MONEY MANAGEMENT USING MOBILE APPLICATION DEVELOPMENT

By

Tay Ye Schao

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman in

partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)
COMMUNICATIONS AND NETWORKING
Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2024

REPORT STATUS DECLARATION FORM

Title: MONEY MANAGEMENT USING MOBILE APPLICATION DEVELOPMENT

Academic Session: Jan 2024

I ____TAY YE SCHAO___

(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,

Tay

(Author's signature)

Farina

(Supervisor's signature)

Address:

No.11 Lorong Wijaya, Jalan Nyior

83700, Yong Peng, Johor

Dr Farina Saffa binti Mohamad Samsamnun

Supervisor's name

Date: 24 April 2024

Date: 26/04/2024

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ___24 April 2024_____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***TAY YE SCHAO***________ (ID No: __***20ACB03821***___) has completed this final year project/ dissertation/ thesis* entitled "*MONEY MANEGEMENT USING MOBILE APPLICATION DEVELOPMENT*" under the supervision of ***Dr Farina Saffa Binti Mohamad Samsamnun*** (Supervisor) from the Department of **Computer and Communication**, Faculty of **Information and Communication Technology**.

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

Yours truly,

Tay

____

(Tay Ye Schao)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Money Management Using Mobile Application Development**" 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 : Tay Ye Schao

Date : 24 April 2024

ACKNOWLEDGEMENTS

I am so grateful that my supervisor, Dr Farina Saffa Binti Mohamad Samsamnun, from the Department of Information and Communication Technologies, gave me the chance to work on this project. I also appreciate how willing she was to take the time to provide me with some insightful comments or suggestions in my project. During these crucial months, I appreciate your support and assistance.

A special thanks goes out to my friends, who are constantly there for me whenever I'm stressed or worried and who are also a great source of guidance as I finish my project. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

In this modern era of globalization, the process of digitization has become increasingly common in our daily lives. When it comes to personal money management, an increasing number of people are turning to mobile app-based money management systems to track their expenses and income. Furthermore, these mobile applications often include an analysis module that allows users to gain a more intuitive understanding of their financial flows. To develop a comprehensive money management system, various existing systems like MyMoney, BudgetApp, and Monefy have been reviewed in terms of their modules, functionalities, strengths, and weaknesses. To address the issues identified in these existing systems, the project objectives for the proposed system have been discussed. The strengths from all of these existing systems have been adopted and integrated into the proposed system to enhance its capabilities. In the proposed money management system, the modules provided include a user authentication module, transaction management module, data virtualization module and user setting module,. The project utilized the Agile Scrum methodology and employed the Flutter programming language to develop a mobilebased money management system. The choice of a mobile application is driven by the fact that nearly everyone uses mobile devices. In the final stages of development, blackbox testing and white-box testing will be conducted to thoroughly test the entire system once all modules have been integrated. Once all modules pass the testing phase, the money management mobile application will be ready for commercialization and use by users.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

UI	User Interface
APP	Application
RAM	Random Access Memory
ROM	Read Only Memory
SSD	Solid State Drive
SDK	Software Development Kit

CHAPTER 1 INTRODUCTION

1.1 Project Background

Personal financial management refers to the practice of recording all expenses and income, analysing them, and setting budgets and savings goals based on these records. This is an essential skill that each of us must master to ensure financial independence and economic freedom without falling into a deficit.

The approach to managing personal finances has evolved significantly over time. In the 1980s, most people recorded their funds in black and white, which was inefficient and time-consuming to check and count. Furthermore, this method had a high error rate, and any accidental statistical error required more time for recalculation. Surprisingly, this method is still widely used today.

In the 1990s, with the advent of Microsoft Excel, some companies began using it for asset statistics. However, personal money management with Excel was not common due to the high cost of computers at the time. Around the year 2000, some individuals, especially office workers, began using Excel for fund management. In the 21st century, with the advancement of digitization and smart technology, the management of personal finances has undergone revolutionary changes. In an era marked by increased financial complexity and the fast-paced nature of modern life, the need for effective financial management tools has grown significantly.

Money management apps on mobile phones provide flexibility compared to using Excel or manual record-keeping. There's no need to carry notebooks, books, and pens to record expenses; just open the mobile app anytime and anywhere to record transactions offline. Additionally, mobile money management apps analyze income and expenses automatically, eliminating the need for manual calculations. Moreover, these apps have a simpler and more user-friendly UI design compared to the businessoriented and complicated design of Excel, making them preferable for modern young people.

However, many existing money management mobile app designs are similar and lack some essential functions. Therefore, during the final stage of the proposed system development, it should aim to provide a comprehensive money management mobile application system for users. The proposed system will include modules such as a login module, transaction records module, statistical analysis module, wallet (type of financial accounts) module, profile settings module.

1.2 Problem Statement and Motivation

Without a proper user design and interface of money management system, even if it provides a variety of tools to help users manage their finances, it will be useless. In recent times, as more and more people use money management apps on mobile devices to handle their finances, we have discovered various challenges in launching a popular money management app.

1.2.1 Sustainable data usage and storage

User data cannot be saved permanently. Many existing money management applications available on the Play Store and App Store only support local storage. Consequently, all transaction records entered by the user will only be accessible on the device where they were originally recorded. Users are unable to transfer their data to another device. Moreover, if a user accidentally clears the data on their device, it results in the permanent loss of the data within the application.

1.2.2 Lack of comprehensive features for optimizing transaction data management

Existing mobile applications lack of provide users with intuitive user interfaces to navigate and optimize the visibility of the transaction data. Most of the existing mobile application is limit at displaying transaction data in a single list or analysis, which impedes users' ability to efficiently view and interact with their financial information and this also greatly limits the user's visibility of the data.

1.2.3 Lack of comprehensive and user-friendly statistical analysis functions

Many of existing money management applications solely offer statistical analysis of all financial transactions recorded by the user. Most of these apps do not permit users to customize their statistical analysis based on their preferences. For instance, they can separate statistics on expenses and income or the ability to conduct data analysis based on different financial accounts. This limitation hampers users' capacity to extract valuable insights from their financial data.

1.3 Motivation

The motivation behind developing a money management mobile app is to enhance users' ability to effectively manage their finances. The Money Manager app is designed to enable users to monitor the flow of funds, enhance their financial well-being, and strategize for their future financial goals.

1.4 Project Objectives

The following are the goals that the development project will attempt to accomplish:

To develop a cloud database-based money manager mobile app.

Developing a cloud database-based money manager application serves the purpose of ensuring data sustainability and scalability. In contrast to relying on local storage, utilizing cloud storage as the application's database enables users to access their data from anywhere with an internet connection, greatly enhancing the application's convenience. By cloud database, users can synchronize their transaction records across all their login devices. Additionally, the choice of cloud storage as the database is motivated by its compatibility across platforms, allowing users to seamlessly manage their finances on both Android and iOS systems. This not only reduces development time but also lowers overall development costs.

To develop a mobile application that can optimize transaction management, improve data visibility and data manipulation.

By create a comprehensive manage transaction module that allowed users to efficiently add, modify, and delete transaction data. Additionally, design intuitive interfaces like multiple view of transaction data lists based on category, financial account and date filtered to provide users with clear visibility and control over their data, enhancing data management capabilities and improving overall system usability.

To develop comprehensive statistical analysis of user transactions data

Apart from offering statistical analysis based on year, month, and week, the proposed application system also provides user with the capability to access customized statistical analysis in categories, financial accounts, income, and expenses according to their preferences. This empowers users to gain a deeper and more detailed insight into their financial situation. Additionally, users can export and share the statistical analysis in the form of Excel, facilitating the ease of generating financial reports or sharing data with financial advisors.

1.5 Project Scope and Direction

The money manager app will encompass the features of user authentication, account management, transactions management, statistical analysis management, type of financial account management, and firebase integration.

i. User authentication

Before processing to user own account, he or she needs to go through the login page to sign in by register email and password. As new user, user can click the register button and will browse to register page and enter their details. In case if user forget their password, he or she can click the forgot password, enter the email and will send a reset password link to the email.

ii. Account management

In this module, user can do some modification on their personal information such as reset password and change profile. There are some functions available in the account page as well. Include account setting, category setting, display mode settings, export excel file and sign out.

iii. Transactions management

In this module, user allow add transactions by enter the category, types of transactions, amount, date and description. After clicking the save button, the transaction will store the data in firebase and user is able to see the transaction detail recorded in the transaction module's dashboard. User can swipe the transaction to delete the transaction or tap the transaction to modify the transaction. All the actions will be updated into firebase as well.

iv. Statistic management

In this module, user can view their total transactions in the view of chart and graph. User is able to summarize it as year, month, weekly or daily view. User will allow choose the statistic display by category, financial accounts, income and expenses.

v. Type of financial account

In this module, user can create different type of the account. For example, cash, deposit card, e-wallet and so on. It will display the current asset for each of the created accounts. It will also have to display the current asset in sum of all accounts.

vi. Firebase integration

After user have signup an account, authentication in firebase will update it at the time. When the transaction have add, modify and delete by user, the data will be delete as well.

1.6 Contributions

People with strong financial management skills contribute significantly to society. An effective money management system empowers users to create budgets tailored to their financial circumstances, reduce expenses, and set achievable savings goals. Furthermore, such a financial management system can cultivate healthy savings habits, accelerating users' journey towards economic independence and financial freedom.

1.7 Report Organization

The details of this research are shown in the following chapters. Chapter 2 provides a literature review related to money management apps, reviewing the current existing application and do compare with the development application. Chapter 3 presents the system design, system architecture, system requirement and project timeline. Chapter 4 delve into preliminary works, include software setup, setting and configuration. Subsequent chapters delve into testing, conclusions, and future work, ensuring a thorough examination of the project's development and outcomes.

CHAPTER 2 Literature Reviews

2.1 Overview

In this literature review chapter will analyze and evaluate existing mobile applications and websites with similar functionalities. The primary objective of this chapter is to identify the strengths and weaknesses of each system and subsequently compare them to inform the development of the proposed system. Additionally, we will critique each module of the existing system in terms of functionality and features. Gathering valuable insights from these existing systems will serve to enhance the proposed system.

2.2 Existing System

2.2.1 Monefy Application [1]

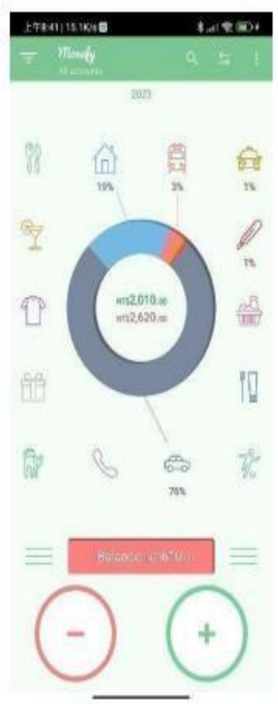


Figure 2.1 Add Transaction with Pie Chart View

Figure 2.2 Date Range Filter

Figure 2.3 User Setting Page Design

Monefy, developed by Reflectly, is a highly popular mobile application available on both Android's PlayStore and Apple's AppStore. With over 5 million downloads and a remarkable rating of 4.5 out of 5 on PlayStore and 4.7 out of 5 on AppStore, it's evident that the majority of users are highly satisfied with the services and functionalities provided by the Monefy mobile application.

Monefy serves as a versatile money management system that supports multiple transaction books. This unique feature enables users to manage their personal finances independently or collectively for family finance management, all within a single application. Monefy is particularly known for its intuitive and user-friendly interface, eliminating any redundancy. All functions and options are seamlessly integrated into a single interface, providing users with a smooth and efficient experience. The application employs a local storage-based database, ensuring data security and accessibility.

Monefy combines the analysis module and transaction book module, simplifying financial management. Users can effortlessly record income and expenses by tapping the plus and minus icons at the bottom of the interface. Selecting expenditure categories is as easy as clicking on the corresponding icons surrounding the chart. The central area of the chart provides a clear overview of total income and total expenditures, while the section below displays the overall balance.

In the upper left corner of the interface, a widget offers chart analysis based on different timeframes and accounts, granting users the flexibility to manage their finances according to their preferences. Monefy's chart allows users to switch between daily, weekly, monthly, yearly, and all-encompassing fund flow analyses. Users can also specify custom time ranges using the provided calendar.

Furthermore, in the upper right corner of the interface, users can access additional functions such as adding categories, financial accounts, currencies, and more. However, it's worth noting that some of these functions require payment for unlocking, a detail we do not comment on.

Monefy's exceptional features and user-centric design have contributed to its outstanding user satisfaction and popularity in the mobile application market.

2.2.2 MyMoney [2]

Figure 2.4 Home Page(MyMoney)

Figure 2.5 Statistic view page(MyMoney)

Figure 2.6 Budget Page (MyMoney)

Figure 2.7 Account Page (MyMoney)

Figure 2.8 Category List Page

MyMoney is a comprehensive personal financial management system, though it is limited to Android users. The UI design of MyMoney is user-friendly and intuitive, making it easy for users, especially first-time users, to navigate.

The main functions are conveniently located at the bottom of the interface as a menu, including Transaction Records, Analysis, Budgets, Accounts, and Categories modules. Additional, less frequently used functions are found in the settings module in the upper left corner of the interface. This design simplifies the user experience.

In the Transaction Records module, MyMoney employs a common design found in many money management applications available on the PlayStore. Transactions are presented in the form of transaction books, displaying total expenditures, total income, and the overall balance. This presentation is generally wellreceived by users.

In the Analysis module, MyMoney offers default statistics in the form of pie charts. However, it distinguishes itself by providing not only an overview of expenses and income but also detailed expenses and income flow charts, along with account analysis. This thoughtful design enables users to gain deeper insights into their spending patterns and financial habits.

The Budget module allows users to set budgets based on various expense categories, facilitating better monitoring of their spending. MyMoney includes predefined categories like education, food, clothing, etc. Users can also customize categories, including icons and titles, a feature that extends to the Categories module as well.

MyMoney's combination of a user-friendly interface and insightful financial analysis tools makes it a valuable tool for Android users seeking to manage their personal finances effectively.

2.2.3 Money Manager – Budget & Expense Tracker [3]

Figure 2.9 Home Page (MoneyManager)

Figure 2.10 Category List (MoneyManager)

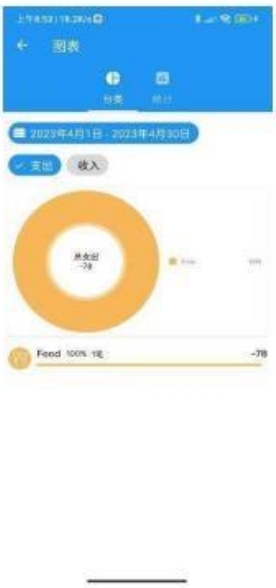


Figure 2.11 Statistic Analysis Page (MoneyManager)

Budget App, developed by HGBGroup Inc., is a mobile application available on both Android's PlayStore and Apple's AppStore. Despite achieving over 1 million downloads, it holds a commendable rating of 4.7 out of 5 in both platforms, showcasing strong user satisfaction.

The Budget App offers core functionalities, including the Budget Setting module, Transaction Records module, Categories Management module, and Analysis module. While it meets the fundamental requirements of a money management system, users have expressed concerns about its user interface (UI) design, describing it as dull and low-quality.

The Budget Setting module and Transaction Records module in the app are integrated into a single interface, organized into upper, middle, and lower sections. The upper portion displays monthly expenses, income, and the total balance. In the middle section, users can set their budgets. However, it's worth noting that the Budget App only allows setting an overall budget amount and doesn't support segmenting budgets by category, which may pose challenges for users who prefer more detailed budget segmentation. The lower part of the interface is dedicated to the Transaction Records module, where users input their financial transactions.

The Analysis module and Categories module in the Budget App offer functionalities that are similar to those found in other money management systems. The Analysis module, in particular, provides users with a pie chart for fund flow analysis, as well as a cash flow feature that offers more intuitive insights into financial amounts on a bi-monthly basis.

While Budget App fulfills essential money management functions, the feedback regarding its UI design suggests room for improvement in creating a more visually appealing and user-friendly interface.

2.3 Comparison between previous works & proposed works

Three existing works (Monefy, MyMoney, and BudgetApp) have been reviewed in the terms of their feature. All reviewed systems have their own strengths and weaknesses and have do comparison with the proposed system in Table 2.1.

	Budget& Expense Tracker	Monefy	MyMoney	Proposed work
Design of UI	Simple and unattractive design	Attractive design	Attractive design	Simple but attractive design
Statistic of transactions	By category	By category	By category	By category
Supported storage	Local storage	Local storage	Local storage	Cloud storage
Advanced search data	Not available	Available	Available	Available
Passcode login	Available	Available	Available	Available
Export data to Excel file	Not Available	Not Available	Not Available	Available
Supported offline status	Supported	Supported	Supported	Not Supported

Table 2.1 Comparison between previous works and proposed works

Chapter 3 SYSTEM METHODOLOGY

3.1 Methodology

Agile methodology (Scrum)

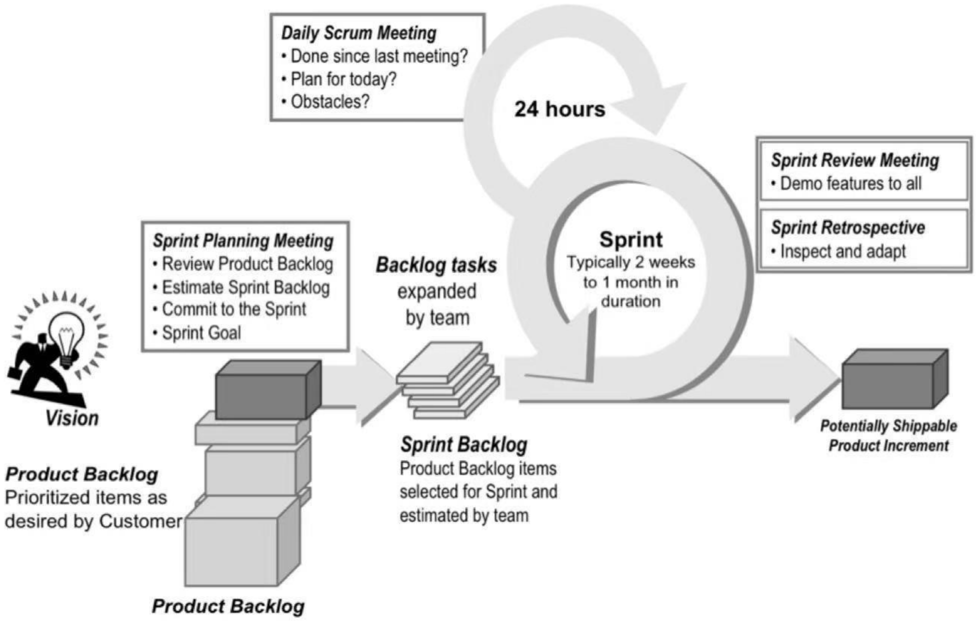


Figure 3.1 Agile Methodology Diagram

Software Development Life Cycle (SDLC) methodology is often used for development of projects. The most suitable methodology for the proposed system is Scrum. Scrum is an agile project management strategy and is suitable for fast, small project development. Scrum is more focused on managing iterative development rather than specific agile. Following are the steps of apply scrum in money manager app development:

1. Product backlog: Product backlog is the initial planning process of the Scrum. In this phase, we should create a backlog of the features, user stories and tasks should be done during the implementation of money manager applications. For example, user registration page, add transaction page, user account preferences page and so on.

2. Sprint planning: In this phase, we should estimate the sprint duration. The sprint duration could be 1 to 2 weeks. We should review back on the product backlog and break down the backlog items into smaller tasks. For example, register page, login page and forgot password page.

3. Daily scrum meeting: During the daily scrum meeting we should review the progress, discuss the problem or obstacles faced and set target works for today.

4. Sprint development: In this stage, need to implement the functions from sprint backlog. We should consider the factors like user experience, functionality and neatness of code during the development. Usually, it will take 2 to 4 weeks in a duration.

5. Sprint review: At the end of the sprint development, we should review and demo the features that what have been accomplished.

6. Product backlog refinement: After having completed the showcase on the functions have developed, continue to refine the product backlog. For example, add a loading animation during the system is process the function.

3.2 System Requirement

3.2.1 Hardware

Hardware Specifications are mentioned below, and the hardware involved is laptop and mobile device. Laptop is used to code the system while final testing will be run in mobile device.

Table 3.1 Specifications of laptop

Description	**Specifications**
Model	Dell Latitude 7420
Processor	Intel Core i7-1185
Operating System	Windows 11
Graphic	Intel Iris Xe
Memory	16GB RAM
Storage	512GB

Table 3.2 Specifications of mobile device

Description	**Specifications**
Model	Xiaomi 11T
Processor	Mediatek MT6893 Dimensity 1200 (6nm)
Operating System	Android 13
UI version	MIUI Global 14.0.4
Memory	8GB RAM
Storage	128GB ROM

3.2.2 Software

Flutter SDK:

Flutter SDK is a free and open-source framework developed by Google for mobile application development. It enables developers to use a single codebase to create applications that can run on different operating systems, eliminating the need to develop separate apps for Android and iOS.

Android Studio:

Android Studio is a free integrated development environment (IDE) that aids developers in building, testing, and debugging mobile applications. It offers various analysis frameworks and facilitates rapid coding through quick iterations, powered by the IntelliJ IDEA. Additionally, Android Studio supports connectivity with Firebase and provides integrated cloud support for backend development. In the proposed system, Android Studio is chosen due to the need for incorporating multiple frameworks, and it is also used to run the Android phone emulator.

Firebase:

Firebase is a cloud-hosted NoSQL database, in contrast to traditional databases that utilize a relational database schema. It is suitable for cloud-based applications like appointment systems and queue management systems, as it offers online database capabilities. However, Firebase has some drawbacks, such as potentially more complex configuration compared to relational databases and the lack of support for the ACID (atomicity, consistency, isolation, and durability) characteristics. In the proposed system, Firebase is selected to implement a virtual queueing system and cloud-based appointment scheduling, aligning with the project's objectives. Firebase's real-time database feature is well-suited for this purpose.

Dart:

Dart is an open-source programming language used in the Flutter project. It supports object-oriented concepts such as inheritance, encapsulation, abstraction, and polymorphism. Dart is versatile, as it can be used on both the client-side and serverside. By implementing Dart in the proposed system, it ensures compatibility with both Android and iOS operating systems, enabling the application to function seamlessly on both platforms.

3.3 Project Milestone

3.3.1 Project timeline (FYP 1)

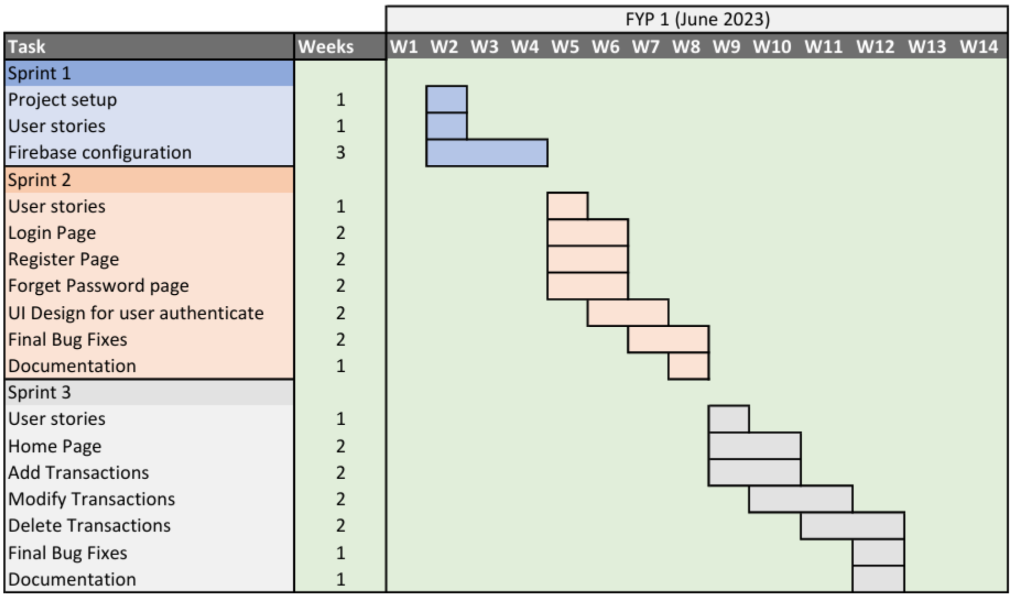


Figure 3.2 Project Gantt Chart for FYP 1

3.3.2 Project Timeline (FYP 2)

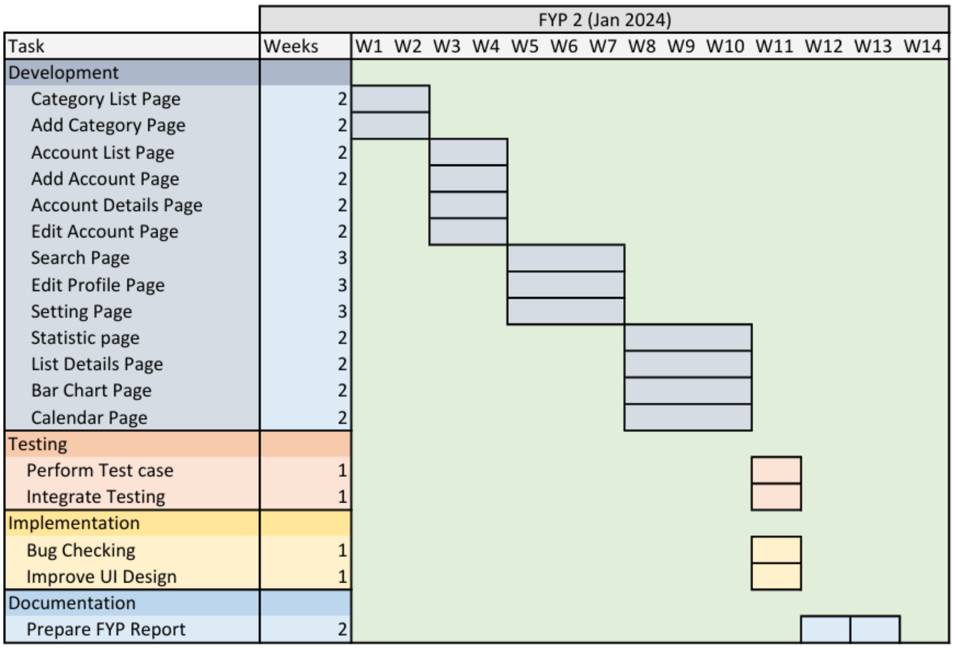


Figure 3.3 Project Gantt Chart for FYP 2

3.4 System Architecture Diagram/ Overview

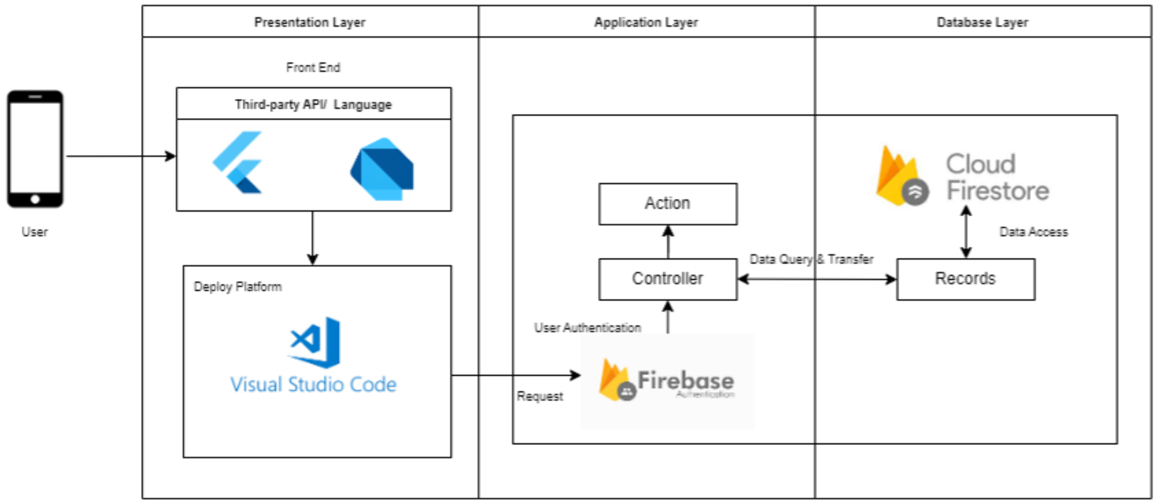


Figure 3.4 System Architecture Diagram

Figure 3.1.1 shows that the system architecture of the proposed system. This system interacts with one API call Pub.Dev which is the official package repository for Dart and Flutter applications development. Pub.Dev is a hub centralized library to share, use and discover packages by include the dependencies into pubspec.yaml file. Another will be the framework called Flutter to build the user interface of the mobile application. Visual Studio Code will be used as deployment platform. When a user starts to use the proposed system, there should register an account and will authenticate through Firebase Authentication at the second layer architecture. Once authenticated successfully, there should be able to start do some transaction records. All the data that user stored will send to the Database layer and fetch the data to the Firebase Firestore. Firebase Firestore will also use for querying the data from database and send to the controller and virtualize the data through Flutter and Dart when every time the user is requested to do data virtualization. Finally, the content will load to the client device to accomplish data virtualization.

