PERSONAL FINANCIAL PLANNING APPLICATION

BY

KUAK CHUN PIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Personal Financial Planning Application

Academic Session: JAN 2024

I KUAK CHUN PIN
(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,

ChunPin
_____
(Author's signature)

tyng
_____
(Supervisor's signature)

Address:
No 61, Lengkok Idaman 4, Taman Idaman,
14100 Simpang Ampat,
Pulau Pinang.

Chai Meei Tyng
Supervisor's name

Date: 18/4/2024

Date: 24/04/2024

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: _____18/4/2024_____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***Kuak Chun Pin*** (ID No: 20ACB0223*4*) has completed this final year project entitled "*Personal Financial Planning Application*" under the supervision of Dr Chai Meei Tyng (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology.

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

Yours truly,

ChunPin

_____

(*KUAK CHUN PIN*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**PERSONAL FINANCIAL PLANNING APPLICATION**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : KUAK CHUN PIN

Date : 18/4/2024

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr Chai Meei Tyng who has given me this bright opportunity to engage in a Personal Financial Planning Application project. A million thanks to you.

Additionally, I want to express my gratitude to my family, especially my parents, for their unwavering support. I am deeply appreciative of their sacrifices as they have invested the financially and emotionally to provide me an excellent education so I able to grow in society. Without their contributions and motivating, I would not have reached this point in my journey.

Lastly, I extend my heartfelt thanks to all my friends and those who have stood by me during my time at UTAR. The emotional support I received was incredibly meaningful, and I am thankful for all the wonderful people I have encountered and the positivity they have brought into my life.

ABSTRACT

Personal financial planning is a strategic approach to achieving financial goals and improving various aspects of individuals' lives. The project's objective is to develop a Personal Financial Planning Application using Flutter, aimed at enhancing users' financial literacy and stability through effective money management and tailored suggestions for optimizing savings. The application not only encompasses essential functionalities but also introduces innovative features like a home screen widget, targeted expense tracking for specific events, and a dedicated saving function. Furthermore, the report addresses five key problem statements associated with the application's development and conducts an in-depth review of seven analogous systems to analyse their strengths, weaknesses, opportunities, and threats (SWOT analysis). Building upon this review, the report proposes solutions to overcome the identified limitations, thereby refining the project's approach. The proposed application encompasses four core modules: the report module, tracking module, saving module, and account module. These modules collectively ensure a comprehensive and user-friendly financial management experience. To achieve efficient development, the Rapid Application Development (RAD) methodology is chosen as the project's guiding approach. The primary coding languages for development are Flutter and Dart, enabling seamless cross-platform compatibility and robust functionality.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Project Background

In this chapter, the introduction, problem statement, motivation, project scope, project objective, and contributions for the personal financial planning application are discussed.

1.1 Introduction

Personal financial planning is producing a set of strategic plans by integrating and analyzing financial data to help people achieve their financial goals. The process of personal financial planning is formulating a financial plan with 6 stages which are (1) identifying the individual's financial situation, (2) determining financial goals, (3) identifying alternatives for investment, (4) evaluating alternatives, (5) putting together a financial plan and implement and (6) reviewing, re-evaluate and monitor the plan, [1]. The plan may require a careful evaluation of strategies for specific areas of personal finance (such as budget, investment, taxes, insurance, retirement, or estate matters) for their side effects on all other areas of personal finance.

Having a financial plan can bring out countless advantages and positive impacts on every aspect of people's lives. Financial plans help the individual create and set goals to work towards. Goals are giving people direction and purpose in their lives and giving them something to concentrate on with clear goals. Financial plans are a great source of motivation and commitment. A financial plan reduces financial uncertainty by providing clear information, stating what people expect to achieve, and improving action and planning progress. [2]. In a financial plan, one of the most recommended products is the emergency fund. Accidents, business losses, or illnesses cannot be predicted. The emergency fund can be used immediately if needed. Getting early retirement is also one of the benefits of financial plans. By setting early retirement as a priority goal in their financial plan, people can save money accordingly and help people develop wealth to maintain the same standard of living after retirement. [3].

Personal Financial Planning Application is a mobile application that helps users to manage all aspects of their finances. Users can record their daily life expenses with different categories to perform their financial planning. The application can also help

users to track their spending, saving, and investing based on specific dates and times. Furthermore, users can create wallets to separate their spending based on their payment method such as credit card, e-wallet, cash, and so on. Users can also create different spending categories (e.g. food, transport, entertainment, household, and others) and set their budget for spending on the specific categories. In addition, the application will generate a chart/graph/trend for users to track total expenses based on each category set by users and to compare the data of expenses with the previous week, month, or year.

1.2 Problem Statement

Without a perfect personal financial planning mobile application, it might be difficult to attract people to manage all aspects of their finances. Many people might think that most personal financial planning mobile applications lack a lot of resources for all aspects of personal finances which brings more inconvenience to people.

Budget rule template

The budget rule template is the guideline of budget allocation for planning the spending and preparing the savings. It is very helpful for people who do not know on to allocate their budget. However, there are not any existing systems that offer this feature, which means that users are required to manually allocate a budget for various aspects of their finances and that may cause the users to face difficulty in managing their finances. For example, the user may allocate most of their budget for daily expenses, resulting in a budget shortfall for housing or other aspects.

Splitting transaction

Most personal financial applications offer only one category of transaction recording functionality at a time. For example, a user must create separate transaction records for each expense in different spending while purchasing several items from a shop. Therefore, keeping track of expenses for users is inconvenient and time-consuming and this is also one of the reasons people do not pay with their financial applications.

Account transfer

A proper personal finance application allows users to select a wallet to note down the payment method in the transaction record. However, the majority of applications do not offer a transfer feature within the wallets which means users have to manually decrease the amount of the wallet and increase the amount in another wallet. There might be inconsistent data of transaction record if there exist any typing error.

Event transaction

Event transactions mean tracking the expenses made in a specific event. In the existing system, they only offer daily expense tracking therefore if a user would like to track the expenses from a specific event, they need to mention that the transaction is for the event

in the description box of every event transaction. In the user aspect, they wish to have an application that offers help instead of making trouble for them.

Receipt Scanning

The receipt scanning feature enables users to import an image of their receipt and extract both the total amount and individual item amounts present on the receipt. This process eliminates the need for users to manually calculate and input these figures into the application, ultimately saving them valuable time. Despite its efficiency-enhancing potential, it's important to note that this functionality is not currently available in any of the existing systems.

1.3 Motivation

The aim of developing a personal financial application is to help each user become financially literate and stable by effectively managing their money and suggesting some adjustments to optimize saving money. Additionally, another motivation is delivering personal financial applications as a tool for people who have a personal financial plan to help them achieve their financial goals. Besides, the motivation for developing personal financial applications is to build up people's habit of tracking expenses to avoid overbudgeting.

1.4 Project Objectives

To provide a budget rule template for the user to set their limit spending of each category.

The proposed application seeks to offer a selection of pragmatic budgeting rule templates. This feature is particularly aimed at users with a lack of financial expertise. These templates serve as instructive guidelines for budget allocation to enable effective expenditure planning and prudent savings for unforeseen circumstances [17]. Users will have the option to seamlessly integrate these templates into their budget management strategy for various spending categories. The available templates encompass examples such as the 50/30/20 rule (advocating the allocation of 50% to necessities, 30% to wants, and 20% to savings and investments), and the 28/36 debt rule (recommending distribution with 28% toward housing and 36% for incurred debts) [18]. This feature aims to empower users with structured financial planning guidance and promote enhanced decision-making in their budget management endeavors.

To improve the efficiency of tracking expenses by using a split transaction function

Users are provided with the flexibility to opt for the split transaction feature within the application while generating a new transaction tracking entry. This capability offers a streamlined and effective approach for users to input multiple expenses across distinct categories within a single transaction and alleviates the need to generate numerous transaction records for tracking multiple items purchased in a shop.

To enhance the precision and effectiveness funds by providing account transfer function

Account transfer function allow users to transfer funds between their various wallets with ease and accuracy. Users are required to select two wallets for transferring a specified amount entered by the user from one wallet to another. The transfer transaction is automatically recorded in the transaction history for users' reference. This function helps users to manage their financial transactions with eliminating the need for manual adjustments and reducing the risk of data inconsistency.

To provide an event option for the user to track expenses about a specific event.

One of the unique features of the proposed application is the inclusion of event-specific transaction options and the ability for users to carefully monitor expenses associated with specific occasions. This requires users to create events that enable them to associate expenses with those events. Upon entering expenditure details, users can effortlessly link the transaction to the designated event they've established. To further enhance the user experience, a comprehensive expenditure report for each event would be generated in the spending report module. The report will provide users with a comprehensive overview of the accumulated costs associated with the selected event and provide insight into the total expenditure incurred.

To optimize time by developing receipt scanning functionality

Upon importing a receipt picture, the application offers a dedicated scanning function that empowers users to scan the receipt effortlessly. This scanning process intelligently extracts the total amount and each item amount presented on the receipt. Once the scanning is complete, users hold the option to select specific items, aggregating their respective amounts to calculate the total sum. This sum is seamlessly returned and displayed within the transaction page, streamlining the overall experience of recording and managing financial transactions.

1.5 Project Scope

The scope of the project is illustrating the modules of the proposed application. The modules of the proposed system include a report module, a tracking module, and an account module.

Report module

In the report module, the system would **integrate all transactions** and **generate the spending and income report** according to year, month, and week and users are allowed to view it. In addition, the system **provides graphs and charts** for users to select and **present an intuitive and understandable report** to the users. In the report, users can compare the report with the previous report to identify which category budget is getting improved or worse. Other than that, the system also **collects the transaction based on the selected event** to **generate an event report** and present it to users in this module.

Tracking module

In the tracking module, the basic functionalities and features are presented for users. For instance, users can **create expenses and income transactions** by **entering the amount, date, and time** of the transaction. Users are also required to **select the category of spending or income** and **select the wallet of the transaction** (payment method) in creating a transaction record. Apart from that, when creating transaction records they are allowed to **capture a photo or insert a photo** of the receipt as evidence of the transaction. After importing the receipt picture, there is a **scanning function** for the user to scan the receipt and capture the total amount and the amount of each item from the receipt. Users can select the items to sum up the total amount of the selected items and return the amount to the transaction page. The **split transaction option** is presented as disabled as default and the user is allowed to enable it to record multiple expenses of different categories in a transaction. There are also provides the **selection of events** if the transactions are made for the specific event and the transactions that selected an event would be forwarded to the report module for generating the event report.

Saving module

The saving module provides the same function as the money box and can also be understood as a virtual money box. In this module, there are allowed users to **create a lot of different money boxes** and **name them by the saving purpose**. In the money box, users can **decide the amount** that they planning to save and the **starting and completing dates** of the saving. In addition, the system will allow **the users to transfer the amount to the money box** accordingly.

Account module

The account module mainly focuses on setting and managing features. Users can **allocate the money to the categories** manually or **select the budgeting rule templates** provided in the module. On top of that, users are allowed to **manage categories of spending and income** such as housing, utilities, food, beverage, and so on. Furthermore, users can **manage the wallet categories** (payment method) to differentiate the payment method of transactions such as cash, e-wallet, bank, and so on. Users can also **add new events** to record the transaction of specific events.

1.6 Contributions

The proposed mobile application would **offer the basic features and functionalities** of the personal financial planning app **such as entering the expense and income record, viewing the transaction history with a graph or chart, creating different categories of expense and income, the budget limit set for every category, and so on.** On top of that, the application also **provided a lot of useful features** for users to perform their tasks efficiently. For example, a **customized calculator** function is provided where user can enter their expenses or incomes and perform some calculations if needed. Moreover, the proposed application will **provide additional features** (**tracking expenses of specific events, and a saving function**), which are not offered by the existing applications. Other than that, some of the new features that are not found in any existing system will be offered in the proposed application such as **receipt scanning** and **budget rule template**.

CHAPTER 2

Literature Reviews

In this literature review chapter, similar existing mobile application is reviewed, and the strength and limitation of each of them are also summarized for comparison. Besides that, there is the proposed solution for the limitation of reviewed applications has been discussed.

2.1 Similar Existing Mobile Application

2.1.1 MAE by Maybank2u

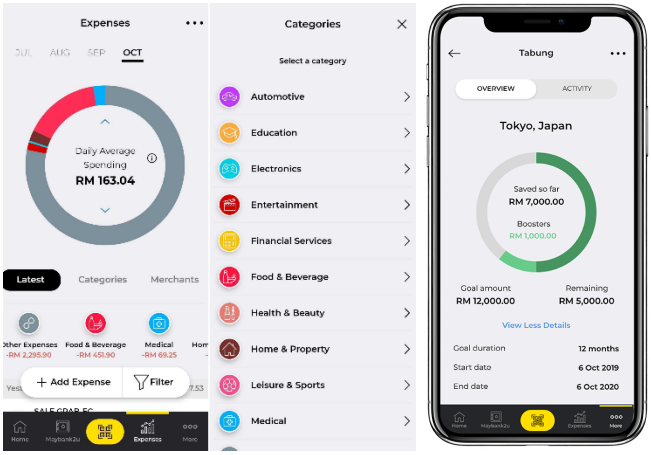


Figure 2.1.1.1

MAE by Maybank2U mobile application is an all-new app launched by Maybank. [4]. The purpose of launching this app is **to help Malaysians manage their daily personal finances instead of regular banking needs** and to provide a fresh, next-generation experience. [5]. Other than that, MAE by Maybank2U mobile application provides a 0% commission food delivery platform called Sama-Sama Lokal, to help local merchants establish their online presence and attract more customers. [6]. It also provides many digital banking functions offered by Maybank such as mobile banking, QR Payments, and the MAE e-wallet, and develops a lot of new features for personal financial management. [5].

In MAE by Maybank2U mobile application, there are two strengths were identified in the personal financial management of the application. **One of the strengths that were found is the Tabung feature which is the most attractive feature**. This feature assists the users to perform more consistently and frequently saving to archive their

financial goals. Users are allowed to set up a Tabung and select a category which is the purpose of the users' saving. Next, users are required to insert the amounts that they are planning to save and the starting date and ending date for saving Tabung. Users can also invite other MAE users into their Tabung and perform the saving together. Furthermore, **another strength of MAE by Maybank2U mobile application is the spending tracker which is a tool to help users track their spending automatically** and offer users a single and holistic view of every spent across Maybank accounts, cards, and QR transactions.

