation	

1. WORK DONE

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

I completed the development and implementation of the notification algorithm. This involved finalizing the logic for triggering notifications based on inventory levels and ensuring that notifications are sent correctly to users. Additionally, I conducted comprehensive testing to verify the functionality of the notification system and its seamless interaction with other application features.

2. WORK TO BE DONE

1. Finalize the development of any remaining features within the application.
2. Conduct thorough testing of all application functionalities to ensure overall performance.

3. PROBLEMS ENCOUNTERED

Encountered issues to notification to ensure they accurately reflect the inventory data and timely alert users.

4. SELF EVALUATION OF THE PROGRESS

Week 10 marked significant progress with the completion of the notification algorithm and its integration.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 12**
Student Name & ID: Beh Wei Jun 20ACB05218	
Supervisor: Mr. Tan Chiang Kang	
Project Title: Inventory Tracker with Estimation	

1. WORK DONE

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

I began drafting the Final Year Project (FYP) 2 report, focusing on documenting the development process, features, and functionality of the application. I also compiled and reviewed testing results to ensure that the report accurately reflects the project's progress and outcomes.

2. WORK TO BE DONE

1. Complete the FYP 2 report.
2. Finalize and check the functionality of application.

3. PROBLEMS ENCOUNTERED

None

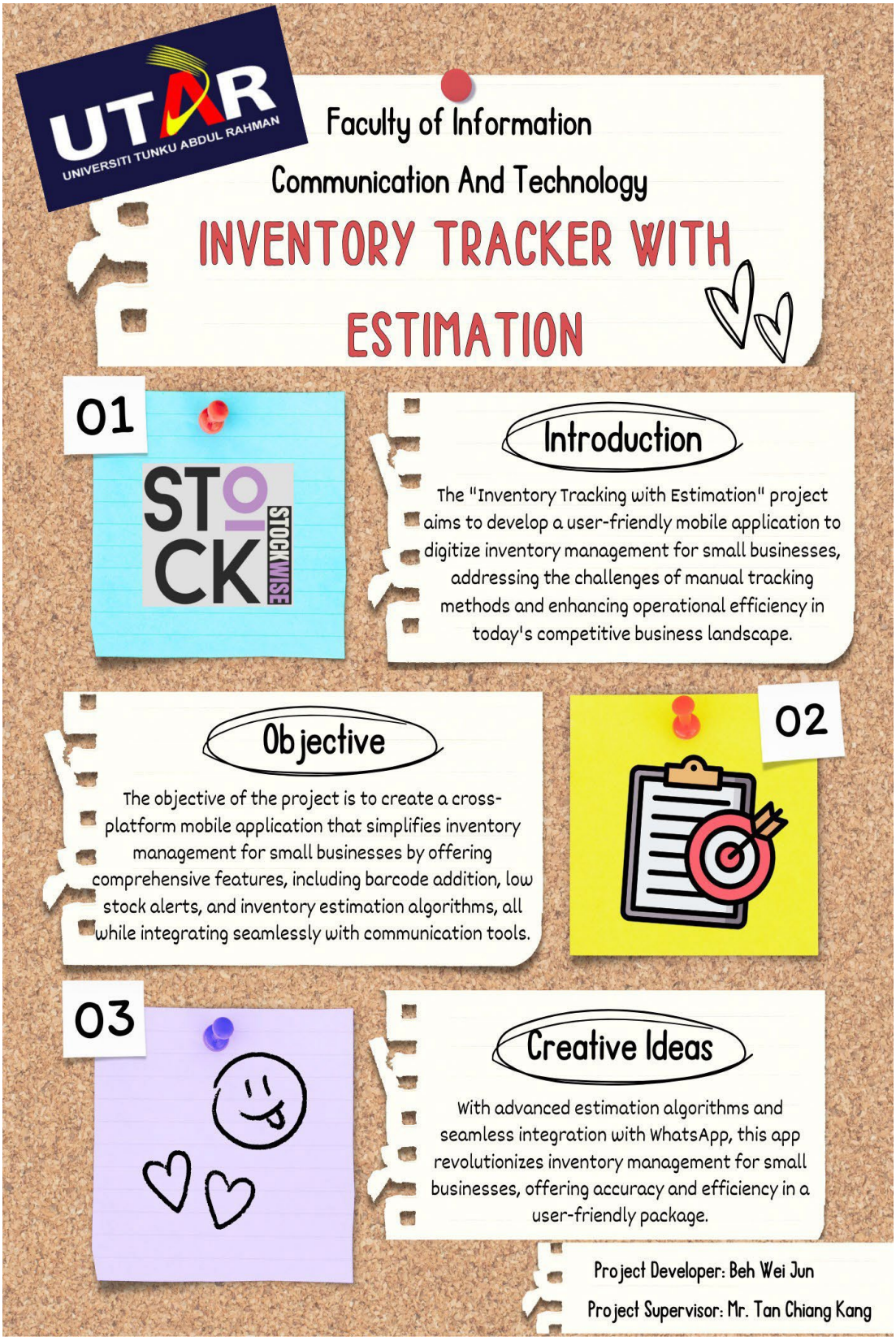
4. SELF EVALUATION OF THE PROGRESS

Week 12 involved a critical step in documenting the project's progress and outcomes through the FYP 2 report.