3.5 Use Case Diagram and Descriptions

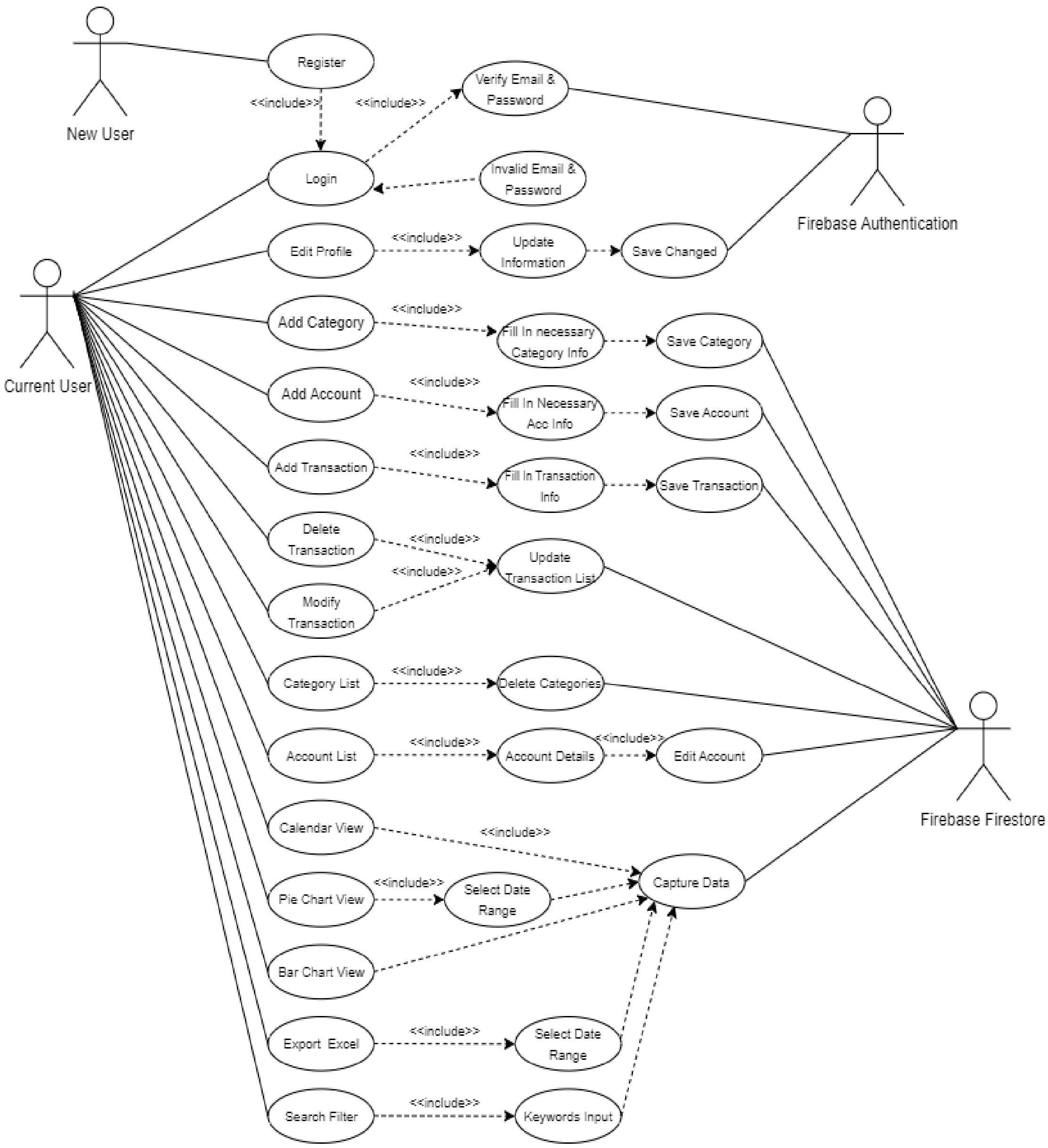


Figure 3.5 Use Case Diagram

Use Case	Register
Aim	Register an account if the user has not had an account.
Actor	New User
Trigger	When user click the Sign-Up button
Main Flow	1. User submits the valid email and password. 2. Browse user back to login page once successful registered. 3. Back to Login page if user already existed account. 4. Use case end.
Alternative Flow	1. System prompt error message if the account was existed. 2. System prompt error message if the password length is no longer than 6 lengths.

Table 3.3 Register Use case descriptions

Use Case	Login
Aim	Enter the registered email and password to proceed the system.
Actor	Current User
Trigger	When user click Sign In button
Main Flow	1. User submits the correct email and password. 2. Browse user to home page once authenticated. 3. Use case end.
Alternative Flow	1. Invalid password and email will prompt error message. 2. Click forget password text to reset password.

Table 3.4 Login Use Case Descriptions

Use Case	Add Category
Aim	To create a category for income or expense.
Actor	Current user
Trigger	When user click done button
Main Flow	1. User selects the type of category (income/expense) 2. User inserts the category name. 3. User select the icon used for the category. 4. Use case end.
Alternative Flow	1. Error message prompt out if did not name the category. 2. Error message prompt out if did not select an icon. 3. Error message prompt out if the category name already existed.

Table 3.5 Add Category Use Case Descriptions

Use Case	Add Account
Aim	To create a transaction account of the cash flow income and expense.
Actor	Current User
Trigger	When user click add button from the account list
Main Flow	1. User selects the icon used for the account. 2. User enters an account name. 3. User enters a starting balance of account. 4. Use case end
Alternative Flow	1. Error message prompt out if did not name the account. 2. Error message prompt out if did not select an icon.

Table 3.6 Add Account Use Case Description

Use Case	Edit Profile
Aim	To reset the password and update the profile information.
Actor	Current User
Trigger	When user click save button
Main Flow	1. User able to update their username. 2. User able to update their password.
Alternative Flow	1. Error message of invalid current password prompt out during update the password. 2. Error message of invalid new password prompt out if the length of new password no longer than 6 lengths.

Table 3.7 Edit Profile Use Case Description

Use Case	Add Transaction
Aim	To add a transaction data into database
Actor	Current User
Trigger	When user click save button
Main Flow	1. User able to select the type of transaction. 2. User able to select the category of the transaction that created by user. 3. User able to select the account of the transaction that created before. 4. User able to insert the amount of transaction. 5. User can give description to the transaction data. 6. User can select the date of transaction save via calendar. 7. Use case end.
Alternative Flow	1. Error message of did not select the type of transaction. 2. Error message dialog prompt out if did not enter amount. 3. Error message of did not select category of transaction.

Table 3.8 Add Transaction Use Case Description

Use Case	Delete Transaction
Aim	Able to delete the transaction data that stored from database
Actor	Current User
Trigger	By swipe the transaction data from home page
Main Flow	User able to delete the transaction data created.
Alternative Flow	Confirm dialog prompt out to confirm delete the transaction.

Table 3.9 Delete Transaction Use Case Description

Use Case	Modify Transaction
Aim	Able to modify the transaction data made before.
Actor	Current User
Trigger	By click the transaction data from home page
Main Flow	User able to update the transaction data type, category, amount, date, account and description created before.
Alternative Flow	None

Table 3.10 Modify Transaction Use Case Description

Use Case	Category List
Aim	Able to review and delete the category created from add category page.
Actor	Current User
Trigger	By click delete button for the category
Main Flow	1. User able to review the category list separate as income category list and expense category list created before. 2. User able to delete the category from list.
Alternative Flow	1. Confirm delete dialog will prompt out to delete selected category. 2. User can click add button to proceed to the step add category.

Table 3.11 Category List Use Case Description

Use Case	Account List
Aim	Able to review account details and delete the account created from add account page.
Actor	Current User
Trigger	By long press to delete the account or select the account
Main Flow	1. User able to review the account list created. 2. User able to delete the account created. 3. User able to check the cashflow of the account.
Alternative Flow	1. Confirm delete dialog will prompt out when delete selected account. 2. User can click add button to proceed to the step add account.

Table 3.12 Account list Use Case Description

Use Case	Calendar Page
Aim	Able to review transaction data by calendar mode.
Actor	Current User
Trigger	By select the date from the calendar provided
Main Flow	1. User able to check the transaction data by select the date from the calendar provided. 2. Transaction data will only display the date selected.
Alternative Flow	Transaction not found displayed if selected date has no transaction data for the date.

Table 3.13 Calendar Page Use Case Description

Use Case	Pie Chart View
Aim	To review the statistic of the transaction data in pie chart view
Actor	Current User
Trigger	By select options date and type of transaction
Main Flow	1. User able to review the statistic analysis of transaction data in pie chart form. 2. User able to select the date range of the statistic view. 3. User able to check the summarize of category. 4. User able to select pie chart for income transaction or expense transaction.
Alternative Flow	1. User can click bar chart icon to view the weekly bar chart statistical analysis. 2. User can click the summarize of category to review the details of the category spend.

Table 3.14 Pie Chart View Use Case Description

Use Case	Bar Chart View
Aim	To review the weekly statistic analysis of transaction data in bar chart view.
Actor	Current User
Trigger	By select bar icon from Pie chart View
Main Flow	1. User able to review the statistical analysis of transaction data in bar chart view. 2. User able to click next and previous icon to check the weekly from different week.
Alternative Flow	No transaction data for this week displayed if there is no any transaction made for selected week.

Table 3.15 Bar Chart View Use Case Description

Use Case	Export Excel
Aim	To export the data into a excel file sheet
Actor	Current User
Trigger	Select Export as excel from setting page
Main Flow	1. User able to select the year of data to export. 2. User able to select months of data to export.
Alternative Flow	Error message prompt out if user has not selected any month.

Table 3.16 Export Excel Use Case Description

Use Case	Search Filter
Aim	To search the specific transaction data added before
Actor	Current User
Trigger	By enter the keywords search
Main Flow	1. User able to search the transaction data by entered characteristics of the transaction data include, category name, account name, amount and type of data. 2. User able to search the transaction data by select the start date range and end date range.
Alternative Flow	None