2.1.2 Expensify

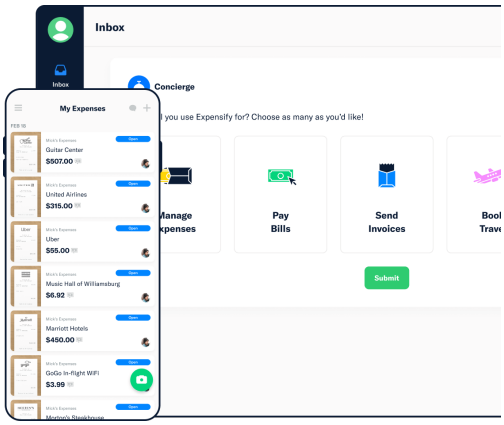


Figure 2.1.2.1

Expensify application is an expense tracker app and also a business travel mobile application as it includes scan receipts, tracking business, and personal expenses, and booking travel in one app. One of the aims of this application is to **help people to manage their spend with stress-free**. It allows user to track their expenses by scanning receipts or manually creating new expense records. Besides, it supports iOS, Android, and Website systems and obtained high ratings in the Apple App Store and Google Play Store which are 4.7/5.0 and 4.5/5.0 respectively. Currently, Expensify is offering RM19.90 / month only for the upgrade to receive unlimited SmartScans but they are free to use for up to 25 SmartScans of receipts per month. In Addition, the Expensify app has been awarded a lot of awards such as Top 50 Finance Products, Editors' Choice, Top-Rated Expense Management Software, and so on. [8], [9].

Although the Expensify application provides a lot of features such as expense management, multiple currency conversion, travel booking, custom report exports, and so on, **the unique feature of this expense tracker application is the SmartScan** which the system will read the receipts that users insert and capture all the details of the expenses quickly and accurately. This feature helps users to reduce the lot of time consumed on manually tracking transactions. On the other hand, the Expensify application also known as a business travel application provides a user-mile tracking feature for calculating the total amount of traveling at a specified distance rate.

2.1.3 Monefy

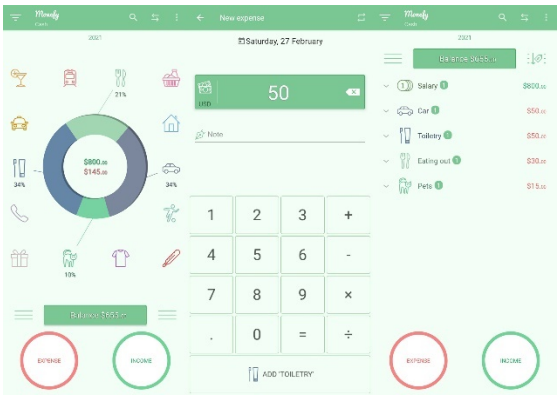


Figure 2.1.3.1

Monefy application is a personal finance application that makes money management easier by p**roviding a simple, intuitive, and understandable way of the user interface**. Users of Monefy can rapidly start to recognize the key spending patterns and effectively begin saving money. For instance, Monefy simplifies the process of inserting new transactions and makes it faster, easier, and simpler for users. Other than that, this application also provides most of the basic functionalities such as the overview of incomes and expenses on a pie chart, a detailed view of all records, multi-currency support, safe data synchronization for the backup, a built-in calculator, and so on. Besides, the objective of the Monefy application is **designed to streamline expense tracking and help people to save money**. This application can be seen as a helpful personal financial management application tool as it has obtained a 4.7/5.0 rating in the Apple App Store and a 4.3/5.0 rating in the Google Play Store. [10], [11].

2.1.4 Money Manager Expense & Budget

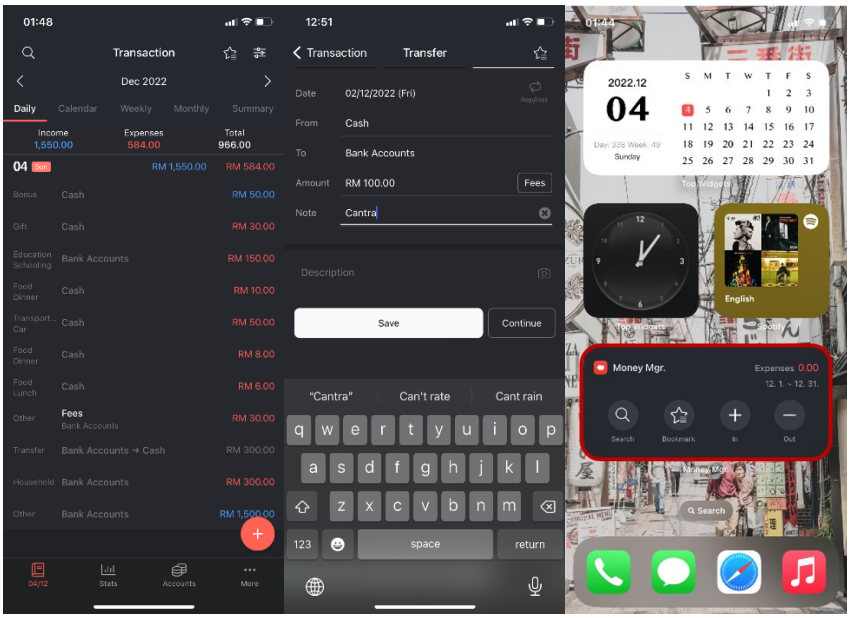


Figure 2.1.4.1

Money Manager Expense & Budget is an application that provides a lot of easy and various features for people to implement financial planning, review, expense tracking, and personal asset management. It is available for iOS, and Android systems, and people can install it with the App Store or Google Play Store. [12], [13]. In this application, there is a feature for users to preset the budget amount of each category. Users can use the feature to allocate their money to different spending categories and control spending according to their allocated amount. The application also provides two different styles which are light mode and dark mode for users to select based on their preferences.

There are a lot of strong features that are identified in the Money Manager application. **One of the strength features is the PC manager function** which data the bookkeeping can synchronize with the PC and users can modify and sort the data by date, category, or wallet group on the screen of their PC. Users are also allowed to view fluctuations of their accounts indicated on graphs on their PC. Other than that, **another strength feature is the home screen widget feature** to allows users to insert a small app instance on the home screen. This feature supports the quickest way and reduces a lot of time consumption by eliminating a few steps to perform the basic function and viewing the amount that had been spent.

2.1.5 Money Lover

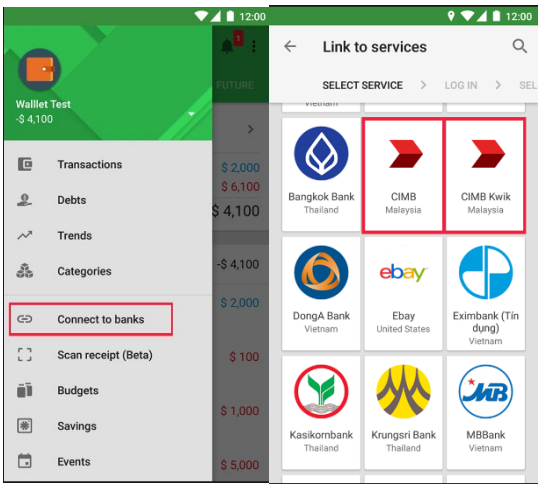


Figure 2.1.5.1

Money Lover application is also an expense tracking application developed by a company from Vietnam called Finsify. [14]. Money Lover contains most of the basic financial management tools for users to set up several "Wallets in the application and add transactions (as an expense or income) to each of the wallets at any time. The application also allows users to categorize and keep a tab on their expenses across weeks, months, and years in a simple way. In addition, many other built-in features are handy such as a calculator, foreign currency exchange rates, the ability to scan receipts, and so on.

Besides, Money Lover is also containing a strength feature that is **connected to banks' features**. This feature is included in the subscription service which means that users are required to pay for the premium version if they request this feature. This feature not only supports the connection with foreign banks but also certain banks and services in Malaysia such as CIMB, Maybank, Public Bank, RHB, AmBank, Bank Islam, and Hong Leong Bank, as well as PayPal. Recently, they also added cryptocurrency to the list of linked services. The purpose of the feature is to track the transactions of the users more accurately and ensure that no transactions are forgotten to record.

2.1.6 Monny

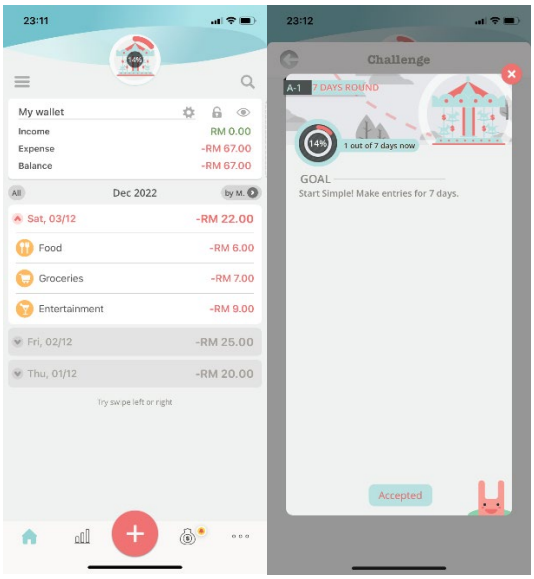


Figure 2.1.6.1

Monny is an expense-tracking application that records all daily or monthly transactions and creates reports to show users where their money is being spent. [15]. The logo icon and the user interface both are designed in a cute style, and it may attract more female users. Monny is free to use and provides users with enjoy free trial on featured analytical reports and charts, but users can always upgrade to Monny Premium to obtain unlimited access such as the Top 10 expenses ranking chart, monthly category expenses ranking chart, annual trend charts, and passcode lock feature.

Moreover, the strength of the Monny application is that the application **provides a drop-down list on every date** to display all of the expenses and income on that date. This allows users effortlessly to view all the amounts and categories of the transaction on a specific date. Another strength of the application is the **challenge feature** which allows users to challenge for keys in their transactions every day. That is a helpful feature for users to cultivate bookkeeping habits if they users never used expense-tracking applications before.

2.1.7 Goodbudget

Figure 2.1.7.1

Goodbudget is a personal finance application that is a perfect option for couples who want to share their budgeting process. [16]. The reason why it is a perfect option for a couple of users is that it has having **synchronization feature across multiple devices and different operating systems**. Other than that, Goodbudget is using the familiar **envelop budgeting philosophy** to power the users' proactive budget for their all transactions. Users are allowed to make envelopes for all their budgeting categories such as household, entertainment, utilities, food, beverages, and so on. After that, user can divide their money into separate envelopes for different expense categories. In addition, Goodbudget provides the **split expense transactions feature** which means it allows users to split a transaction into multiple expense categories from the same transaction.

2.1.8 Functionality Comparison

Table 2.1.8.1 Functionality Comparison

	MAE by Maybank2U	Expensify	Monefy	Money Manager Expense & Budget	Money Lover	Monny	Goodbudget	Proposed Application
Provide widget feature				√	√			
Provide calculator feature			√	√	√	√		√
Provide travel model feature to record the transaction of the specific event					√			√
Support multi-currency		√	√	√	√	√	√	
Provide a graph/chart to view the history of the transaction	√		√	√	√	√	√	√
Allow setting spending limit for each transaction category	√			√	√		√	√
Provide alarm feature (remind users record daily transaction)		√		√	√	√		√
Provide backup feature including export data to cloud storage		√	√	√	√	√	√	
Provide wallet category which to record the payment method of the transaction	√		√	√	√	√	√	√
Provide passcode feature	√	√	√	√		√		

Allow insert picture of receipt for expenses		√		√	√	√		√
Provide money box for saving purpose	√						√	√
Provide recurring transactions feature			√	√	√	√		

2.2 Limitations of Previous Studies

The previous studies did not cover the weakness of similar existing systems therefore it will be discussed in this section. In the **MAE by Maybank2U** mobile application, there is a weakness that exists in the system which is **not accurate to track the spending categories in the spending tracker feature**. For instance, ordering food delivery using the GrabFood application is categorized under Transportation instead of Food & Beverage, and the expenses that were spend in the Shopee application were included under the Finances category instead of the Shopping category. [7]. These may be causing users to be misled while tracking their expenses.

Apart from that, one of the limitations of the **Expensify application** is there is **no user-friendly interface**. The graphic user interface design of the application is complex and unintuitive for users. Although the application can be working well it is time-consuming for the user to be familiar with it. **Another limitation of the application is the lack of some basic functionality** such as the ability to filter receipts based on date, merchant, or category and the ability to search for a receipt based on different parameters.

Another existing system in previous studies is the **Monefy** application and there are some limitations in the application. **One of the limitations is the lack of budget features for each category.** For example, setting 500 dollars for food and beverages, 300 dollars for household, 100 dollars for utilities, and so on. Without setting budget features, users will have more difficulty in controlling their spending and achieving their financial goals. **Another limitation of the application is the lack of multiple-device support within the different operating systems**. For example, they are not allowed to connect iPhone and Android devices with a single user account.

In the **Money Manager Expense & Budget** application, there is a limitation that does not exist in the app which is the application's **lack of a split transaction feature**. This limitation was also mentioned by a lot of users in their comments on the Apple App Store and Google Play Store. An example provided by a user, the system requires users to create separate transaction records for each expense in different spending categories while purchasing several items from a shop. [12], [13].

Besides, the **Money Lover** application **lacks a saving feature** that can guide the user to save up their money. This is one of the helpful personal financial tools to save and achieve their financial goals. For example, a user who is planning to buy

something or planning a trip, the user allowed to create a fund and enter the name of the fund's purpose and the amount that planning to save. After that, the user is required to select the date to start saving the fund and the date to complete saving the fund. The system will calculate the amount that the user should save up every day, week, month, or year.

The limitation that exists in the Monny application is its **lack of a transfer feature** that allows users to transfer money from one wallet to another. Without this feature, users are required to manually decrease the amount of the wallet and increase the amount of another wallet. Expect this, the application **does not allow users to record the transaction of the specific event**, it only allows users to record their daily expenses. If the users would like to record the expenses of a trip the users are required to manually specify it in the description box of each transaction.

Last but not least, there is a lot of limitation of the **Goodbudget** application. One of the limitations is there is **no user-friendly application** for first-time users. It's confusing at first, with all the random buttons and different features that could make it seem confusing. The font size and style look unprofessional and difficult to read. Another limitation of the application does not provide a **home screen widget feature**. Personal finance application is used to track more than one expense per day which means that users without widget features need to open the application with many frequencies every day.