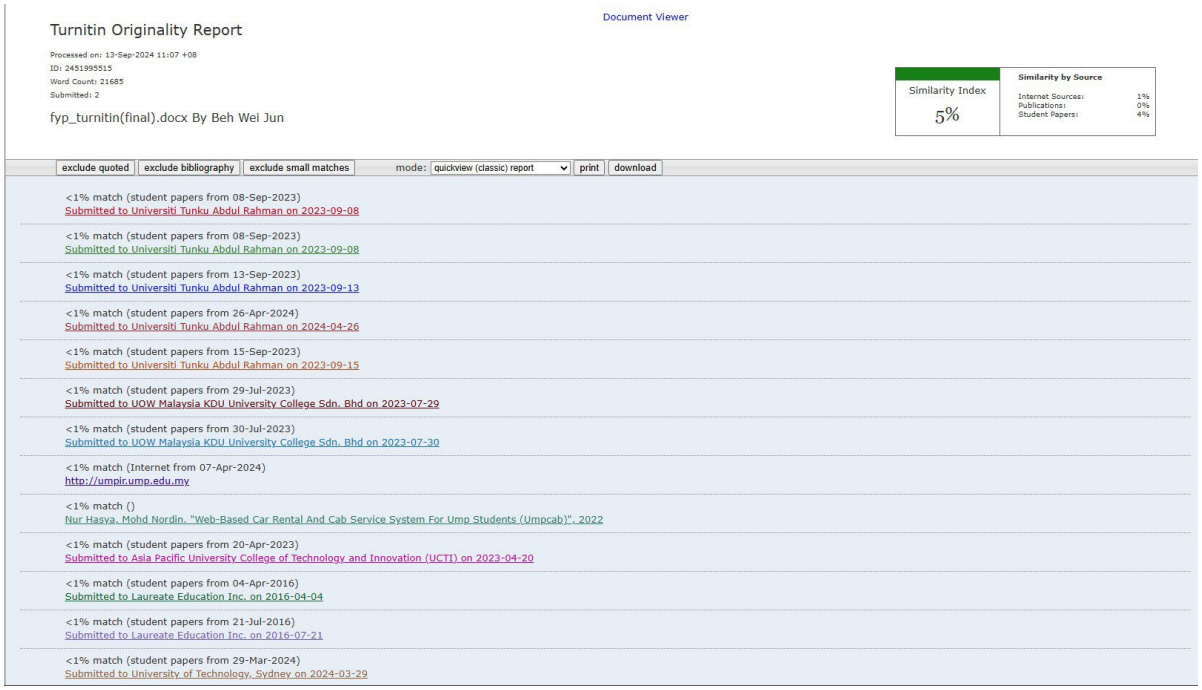
Supervisor's signature

Student's signature

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information
Communication And Technology
INVENTORY TRACKER WITH ESTIMATION
01
STOCKWISE
Introduction
The "Inventory Tracking with Estimation" project aims to develop a user-friendly mobile application to digitize inventory management for small businesses, addressing the challenges of manual tracking methods and enhancing operational efficiency in today's competitive business landscape.
Objective
The objective of the project is to create a cross-platform mobile application that simplifies inventory management for small businesses by offering comprehensive features, including barcode addition, low stock alerts, and inventory estimation algorithms, all while integrating seamlessly with communication tools.
02
03
Creative Ideas
With advanced estimation algorithms and seamless integration with WhatsApp, this app revolutionizes inventory management for small businesses, offering accuracy and efficiency in a user-friendly package.
Project Developer: Beh Wei Jun
Project Supervisor: Mr. Tan Chiang Kang

PLAGIARISM CHECK RESULT



Turnitin Originality Report

Processed on: 13-Sep-2024 11:07 +08
ID: 2451995515
Word Count: 21685
Submitted: 2

fyp_turnitin(final).docx By Beh Wei Jun

Document Viewer

Similarity Index
5%

Similarity by Source
Internet Sources: 1%
Publications: 0%
Student Papers: 4%

exclude quoted | exclude bibliography | exclude small matches | mode: quickview (classic) report | print | download

<1% match (student papers from 08-Sep-2023)
Submitted to Universiti Tunku Abdul Rahman on 2023-09-08

<1% match (student papers from 08-Sep-2023)
Submitted to Universiti Tunku Abdul Rahman on 2023-09-08

<1% match (student papers from 13-Sep-2023)
Submitted to Universiti Tunku Abdul Rahman on 2023-09-13

<1% match (student papers from 26-Apr-2024)