Table 3.17 Search Filter Use Case

3.6 Activity Diagram

3.6.1 User Authentication

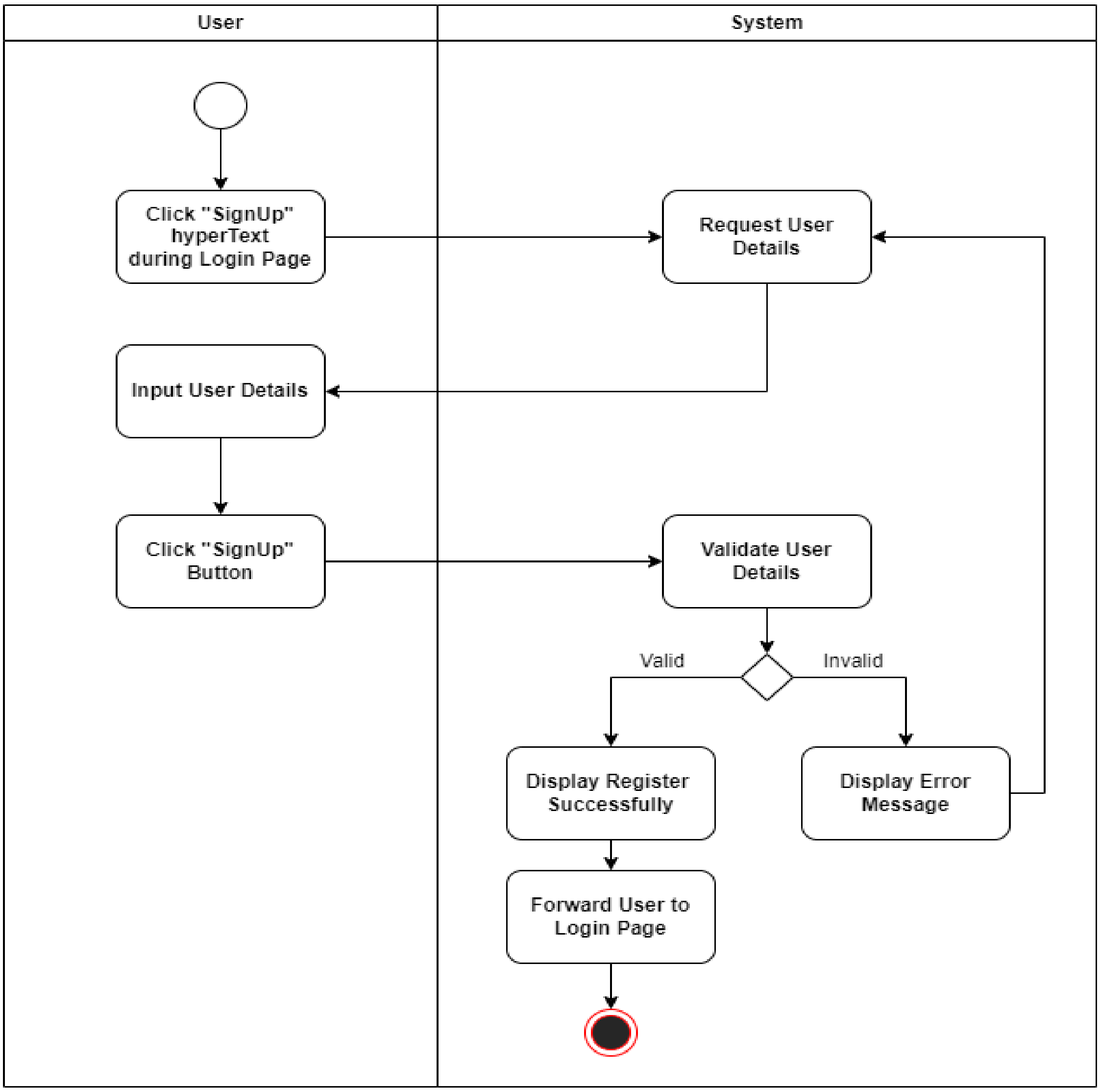


Figure 3.6 Sign Up Activity Diagram

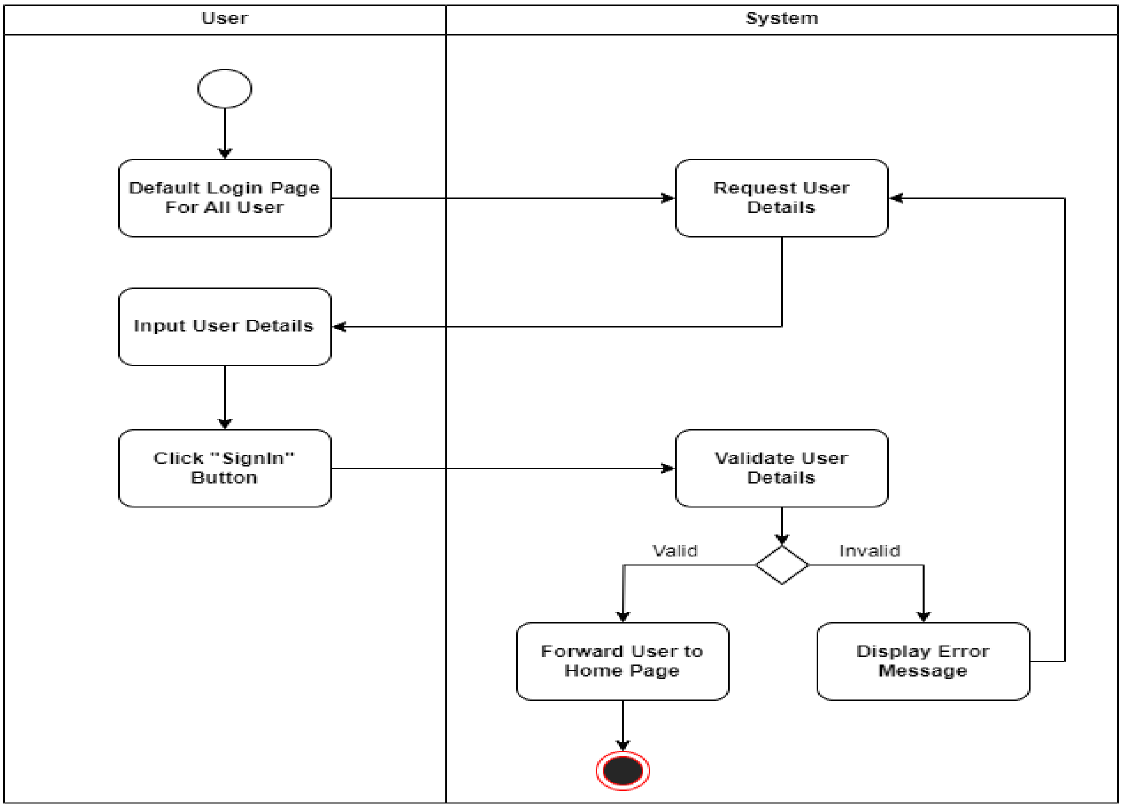


Figure 3.7 Sign In Activity Diagram

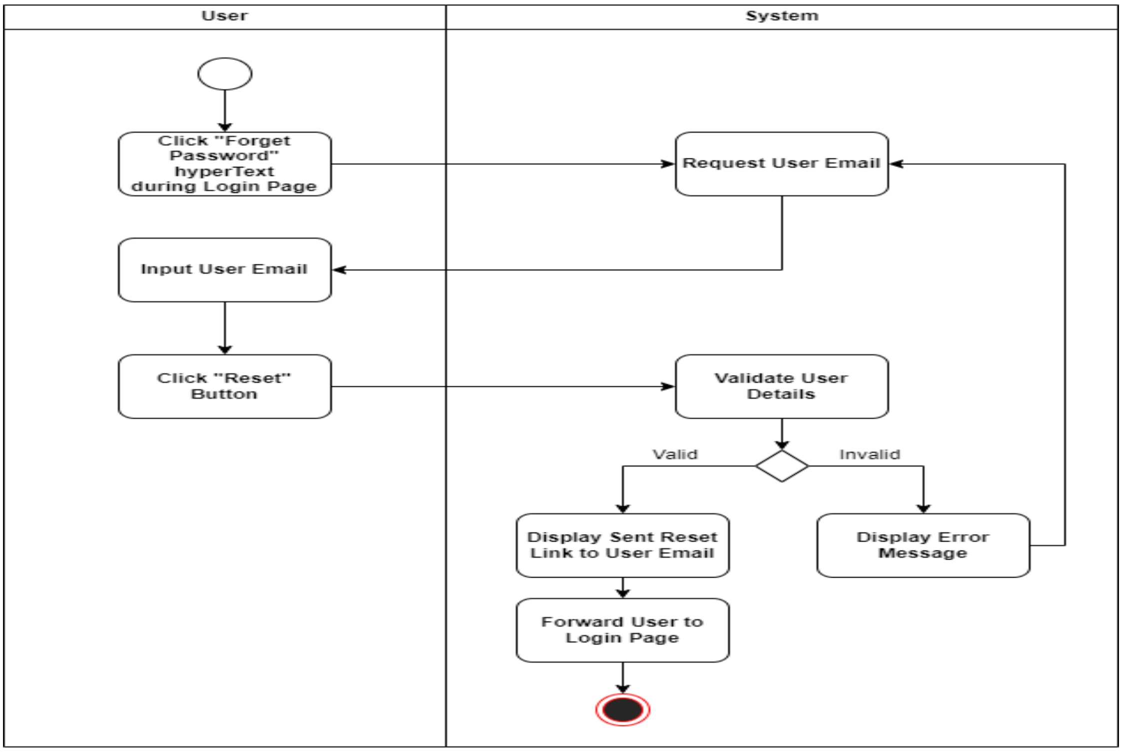


Figure 3.8 Reset Password Activity Diagram

3.6.2 Manage Transaction Data

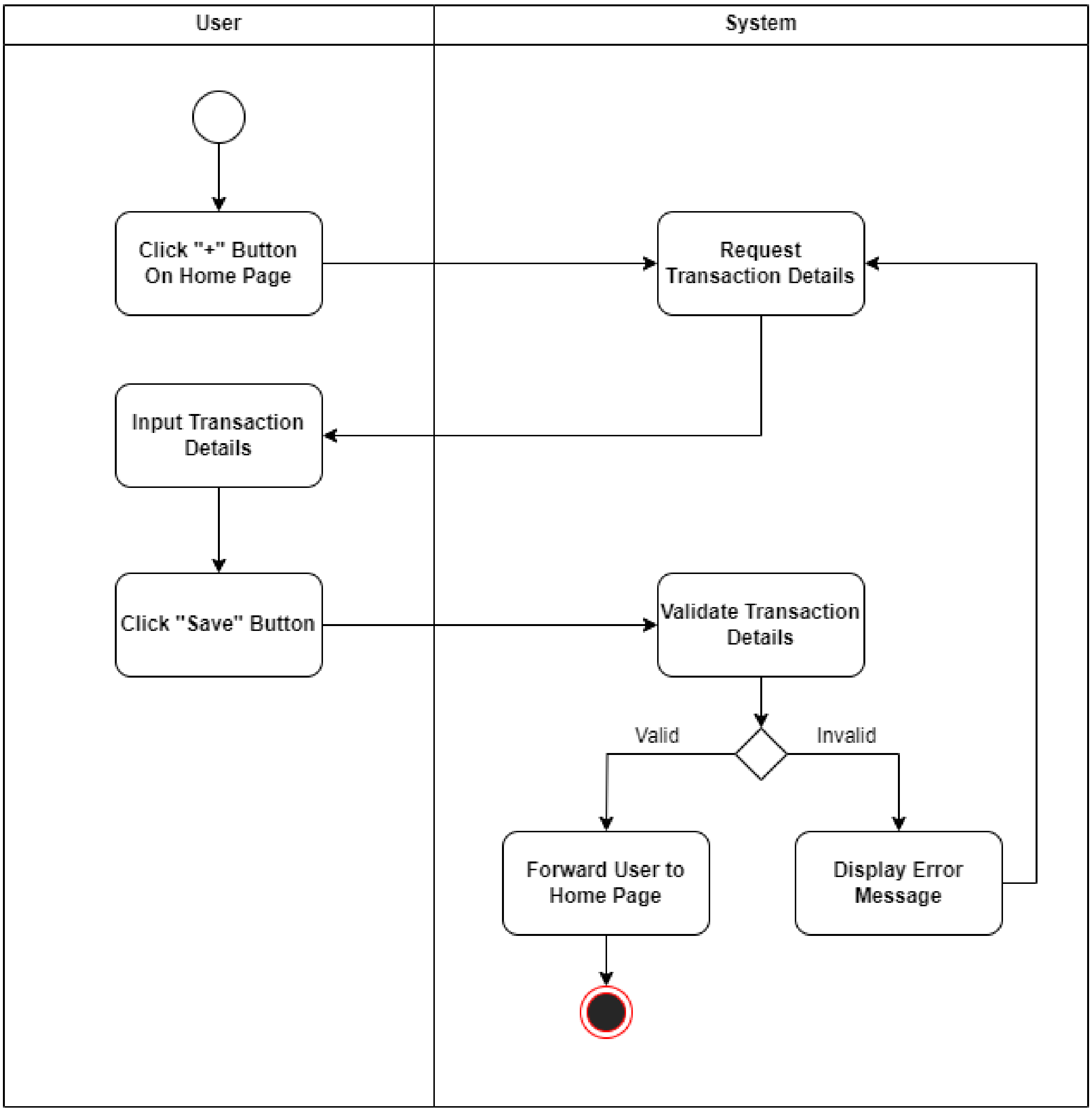


Figure 3.9 Add Transaction Activity Diagram

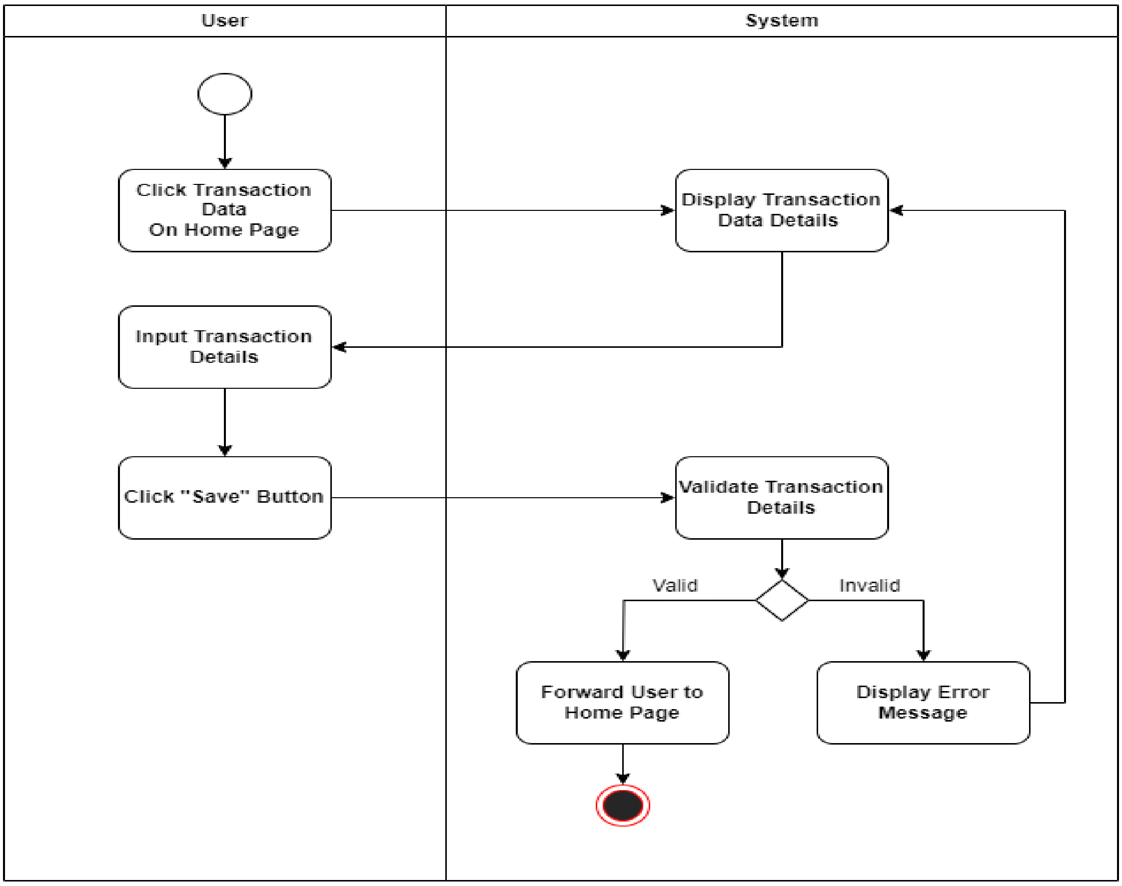


Figure 3.10 Modify Transaction Activity Diagram

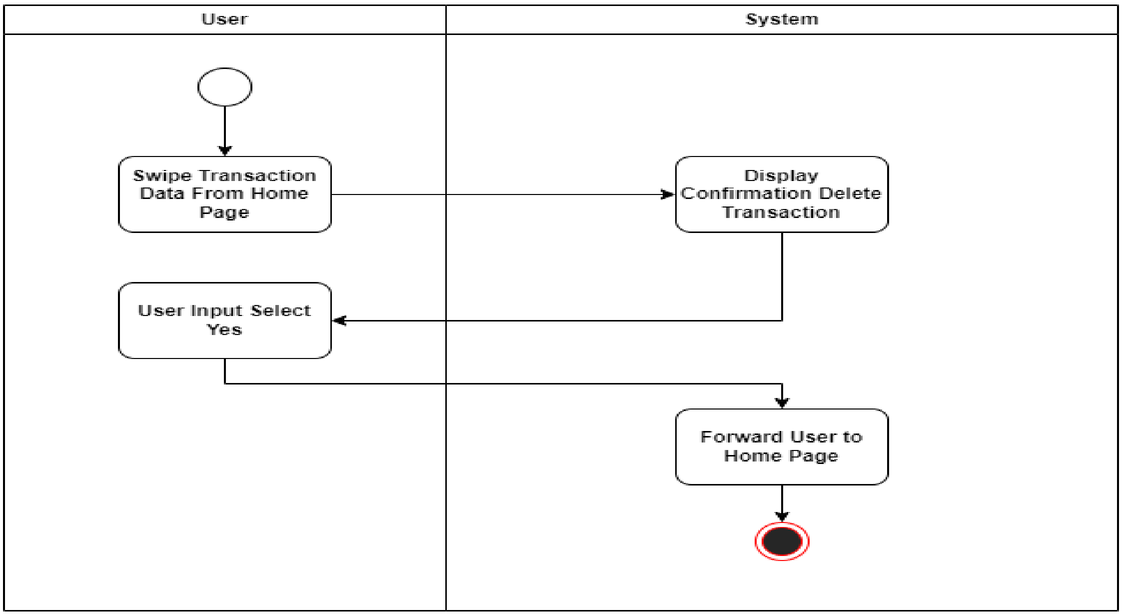


Figure 3.11 Delete Transaction Activity Diagram

3.6.3 Manage Data Virtualization

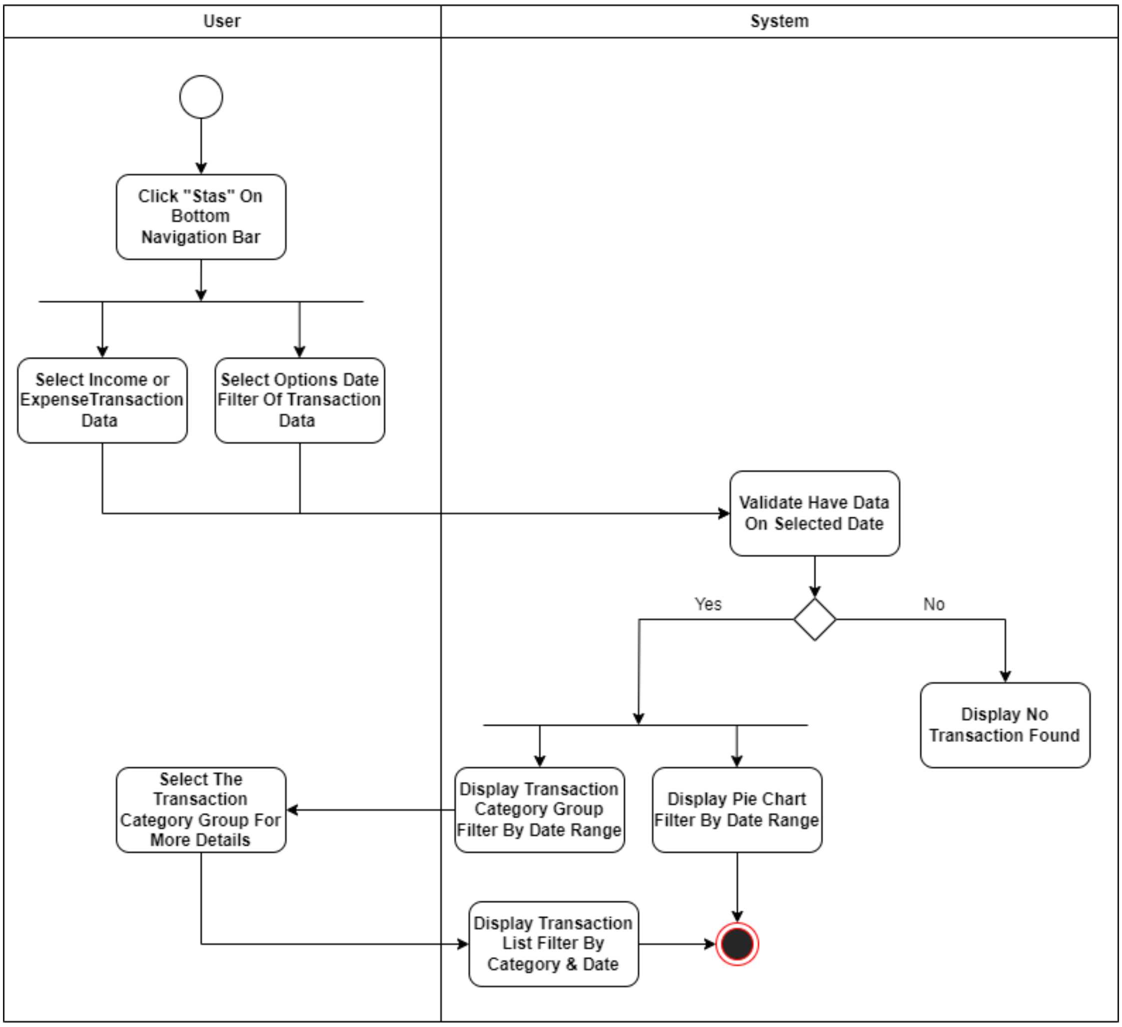


Figure 3.12 Pie Chart View Activity Diagram

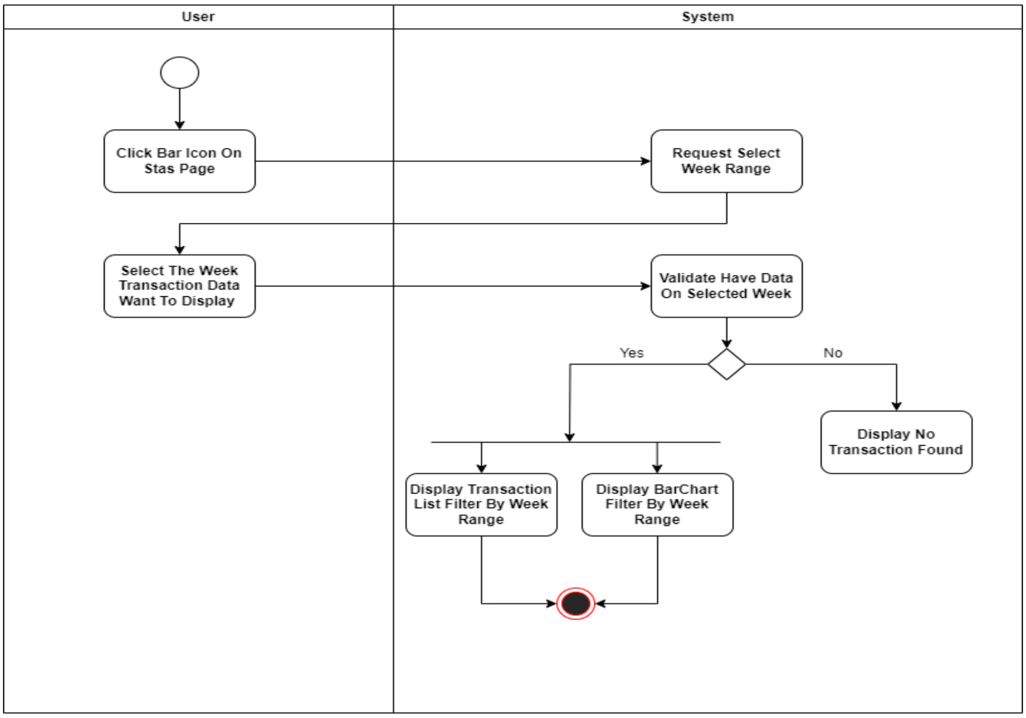


Figure 3.13 Weekly Bar Chart View Activity Diagram

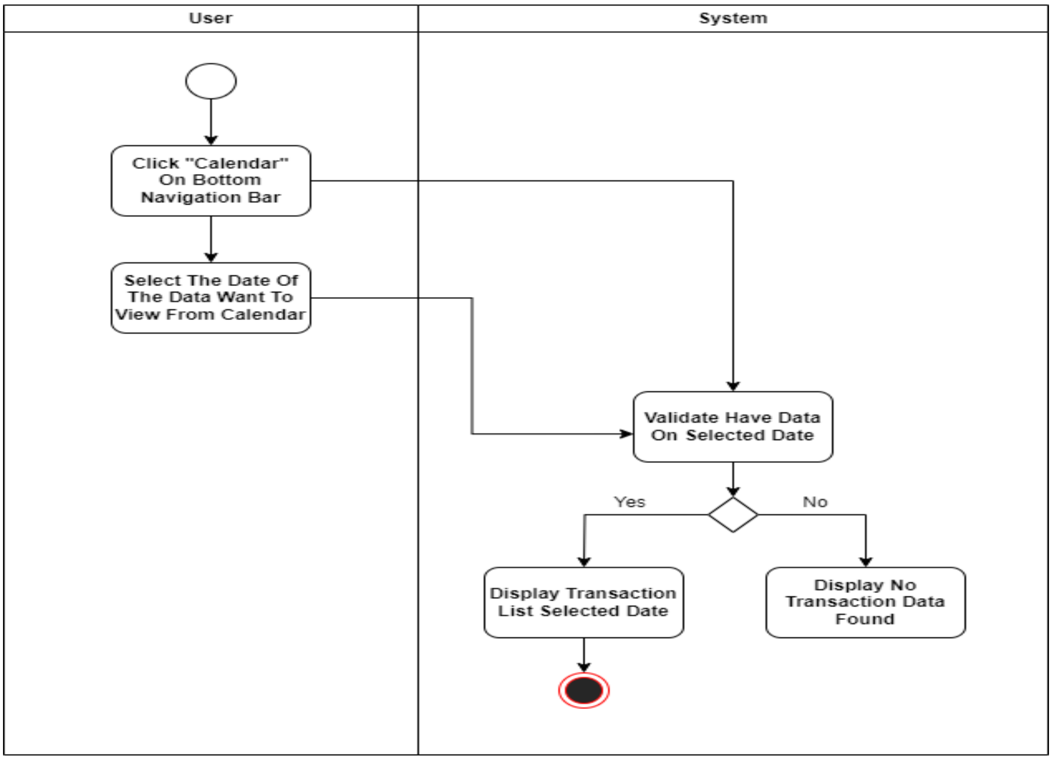


Figure 3.14 Calendar View Activity Diagram

3.6.4 Manage User Account Setting

3.6.4.1 Edit Profile

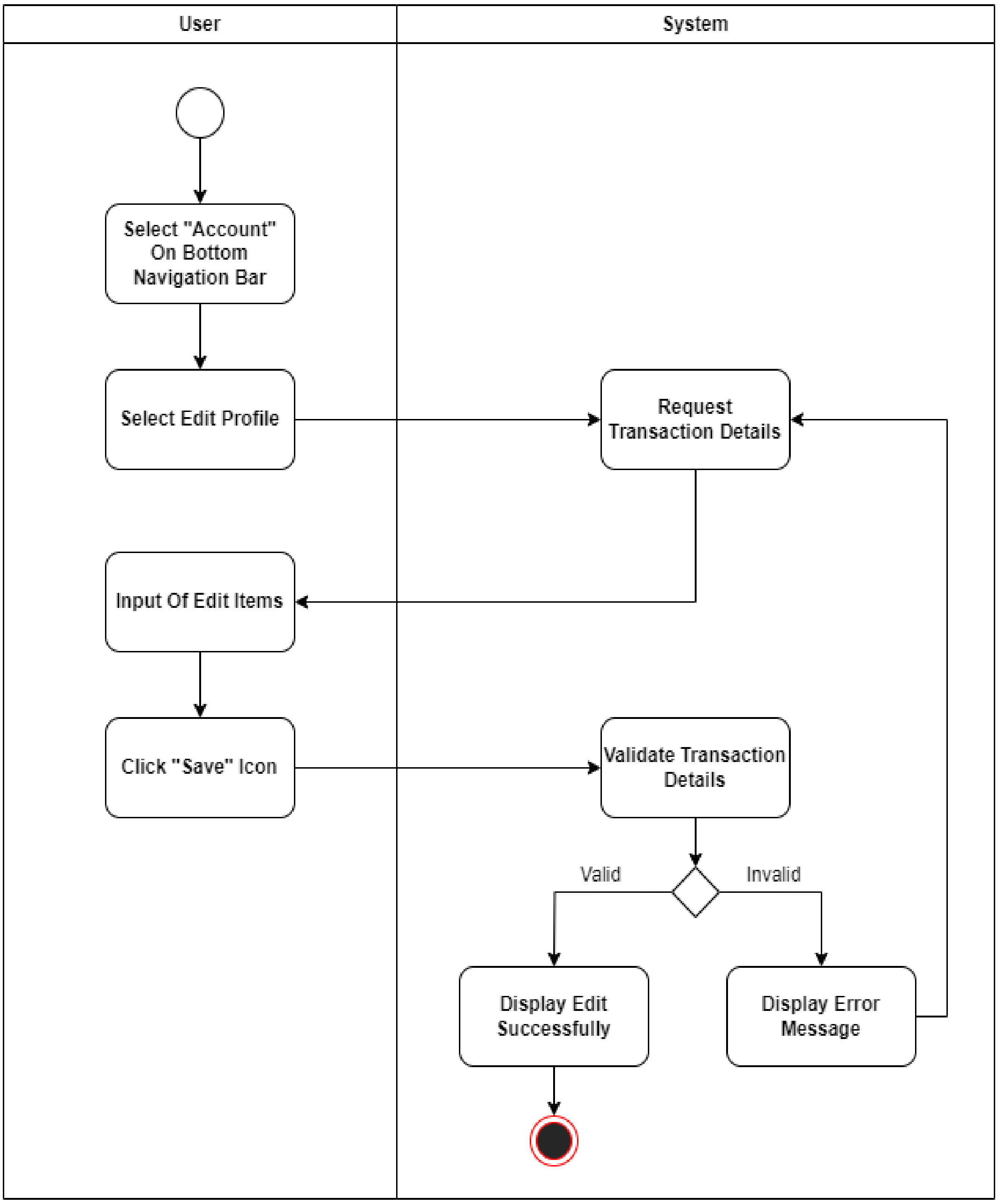


Figure 3.15 Edit Profile Activity Diagram

3.6.4.2 Transaction Data Search

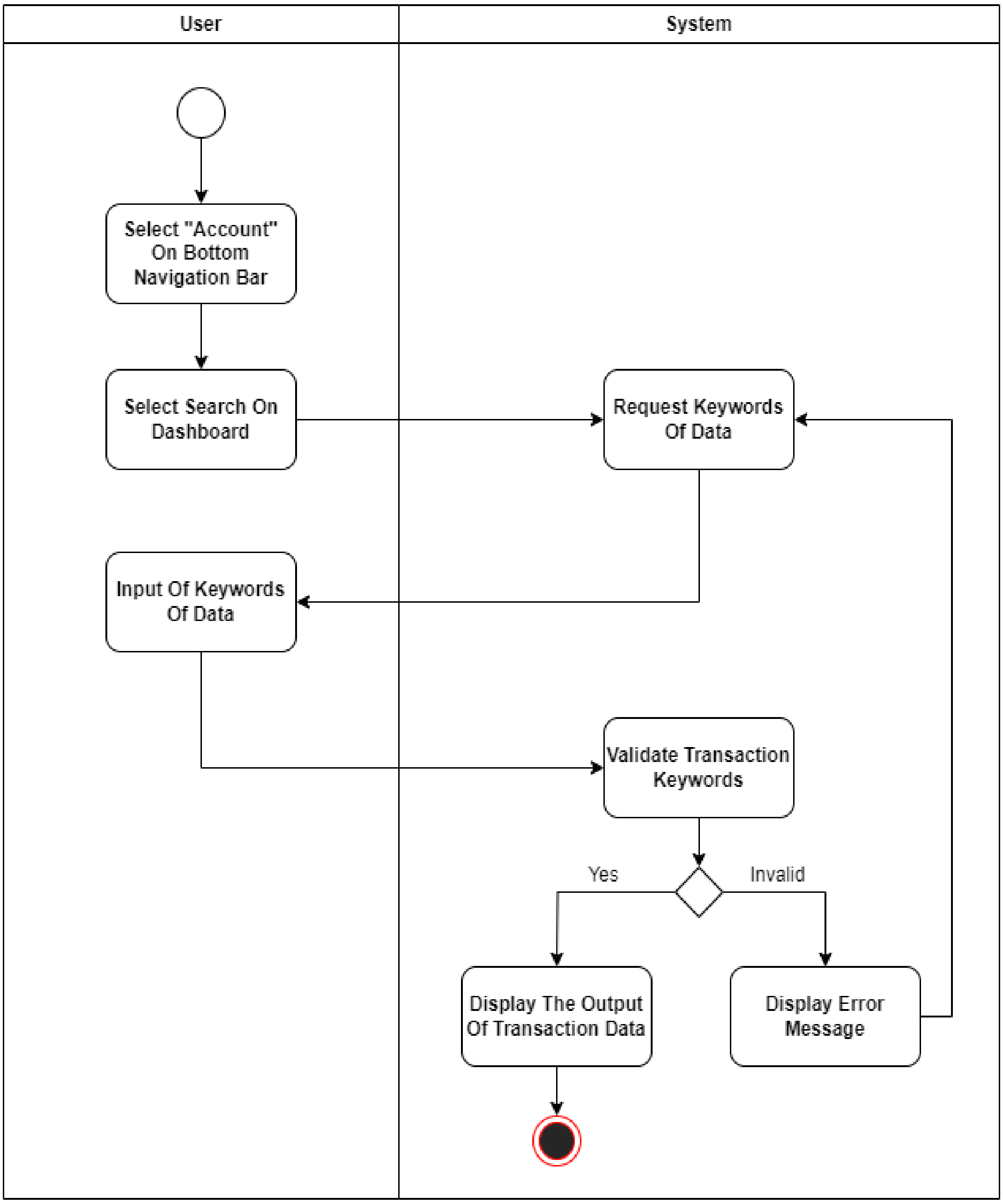


Figure 3.16 Search Data Activity Diagram

3.6.4.3 Manage Transaction Account

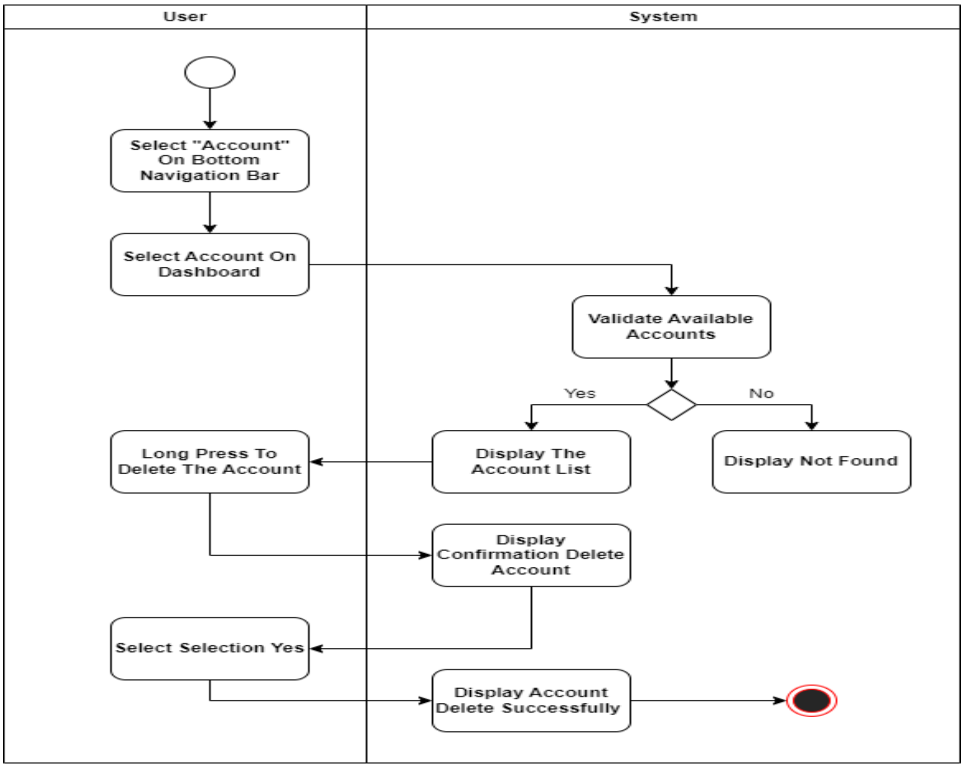


Figure 3.17 Review List & Delete Account Activity Diagram

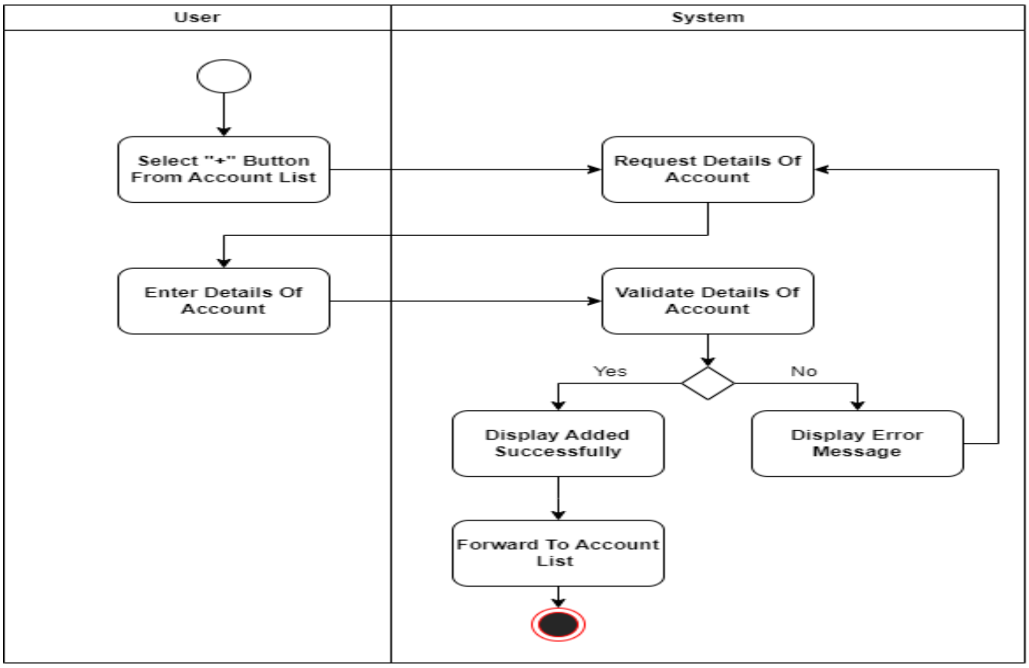


Figure 3.18 Add Transaction Account Activity Diagram

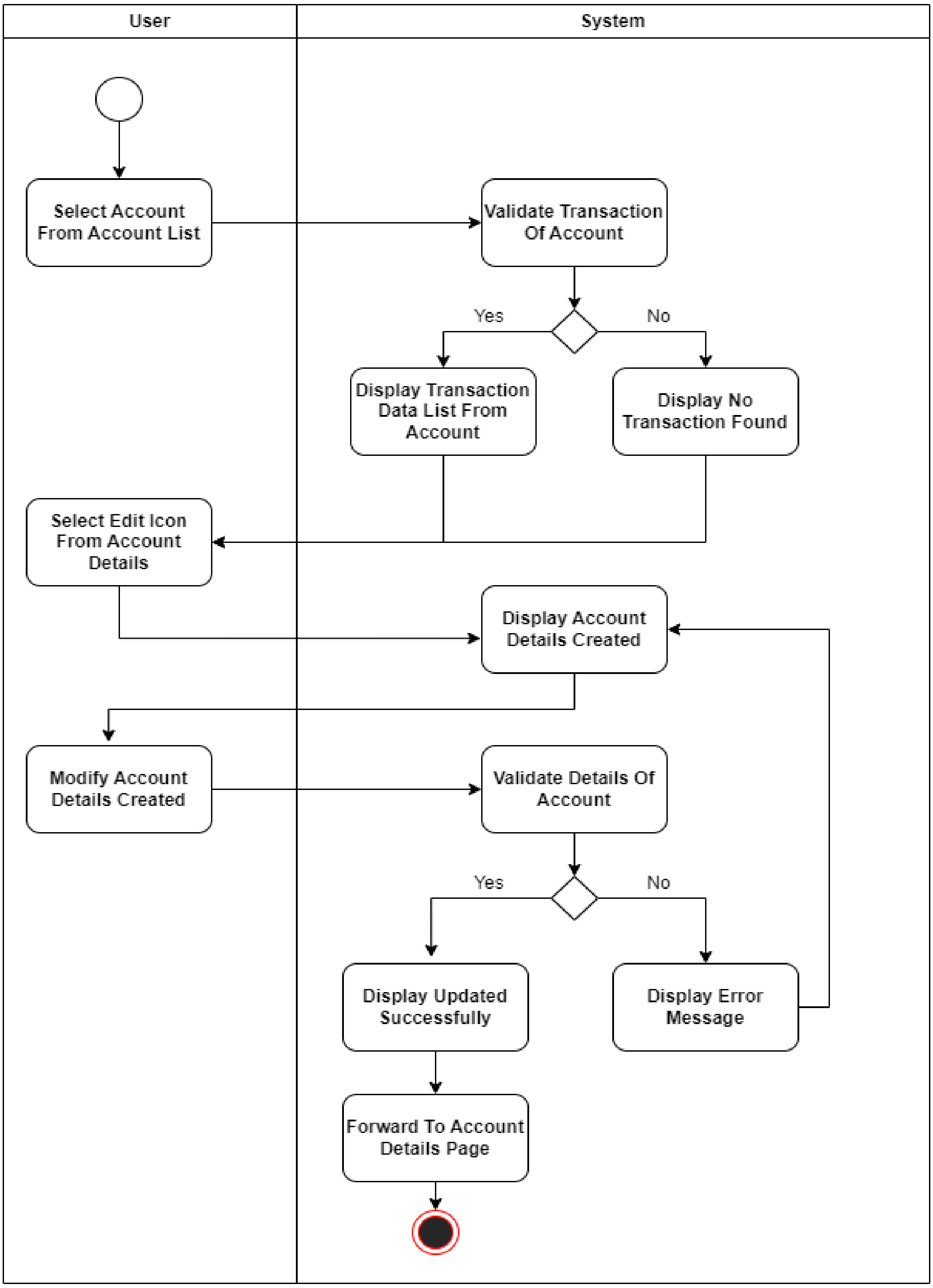


Figure 3.19 Modify Transaction Account Activity Diagram

3.6.4.4 Manage Transaction Category

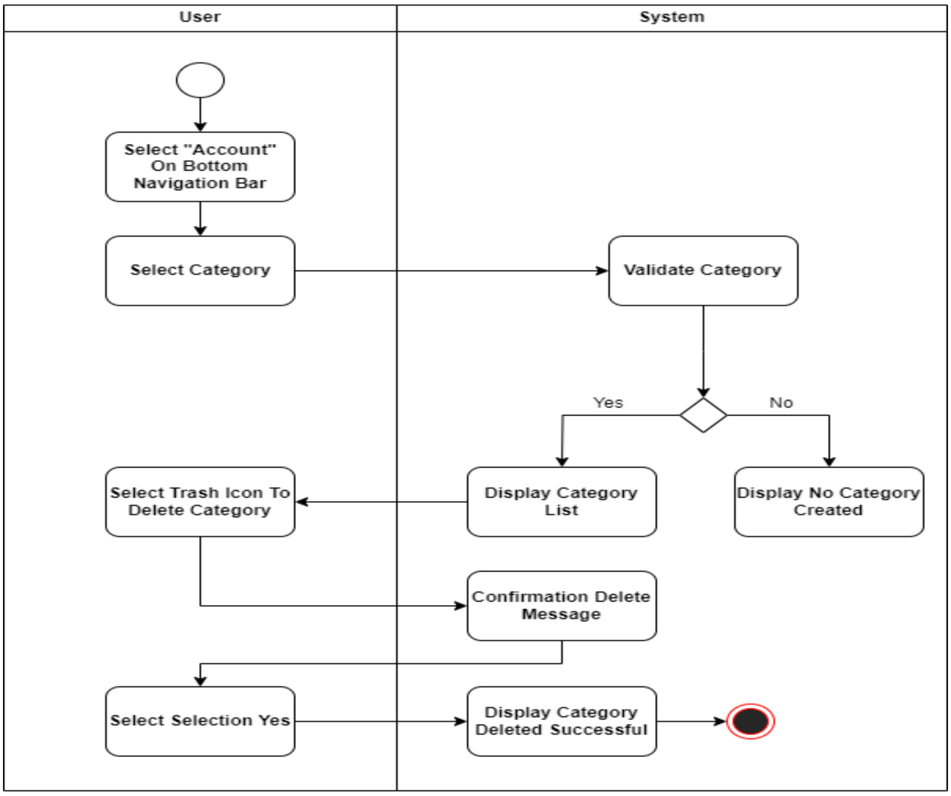


Figure 3.20 Review Category List & Delete Category Activity Diagram

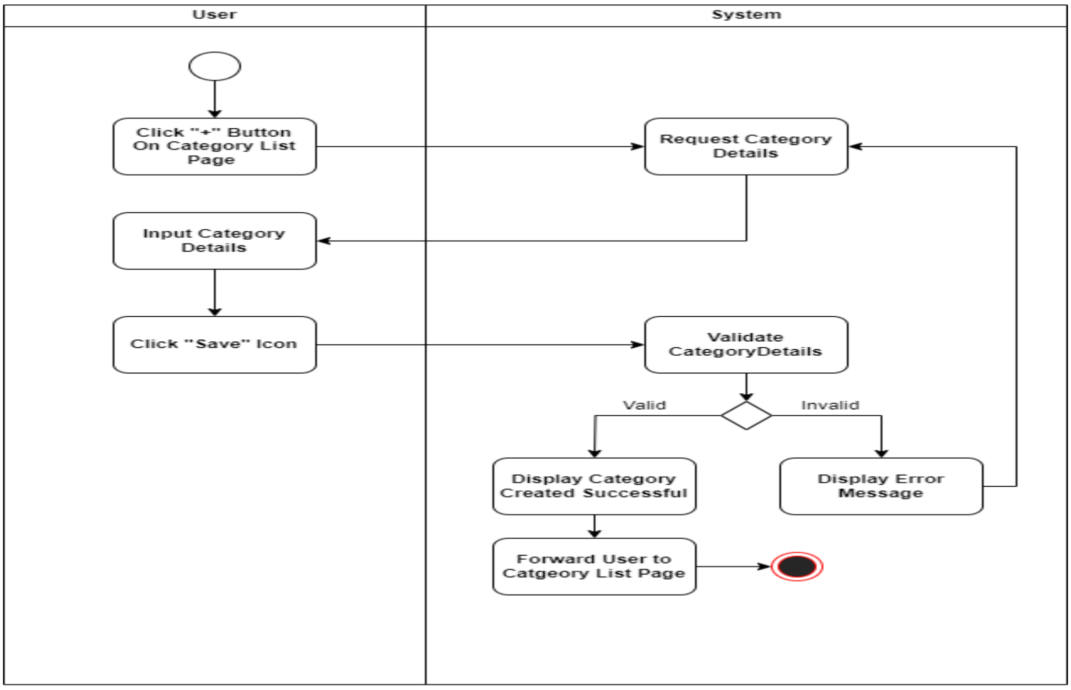


Figure 3.21 Add Category Activity Diagram

3.6.4.5 Export Data Setting