2.3 Proposed Solutions

This project is going to propose a solution for improving some of the limitations of the reviewed system discussed in the previous section. All the solutions that had been proposed might be included in the proposed application.

First and foremost, the solution that would like to propose for the limitation of the transfer function is to **create a new transaction category called transfer**. When the system provides a wallet category to differentiate the payment method, there should be a transfer function that allows users to transfer the amount from one wallet to another wallet. In this proposal report, the transfer means moving the money record between wallets in the application instead of transferring money from one bank account to another. For example, a user withdraws money from the bank and the user can use the transfer feature to manage the remaining amount of cash wallet and bank wallet. In this category, it should include an amount column for the user to fill in the amount that wants to transfer, a wallet selection for the amount transferred from and to, as well as a date and description.

On top of that. **developing an event feature** is the best solution for users who would like to track the transaction of a specified event easily. Instead of manually specified in the description box, this feature will create a manage event page for users to create events and insert an event selection column in the tracking page for users to select the event of the transaction. The event selection can be nullable to record the daily transaction.

To solve the limitation of creating multiple transactions to separately track a transaction, the application can be improved by **adding the split transaction option**. The default of the split transaction option will be disabled therefore when users enable it, they can enter multiple expenses of different categories in a transaction. In the split transaction feature, there would be a lot of amount columns with the corresponding category selection column for fill-in purposes.

Lastly, the solution for the lack of saving feature limitation is to **develop a saving module**. In this module, users can create a money box and rename it for saving purposes. In the money box, users are allowed to set the dates of start and complete saving as well as the amount that planning to save. After that, the system will help the users to schedule weekly, monthly, or annual recurring deduct the amount from the

wallet, and move them into the money box. In this approach, users can easily record their savings amount and avoid overspending to achieve the savings plan.

CHAPTER 3

3.1 System Methodology

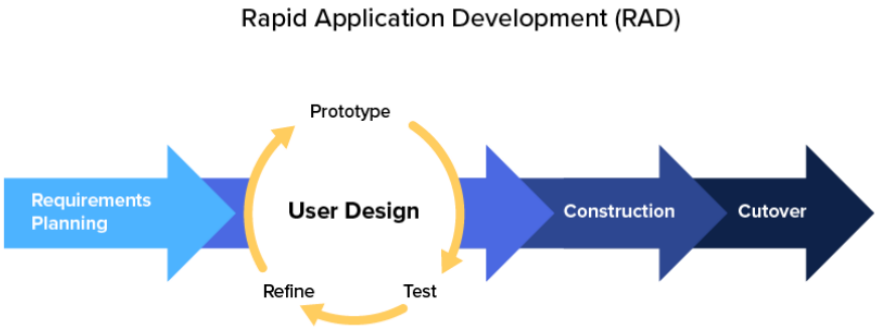


Figure 3.1.1

The methodology proposed for the project is the Rapid Application Develop (RAD) methodology. The RAD model is a system development method based on prototyping without any specific planning. In the RAD model, there is more focus on development tasks instead of paying attention to planning. It targets developing a system in a short period therefore it is the most suitable methodology for the proposed application.

The RAD model contains four process phases which are requirements planning, prototyping, rapid construction, and implementation.

Phase1 – Requirement Planning

Rapid application development distinguishes itself from typical software development techniques from the very beginning. They are only asked for a general requirement instead of being required to sit together with end users and obtain a comprehensive list of requirements. Due to the wide requirements, the developer can spend their time segmenting certain criteria at various stages of the development cycle.

Phase2 – Prototyping

The prototyping phase is where the actual development happens. Developers produce prototypes efficiently with a variety of features and functionality rather than adhering to a particular set of requirements. After that, developers would present the prototype to the client to recommend the improvement of the features and functionalities. These prototypes would be developed in a short time to highlight the most important characteristics and the final prototypes are only developed when the client and the developer are on the same page regarding the result, or during the finalization stage.

Phase3 – Rapid Construction

The vital phase of development is the construction phase which the coding, testing, and integration will be involved to convert the prototypes into a working phase. Feedback and evaluations are vital at this stage and most bugs, issues, and adjustments are resolved during this stage. This phase may take a long time especially when clients change their minds if the feedback is detailed.

Phase4 – Implementation

The final stage of the RAD model involves deploying the constructed system into a real-time production environment. In-depth scale testing, technical documentation, issue tracking, final modifications, and system simulation are all part of the deployment phase. Before the application goes live, developers also spend time troubleshooting it and performing final upgrades and maintenance.

3.2 System Design Diagram

3.2.1 Use Case

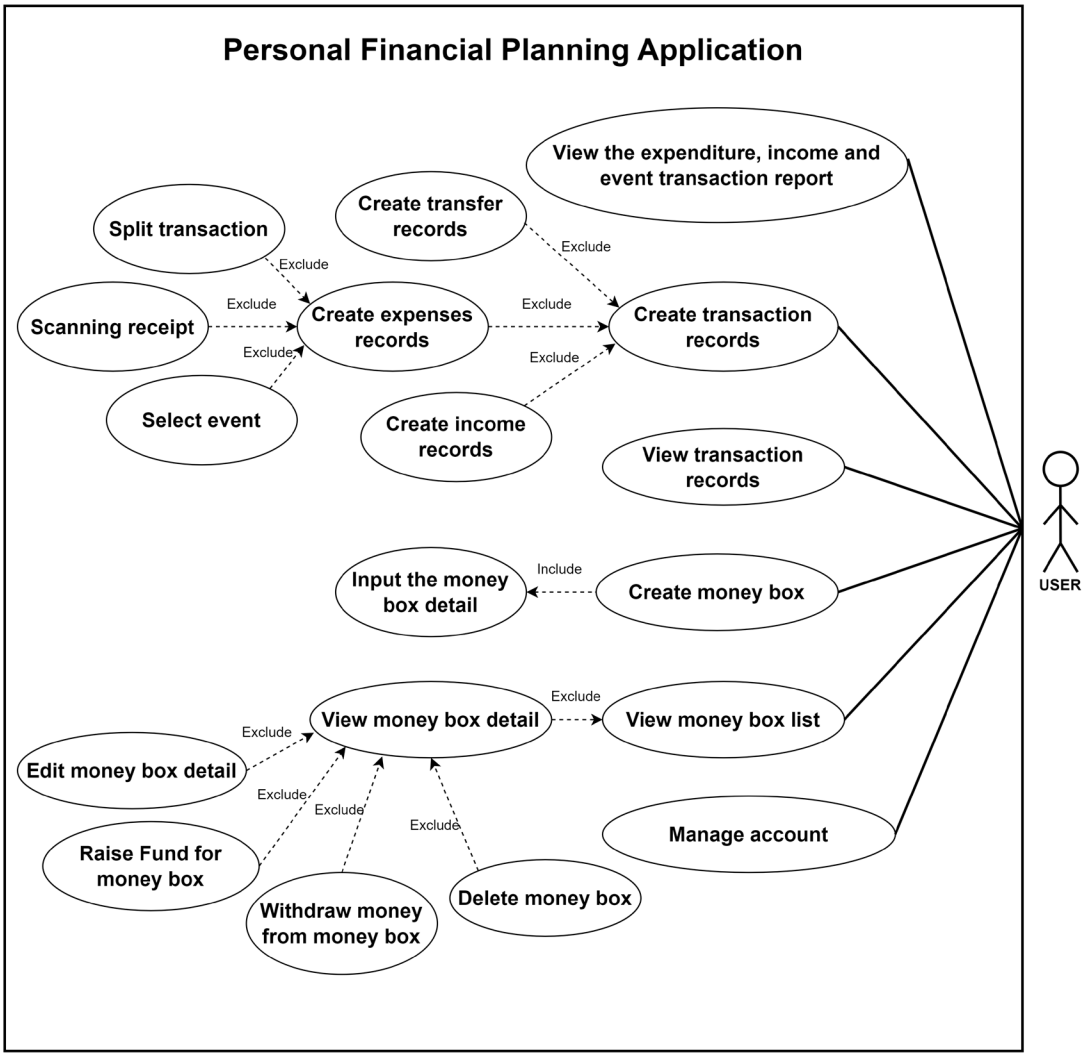


Figure 3.2.1.1

Description

The figure depicted above illustrates the use case diagram for the proposed Personal Financial Planning Application. This use case diagram underscores the core functionality that will be accessible to the users by offering them a comprehensive set of tools meticulously crafted to manage diverse facets of personal finance. Among the standout features is the ability to access detailed reports encompassing expenditures, income, and transactions linked to specific events. Users are provided with the flexibility to opt for either a list view or a pie chart representation when perusing these reports. Besides, the reporting function allows users to arrange the transaction list based on varying timeframes such as year, month, and week.

Another critical functionality of the proposed application is the creation of transaction records that encompass three distinct types which are transfer, expense, and income transactions. Transfer transaction records enable the transfer of funds between different wallets such as withdrawing RM200 in cash from a bank, which would lead to a deduction of RM200 from the bank wallet and an increase of RM200 in the cash wallet. Furthermore, the income transaction records permit users to augment wallet balances based on the method through which they've received money. Conversely, the expense transaction records encompass three additional functionalities beyond those of transfer and income records which are split transaction (dividing a single transaction across multiple categories), event-specific transaction (monitoring expenses linked to specific events), and receipt scanning. Irrespective of type, all transaction records include essential details like date, amount, notes, and imported images, providing a comprehensive record-keeping application.

Apart from that, a distinctive offering within the application is the "money box" concept which introduces a virtual container where users can allocate funds for specific purposes. This feature enables users to input, monitor, and modify details for each money box to adapt to evolving financial objectives. The money box introduces two essential functions which are "raise fund for money box" and "withdraw money for money box." Through the "raise funds" function, users can channel surplus funds that are not immediately required into their designated money box. Conversely, the "withdraw money" function allows users to withdraw funds from the money box in case of emergencies. Additionally, users retain the capability to delete money boxes as necessary.

In addition, the proposed application provides a comprehensive account management feature. Users can customize and create categories for income and expenses and have the flexibility to set spending limits for each category. This functionality extends to creating and modifying wallets and events. In addition, users can use the alarm function to set reminders to ensure timely creation of transaction records. All in all, the use case demonstrated that the proposed application aims to provide a comprehensive financial planning functionality that covers a wide range of needs from day-to-day expense tracking to long-term financial planning.

3.2.2 Wireframe

Figure 3.2.2.1 Home Page

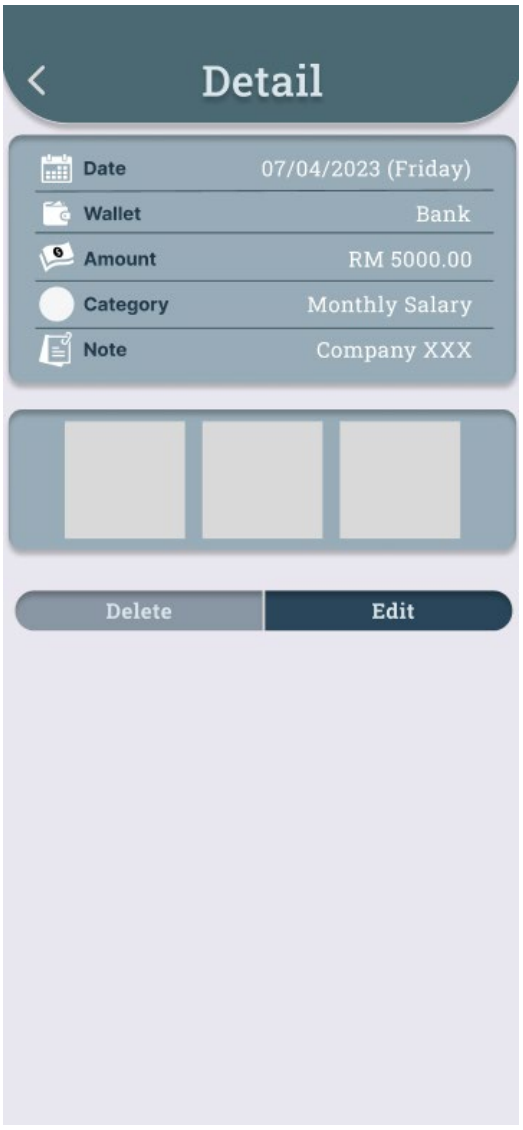


Figure 3.2.2.2 Detail Page

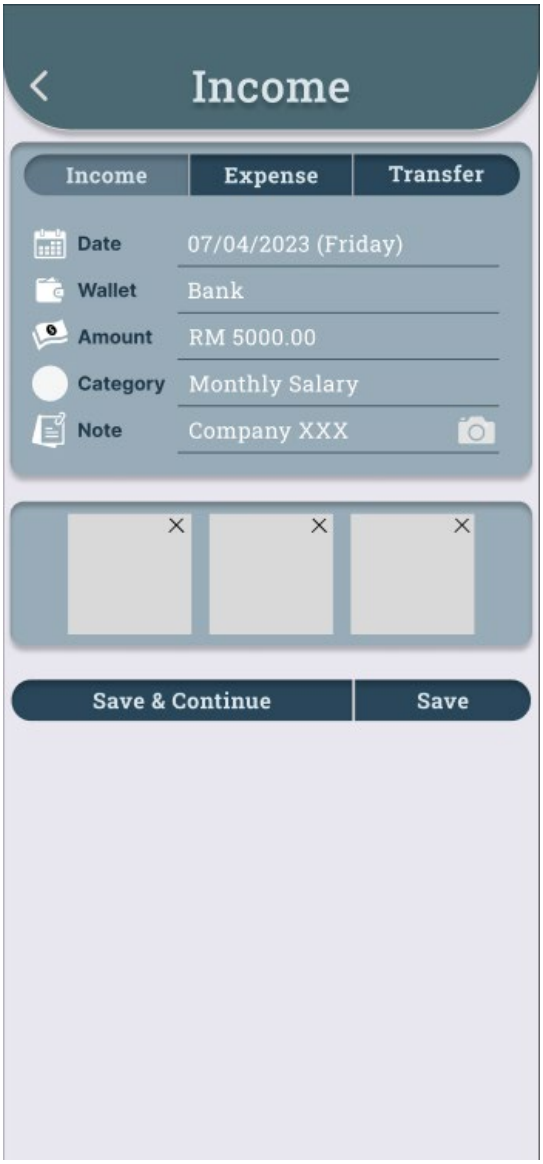


Figure 3.2.2.3 Transaction Page (Income Tab)