Submitted to Universiti Tunku Abdul Rahman on 2024-04-26

<1% match (student papers from 15-Sep-2023)
Submitted to Universiti Tunku Abdul Rahman on 2023-09-15

<1% match (student papers from 29-Jul-2023)
Submitted to UOW Malaysia KDU University College Sdn. Bhd on 2023-07-29

<1% match (student papers from 30-Jul-2023)
Submitted to UOW Malaysia KDU University College Sdn. Bhd on 2023-07-30

<1% match (Internet from 07-Apr-2024)
http://umpir.ump.edu.my

<1% match ()
Nur Hasya, Mohd Nordin, "Web-Based Car Rental And Cab Service System For Ump Students (Umpcab)", 2022

<1% match (student papers from 20-Apr-2023)
Submitted to Asia Pacific University College of Technology and Innovation (UCTI) on 2023-04-20

<1% match (student papers from 04-Apr-2016)
Submitted to Laureate Education Inc. on 2016-04-04

<1% match (student papers from 21-Jul-2016)
Submitted to Laureate Education Inc. on 2016-07-21

<1% match (student papers from 29-Mar-2024)
Submitted to University of Technology, Sydney on 2024-03-29

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.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Beh Wei Jun
ID Number(s)	20ACB05218
Programme / Course	Bachelor of Computer Science
Title of Final Year Project	Inventory Tracker with Estimation

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 5 **%** **Similarity by source** Internet Sources: 1 % Publications: 0 % Student Papers: 4 %	
Number of individual sources listed of more than 3% similarity: 0	
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** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

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: Tan Chiang Kang

Date: 13/9/2024

Signature of Co-Supervisor

Name:

Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB05218
Student Name	Beh Wei Jun
Supervisor Name	Mr. Tan Chiang Kang

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.
√	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: 13 September 2024