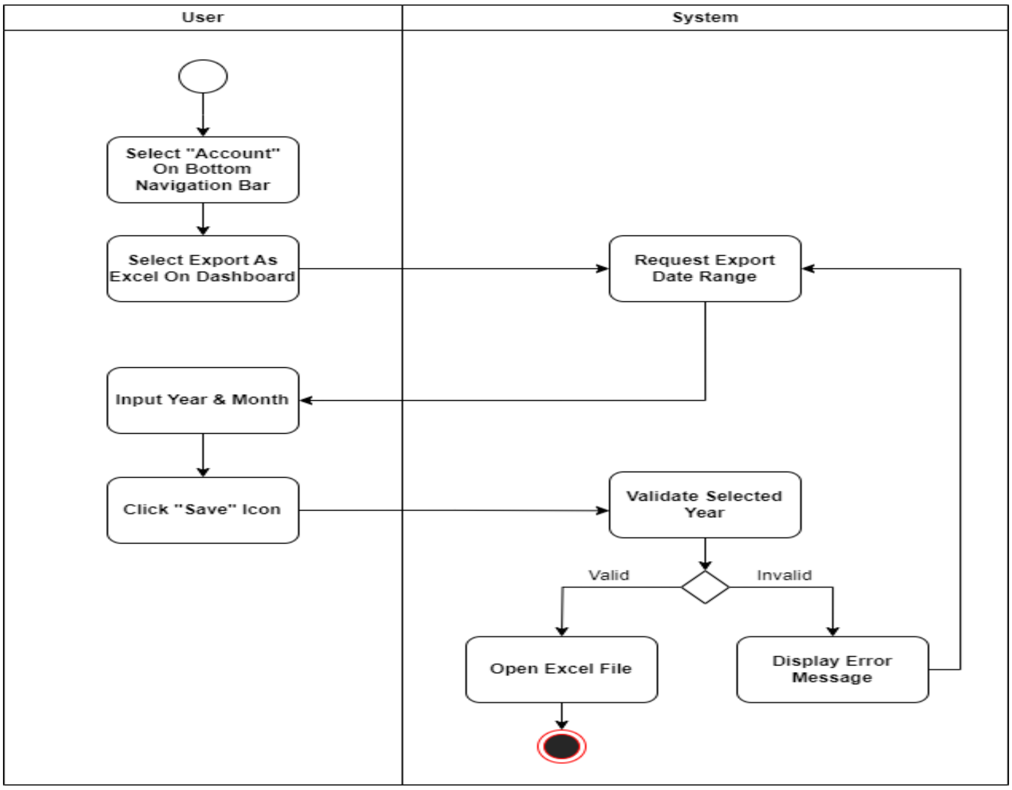


Figure 3.22 Export Excel File Activity Diagram

3.6.4.6 User Sign Out

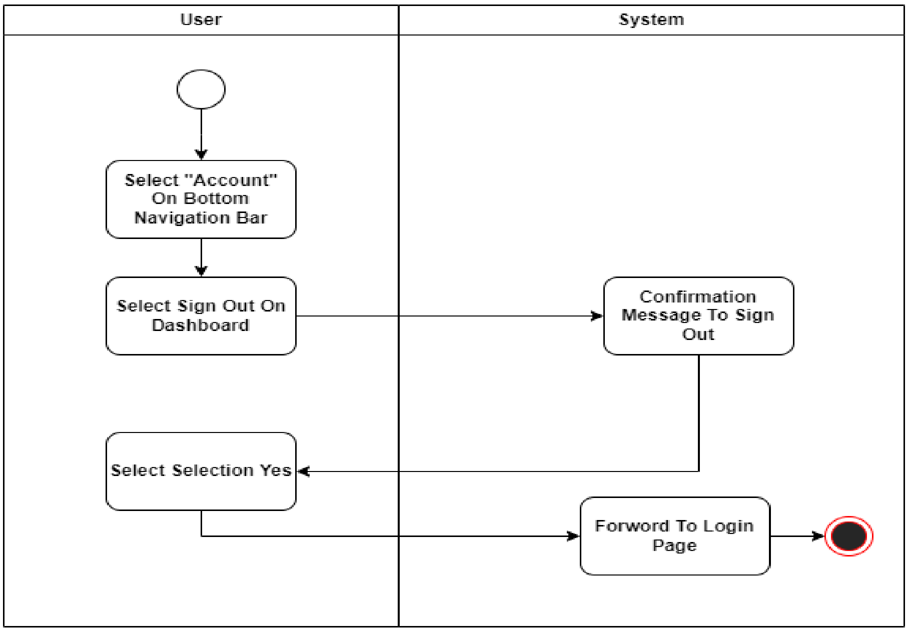


Figure 3.23 User Sign Out Activity Diagram

CHAPTER 4 SYSTEM DESIGN

4.1 System Block Diagram

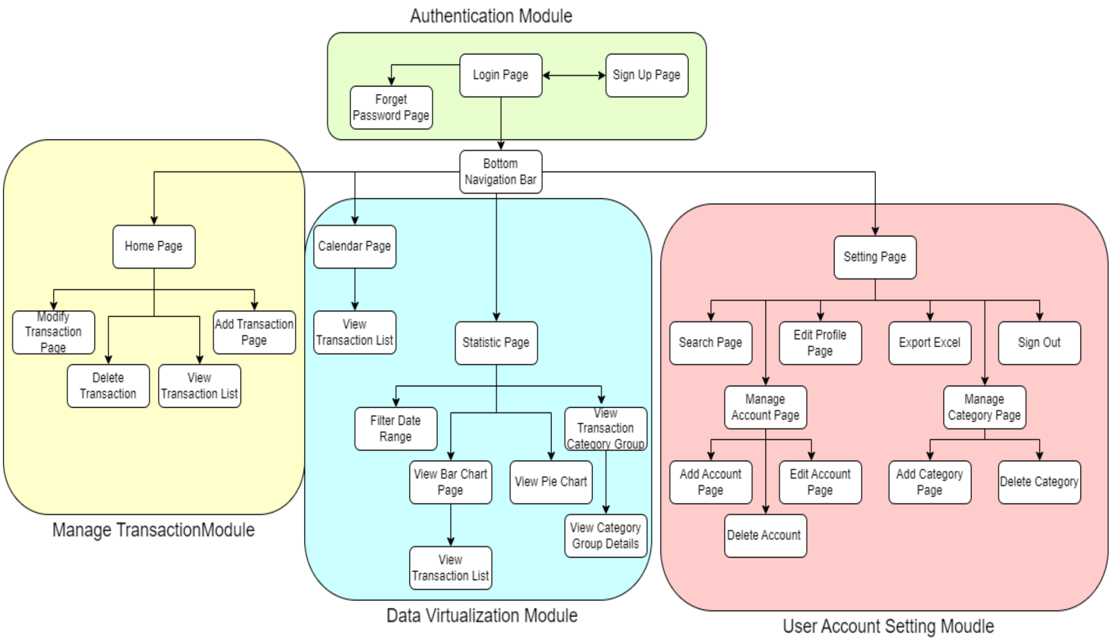


Figure 4.1 System Block Diagram

Above figure showed the overall block diagram of the proposed system. The system can be separate to 4 module which included authentication module, manage transaction module, data virtualization module and user account setting module. Authentication module is a module to register and verify the identity of users before access to the system. This module including login page which a page to validate identity of the user, sign up page which a page creates an access account to gain access to the system and forget password page is a page to reset password if user forgotten them. In addition, manage transaction module is a module that allow user to perform add data, modify data and delete the transaction data. In order to perform modifying and deleting transaction data, a list to view the transaction data will be provided. While data virtualization module is a module allowed user to manipulate transaction data and display in several ways. In this module, user able to view data in bar chart, pie chart, calendar view, category group details and advanced filter the date of data to perform the data analysis. Last, user account setting module will be a module more likely preferences setting of the system which allow user to create, export and search function. In this module users allowed to conduct create category, create account, edit profile, advanced search transaction data, export data and sign out the system fully based on user preferences.