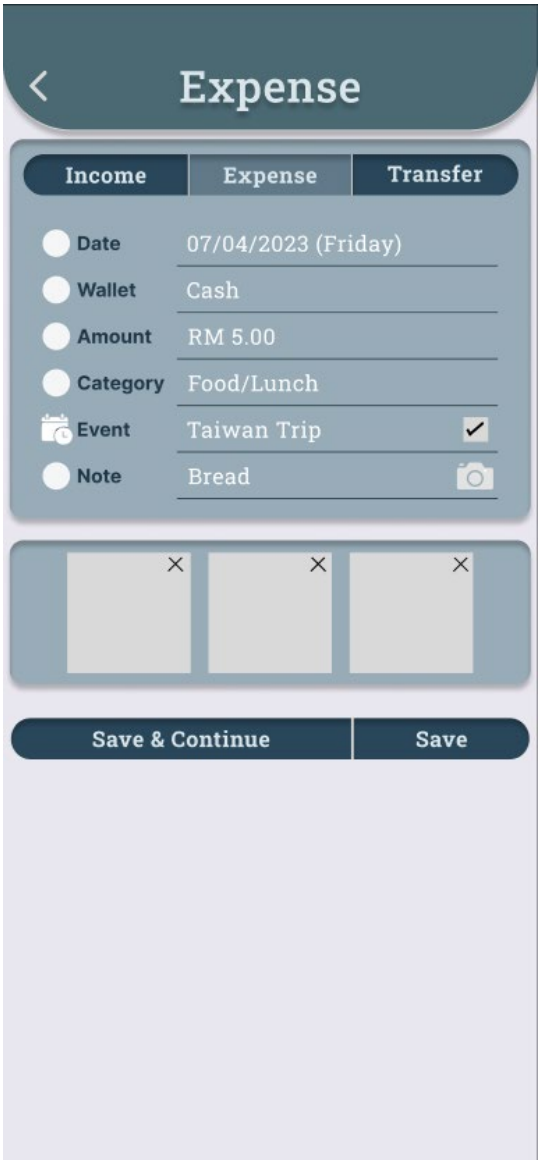


Figure 3.2.2.4 Transaction Page (Expense Tab)

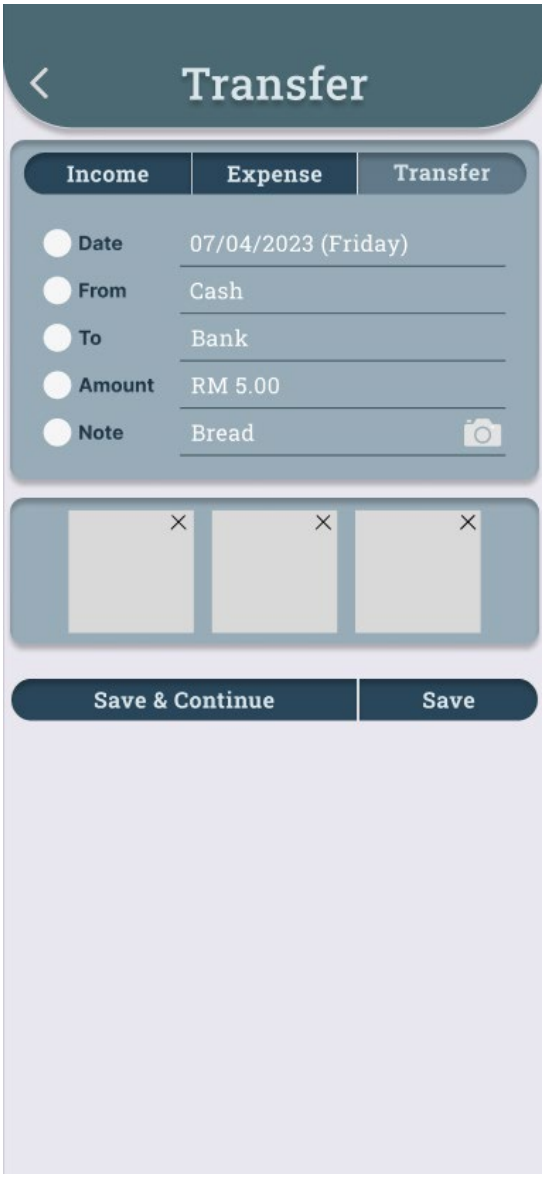


Figure 3.2.2.5 Transaction Page (Transfer Tab)

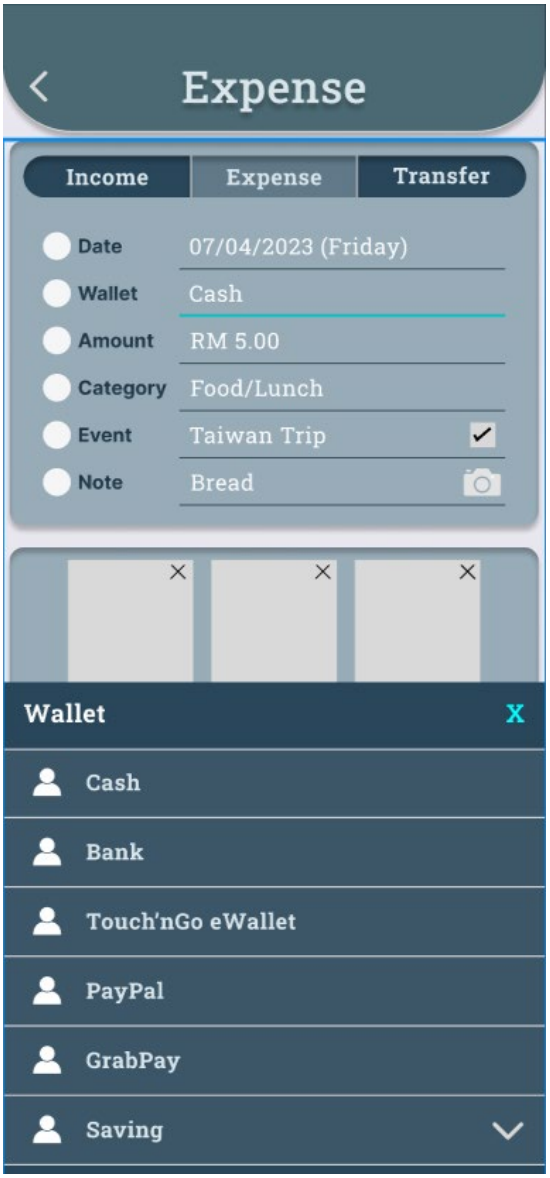


Figure 3.2.2.6 Transaction Page (Wallet Selection)

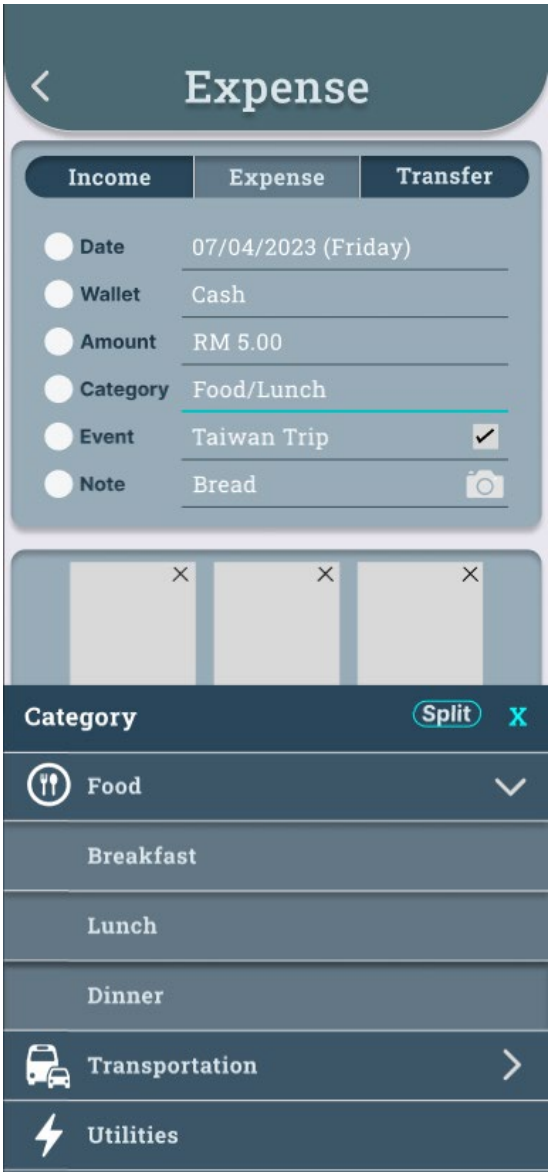


Figure 3.2.2.7 Transaction Page (Category Selection)

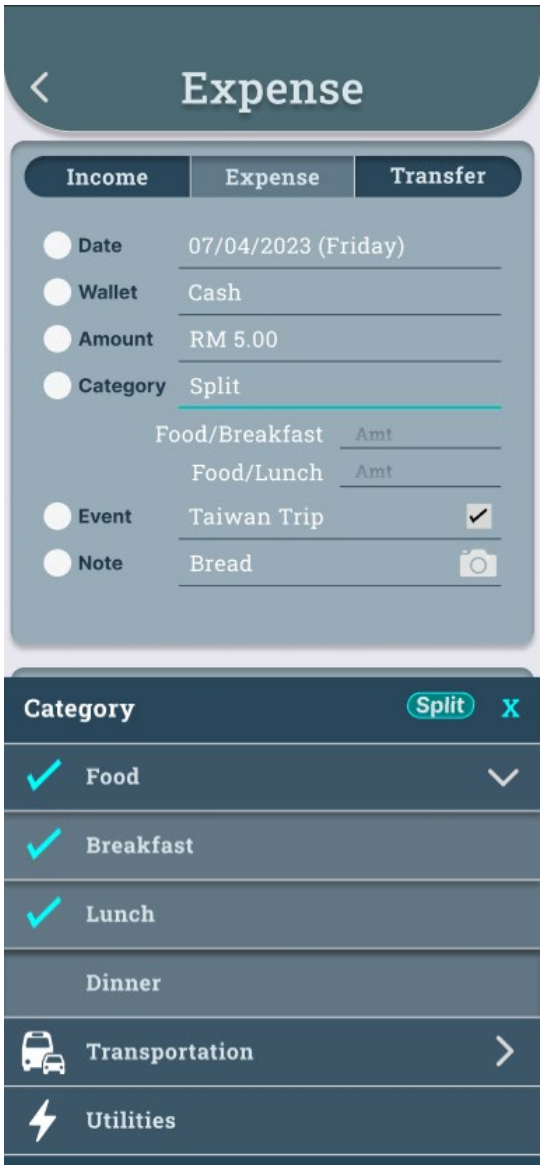


Figure 3.2.2.8 Transaction Page (Split Category)

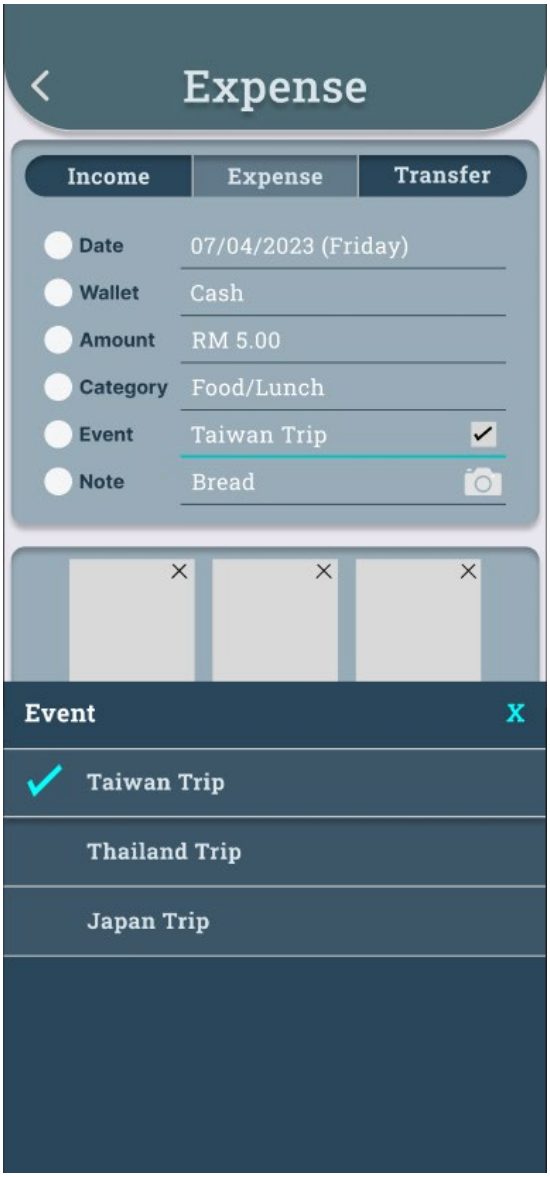


Figure 3.2.2.9 Transaction Page (Event Selection)

Figure 3.2.2.10 Report Page (List Tab)

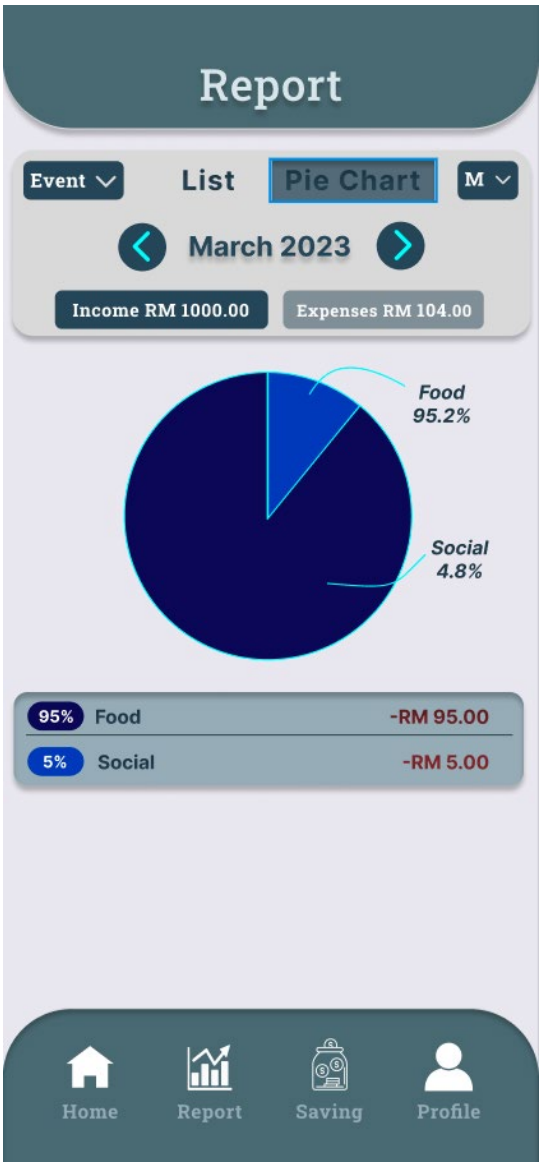


Figure 3.2.2.11 Report Page (Pie Chart)

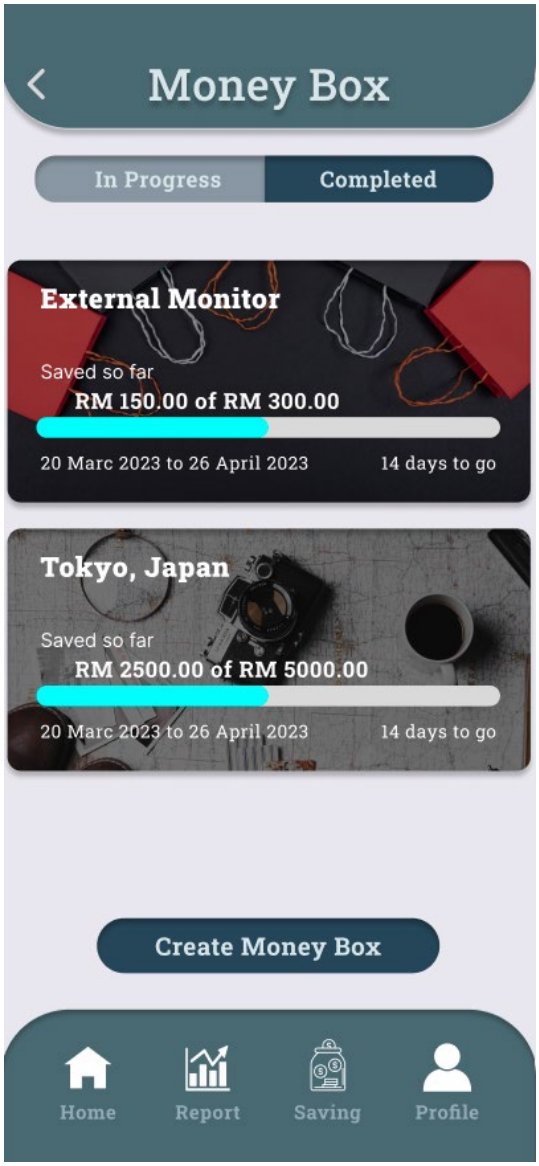


Figure 3.2.2.12 Saving Page (In Progress Tab)

Figure 3.2.2.13 Saving Page (Completed Tab)

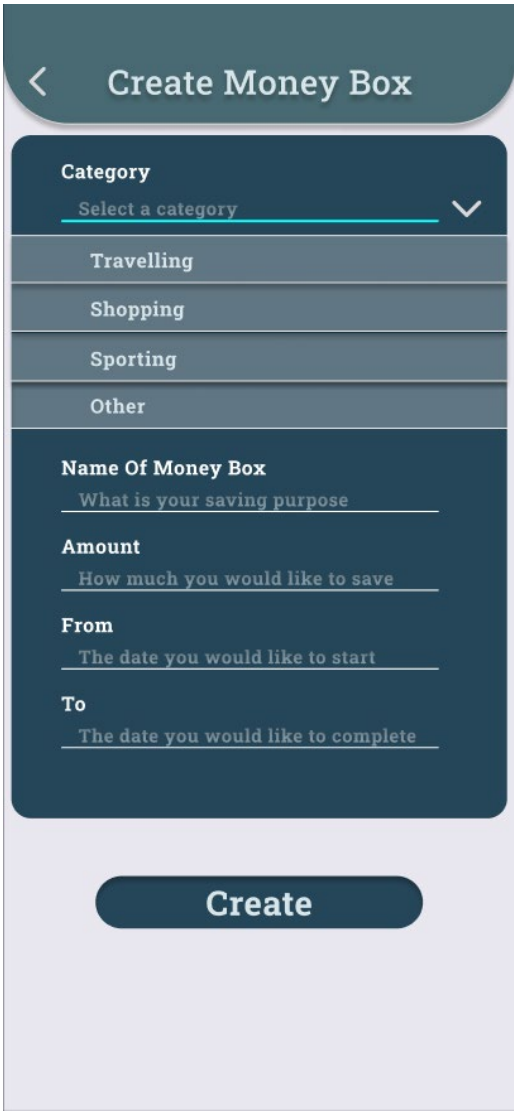


Figure 3.2.2.14 Create Saving Page

Figure 3.2.2.15 Saving Detail Page

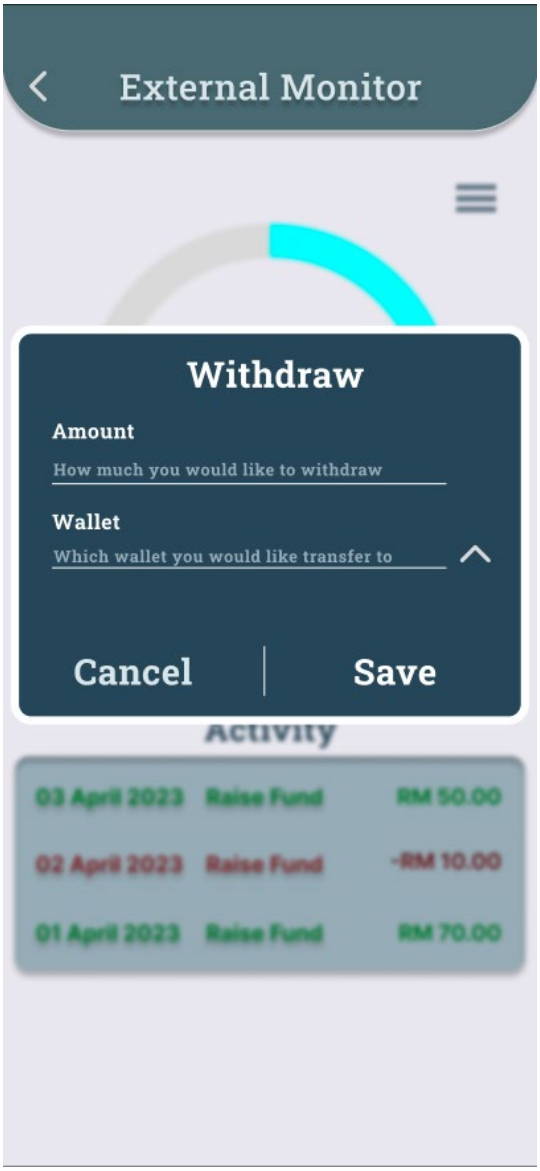


Figure 3.2.2.16 Saving Detail Page (Withdraw)

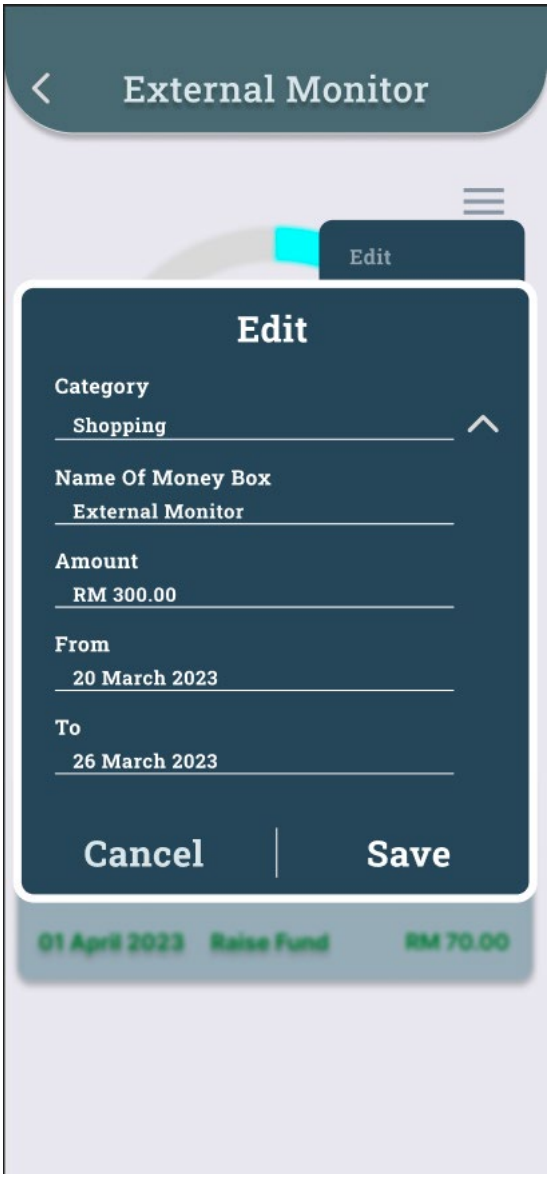


Figure 3.2.2.17 Saving Detail Page (Edit)

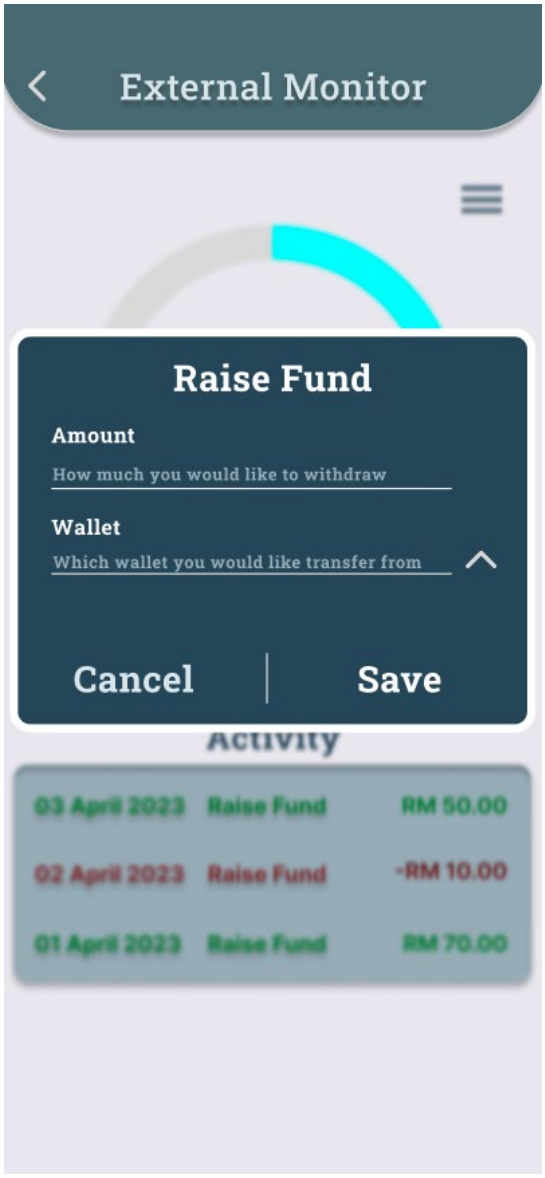


Figure 3.2.2.18 Saving Detail Page (Raise Fund)

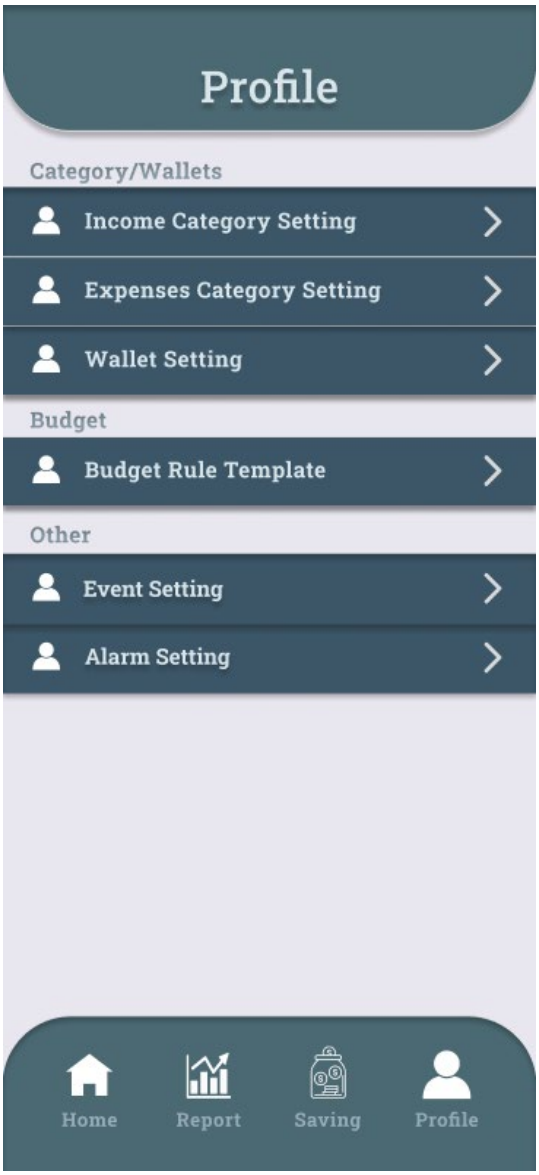


Figure 3.2.2.19 Profile Page

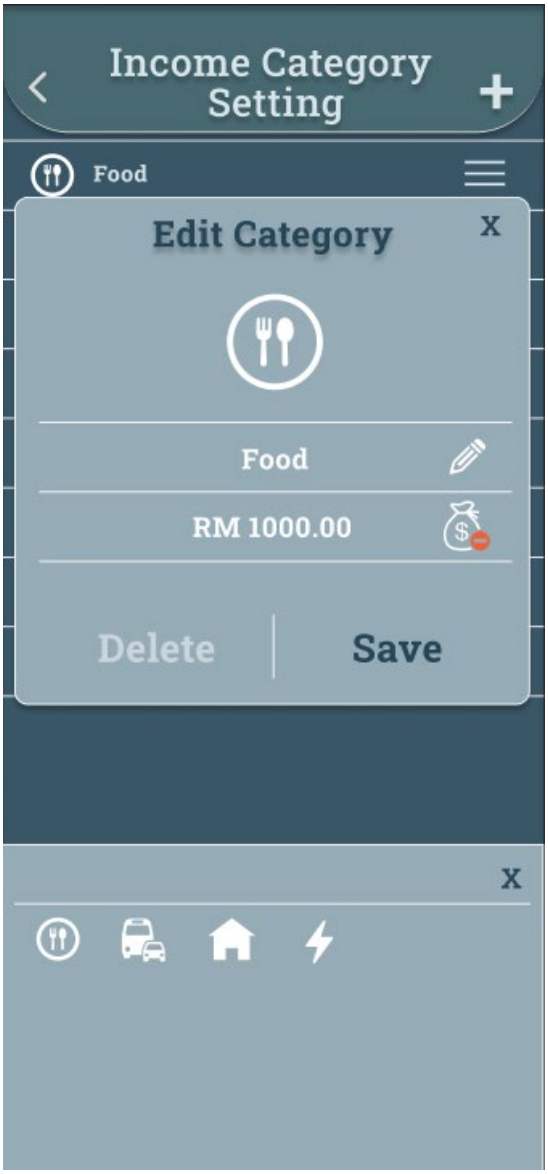


Figure 3.2.2.20 Income Category Setting Page (Edit)