4.2 System Flowchart

4.2.1 Flowchart of Authentication Module

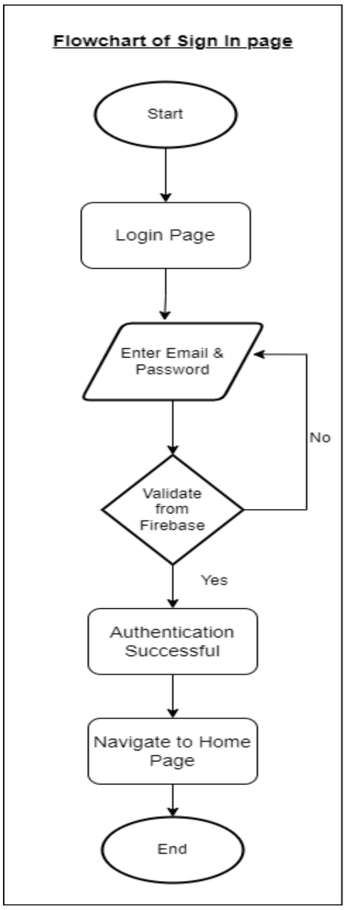


Figure 4.2 Sign In Flowchart

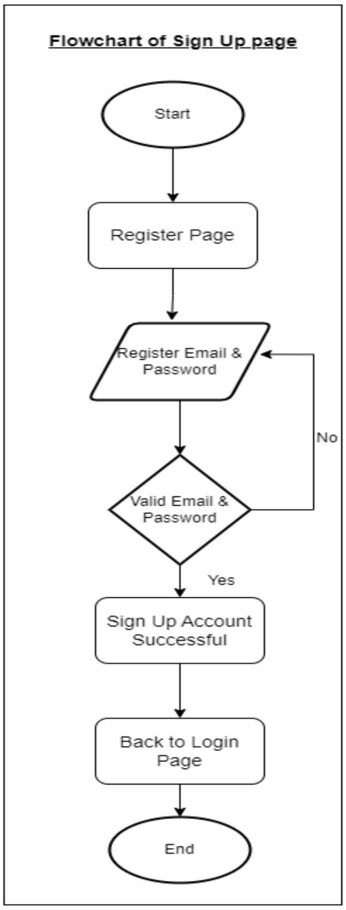


Figure 4.3 Sign Up Flowchart

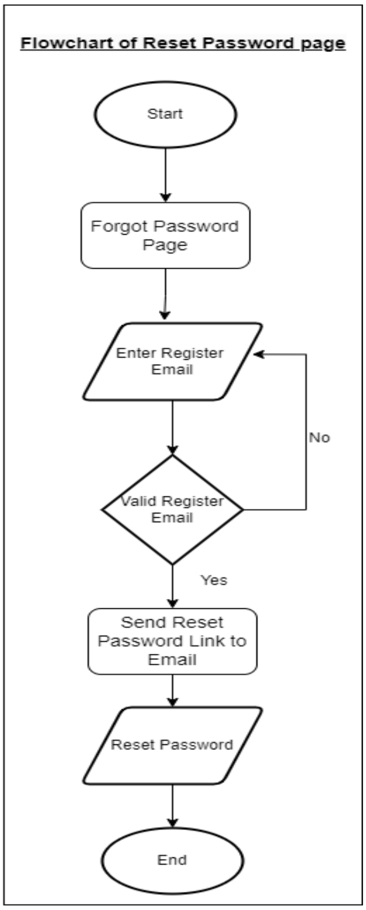


Figure 4.4 Forget Password Flowchart

4.2.2 Manage Transaction Module

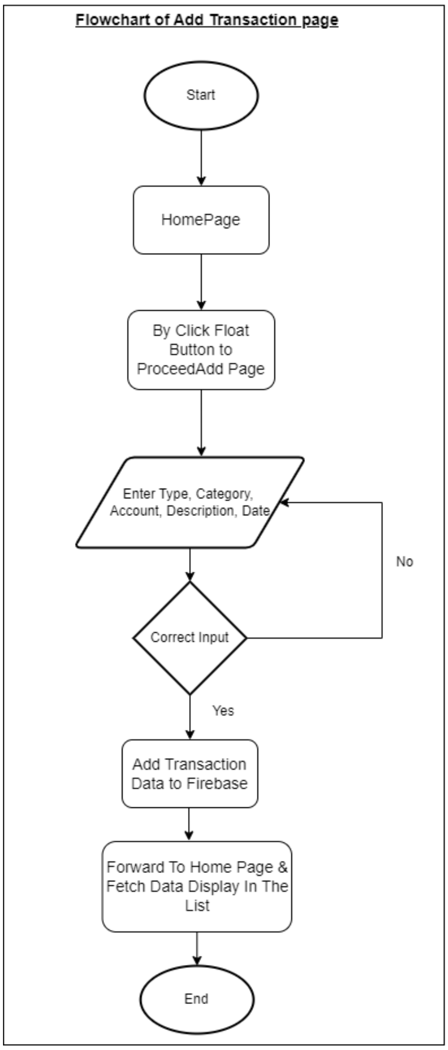


Figure 4.5 Add Transaction Page Flowchart

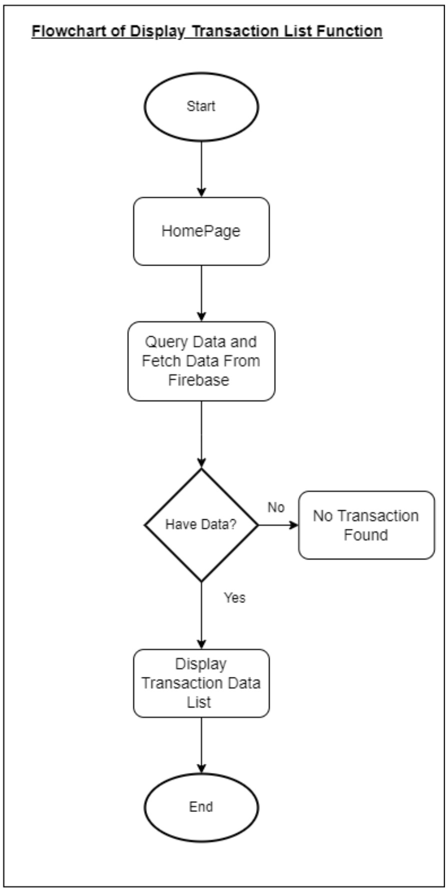


Figure 4.6 Display Transaction List Flowchart

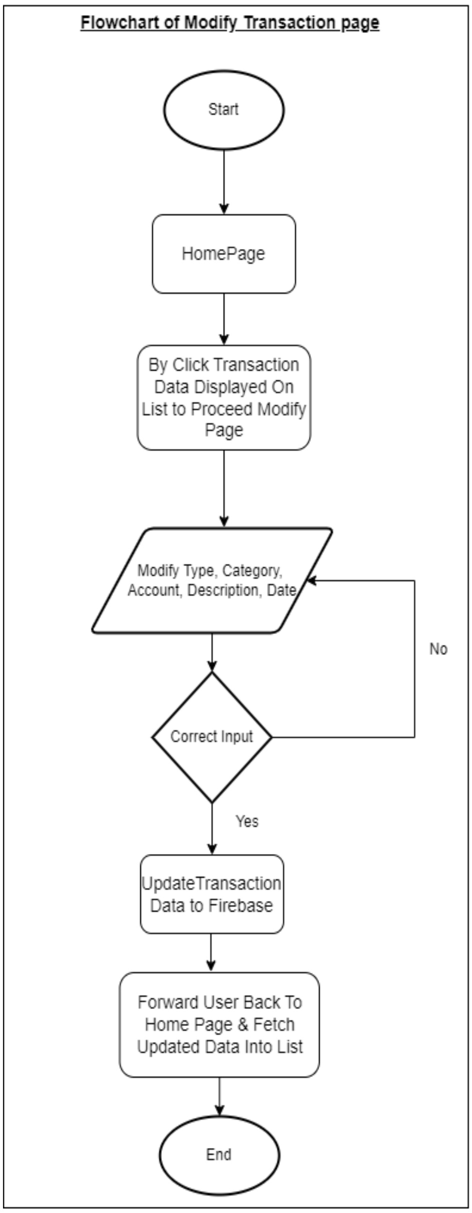


Figure 4.7 Modify Transaction Flowchart

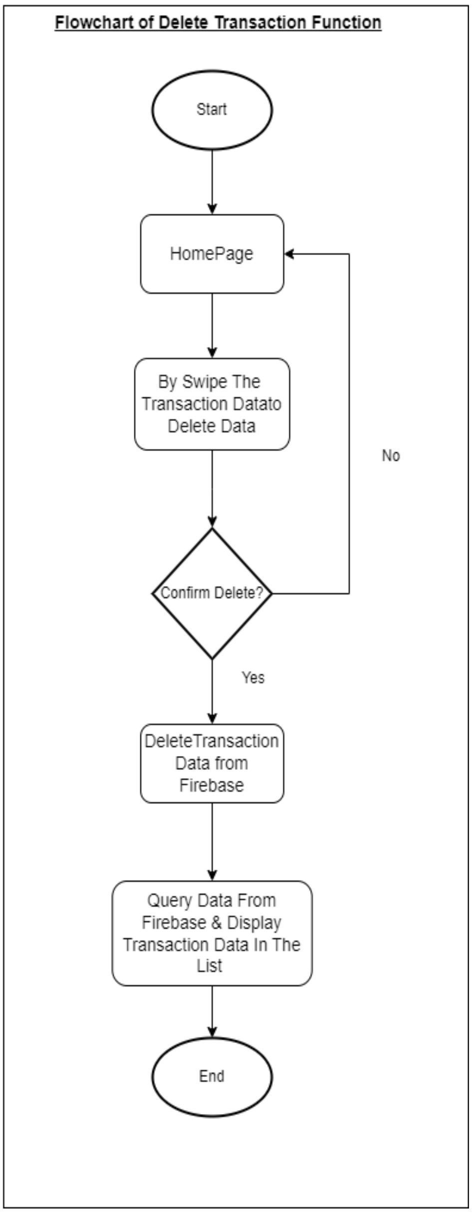


Figure 4.8 Delete Transaction Flowchart

4.2.3 Data Virtualization Module

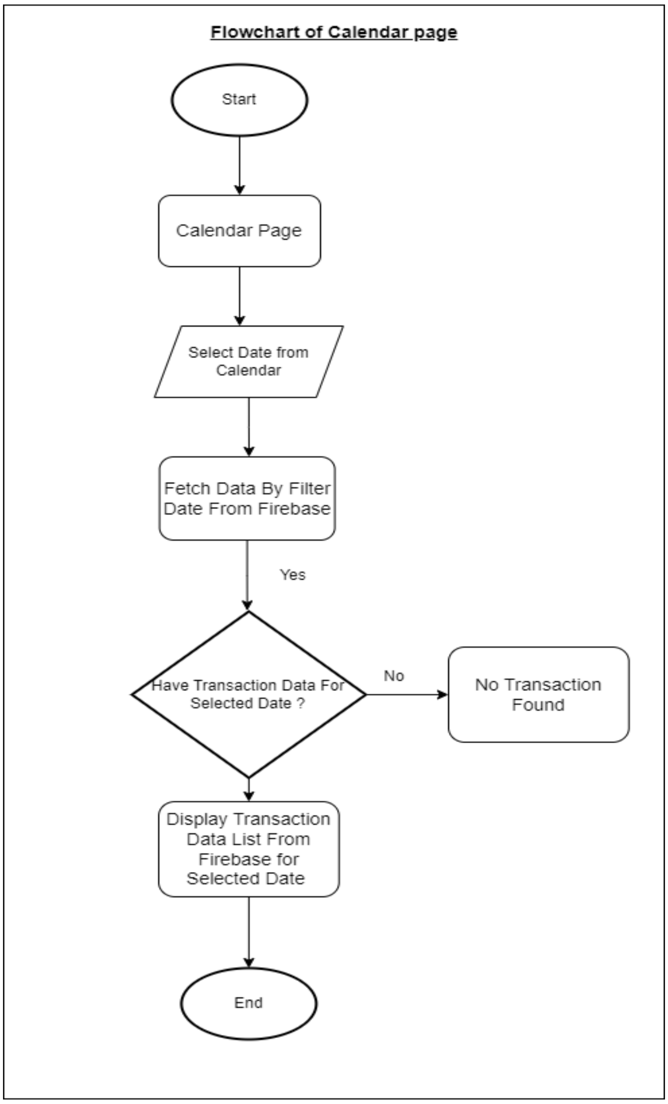


Figure 4.9 Calendar View Flowchart

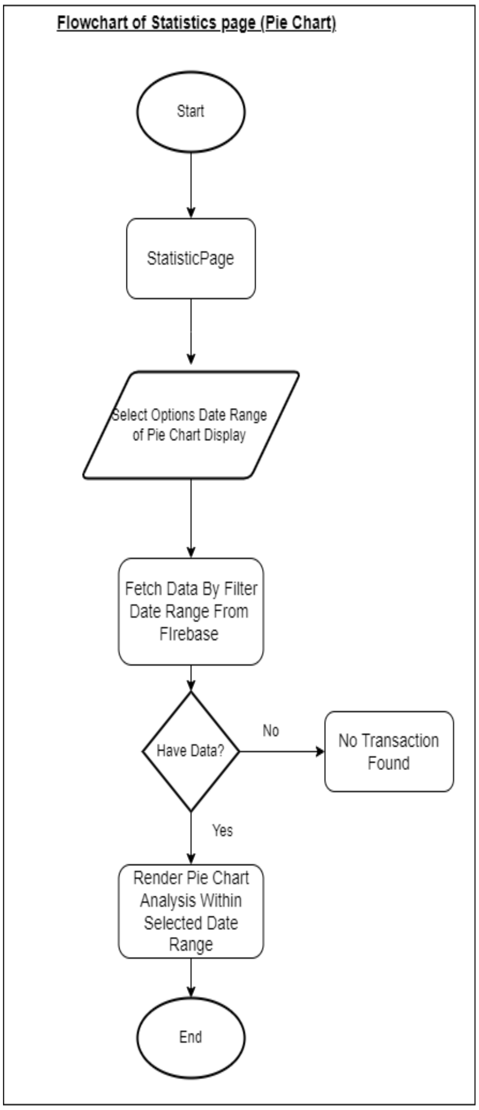


Figure 4.10 Display Pie Chart Flowchart

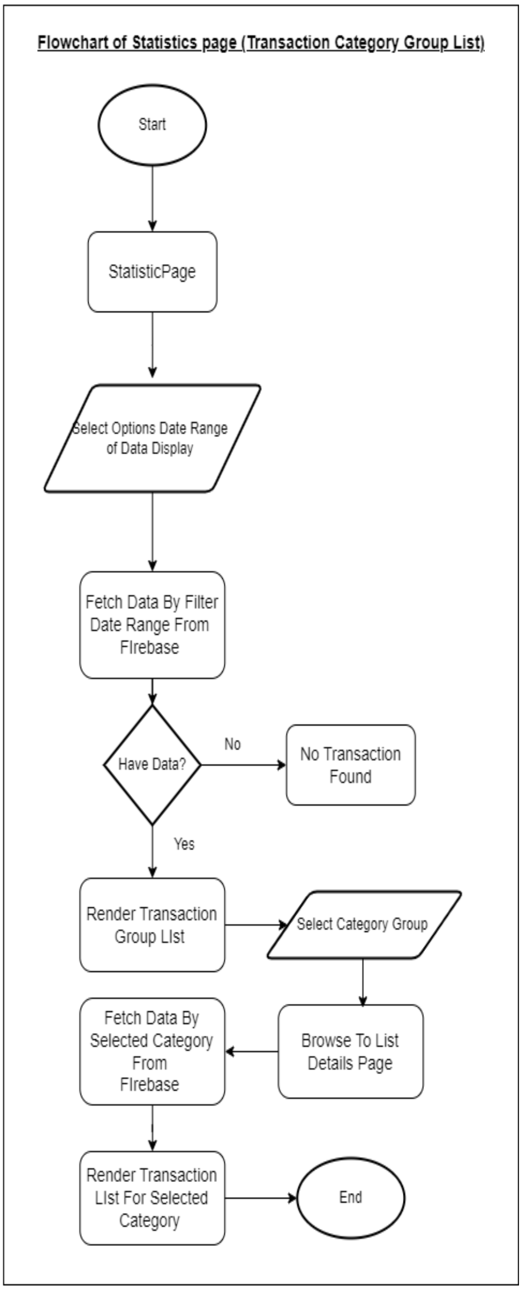


Figure 4.11 Display Transaction List by Category Flowchart

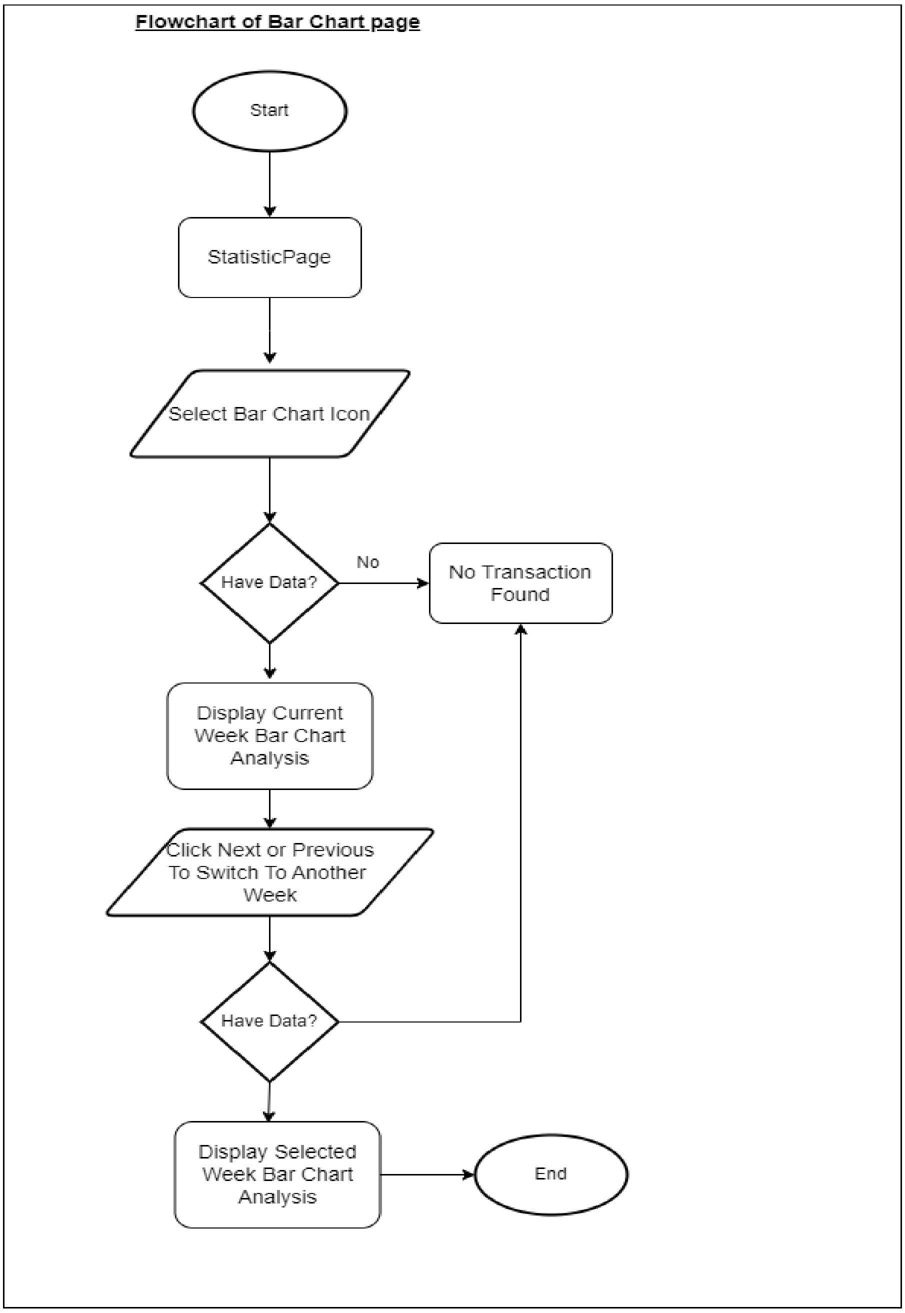
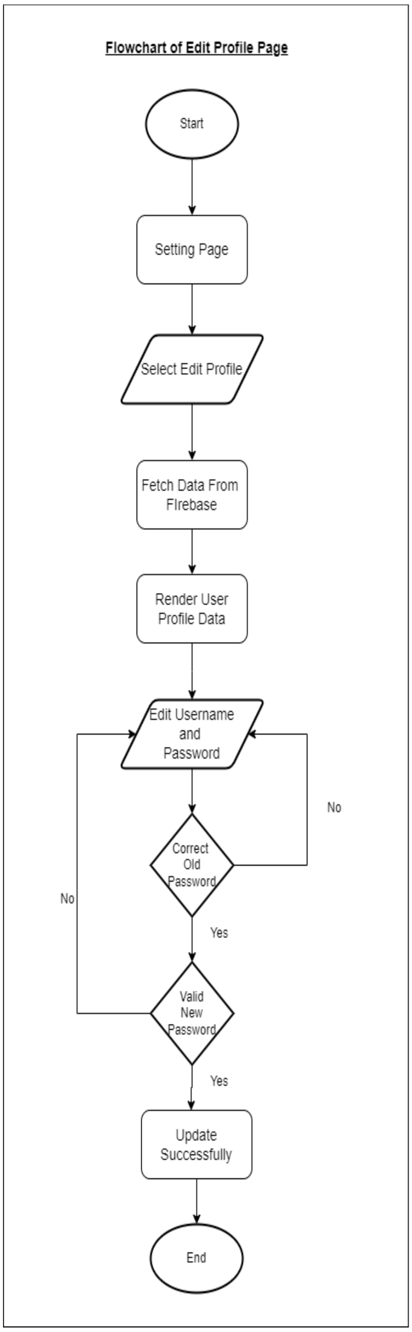


Figure 4.12 Weekly Bar Chart View Flowchart

4.2.4 User Account Setting Module



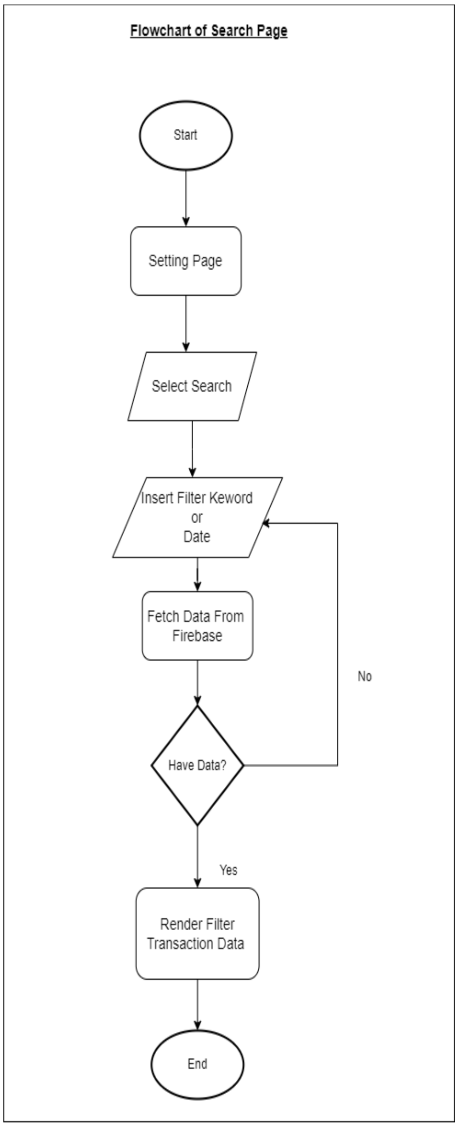


Figure 4.13 Edit Profile Flowchart

Figure 4.14 Search Transaction Flowchart

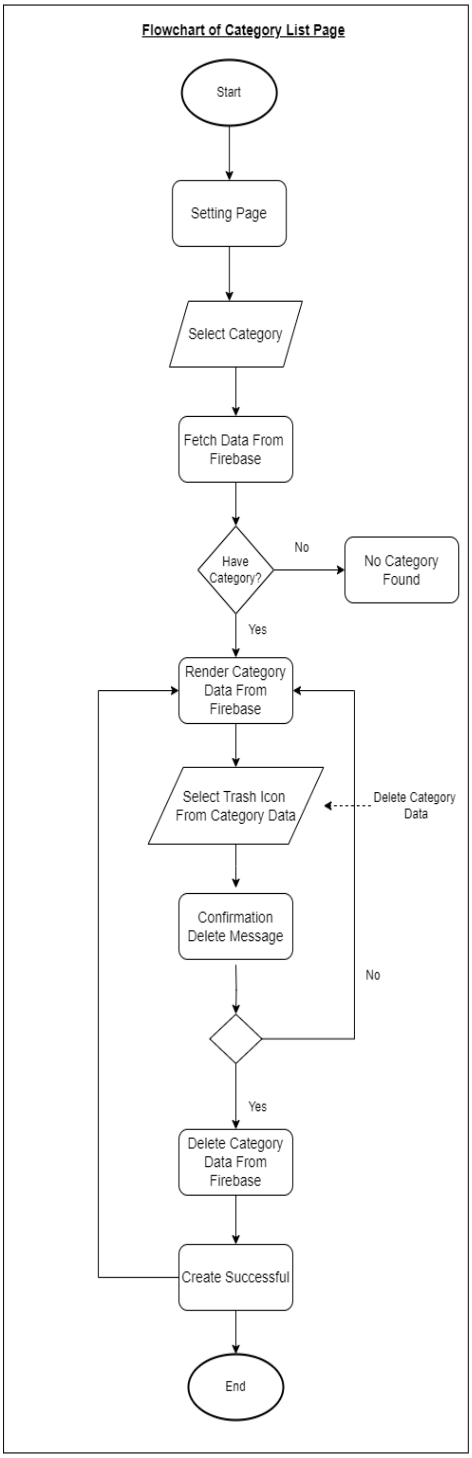


Figure 4.15 Category List Flowchart

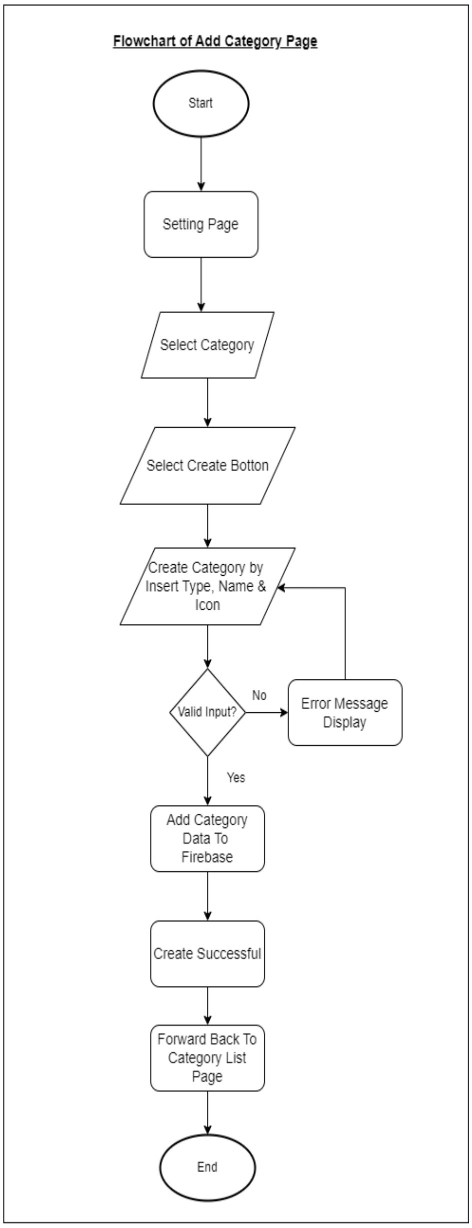


Figure 4.16 Add Category Flowchart

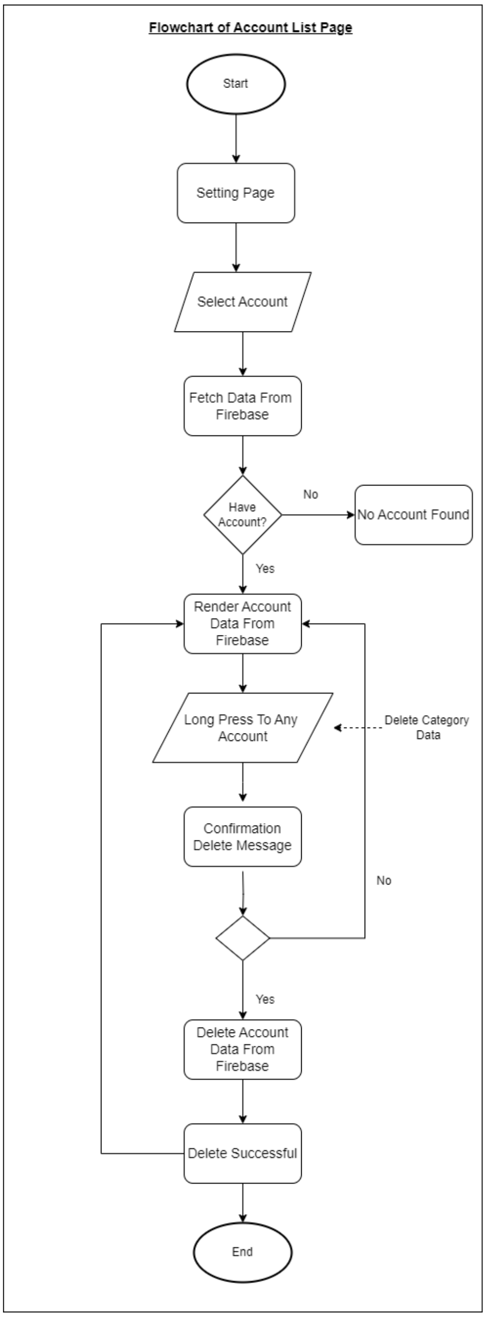


Figure 4.17 Account List Flowchart

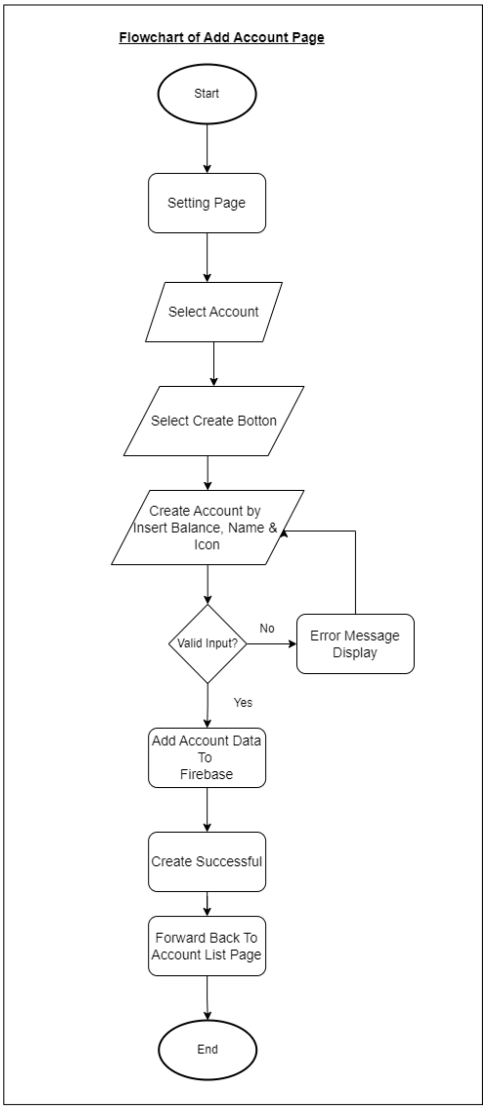


Figure 4.18 Add Account Flowchart

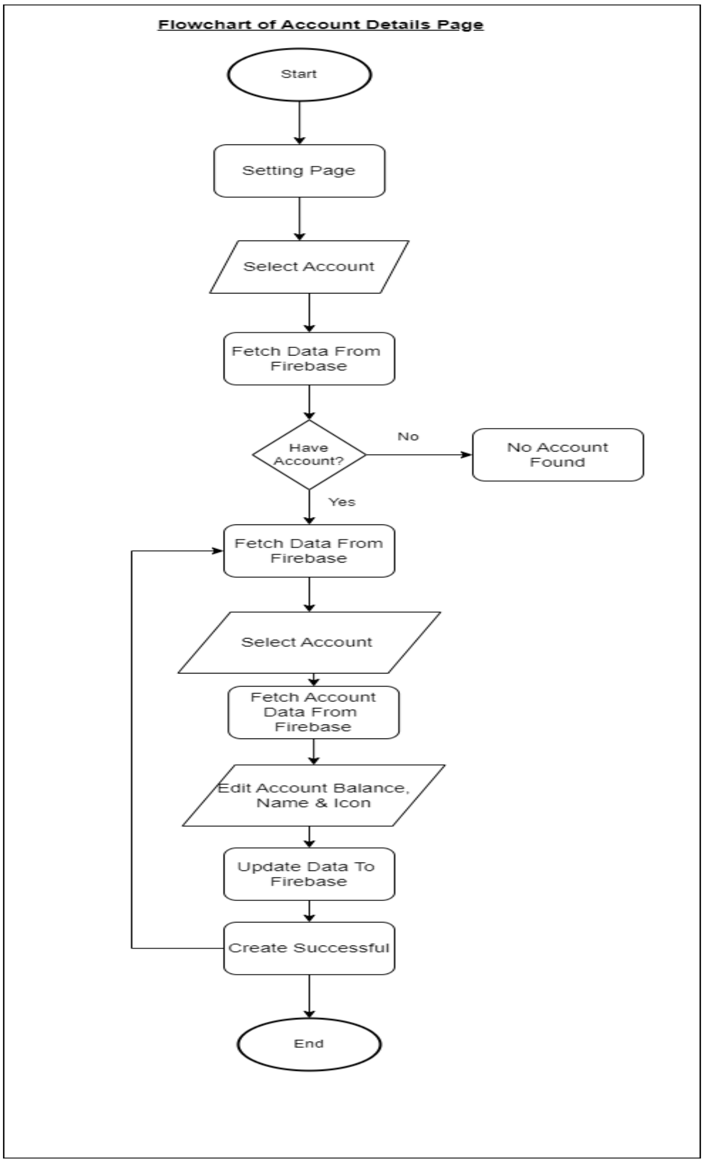


Figure 4.19 Account Details Flowchart

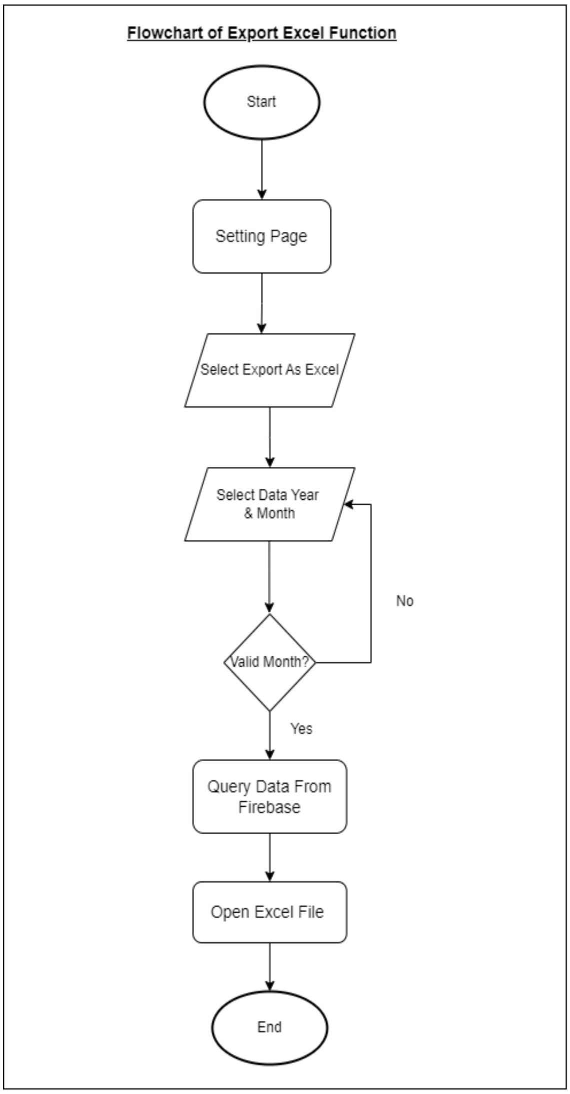


Figure 4.20 Export Excel Flowchart

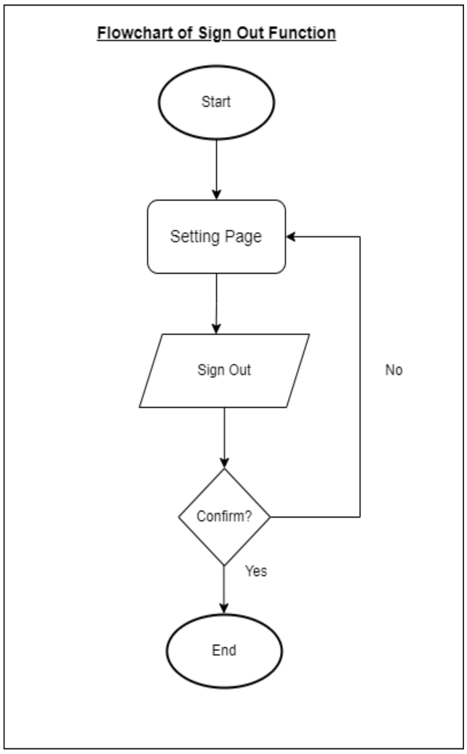


Figure 4.21 Sign Out Flowchart

4.3 Entity Relationship Diagram

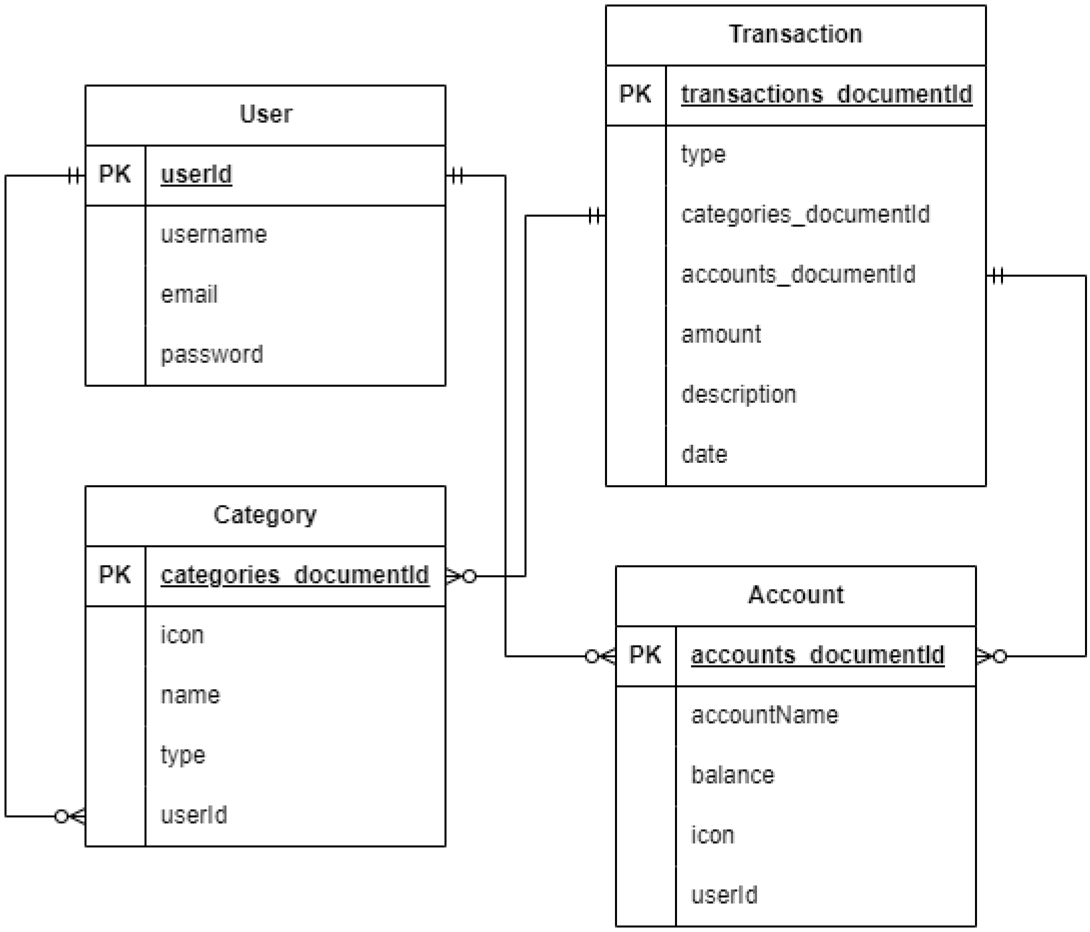


Figure 4.22 Database Design Diagram

Figure 4.22 showed database diagram as a backend of the system design. This proposed system functionality is more about the front end based which about data virtualization. Therefore, the backend of the system designed will be more simpler compared with other systems. First, when user registered an account, Firebase will record the session and fetch the username and password into user ID as a primary key. Before user start to add transactions data, there needs to create at least one transaction account and category. User can create the account and category at Setting page. Once user create an account and category, it will add into the Firebase and generate Category ID and Account ID as primary key. At the end, when user try to add transaction data into Firebase, it will retrieve the data from user ID, category ID and account ID like where account ID and category store amount and type into transaction database from user ID.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Software Setup

Before we start to do the development, some of the software should be install first. Which include Android Studio, Node.js, Visual Studio, Git, Flutter SDK and Firebase configuration.

5.1.1 Android Studio

Open browse search https://developer.android.com/studio . Download the android studio and install in the device.

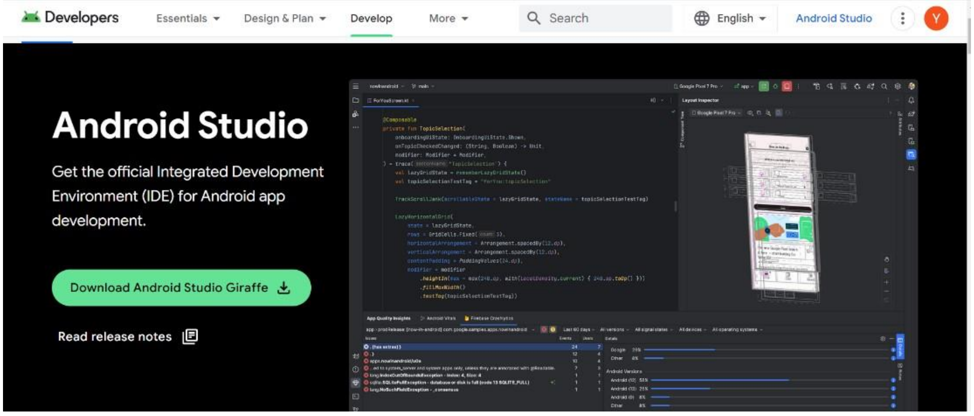


Figure 5.1 Download Android Studio

Once successful downloaded, open Android studio, select plugins search the Flutter and install it inside Android Studio.

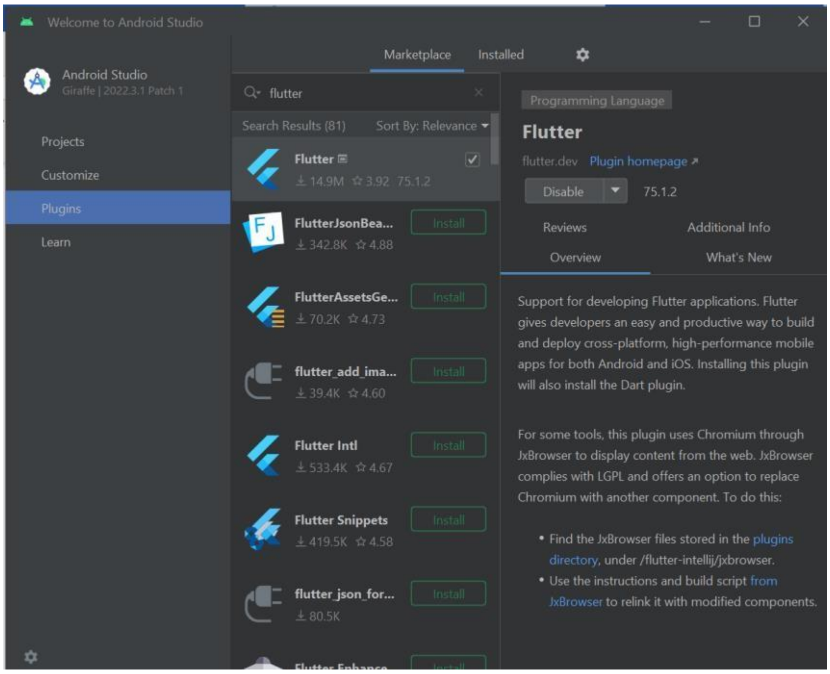


Figure 5.2 Installing Flutter framework in Android Studio

5.1.2 Flutter SDK

Open browse search https://docs.flutter.dev/get-started/install. Download the Flutter SDK and extract the folder.

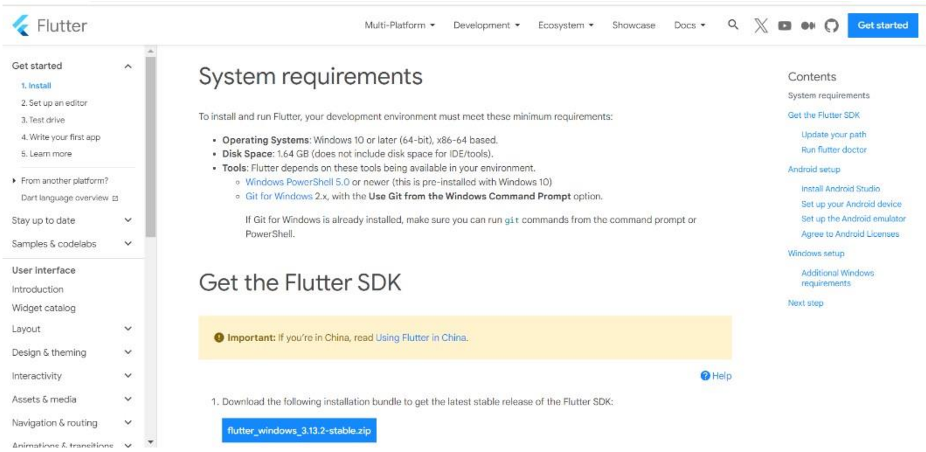


Figure 5.3 Download Flutter SDK

Open "flutter_console" in the folder and run the command "flutter doctor" to check the status. Make sure that all the summary should have a green color tick and display as "no issues found". If no, please download the following requirements.

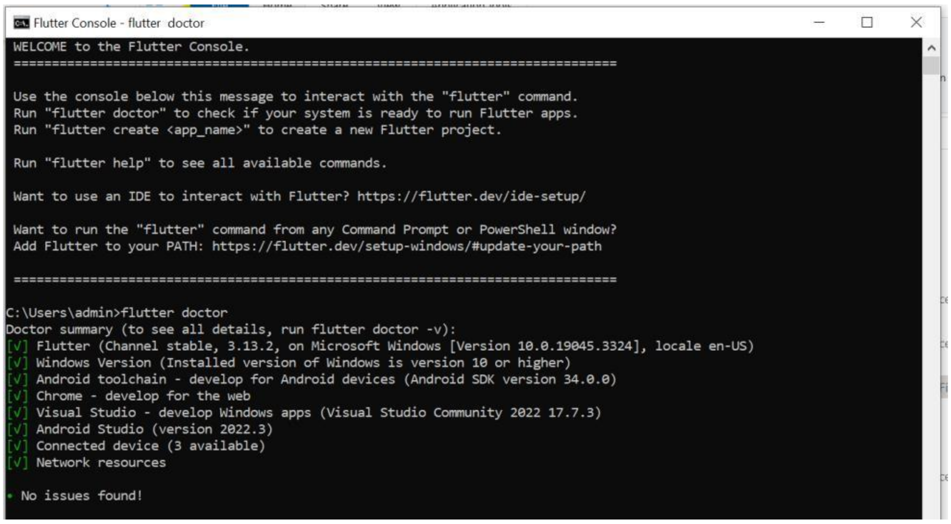


Figure 5.4 Command "flutter_console"

Requirements:

1. Chrome
2. Android Studio
3. Visual Studio with installing "Desktop development with C++"
4. Android toolchain (Open Android studio, select SDK manager > SDK tools > Select Android SDK Command-line Tools and click apply)

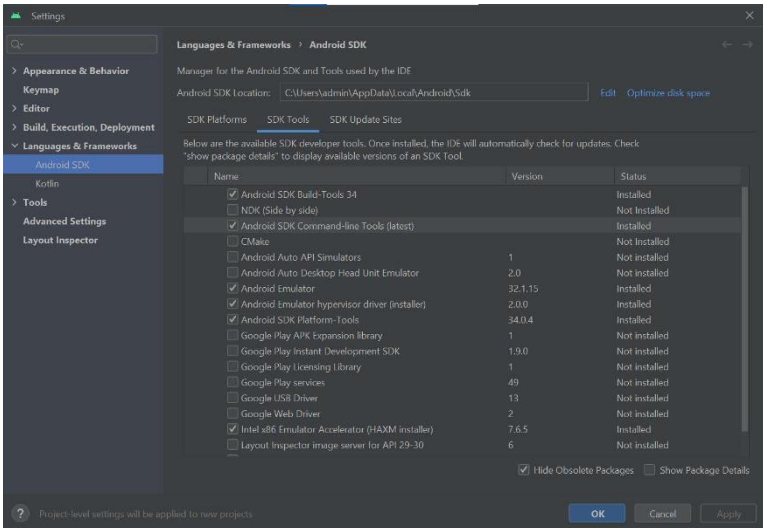


Figure 5.5 Download Android SDK command-line tools

Open edit the system environment variable, select the environment variable and double click the Path.

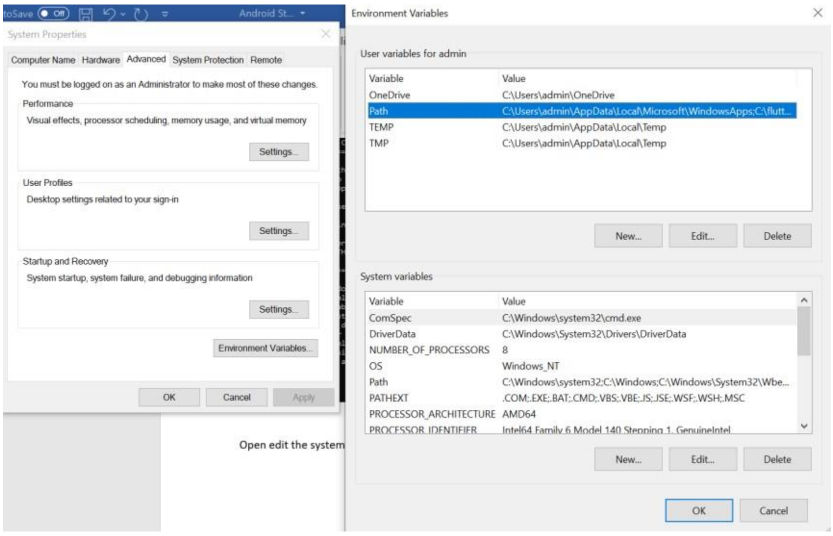


Figure 5.6 Create a shortcut path in edit environment window.

Add C:\flutter\bin in the Path and apply it. This is use for create a quick command access we can create the flutter project in command prompt.

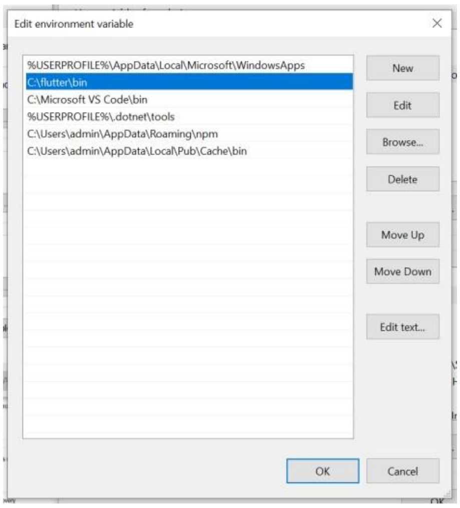


Figure 5.7 Enter the path.

Now we can create a flutter project file by command prompt.

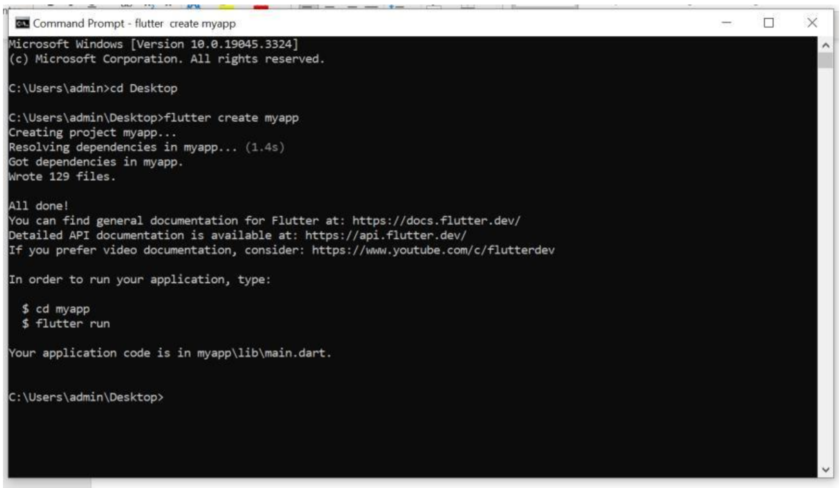


Figure 5.8 Create flutter project file.

5.2 Setting and Configuration

Firebase

Prepare workspace:

1. Open browse search https://console.firebase.google.com/ . Select add project to create a new database.
2. Once finish create open the database, open the database and select flutter inside the database dashboard.
3. Open browse and search https://firebase.google.com/. Download the firebase CLI based on device operating system. (Before download firebase CLI please make sure that Node.js has been installed)
4. Once successful downloaded, open command prompt, enter the command "*npm install -g firebase-tools*".
5. After download all the packages, enter the command "*firebase login*" and select the google account that want to use for login.

Install and run the FlutterFire CLI:

6. Open command prompt, enter command "*dart pub global activate flutterfire_cli*". Please make sure that "git" already installed in the device.
   Otherwise, the command prompt will close after enter this command line.

7. Open the flutter project inside the Android Studio, in terminal enter "*flutterfire configure --project=moneyapp-bdd8d*" and select the platforms.
8. Once done installed, open "pubspec.yaml" and enter "firebase_core: ^2.15.1" > save the file > terminal run command "*flutter pub get*".
9. Once all the steps have done, you can start to connect and use the firebase.

5.3 System Operation

In this chapter will share the capture of the development used in money manager app. It will include login activity, register activity, reset password activity, home activity, add transaction activity, modify activity and bottom navigation bar activity.

5.3.1 Login Activity

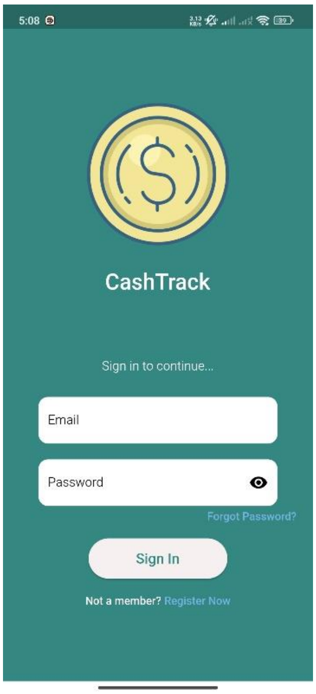


Figure 5.9 Login Page

When the application started, Login page will be the default page displayed to the new user. For the users have login into account before and haven't s sign out, it will direct to home page. For new users or logout users, there requires entering the register email and password as showed in Figure 5.3.1.1. if the email or password incorrectly, system will display the error message to the user. In case that user have not registered yet or forget the password, there can click the hypertext of "Forgot Password" and "Register Now" to proceed.

5.3.2 Sign Up Activity

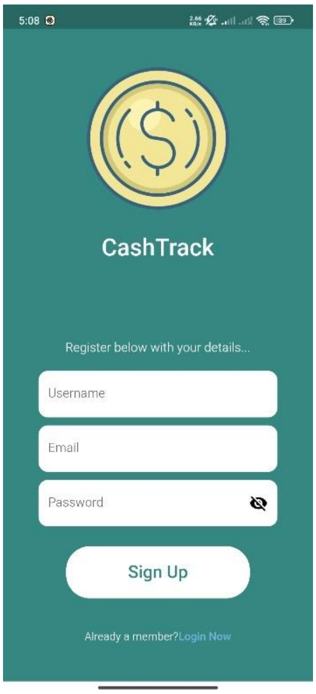


Figure 5.10 Register Page

As a new user, there need to register an account by entering username, email and password. Once done entered their account information and proceed with click Sign Up the system will check on the requirement statement. If the password lengths were not long than 6 digits and the email address has been created before, system will prompt the error message to the user and request to fill-in again. Once registered successfully, it will show a successful message and forward user back to login page.

5.3.3 Reset Password Activity

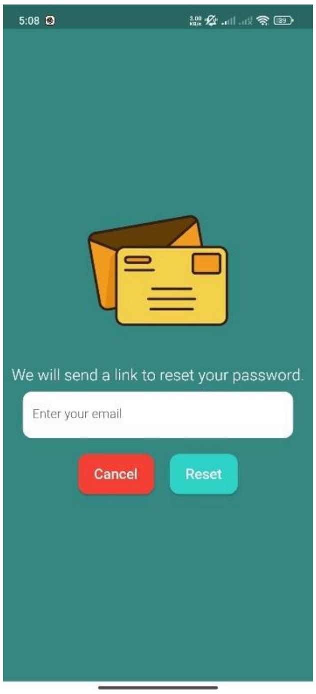


Figure 5.11 Forget Password Page

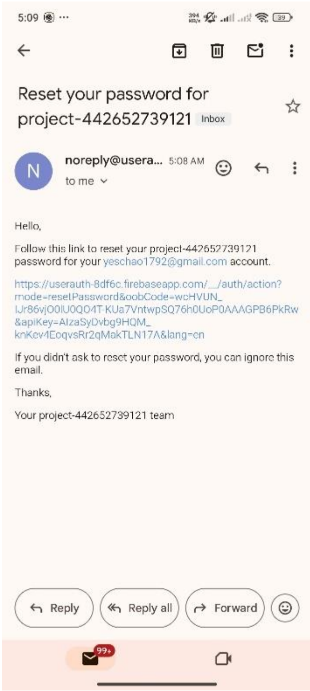


Figure 5.12 Reset Password Email

In case that user forget the password, user can click hypertext on login page to proceed with reset password page. In this page, user needs to enter the registered email address and click reset. System will fetch the user authentication data store in Firbase. Firebase will send reset password link to the registered email and user can click the link sent and reset it.

5.3.4 Home Page Activity

5.3.4.1 Add Transaction Activity

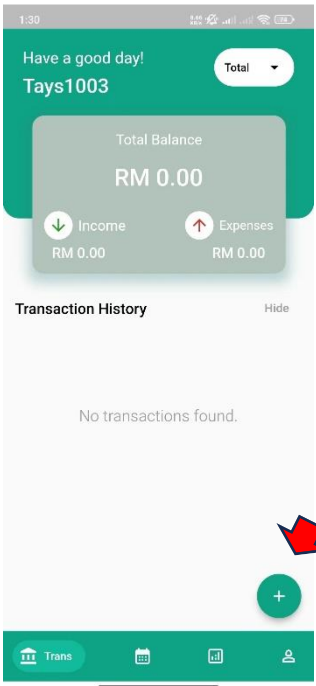


Figure 5.13 Home Page

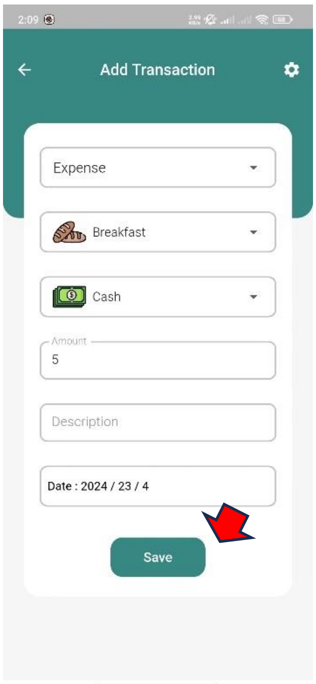


Figure 5.14 Add Page

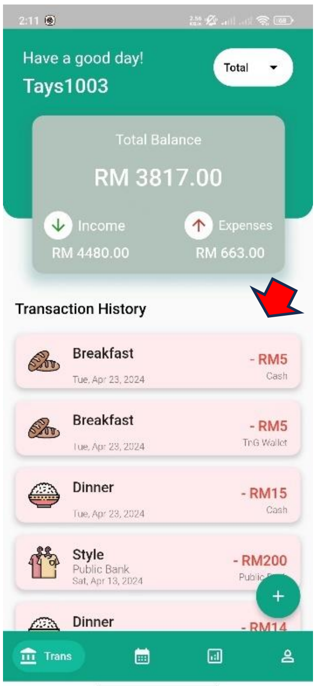


Figure 5.15 Updated Home Page

After user login to the account, the default page displays will be the home page. To start to record the transaction, click add button that provided at bottom right corn of page, it will forward user to add transaction page. Inside add transaction page, User need to enter all requirement information for transaction data included type of transaction data, category, cashflow account, amount and date. Date will direct select the current user's date and for the description is optional information. Once finished, click save system will forward user back to home page and the data will save to the Firebase. All the data added before by user will immediately display in the transaction history list on the home page.

5.3.4.2 Modify Transaction

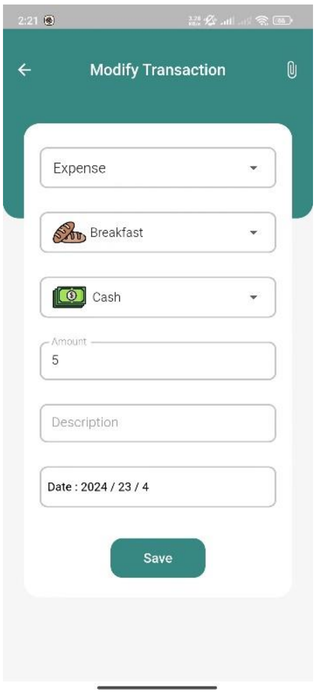


Figure 5.16 Modify Page

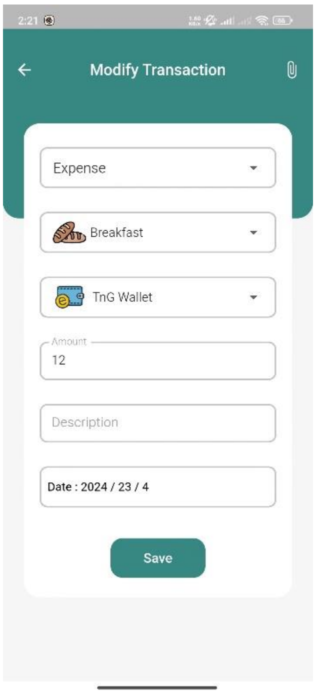


Figure 5.17 Modify Data

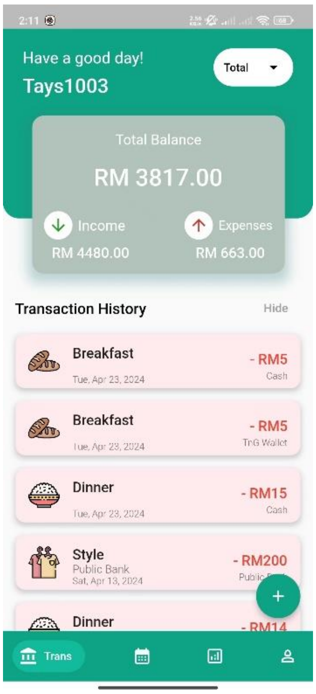


Figure 5.18 Latest Home Page

Users are allowing to modify the data created before by click the data that want to modify from the transaction history. System will browse user to the data into modify transaction page. Inside of page will be the data previously created and stored in Firebase. Once user has done to modification click save button. The system will update the data in Firebase and forward user back to home page. The data modified will immediately updated into transaction history page.