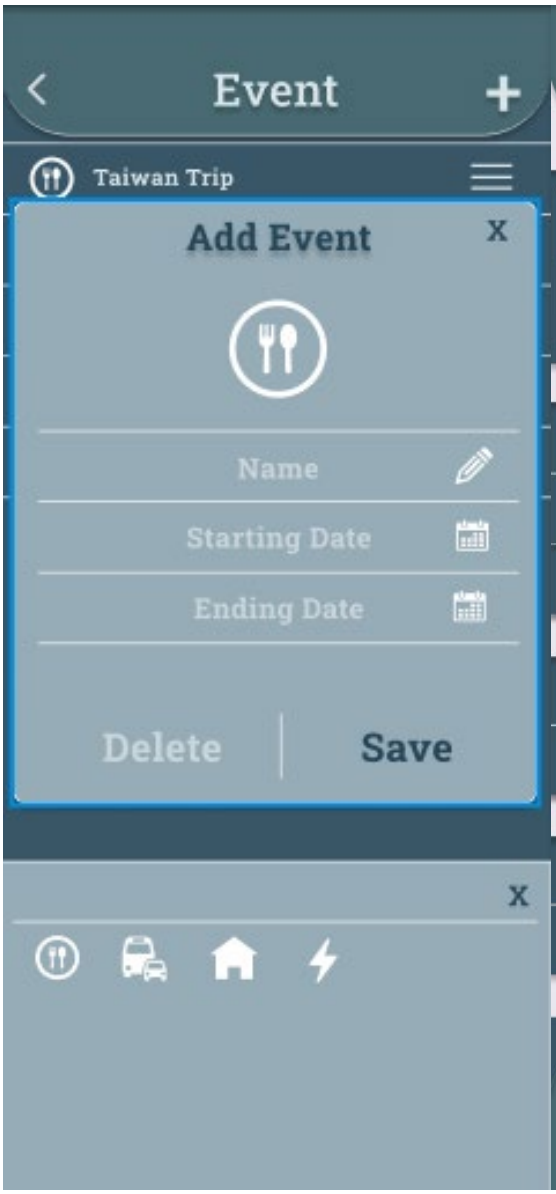


Figure 3.2.2.21 Event Page (Create)

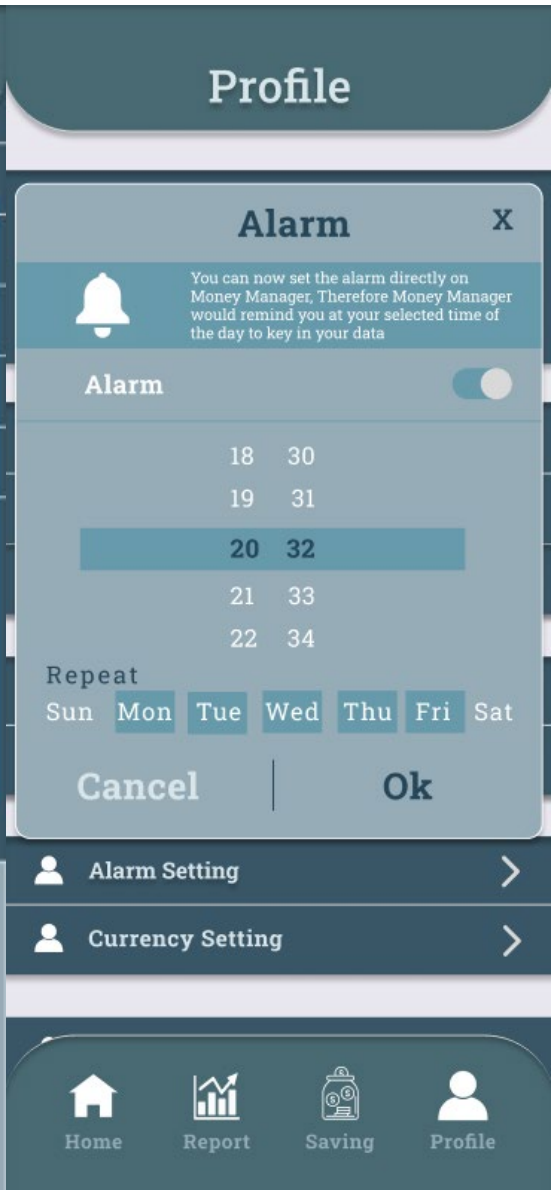


Figure 3.2.2.22 Profile Page (Alarm)

CHAPTER 4

4.1 Block Diagram

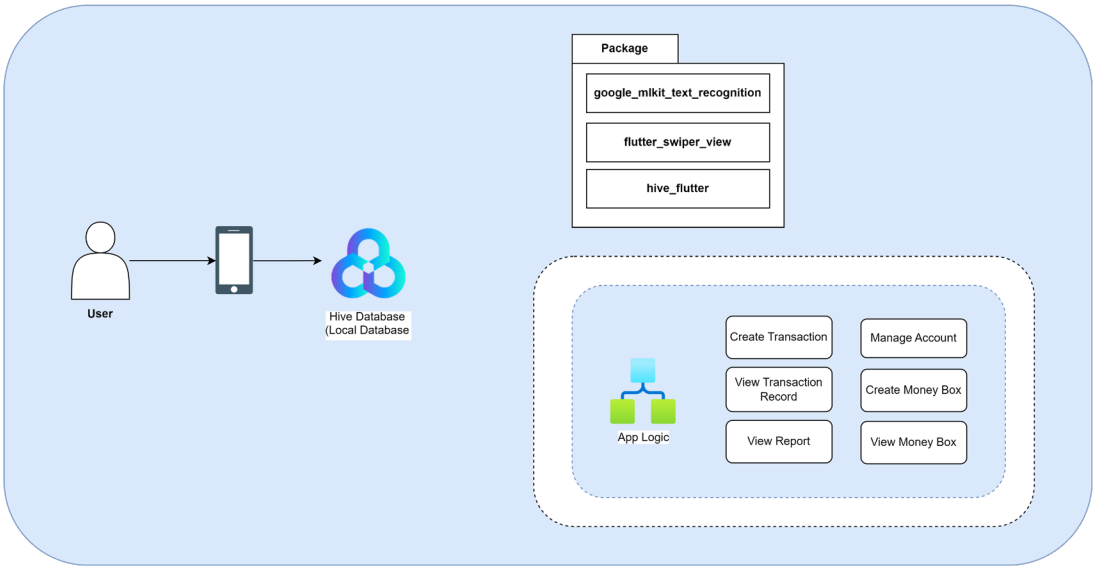
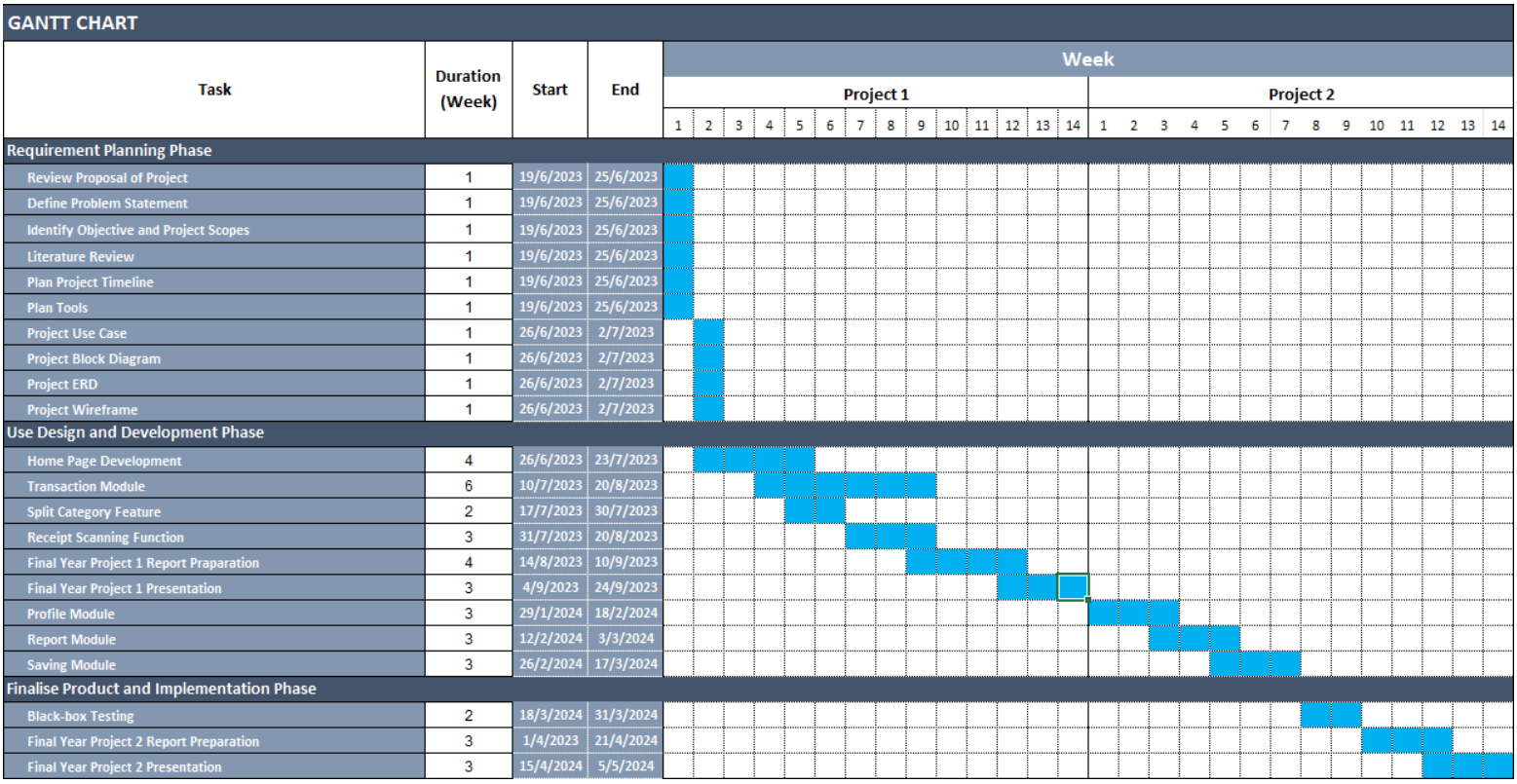


Figure 4.1.1

Figure 4.1.1 presents the block diagram of the proposed Personal Financial Planning Application. First and foremost, users will be accessing the application by using their smartphones. However, this proposed application contains two of the databases Hive Database (local database). Local database is used to support users to access the application without the internet. Apart from that, the application contains a lot of app logic to support the application's operations such as creating transactions, viewing transaction records, viewing reports, managing accounts, creating money boxes, and viewing money box. Moreover, the proposed application also imports some of the local packages of Flutter to gain the features offered by Flutter. For instance, google_mlkit_text_recognition (a package for text recognition), flutter_swiper_view (provides the best swiper for flutter, with multiple layouts and infinite loop to beautify the user interface), and hive_flutter (provides a local database).

4.2 Gantt Chart



GANTT CHART

Task	Duration (Week)	Start	End
Requirement Planning Phase			
Review Proposal of Project	1	19/6/2023	25/6/2023
Define Problem Statement	1	19/6/2023	25/6/2023
Identify Objective and Project Scopes	1	19/6/2023	25/6/2023
Literature Review	1	19/6/2023	25/6/2023
Plan Project Timeline	1	19/6/2023	25/6/2023
Plan Tools	1	19/6/2023	25/6/2023
Project Use Case	1	26/6/2023	2/7/2023
Project Block Diagram	1	26/6/2023	2/7/2023
Project ERD	1	26/6/2023	2/7/2023
Project Wireframe	1	26/6/2023	2/7/2023
Use Design and Development Phase			
Home Page Development	4	26/6/2023	23/7/2023
Transaction Module	6	10/7/2023	20/8/2023
Split Category Feature	2	17/7/2023	30/7/2023
Receipt Scanning Function	3	31/7/2023	20/8/2023
Final Year Project 1 Report Praparation	4	14/8/2023	10/9/2023
Final Year Project 1 Presentation	3	4/9/2023	24/9/2023
Profile Module	3	29/1/2024	18/2/2024
Report Module	3	12/2/2024	3/3/2024
Saving Module	3	26/2/2024	17/3/2024
Finalise Product and Implementation Phase			
Black-box Testing	2	18/3/2024	31/3/2024
Final Year Project 2 Report Preparation	3	1/4/2023	21/4/2024
Final Year Project 2 Presentation	3	15/4/2024	5/5/2024

Figure 4.2.1

4.3 Database (Non-Relationship)

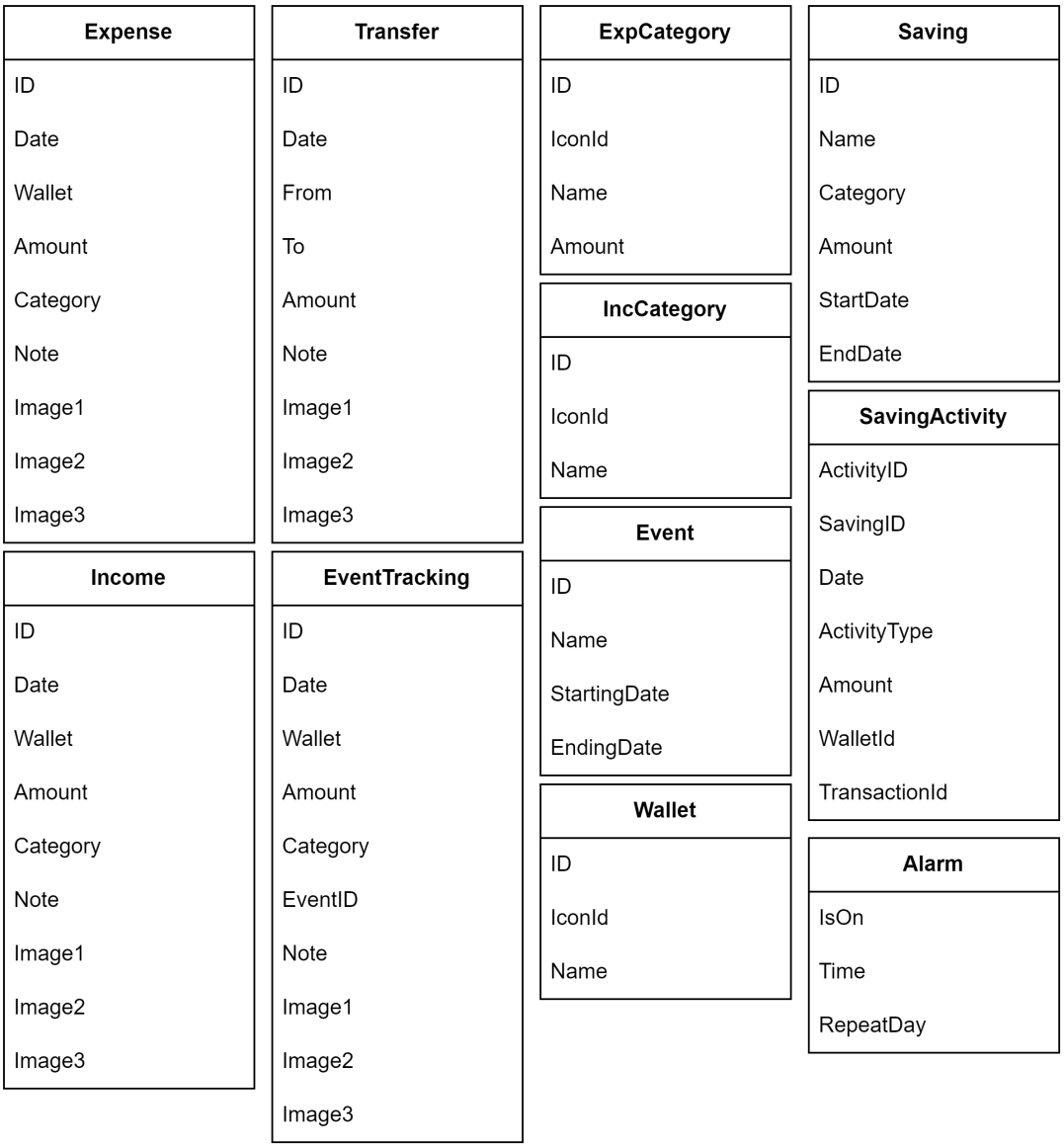


Figure 4.3.1

In this proposed application, there would be applied non-relationship databases which are Hive for local storage and Firebase online storage instead of the traditional relational database. Non-relational databases, often known as NoSQL databases, are contemporary and varied data management solutions that depart from the conventional relational database model. There are several reasons why choosing the non-relational databases to develop the proposed application.

First of all, since NoSQL databases are designed for speedy read and write operations, they are excellent for high-speed data access, which is crucial for real-time applications like mobile financial planning application. Unlike relational databases, NoSQL databases do not require complicated joins, making queries more effective. In

addition, NoSQL databases excel at offline functionality, which is a feature relevant to the proposed application. By leveraging local and online non-relational databases, users can seamlessly interact with applications even without an Internet connection. NoSQL databases are good at managing data synchronization between these two modes to ensure a smooth transition.

Moreover, NoSQL databases also increase the productivity of developers. Due to NoSQL databases being flexible, iterative development may easily modify the data model or structure without affecting previously existing data or running queries. However, the development and testing cycle can undoubtedly be accelerated with this agility. Besides, NoSQL databases are also the greatest option for applications that manage a range of financial transactions, user-generated content, and customized profiles in mobile apps since they are particularly suitable for dynamic or unstructured data circumstances. Their capability to effectively handle a variety of data types provides for its applicability.

In essence, the NoSQL database provides a comprehensive solution in terms of speed, offline capabilities, development efficiency, and compatibility with dynamic data needs, which works seamlessly with the needs of personal finance mobile application.

CHAPTER 5

5.1 Hardware Setup

Table 5.1.1 Hardware requirement

Description	Requirement
Model	Huawei Matebook D15
Processor	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx
Operating System	Windows 11
Graphic	AMD Radeon(TM) Vega 8 Graphics
Memory	8GB RAM
Storage	256GB SSD 1TB SATA HDD

5.2 Software Setup

Table 5.2.1 Software requirement

Description	Requirement
Programming Language	**Flutter** Flutter is a mobile UI framework for creating native apps for iOS and Android devices. Flutter supports a single code base to run on the different operating system (iOS and Android). Therefore, developers allow developing a mobile application with a language to support the different operating systems. **Dart** Dart is a Google-developed framework, comes with a BSD license, and is approved by the ECMA standard. It is a garbage-collected and class-based language with C-style syntax. Dart is an open-source framework which means it is free to use and available on every browser.
Tool	**Visual Studio Code** Visual Studio Code is a free and cross-platform source code editor developed by Microsoft. The software supports syntax highlighting, code auto-completion, code refactoring in an extended way, and has a built-in command line tool and Git version control system.
Database	**Hive** Hive is a lightweight, efficient, and easy-to-use local database solution and a UI toolkit developed by Google. It is particularly useful for storing structured data locally on the device by providing fast read and write operations. Hive is designed to work seamlessly with Flutter and Dart, making it a popular choice for mobile app development.

5.3 Setting and Configuration

5.3.1 Flutter Version

Verifying that the Flutter version matches the specified requirement of 3.10.4 is recommended. To figure out the current version of Flutter, run 'flutter --version' in the terminal. If the version of Flutter in use is different with the specified version, the user is offered with the choice to either downgrade or upgrade Flutter. To ensure that Flutter is running on that most updated version, run 'flutter upgrade' in the terminal. On the contrary, in the case where a downgrade to a particular version of Flutter is required, the command 'flutter downgrade 3.10.4' can be executed in the terminal.

5.3.2 Packages

```
30  dependencies:
31    flutter:
32      sdk: flutter
33    cupertino_icons: ^1.0.2
34    google_nav_bar: ^5.0.6
35    flutter_swiper_view: ^1.1.8
36    google_fonts: ^4.0.5
37    hive: ^2.2.3
38    hive_flutter: ^1.1.0
39    intl: ^0.17.0
40    flutter_spinkit: ^5.2.0
41    lottie: ^2.4.0
42    image_picker: ^0.8.9
43    random_string: ^2.3.1
44    polygon: ^0.1.0
45    google_mlkit_text_recognition: ^0.8.1
46    numberpicker: ^2.1.2
47    fluttertoast: ^8.2.4
48    flutter_neumorphic_plus: ^3.3.0
49    percent_indicator: ^4.2.3
50    syncfusion_flutter_charts: ^21.2.4
```

Figure 5.3.2.1

As illustrated in Figure 5.3.1.1, the 'pubspec.yaml' file provides a list of the installed packages that were employed as libraries in the project. The packages ‘cupertino_icons’, ‘google_nav_bar,’ and ‘flutter_swiper_view’ facilitate smooth navigation, interactive sliding views, and consistent icon depiction, respectively. The utilization of the 'google_fonts' package improves the quality of typographic components, whereas the 'intl' package makes the process of internationalizing elements. Additionally, the 'hive' and 'hive_flutter' packages facilitate the management of data and local database operations.

The 'lottie' and 'flutter_spinkit' utilities enhance the user experience by implementing customizable loading indicators and animations, which are specifically applied during page load times. Furthermore, the 'random_string' package facilitates the

generation of random strings for a variety of purposes, whereas the 'image_picker' package enables the selection of images from the local device files. Besides, the receipt scanning functionality is further enhanced by the 'google_mlkit_text_recognition' and 'polygon' packages, which implement geometric operations on the recognized text data contained within the receipts.

Moreover, the ‘numberpicker’ package provides a user-friendly interface for selecting numbers from a specified range, whereas the ‘fluttertoast’ package enables toast notifications to be displayed. The Flutter Neumorphic Plus package is a design trend that gives user interfaces a three-dimensional, tactile appearance through delicate shadows and highlights. The Percent Indicator package facilitates the creation of linear or circular progress indicators by developers. The package Syncfusion Flutter Charts enables the generation of an extensive variety of graphs and charts. It offers assistance for a wide range of chart types, such as line charts, bar charts, pie charts, and others.

In order to ensure a successful run of the code, it is critical to install all necessary libraries into the project. It can be achieved by carrying out the following steps: Paste the package name and version from Figure 5.3.1.1 into the dependencies section of the 'pubspec.yaml' file. After that, install the packages using the terminal command 'flutter pub get'. Alternatively, you can install each package individually by running the command `flutter pub add package_name` in the terminal, replacing `package_name` with the name of the package you want to install.

5.4 System Development

5.4.1 Local Database

Figure 5.4.1.1

Figure 5.4.1.1 shows the file named `MoneyMgrDatabase` and it manages the data from the local database which Hive database within the application. For storing several forms of financial data, such as account information, expense records, revenue records, transfer records, expense categories, revenue categories, alerts, event monitoring, events, savings, and so on, there are numerous lists included. Data manipulation, insertion, retrieval, and updating actions are all handled through class methods. Additionally, there are techniques for initializing and loading data that use either default

values or the Hive box. It allows you to track events, add income, expense, and transfer records, and make modifications to different account balances based on transactions.

5.4.2 Text Recognition

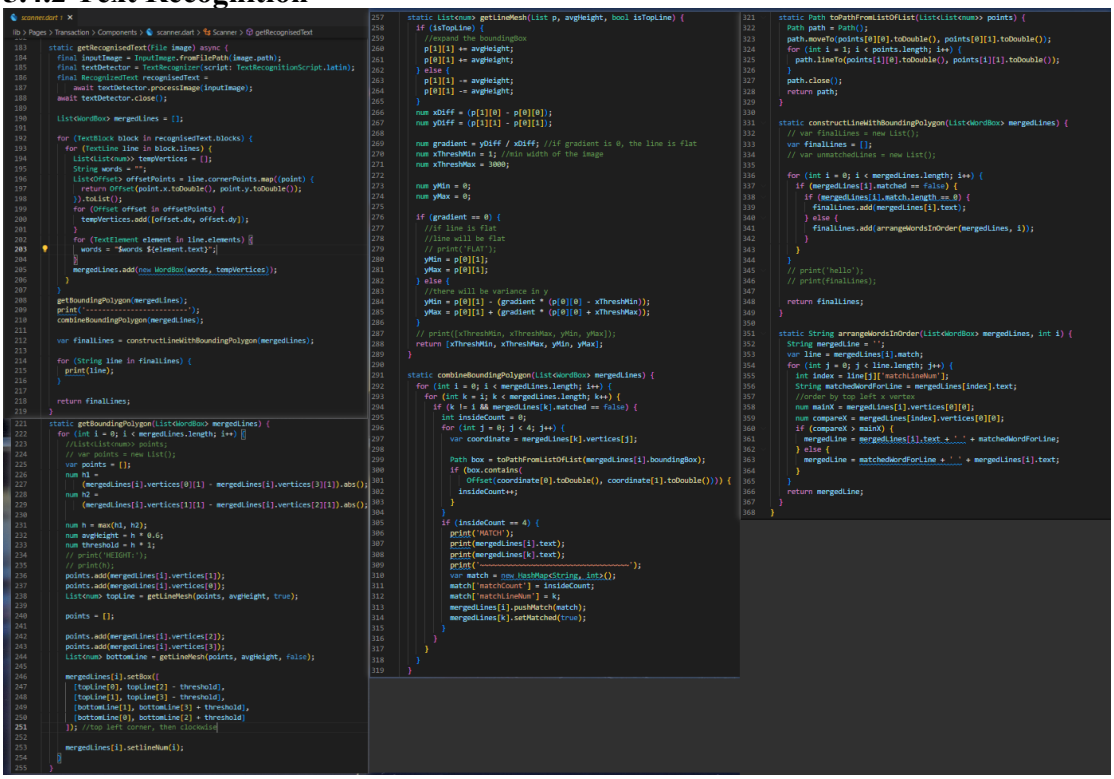


Figure 5.4.2.1

Figure 5.4.2.1 presents the codes of text recognition and there are containing several methods which are getRecognisedText, getBoundingPolygon, getLineMesh, combineBoundingPolygon, toPathFromListOfList, constructLineWithBoundingPolygon, and arrangeWordsInOrder. The purpose of these methods is to process the recognized text from an image of a receipt, identify lines of text that belong together (e.g., the lines of an item entry), and arrange the words in a coherent order based on their relative positions within the bounding polygons. The method essentially tries to structure the recognized text into meaningful content based on spatial relationships.

5.4.3 Regular Expression (Regex)

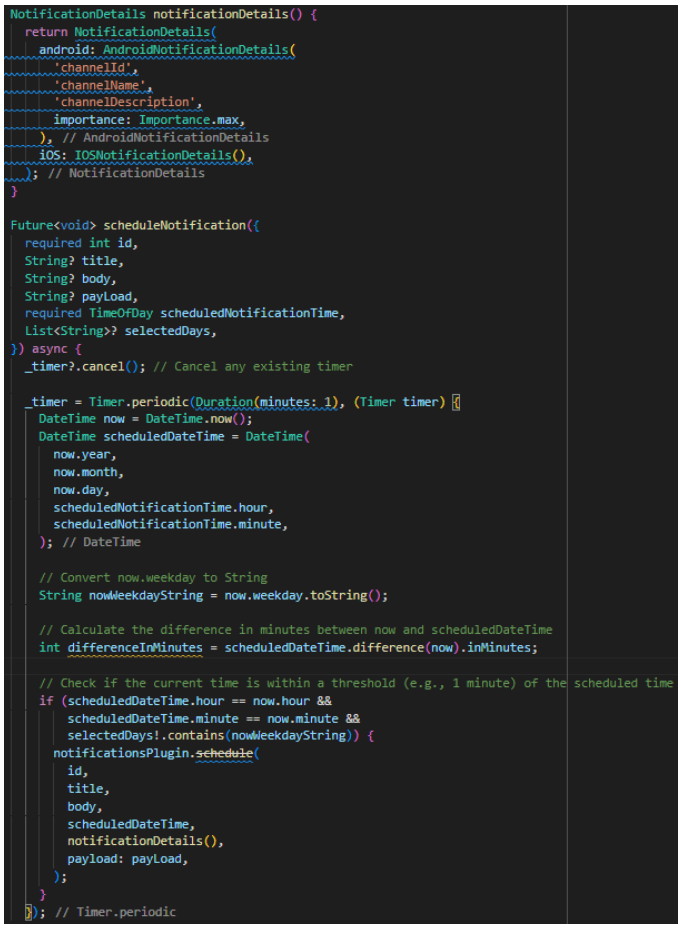
Figure 5.4.3.1

Figure 5.4.3.1 presents the two methods of codes from the scanner which are the findSubtotal and getItems methods. They are applying the regular expression to match the texts needed to obtain and there are several steps needed to perform.