5.3.4.3 Delete Transaction

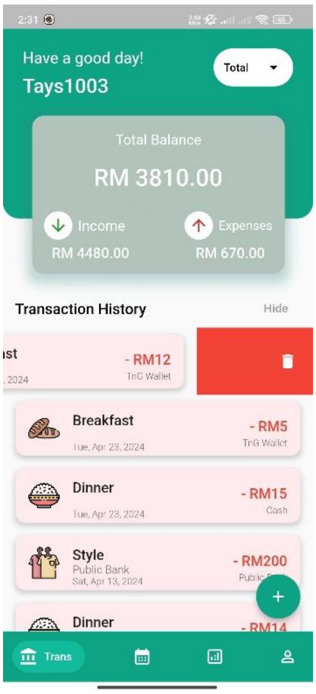


Figure 5.19 Delete Data

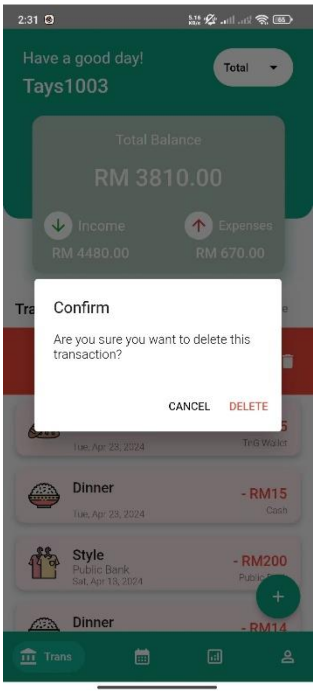


Figure 5.20 Confirmation

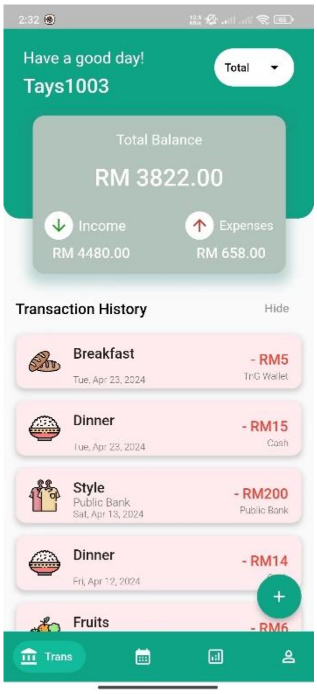


Figure 5.21 Data Deleted

To delete the transaction data, user can just swipe the data that want to delete from the transaction history list. It will display a confirmation message to let user decide to want to delete the selected transaction data. If click No, system will withdraw the delete transaction data action. Vice versa, system will delete the transaction data from firebase and the transaction history will be updated immediately.

5.3.5 Calendar View Activity

Figure 5.22 Have Data Calendar Page

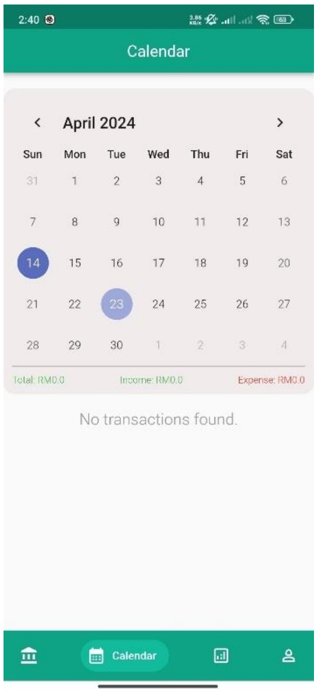


Figure 5.23 No Data Found

To improve the data virtualization of the users, user able to view the data in the calendar mode provided. The default of the selected date will be the current date user using the system. Below of the calendar will display a transaction list for selected date only. It is more convenient for users to view the data of the day according to the selected date. If the date selected has not create transaction data, the transaction list will display no transaction found.

5.3.6 Statistic Analysis Activity

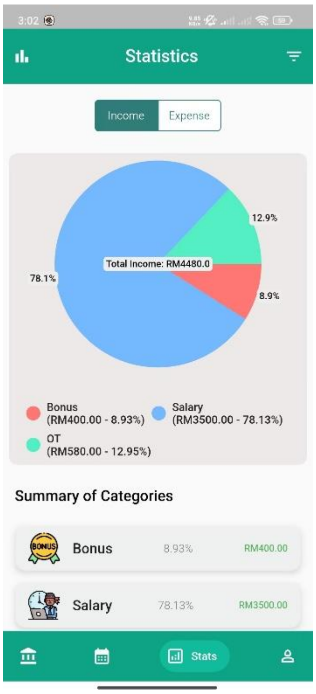


Figure 5.24 Statistic Page

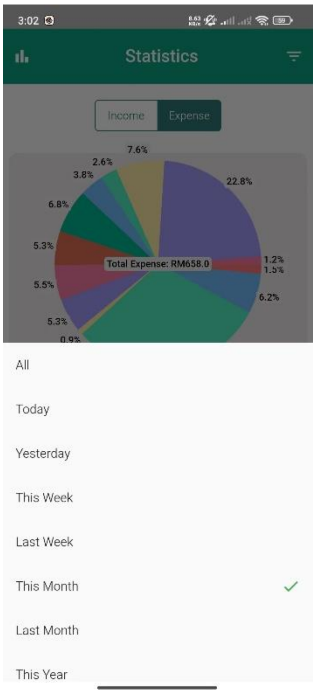


Figure 5.25 Filter Data Date

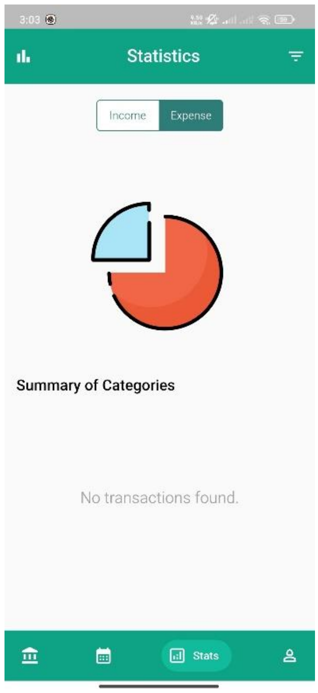


Figure 5.26 No Data Found

Statistic page is used to view transactions data in the chart form. This can let users better understand their cashflow allocation. Inside the statistic page, type of income and expense transactions data will separate displayed. User can select the selection either one to display. To improve the chart can be more diversification analysis, user can click the date filter range to check the data within the date select. In addition to provide fixed date range, user able to customize the display data date on the date filter options. If the selected date range does not have any transaction data made, the pie chart will just display an animation to represent nothing found and the summary of categories will not display anything.

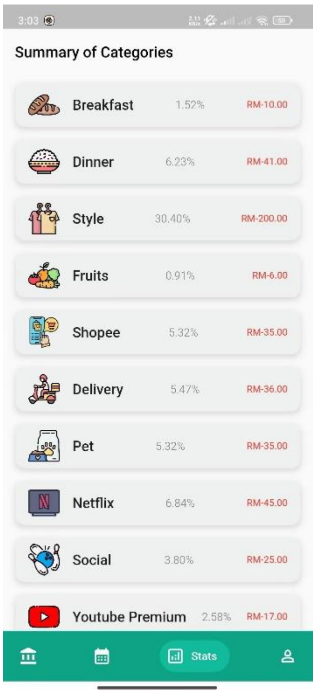


Figure 5.27 Summary of Categories

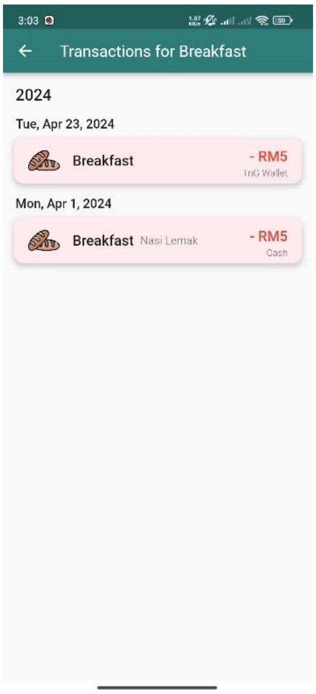


Figure 5.28 Category Details

After applying the date range filter provided, the summary of categories will also be updated as followed the date range selection. The summary of categories will display summarize of the categories that of amount. It will use for purposes let users can check on what is the higher percentage that category for income and expense. To check the details of the category spend or earn, use can click the category and the system will browse user to the category details. Inside the category details page will display every category flow amount and date. The data display will be based on the date range filter query and fetch data from Firebase.

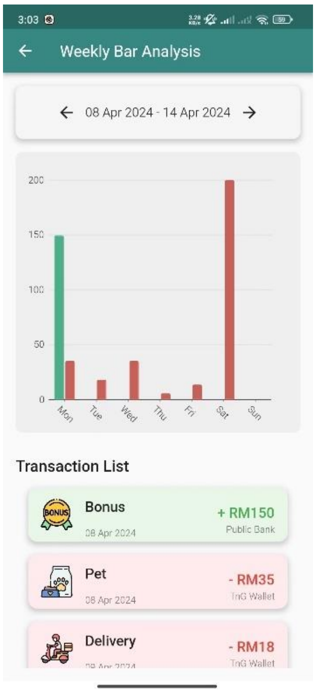


Figure 5.29 Bar Chart Page

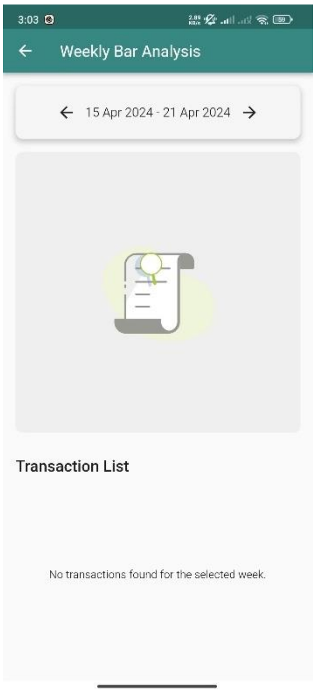


Figure 5.30 Not Found Weekly Data

User able to view weekly bar chart transaction data by click the bar icon that offer at left app bar. This bar chart will used of the purpose to let user view the weekly income and expense cashflow to make up for the shortfall from pie chart as cannot be view income and expense transaction data together. This bar chart will show data between Monday to Sunday as a period of week. User can view the previous or next week by click the controller provided. Below bar chart will display the transaction list of the data made within this week and arrange in the descending order of date. If the selected review week did not have any transaction data found, the bar chart will display an animation of no data found to inform user.

5.3.7 User Setting Activity

5.3.7.1 Setting Page

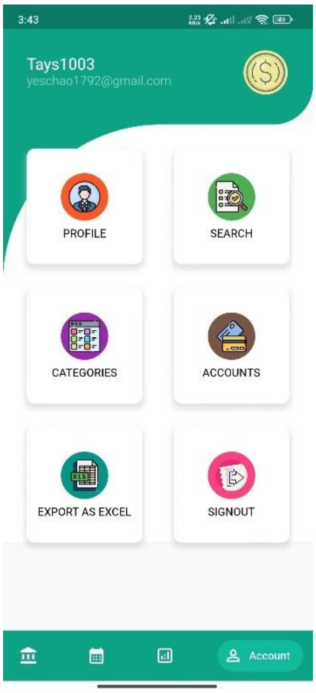


Figure 5.31 User Setting Dashboard

The account page is used to display advanced setting to the user. It includes edit profile page, search transaction data page, category management page, account management page, function to export date to Excel sheet and sign out function.

5.3.7.2 Edit Profile Page

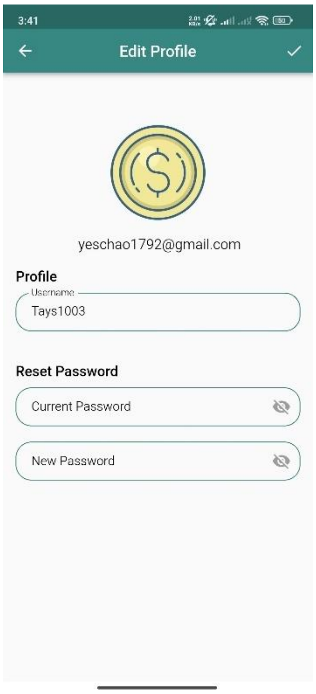


Figure 5.32 Edit Profile Page

Users proceed to this page by select the edit profile from setting page. In edit profile page user allowing to change their username and password. While the email address cannot be changed once registered in database. User can direct change the username without enter current password. But if user want to reset password, there should enter the current password to proceed. The new password needs at least 6 lengths of digits, otherwise system will reject the reset password request.

5.3.7.3 Category Page Activity

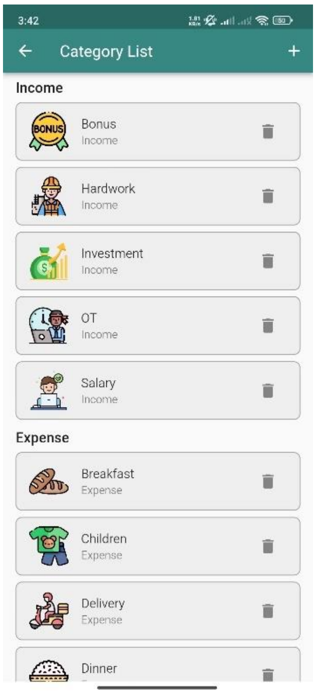


Figure 5.33 Category List Page

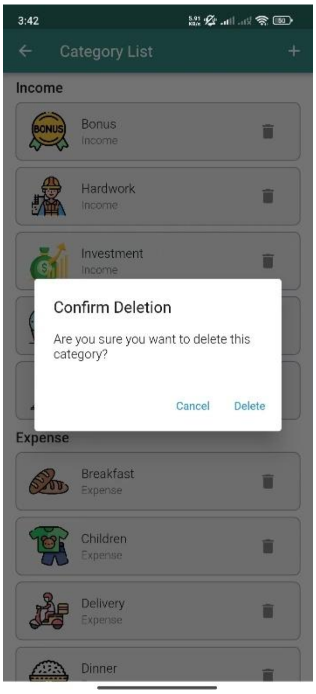


Figure 5.34 Delete Category

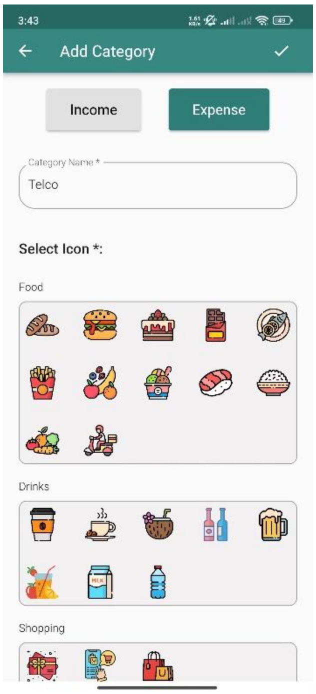


Figure 5.35 Add Category Page

Users proceed to this page by select the category from setting page. Category list page will display all the category created my user own. Category for income and expense will store separately. In this page user also can perform delete category or create a new category. To create a new category, the only thing user need to perform is select the type of category, category name and select an icon to represent of the category. System is prompt out an error message if the category name already exists in the list or missing information of during create a new category. Once finished created, system will prompt successful message to the user and forward user back to category list.

5.3.7.4 Account Page Activity

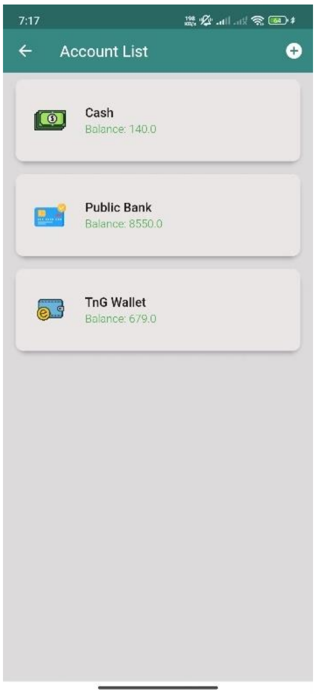


Figure 5.36 Account List Page

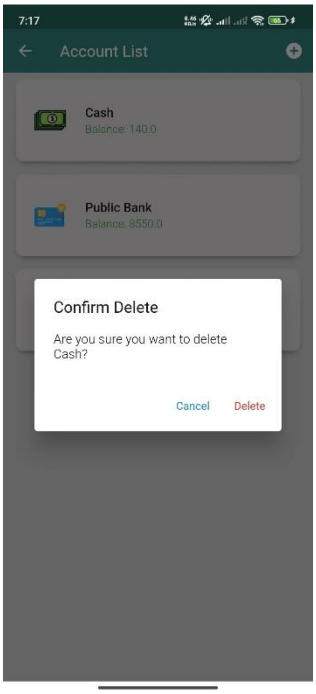


Figure 5.37 Delete Account

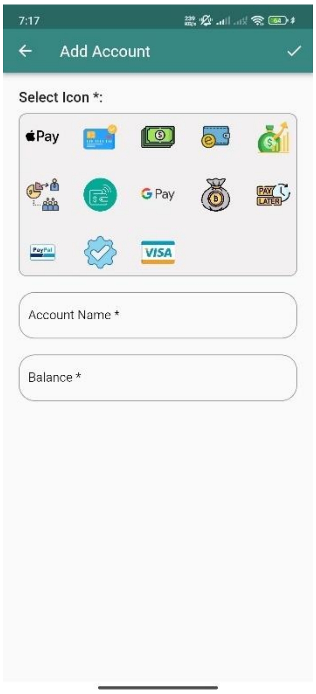


Figure 5.38 Add Account

Users proceed to this page by select the accounts from setting page. Account list page will used to display all the accounts that created by user. The activities that user can perform in this page include review the account list created, delete a selected account and add account. The list of the account will display for each account have included icon, account name and balance that after calculated with income and expense. To delete a specific account, user can just long press to the account and the confirmation of the delete account message will be display. Take note that the transaction data from account deleted will not affected to the transaction data made before. To add a new account, user can click add button to proceed. Same as the create new category, to create an account user need to select an icon for the account, name and the balance of the account. Once finished added account, system will forward user back to account list and now the new account created available on the list.

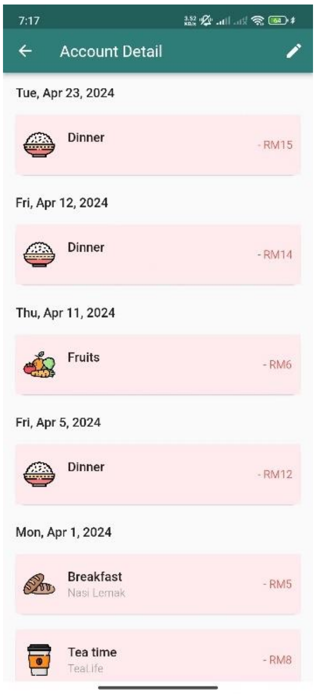


Figure 5.39 Account Details Page

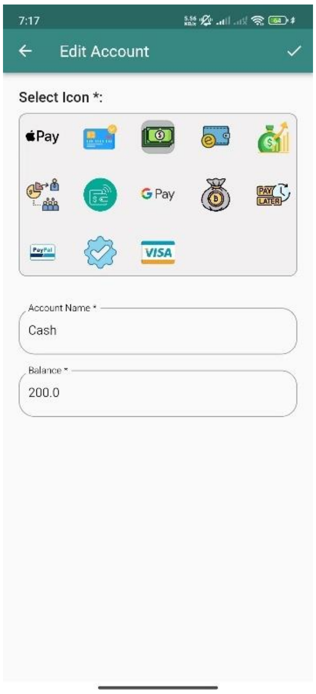


Figure 5.40 Edit Account Page

In addition, user able to view the account cashflow by click the account at account list page. This page will use for displaying all transaction data that select the account from account selected during add transaction stage. Moreover, user also allowed to edit the account includes reset the icon used, rename the account or reset the balance.

5.3.7.5 Export Excel Function

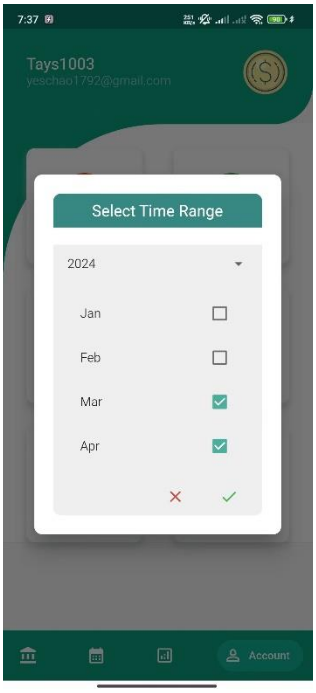


Figure 5.41 Select Export Date Dialog

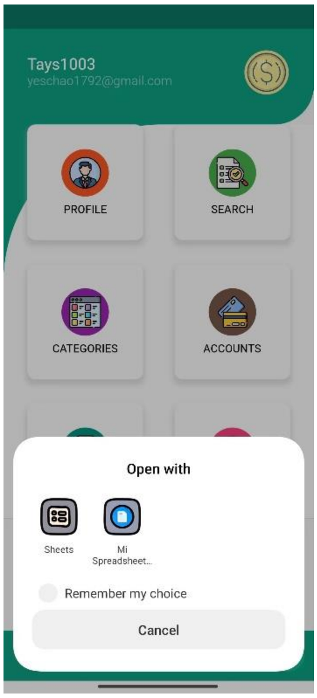


Figure 5.42 Open Excel File

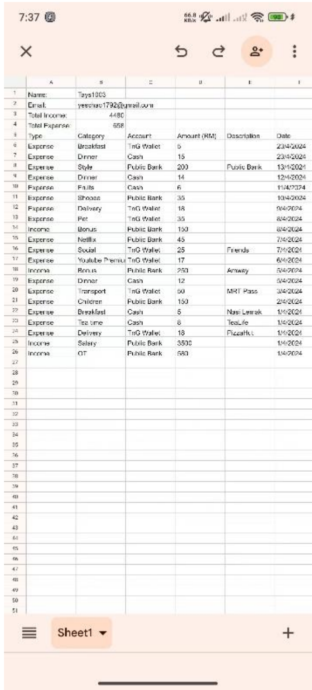


Figure 5.43 Excel File Data

User can export data to Excel file by select export as Excel from setting page. Once user click it, a dialog box will pop up to let the user select the export data for which year, and months. Users are allowing to select multiple months of the data. Once done selected click the done button, the system will pop up to select the source apps to open the file. At the end, the data will export into excel file and arranged in descending order date.

5.3.7.6 Sign Out Function

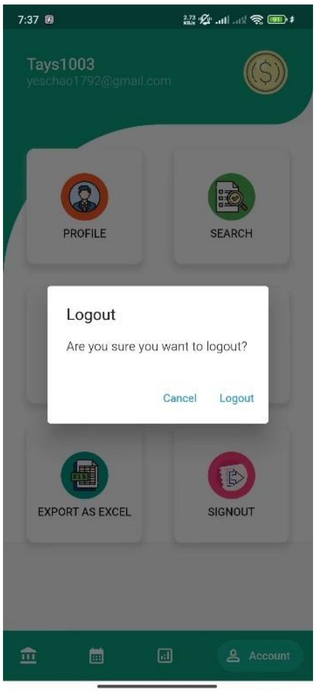


Figure 5.43 Sign Out Dialog

User able to logout the current account by select the Sign out from the setting page. Once click the dashboard, a confirmation dialog will pop up and when user click Logout, the system will exit current user account and forward user back to the login page.

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing & Result

This chapter will conduct various test cases will be executed to ensure that the application runs smoothly and as per the requirements.

Sign Up Test Case				
Role: New User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click Sign Up without fill in any input fields	Error message conduct cannot be blank.	Username cannot be empty.	Pass
2.	Click Sign Up without fill password and email	Error message conduct cannot be blank.	Invalid email	Pass
3.	Click Sign Up without fill username	Error message conduct username cannot blank	Username cannot be empty.	Pass
4.	Click Sign Up without fill password	Error message conduct password cannot be blank.	Password must be at least 6 characters	Pass
5.	Click Sign Up without fill 5 characters of password	Error message conduct password format	Password must be at least 6 characters	Pass
6.	Use email address already registered	Error message conduct email has been registered	The email address is already in use by another account	Pass
7.	Use email address in invalid format	Error message of invalid email format	Invalid email format	Pass
8.	Press “show eye” icon	Show password	Password displayed	Pass
9.	Press “hide eye” icon	Hide password	Hidden password	Pass
10.	Sign up with valid email, password and username	Show successfully register message	Registration successfully	Pass
11.	Click hypertext Login Now	Back to Login Page	Back to Sign In page	Pass

Table 6.1 Sign Up Page Test Case

Sign In Test Case				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click hypertext Register Now.	Forward to Register Page	Browse to Sign Up page	Pass
2.	Click hypertext Forgot Password	Forward to Reset Password page	Browse to Reser Password page	Pass
3.	Click Sign In without fill in any input fields	Error message conduct cannot be blank.	An error occurred. Please try again.	Pass
4.	Click Sign In without fill password	Error message conduct cannot be blank.	An error occurred. Please try again.	Pass
5.	Click Sign In without fill email	Error message conduct cannot be blank.	An error occurred. Please try again.	Pass
6.	Use Invalid email address	Error message conduct email invalid.	An error occurred. Please try again.	Pass
7.	Use wrong password to Sign In	Error message conduct invalid password.	Incorrect password	Pass
8.	Sign In using correct email and password	Successful login to system	Browse to Home page	Pass

Table 6. 2 Sign In Page Test Case

Reset Password Test Case				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click Reset using invalid email format	Show error message conduct email format	Invalid Email format	Pass
2.	Click Reset using invalid email	Show error message conduct invalid email	There is no user record corresponding	Pass

			to this identifier	
3.	Click Reset using valid email	Show successful message and receive email	Password reset link sent & received email	Pass
4.	Click Cancel button	Back to Login Page	Back to Sign In page	Pass

Table 6. 3 Reset Password Page Test Case

Home Page Test Case				
Use Case: Delete Transaction, Forward Modify Page, Forward Add Page				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click Add button	Forward to Add page	Forward to Add page	Pass
2.	Click "See all" text	Display all the transaction history	Display all the transaction history	Pass
3.	Click "Hide" text	Hide all the transaction history	Hide all the transaction history	Pass
4.	Select Daily from drop box	Display today transaction history data	Display total transaction history data	Pass
5.	Select Weekly from drop box	Display this week transaction history data	Display this week transaction history data	Pass
6.	Select Monthly from drop box	Display this month transaction history data	Display this month transaction history data	Pass
7.	Select Yearly from drop box	Display this year transaction history data	Display this year transaction history data	Pass
8.	Select Total from drop box	Display this all-transaction history data	Display this all-transaction history data	Pass
9.	Click one of transaction data	Forward to Modify page	Forward to Modify page	Pass
10.	Swipe transaction data	Confirmation messages delete transaction data	Confirmation of message to delete transaction data	Pass

11.	Click Delete from confirmation of message	Delete the data from the transaction history list	Data remove from transaction history list	Pass
12.	Click Cancel from confirmation of message	Cancel action of delete data	The data will still appeal	Pass

Table 6. 4 Home Page Test Case

Add Transaction Page Test Case				
Use Case: Add Transaction, Forward to Home Page, Forward to Category List Page				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click Save button without fill input field	Error message as missing field	Error. Please insert the required fields.	Pass
2.	Click “Type” input field	Display a drop list of income and expense	Drop list displayed with income and expense	Pass
3.	Select Income Type	Income Category list displayed	Income Category list displayed	Pass
4.	Select Expense Type	Expense Category list displayed	Expense Category list displayed	Pass
5.	Select One of the categories from list	Display the selected category in the input field	Display the selected category in the input field	Pass
6.	Select Monthly from drop box	Display this month transaction history data	Display this month transaction history data	Pass
7.	Click “Account” input field	Display a drop list of account	Display a drop list of account	Pass
8.	Select One of the accounts from list	Display the selected account in the input field	Display the selected account in the input field	Pass
9.	Insert description	Description input field display user input	Description input field display user input	Pass

10.	Click “Date” input field	Display a calendar	Display a calendar	Pass
11.	Select a date from calendar	The date input field display the selected date	The date input field display the selected date	Pass
12.	Click Save without selected date	Save data as today date	Save data as today date	Pass
13.	Click Save without selected Description	Data saved	Data saved and forward to home page	Pass
14.	Click Save by field all input field	Data saved	Data saved and forward to home page	Pass
15.	Click save without select category	Error message as missing field	Error. Please insert the required fields.	Pass
16.	Click save without select amount	Error message as missing field	Error. Please insert the required fields.	Pass
17.	Click save without select account	Error message as missing field	Error. Please insert the required fields.	Pass
18.	Fresh User yet create income category at add category page	Empty income categories drop list	Drop list will not appeal	Pass
19.	Fresh User yet create expense category at add category page	Empty expense categories drop list	Drop list will not appeal	Pass
20.	Fresh User yet create account at add account page	Empty account drop list	Drop list will not appeal	Pass
21.	Click “setting” icon	Browse to category list page	Browse to category list page	Pass
22.	Click “back” icon	Back to home page	Back to home page	Pass

Table 6.5 Add Transaction Page Test Case

Calendar Page Test Case				
Use Case: Calendar view data				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click the date that have data from calendar	Display data from selected date	Display transaction data list from selected date only	Pass
2.	Click the date that did not have data from calendar	Display a message no data found to notice user	Display No Transactions Found	Pass
3.	Swipe calendar to left	Calendar displays the next month	Calendar displays the next month	Pass
4.	Swipe calendar to right	Calendar displays the previous month	Calendar displays the previous month	Pass

Table 6.6 Calendar Page Test Case

Statistic Page Test Case				
Use Case: Pie chart view, Forward to Bar Chart page, Summary of Categories, Forward to Details List Page				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click selection “Income”	Display pie chart and summary of categories for type data is income	Display pie chart and summary of categories for type data is income	Pass
2.	Click selection “Expense”	Display pie chart and summary of categories for type data is expense	Display pie chart and summary of categories for type data is expense	Pass
3.	Click “filter” icon	Display a filter date of list	Display a filter date of list	Pass
4.	Click “bar chart” icon	Forward to Bar Chart Page	Forward to Bar Chart Page	Pass
5.	Under filter date of list select All	Display pie chart and summary of categories that include all data	Display pie chart and summary of categories that include all data of date	Pass
6.	Under filter date of list select Today	Filter data that within today to build the display pie chart and summary of categories	Filter data that within today to build the display pie chart and summary of categories	Pass
7.	Under filter date of list select Yesterday	Filter data that within yesterday to build the display pie chart and summary of categories	Filter data that within yesterday to build the display pie chart and summary of categories	Pass
8.	Under filter date of list select This Week	Filter data that within this week to build the display pie chart and summary of categories	Filter data that within this week to build the display pie chart and summary of categories	Pass
9.	Under filter date of list select Last Week	Filter data that within last week to build the display pie chart and summary of categories	Filter data that within last week to build the display pie chart and summary of categories	Pass
10.	Under filter date of list select This Month	Filter data that within this month to build the display pie	Filter data that within this month to build the display pie	Pass

		chart and summary of categories	chart and summary of categories	
11.	Under filter date of list select Last Month	Filter data that within last month to build the display pie chart and summary of categories	Filter data that within last month to build the display pie chart and summary of categories	Pass
12.	Under filter date of list select This year	Filter data that within this year to build the display pie chart and summary of categories	Filter data that within this year to build the display pie chart and summary of categories	Pass
13.	Under filter date of list select Last Year	Filter data that within last year to build the display pie chart and summary of categories	Filter data that within last year to build the display pie chart and summary of categories	Pass
14.	Under filter date of list select Last 7 days	Filter data that within last 7 days to build the display pie chart and summary of categories	Filter data that within last 7 days to build the display pie chart and summary of categories	Pass
15.	Under filter date of list select Last 30 days	Filter data that within last 30 days to build the display pie chart and summary of categories	Filter data that within last 30 days to build the display pie chart and summary of categories	Pass
16.	Under filter date of list select Last 90 days	Filter data that within last 90 days to build the display pie chart and summary of categories	Filter data that within last 90 days to build the display pie chart and summary of categories	Pass
17.	Under filter date of list select Custom Date Range	Filter data that within user customize the start and end date to build the display pie chart and summary of categories	Filter data that within user customize the start and end date to build the display pie chart and summary of categories	Pass
18.	Select One of the summary categories group	Forward to category details page	Forward to category details page	Pass
19.	Filter date that doesn’t have data made	Display no found data message	No transactions found	Pass

Table 6.7 Statistic Page Test Case

Details List Test Case				
Use Case: Display selected summary category details				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click “Back” icon	Back to Statistic page	Back to Statistic page	Pass

Table 6.8 Details List Page *Test Case*

Bar Chart Page Test Case				
Use Case: Bar chart view				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click “left arrow” icon	Display previous week bar chart data	Display previous week bar chart data and display transaction data list for the week	Pass
2.	Click “right arrow” icon	Display next week bar chart data	Display next week bar chart data and display transaction data list for the week	Pass
3.	Selected week does not have data	Display No data found message	No Transactions found for the selected week	Pass
4.	Click “Back” icon	Back to Statistic page	Back to Statistic page	Pass

Table 6.9 Bar Chart Page Test Case

User Setting Page Test Case				
Use Case: Export data to Excel, sign out, Forward to edit profile page, forward to search page, forward to category list page and forward to account list page				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click “Profile” dashboard	Forward to edit profile page	Forward to edit profile page	Pass
2.	Click “Search” dashboard	Forward to search page	Forward to search page	Pass
3.	Click “Categories” dashboard	Forward to categories list page	Forward to categories list page	Pass
4.	Click “Accounts” dashboard	Forward to account list page	Forward to account list page	Pass
5.	Click “Export as excel” dashboard	Dialog for select year and month	A dialog to select year and month pop out	Pass
6.	Select the month	The checkbox of month will tick	The checkbox of month will tick	Pass
7.	Select multiple month	The checkbox of months will tick	The checkbox of months will tick	Pass
8.	Select the year	The year will change to the year selected into input field	Input field of year change to selected year	Pass
9.	Click “Done” icon with select month	Export data of selected month to excel	Open excel file directly	Pass
10.	Click “Done” icon without selected month	Error message as invalid input of month	Please select at least one month	Pass
11.	Click “Cancel” icon	Back to User setting page	Back to User setting page	Pass
12	Click “Sign out” dashboard	Dialog of confirm sign out	Display a confirmation to sign out the account	Pass

13	Select “Logout”	System logout	Sign out from current account and back to login page	Pass
14	Select “Cancel”	Back to User setting page	Back to User setting page	Pass

Table 6.10 User Setting Page Test Case

Category List Test Case				
Use Case: Delete Category, Forward to add category page, review category list created				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	User did not have create category	Category list will display no found any category	No Category Found	Pass
2.	Users create income category only	Display Income category list and blank of expense category list	Display Income category list only	Pass
3.	Users create expense category only	Display Expense category list and blank of income category list	Display Expense category list only	Pass
4.	User clicks “trash” icon	Display confirmation of delete selected category	Display Confirmation to delete the selected category	Pass
5.	User selects “Delete”	Remove the category from the list	Remove the category from the list	Pass
6.	User selects “Cancel”	Cancel remove the category from list	Cancel remove the category from list and back to category list page	Pass
7.	User clicks “Back” Icon	Back to previous page	Back to Setting Page	Pass
8.	User clicks “Add” Icon	Forward to Add Page	Forward to Add Page	Pass

Table 6.11 Category List Page Test Case

Add Category Page Test Case				
Use Case: Create Category				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click “Done” icon without select icon and insert name	Error message of missing field	Please provide a category name and select an icon	Pass
2.	Click “Done” icon without select icon	Error message of missing field	Please provide a category name and select an icon	Pass
3.	Click “Done” icon without insert name	Error message of missing field	Please provide a category name and select an icon	Pass
4.	Click “Done” icon with select the type of income, insert name and select icon	Display successful create message	Category saved successfully and display the category into income category list	Pass
5.	Click “Done” icon with select the type of expense, insert name and select icon	Display successful create message	Category saved successfully and display the category into expense category list	Pass
6.	Click “Done” icon with insert same name and same type created before	Error message prompt out	Category with the same name already exists.	Pass
7.	Click “Done” icon with insert same name but different type created before	Error message prompt out	Category with the same name already exists.	Pass
8.	Click “Done” icon with select used icon created before	Display successful create message	Category saved successfully	Pass
9.	Click “Back” icon	Back to previous page	Back to category list page	Pass

Table 6.12 Add Category Page Test Case

Account List Page Test Case				
Use Case: Delete account, review account list, forward to Add account page and forward to account details page				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	User did not have create account before	Account list will display no found any account	No Found Any Account Created Before	Pass
2.	Have create account before	Account list will display all created account	Account list will display all created account	Pass
3.	Long press on any account	Confirmation message to delete the account prompt out	Confirm Delete the account?	Pass
4.	Select "Delete" from confirmation message	Delete account from account list	Account deleted successfully	Pass
5.	Select "Cancel" from confirmation message	Cancel action of delete account	Close the confirmation message and back to account list page	Pass
6.	Click "Add" icon	Forward to Add account page	Forward to Add account page	Pass
7.	Click "Back" icon	Back to previous page	Back to setting page	Pass
8.	Click any account from the list	Forward to display account details page	Forward to display account details page of selected account	Pass

Table 6.13 Account List Page Test Case

Add Account Page Test Case				
Use Case: Create Account				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Click “Done” icon without select icon, insert name and balance	Error message of missing field	Please provide an account name and select an icon	Pass
2.	Click “Done” icon without select icon	Error message of missing field	Please provide a category name and select an icon	Pass
3.	Click “Done” icon without insert name	Error message of missing field	Please provide a category name and select an icon	Pass
4.	Click “Done” icon without insert balance	Account saved successful	Account saved successfully and forward back to account list	
5.	Click “Done” icon with select the balance, insert name and select icon	Display successful create message	Account saved successfully and forward back to account list	Pass
6.	Click “Done” icon with insert same name created before	Error message prompt out	Account with the same name already exists.	Pass
8.	Click “Done” icon with select same icon created before	Display successful create message	Account saved successfully and forward back to account list	Pass
9.	Click “Back” icon	Back to previous page	Back to category list page	Pass

Table 6.14 Add Account Page Test Case

Account Details Page Test Case				
Use Case: Review the account cashflow and forward to edit account page				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Have transaction data from account	Display the transaction data list	Display the transaction data list from selected account	Pass
2.	Not have transaction data from account	Display no found any transactions message	No transactions found	Pass
3.	Click “Edit” icon	Forward to edit account page	Forward to edit account page	Pass
4.	Click “Back” icon	Back to previous page	Back to account list page	Pass

Table 6.15 Account Details Page Test Case

Edit Account Page Test Case				
Use Case: Edit selected account information				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Re-select icon and click “Done” icon	Display Account updated successfully	Account updated successfully and back to setting page	Pass
2.	Rename the account and click “Done” icon	Display Account updated successfully	Account updated successfully and back to setting page	Pass
3.	Reset the balance and click “Done” icon	Display Account updated successfully	Account updated successfully and back to setting page	Pass
4.	Leave the name blank and click “Done” icon	Error message prompt out	Please provide an account name and select an icon	Pass
5.	Rename the account to the same name created before	Error message prompt out	Account with the same name already exists.	Pass

6.	Click “Back” icon	Back to previous page	Back to account list page	Pass

Table 6.16 Edit Account Page Test Case

Edit Profile Page Test Case				
Use Case: Edit profile				
Role: Current User				
No.	**Test Cases**	**Expected Results**	**Actual Results**	**Pass/Fail**
1.	Rename the username	Display updated successfully	Profile updated successfully	Pass
2.	Empty the username	Error message prompt out	Invalid username	Pass
3.	Nothing changes on the data and click done icon	Display updated successfully	Profile updated successfully	Pass
4.	Enter current password but empty the new password field	Display updated successfully	Profile updated successfully	Pass
5.	Enter new password but empty the current password field	Display updated successfully	Profile updated successfully	Pass
6.	Enter new password and current password field but incorrect current password	Error message prompt out	Incorrect current password. Please try again.	Pass
7.	Enter new password with 5 characters and current password field correct current password	Error message prompt out	Password must be at least 6 characters. Please try again	Pass
9.	Enter new password with 6 characters and current password field correct current password	Display updated successfully	Profile updated successfully	Pass
10.	Click “Back” icon	Back to previous page	Back to setting page	Pass

Table 6.17 Edit Profile Page Test Case

6.2 Project Challenges

During implement this project, some of the issues as well as challenges are encountered. One of the challenges is allowing user create account and category by select an icon provided. When users use this system, the transaction data has been displayed but the category icon is delayed or cannot be displayed due to network delay problems. The reason made this happens is because all the selected icons are pre-stored in the asset images file in advance. Whenever an icon is selected as the display icon by users, the system will only save the selected icon path in Firebase. Whenever an icon needs to be displayed, the system will fetch the icon path from Firebase and put it into asset images for comparison to display the icon selected by the user. If there is a delay in the network during this process, the time required to extract the displayed icon will increase, resulting in the problem of delayed display of the icon.

Moreover, this project was corporate with several API such as Pie chart API, Excel API and Calendar API. These APIs will become one of the presents ongoing challenges for the system. This is because of these APIs are undergo periodic updates. Administrator requires manual updates to the API versions to ensure optimal system performance. Another will be risk of API downtime or discontinuation. If the API providers stop providing API access, it would render certain system functions unusable. Therefore, it is necessary to always pay attention to the trends of API providers and prepare backup APIs as a substitute to avoid these risks.

6.3 Objective Evaluation

1. To develop a cloud database-based money manager mobile app

The proposed system has achieved this objective. The proposed system will store the user transaction data, user detail, category data and account data to the Firebase. Beside that, the data that stored in Firebase will retrieving by system when user decided to view data from anywhere.

2. To develop a mobile application that can optimize transaction management, improve data visibility and data manipulation.

The objective of improve data visibility and data manipulation have also accomplished in this project. The proposed application only allowed user add transaction, delete transaction data, and modify transaction. This able to let user effortlessly access and manipulate their financial information. While on improving data visibility, proposed system will offer multiple views of transaction data that includes categorizing transactions by category, financial account, and date filters to providing users with enhanced visibility into their financial transactions activities.

3. To develop comprehensive statistical analysis of user transactions data

This proposed system also achieved this objective by providing a weekly bar chart analysis and pie chart analysis. Utilize advanced analytics tools to generate comprehensive reports, such as pie charts for income and expenses and weekly bar charts for spending patterns. This allows user to customize date filters and category summaries to gain deeper insights into their financial behavior.

CHAPTER 7 CONCLUSION AND FUTURE WORK

7.1 Conclusion

The development of the Money Manager mobile application represents a significant step forward in the domain of personal financial management. This comprehensive report has outlined the journey undertaken to address key challenges in the realm of financial management and offer a solution that empowers individuals to take control of their finances.

The objectives of this project were carefully designed to tackle these challenges head-on. We sought to develop a cloud database-based money manager app to ensure data sustainability and scalability. This objective was met with the implementation of Firebase integration, allowing users to access their data seamlessly from any device with an internet connection.

Furthermore, the Money Manager app was engineered to offer comprehensive statistical analysis functions. Users can now gain valuable insights into their financial data, customize their analyses based on preferences, and export data for reporting purposes.

Effective personal financial management skills are a cornerstone of financial independence and economic freedom. The Money Manager app aspires to contribute significantly to society by empowering users to budget effectively, reduce expenses, and set achievable savings goals. Cultivating healthy savings habits is at the core of our mission, accelerating users' journeys towards financial prosperity.

In closing, this report has documented the journey of creating a money manager app that not only addresses the challenges of today but also anticipates the financial needs of tomorrow's users. The Money Manager app is more than just a tool; it is an enabler of financial empowerment.

7.2 Recommendation

Although the current proposed system has roughly satisfied the basic functions and design of a money tracker application. But there is still have some recommendations for improvement in the future.

The first thing that must be enhanced is system security. All transaction data entered by the user will be queried into the database, which is the user's privacy detailed of financial cashflow. Therefore, the protection performance of the database must be strengthened to reduce the possibility of being cracked by hackers and causing data leakage.

The second recommendation is to add additional functions to the proposed system. Although current systems have satisfied users as they can record and analyze transaction data daily. But just offers lesser extra features. Therefore, in the future, some additional functions can be added to the original proposed system, such as budget planning, periodic bill, exchange currency or etc to increase the user activity action can perform during using proposed system.

Last, is highly recommend important selling points that can be added to the system in the future. Current proposed system is similar to the existing money manager app that available on the Play Store or App Store. To attract users to choose the proposed system, it needs a special feature that other competitors do not have. Therefore, in the future, we can enhance the key functions to the proposed system, such as read user transactional SMS messages sent by banks and will auto add into the system and show them in the transactions data list. This can be a one key function to stand out from many similar systems.

REFERENCES

[1] Google Play. (2021). Monefy: Money Manager & Expense Tracker. [Mobile application software]. Retrieved from https://play.google.com/store/apps/details?id=com.monefy.app.lite&pli=1

[2] Google Play. (Year). My Money - Expense Manager. [Mobile application software]. Retrieved from https://play.google.com/store/apps/details?id=com.raha.app.mymoney.free&hl=en_US

[3] Google Play. (Year). Budget App - Money Manager & Expense Tracker. [Mobile application software]. Retrieved from https://play.google.com/store/apps/details?id=com.hg.moneymanager.budgetapp&hl=en_US

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 2**
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

1. WORK DONE

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

I have done created functional category list page and add category page. But the UI design for those pages yet to integrate it

2. WORK TO BE DONE

Create all works to the financial account included, add account, display account, edit account, delete and account details.

3. PROBLEMS ENCOUNTERED

The images of category cannot be display as normal way.

4. SELF EVALUATION OF THE PROGRESS

Tay

Farina

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: Y3S3	**Study week no.: 3**
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

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

I have done created account list, delete account function and add account page. I also fixed the error of the cannot display the category images in the transaction list.

2. WORK TO BE DONE

Continue to do the account details page, edit account page.

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

farina

Supervisor's signature

Tay

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 4**
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

1. WORK DONE

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

I have done created account details page and edit account page. But still have some logic error need to fix.

2. WORK TO BE DONE

Plan to do the search page

3. PROBLEMS ENCOUNTERED

The account balance will not updated after have cashflow into the account.

4. SELF EVALUATION OF THE PROGRESS

farina

Supervisor's signature

Tay

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 8**
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

1. WORK DONE

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

I have done created setting page, search page and edit profile page.

2. WORK TO BE DONE

Plan to do statistic page and bar chart page.

3. PROBLEMS ENCOUNTERED

The custom date range function inside the search page cannot be capture the data within date range selected.

4. SELF EVALUATION OF THE PROGRESS

farina

Supervisor's signature

Tay

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 9**
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

1. WORK DONE [Please write the details of the work done in the last fortnight.] I have done created statistic page with summary of categories part. The date picker in side the search page has been solved.
2. WORK TO BE DONE Continue to develop the bar chart page.
3. PROBLEMS ENCOUNTERED The pie chart was failed to display the income and expense separately.
4. SELF EVALUATION OF THE PROGRESS

Farina

Supervisor's signature

Tay

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 10**
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

1. WORK DONE [Please write the details of the work done in the last fortnight.] I have done create the bar chart page and solve the problem faced before.
2. WORK TO BE DONE Plan to develop the last system page which is calendar page
3. PROBLEMS ENCOUNTERED The bar chart of X-axis cannot display from Monday to Sunday.
4. SELF EVALUATION OF THE PROGRESS

Farina

Supervisor's signature

Tay

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 11
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

1. WORK DONE [Please write the details of the work done in the last fortnight.] I have done all development phase and integrate the UI of the system
2. WORK TO BE DONE Start writing FYP 2 Report.
3. PROBLEMS ENCOUNTERED Some of the bug like same name of category display in the list yet to fix.
4. SELF EVALUATION OF THE PROGRESS

Farina

Supervisor's signature

Tay

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 12**
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

1. WORK DONE

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

Done to draw activity diagram, use case, architecture diagram, and flowchart.

2. WORK TO BE DONE

Continue to write the system operation, draw ERD diagram.

3. PROBLEMS ENCOUNTERED

None

4. SELF EVALUATION OF THE PROGRESS

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 13**
Student Name & ID: Tay Ye Schao 20ACB03821	
Supervisor: Dr Farina Saffa Binti Mohamad Samsamnun	
Project Title: Development of money management mobile application	

1. WORK DONE [Please write the details of the work done in the last fortnight.] Almost done to write the report
2. WORK TO BE DONE Redesign the table content and write, project challenges and objective evaluation
3. PROBLEMS ENCOUNTERED None
4. SELF EVALUATION OF THE PROGRESS

farina

Supervisor's signature

Tay

Student's signature

POSTER


Money Management Mobile APP
Support IOS and Android
Introduction
Keep track on your income and expenses
Statistical analysis on your financial flows
Develop a good habit of saving
Objectives
Cloud Storage supported
Schdedule own notifcation and message
Support Multiple of Statistical Analysis
CONCLUSION:
Done record within 15 sec
User Freiendly design
Access in different device
Attractive design
Intuitively view your financial flows
By: Tay Ye Schao

PLAGIARISM CHECK RESULT

20ACB03821_FYP2.pdf

ORIGINALITY REPORT

6%	4%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	georgia.gov Internet Source	1%
2	Sara, . "List Of Tables", Site Assessment and Remediation Handbook Second Edition, 2003. Publication	1%
3	mafiadoc.com Internet Source	1%
4	Navulur, Kumar. "List of Tables", Remote Sensing Applications Series, 2006. Publication	1%
5	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
6	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
7	Submitted to Universiti Sains Islam Malaysia Student Paper	<1%
8	umpir.ump.edu.my Internet Source	<1%

9	Submitted to HELP UNIVERSITY Student Paper	<1 %
10	Submitted to Middlesex University Student Paper	<1 %
11	Submitted to Royal Melbourne Institute of Technology Student Paper	<1 %
12	media.proquest.com Internet Source	<1 %
13	Submitted to Universiti Teknologi Petronas Student Paper	<1 %
14	Submitted to University of London External System Student Paper	<1 %
15	Williams, David Brian, Webster, Peter Richard. "Experiencing Music Technology", Experiencing Music Technology, 2022 Publication	<1 %
16	Submitted to City University Student Paper	<1 %
17	Submitted to University of Westminster Student Paper	<1 %
18	Submitted to Multimedia University Student Paper	<1 %
	brage.bibsys.no	

19	Internet Source	<1 %
20	Submitted to Bahrain Polytechnic Student Paper	<1 %
21	Submitted to GradeGuru Publication	<1 %
22	Submitted to National University of Ireland, Galway Student Paper	<1 %
23	Submitted to Universiti Tenaga Nasional Student Paper	<1 %
24	Submitted to Gusto International College Student Paper	<1 %
25	Submitted to University of Greenwich Student Paper	<1 %
26	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1 %
27	eprints.usm.my Internet Source	<1 %
28	ASP NET MVC 4 Recipes, 2013. Publication	<1 %
29	Grant Allen. "Android for Absolute Beginners", Springer Science and Business Media LLC, 2021	<1 %

Form Title: Supervisor's Comm ents 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)	Tay Ye Schao
ID Number(s)	20ACB03821
Programme / Course	Communication and Networking
Title of Final Year Project	Development of money management mobile application

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 6 % **Similarity by source** Internet Sources: 4 % Publications: 2 % 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.

farina
Signature of Supervisor

Signature of Co-Supervisor

Name: Dr Farina Saffa binti Mohamad Samsamnun

Date: 26/04/2024

Name: ______________

Date: ______________

UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)
CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB03821
Student Name	TAY YE SCHAO
Supervisor Name	Dr Farina Saffa Binti Mohamad Samsamnun

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.

Tay

____
(Signature of Student)
Date: 26 APRIL 2024