1. Defines regular expressions (RegExp) for patterns like money values, total indicators ("total"), tax indicators ("tax"), and quantity indicators.
2. Scan the lines to find the highest money value, which is considered the total cost.
3. Removes lines after the total line, assuming they are not relevant to the calculation.
4. Removes lines containing "total" indicators, assuming they are not relevant.
5. Calculates potential tax based on the difference between the total and subtotal.

6. Scans for lines indicating tax or other charges, add their values to scanTax,and remove them.
7. Iterates through the lines to find items based on money values and quantities, considering only those items whose cumulative cost doesn't exceed the total.
8. For each eligible item, it calculates adjusted costs and quantities, removes the cost from the line, and constructs an Item object.
9. Prepares final values for total and tax based on the scanned values.

5.4.4 Notification Service



```
NotificationDetails notificationDetails() {
  return NotificationDetails(
    android: AndroidNotificationDetails(
      'channelId',
      'channelName',
      'channelDescription',
      importance: Importance.max,
    ), // AndroidNotificationDetails
    iOS: IOSNotificationDetails(),
  ); // NotificationDetails
}

Future<void> scheduleNotification({
  required int id,
  String? title,
  String? body,
  String? payLoad,
  required TimeOfDay scheduledNotificationTime,
  List<String>? selectedDays,
}) async {
  _timer?.cancel(); // Cancel any existing timer

  _timer = Timer.periodic(Duration(minutes: 1), (Timer timer) {
    DateTime now = DateTime.now();
    DateTime scheduledDateTime = DateTime(
      now.year,
      now.month,
      now.day,
      scheduledNotificationTime.hour,
      scheduledNotificationTime.minute,
    ); // DateTime

    // Convert now.weekday to String
    String nowWeekdayString = now.weekday.toString();

    // Calculate the difference in minutes between now and scheduledDateTime
    int differenceInMinutes = scheduledDateTime.difference(now).inMinutes;

    // Check if the current time is within a threshold (e.g., 1 minute) of the scheduled time
    if (scheduledDateTime.hour == now.hour &&
        scheduledDateTime.minute == now.minute &&
        selectedDays!.contains(nowWeekdayString)) {
      notificationsPlugin.schedule(
        id,
        title,
        body,
        scheduledDateTime,
        notificationDetails(),
        payload: payLoad,
      );
    }
  }); // Timer.periodic
```

Figure 5.4.4.1

Figure 5.4.4.1 is showing the codes of notification service development. The function ‘notificationDetails()’ that returns notification details specific to the Android and iOS platforms. For Android, it sets up notification details such as the channel ID, name, description, and importance level. For iOS, default notification details are used. The ‘scheduleNotification()’ function is defined to schedule notification. First, canceled any existing timer with using ‘timer_?.cancel()’ to prevent multiple timers from running simultaneously. Next, a new timer is initiated using ‘Timer.periodic(Duration(minutes: 1), ...)’. This timer triggers a callback function every minute to check if it's time to send a notification. It will check the current time, hour, and weekday are matched with the selected minute, hour, and day, if matched then a notification is scheduled using the ‘notificationsPlugin.schedule()’ method.

5.4.5 Progress Indicator (Saving Module)

```
1596  child: CircularPercentIndicator(
1597    radius: 125.0,
1598    animation: true,
1599    animationDuration: 2500,
1600    lineWidth: 25.0,
1601    percent: progress,
1602    center: Column(
1603      mainAxisAlignment: MainAxisAlignment.center,
1604      crossAxisAlignment: CrossAxisAlignment.center,
1605      children: [
1606        new Text(
1607          "Saved so far",
1608          textAlign: TextAlign.center,
1609          style: new TextStyle(
1610            fontWeight: FontWeight.bold,
1611            fontSize: 12.0,
1612            color: Color.fromRGBO(41, 70, 90, 1),
1613          ), // TextStyle
1614        ), // Text
1615        new Text(
1616          "\$ ${(savedAmt < 0 ? 0 : savedAmt).toStringAsFixed(2)} of \$ ${widget.saving[3]}",
1617          textAlign: TextAlign.center,
1618          style: new TextStyle(
1619            fontWeight: FontWeight.bold,
1620            fontSize: 14.0,
1621            color: Color.fromRGBO(41, 70, 90, 1),
1622          ), // TextStyle
1623        ), // Text
1624        Padding(
1625          padding: const EdgeInsets.only(top: 12.0),
1626          child: new Text(
1627            "Remaining",
1628            textAlign: TextAlign.center,
1629            style: new TextStyle(
1630              fontWeight: FontWeight.bold,
1631              fontSize: 12.0,
1632              color: Color.fromRGBO(41, 70, 90, 1),
1633            ), // TextStyle
1634          ), // Text
1635        ), // Padding
1636        new Text(
1637          "\$ ${remaining.toStringAsFixed(2)}",
1638          textAlign: TextAlign.center,
1639          style: new TextStyle(
1640              fontWeight: FontWeight.bold,
1641              fontSize: 14.0,
1642              color: Color.fromRGBO(8, 120, 39, 1)), // TextStyle
1643        ), // Text
1644      ],
1645    ), // Column
1646    circularStrokeCap: CircularStrokeCap.round,
1647    backgroundColor: Color.fromRGBO(217, 217, 217, 1.0),
1648    progressColor: Color.fromRGBO(139, 176, 201, 1),
1649  ), // CircularPercentIndicator
```

Figure 5.4.5.1

Figure 5.4.5.1 presents the circular progress indicator development in saving module. The radius of the circular progress indicator is set with 125 pixels, animation of the progress indicator is enable, duration time is 2500 milliseconds, and the line width with 25 pixels. The ‘percent’ property determines the progress displayed by the indicator, with its value assigned through the ‘progress’ variable. Within the ‘Column’ widget, the included code defines the text to be centered within the progress indicator.

‘CircularStrokeCap.round’ is set the shape of the ends of the progress indicator line to be round. 'backgroundColor' sets the background color of the circular progress indicator to a light gray and 'progressColor' sets the color of the progress indicator line to a shade of blue .

5.4.6 Bar Chart

```
child: SfCartesianChart(
  title: ChartTitle(
    text: 'Overall',
  ), // ChartTitle
  legend: Legend(
    isVisible: true,
    overflowMode: LegendItemOverflowMode.wrap,
    position: LegendPosition.bottom,
  ), // Legend
  primaryXAxis: CategoryAxis(
    labelRotation:
        -90, // Rotates labels for better fit
    interval: 1, // Display every label
    majorGridLines: MajorGridLines(
        width:
            0), // Optional: Hides grid lines for cleaner look // MajorGridLines
    labelAlignment: LabelAlignment
        .start, // Aligns labels to start
  ), // CategoryAxis
  series: <ChartSeries>[
    // Renders bar chart
    ColumnSeries<OverallData, String>(
        dataSource: getOverallChartData(),
        xValueMapper: (OverallData data, _) =>
            data.month,
        yValueMapper: (OverallData data, _) =>
            data.income,
        name: 'Incomes',
        color: Color.fromRGBO(162, 208, 220, 1.0)), // ColumnSeries
    ColumnSeries<OverallData, String>(
        dataSource: getOverallChartData(),
        xValueMapper: (OverallData data, _) =>
            data.month,
        yValueMapper: (OverallData data, _) =>
            data.expense,
        name: 'Expenses',
        color: Color.fromRGBO(255, 104, 109, 1.0)), // ColumnSeries
  ], // <ChartSeries>[]
), // SfCartesianChart
```

Figure 5.4.6.1

Figure 5.4.6.1 shows the code utilized for developing a bar chart within the report module to visualize the overview transactions data. However, a similar set of code is employed within the budget report section to present budget data using a bar chart.

1. Chart Title is defining the pie chart title.
2. The ‘Legend’ widget is used to set the label to be visible, position at bottom, and ‘LegendItemOverflowMode.wrap is the items will wrap to the next line if they do not fit within the available space.
3. In ‘primaryXAxis’, there are defined the rotation of label to be -90 degrees, display every label, hides grid lines, and align labels to start.
4. Two of the ‘ColumnSeries’ class are used to display the incomes and expenses data side by side within the chart. These series extract the monthly income and expense amounts, respectively from the ‘getOverallChartData()’ function.

Data Label Setting is describes the settings related to data labels. Set the label as visible, label position, connector line setting, and the text style.

5.4.7 Pie Chart

```
child: SfCircularChart(
  title: ChartTitle(
    text: chartTitle,
  ), // ChartTitle
  series: <CircularSeries>[
    DoughnutSeries<Map<String, dynamic>, String>(
      dataSource: summary.entries.map((entry) {
        return {
          'label':
              '${entry.key}\n${entry.value.toStringAsFixed(2)}%',
          'value': entry.value,
        };
      }).toList(),
      xValueMapper: (data, _) =>
          data['label'] as String,
      yValueMapper: (data, _) =>
          data['value'] as double,
      dataLabelMapper: (data, _) => data['label'],
      pointColorMapper: (data, index) {
        // Retrieve color from categoryColors map using modulo operation
        int colorIndex =
            index % categoryColors.length;
        String category = categoryColors.keys
            .toList()[colorIndex];
        return categoryColors[category] ??
            Colors
                .grey; // Default color if not found
      },
      dataLabelSettings: DataLabelSettings(
        showZeroValue: false,
        isVisible: true,
        // useSeriesColor: true,
        overflowMode: OverflowMode.shift,
        // labelIntersectAction: LabelIntersectAction.shift,
        labelPosition:
            ChartDataLabelPosition.outside,
        connectorLineSettings:
            ConnectorLineSettings(
              type: ConnectorType.curve,
              length: '15%'), // ConnectorLineSettings
        labelAlignment:
            ChartDataLabelAlignment.outer,
        textStyle: TextStyle(
          fontSize: 12,
        ), // TextStyle
      ), // DataLabelSettings
    ), // DoughnutSeries
  ], // <CircularSeries>[]
) // SfCircularChart
```

Figure 5.4.7.1

Figure 5.4.7.1 presents the code of pie chart development that using in report module for display the data of income, expense, and event transactions.

5. Chart Title is defining the pie chart title and text is passing from a declared variable.
6. In the series, there defined ‘DoughnutSeries’ for presenting data in the form of a doughnut chart which is pie chart.
7. The data source for the pie chart is specified using the ‘dataSource’ property. It is generated by mapping each entry of a ‘summary’ map into a list of maps.

8. The xValueMapper and yValueMapper properties are used to extract labels (Categories) and values (Percentages) for each data point from the data source, respectively.
9. The ‘pointColorMapper’ is used to dynamically assign colors for each slice.
10. Data Label Setting is describes the settings related to data labels. Set the label as visible, label position, connector line setting, and the text style.

5.5 System Operation (with Screenshot)

5.5.1 Home Page

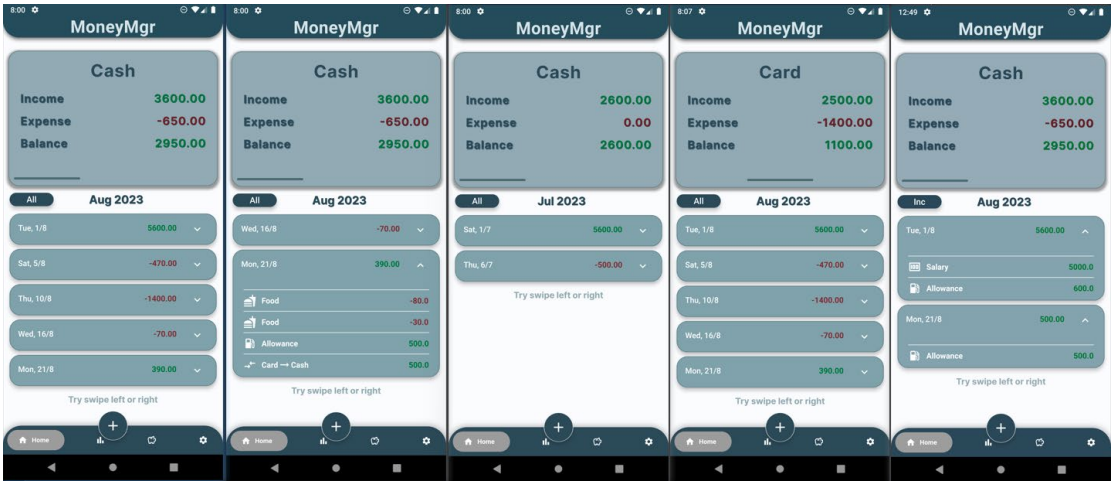


Figure 5.5.1.1

In Figure 5.5.1.1 is the home page's user interface of the application. The app bar plays the name of the application, and the bottom navigation bar is used to navigate to other pages. However, the app bar and bottom navigation bar will be displayed on every page of the bottom navigation bar. On the home page, they are differentiated into three sections which are top, middle, and bottom.

In the top section, there are the amounts used to show the income, expense, and balance of each wallet. The amounts of income and expenses are the sum of all income and expenses for the month shown in the middle section, respectively. Then, calculate the balance of the total income minus the total expenses. Users can swipe the section to view every wallet amount. After that, the middle section presents a button and a date. The purpose of the button is to control the type of transaction record displayed at the bottom. The purpose of the date is to control which month the transaction records need to be displayed and which month the amount of each wallet needs to sum.

Next, the bottom section displays every date of the month shown in the middle section that has any transaction records. There are also displaying the balance amount of every date. Each date can be expanded to view each transaction record in the date. Users can also swipe left or right of this section to view previous or next month's transaction records and the date shown in the middle section will be changed at the same time. At the end of the section, there are the hints for user to know the section can be swiped.

5.5.2 Transaction Page

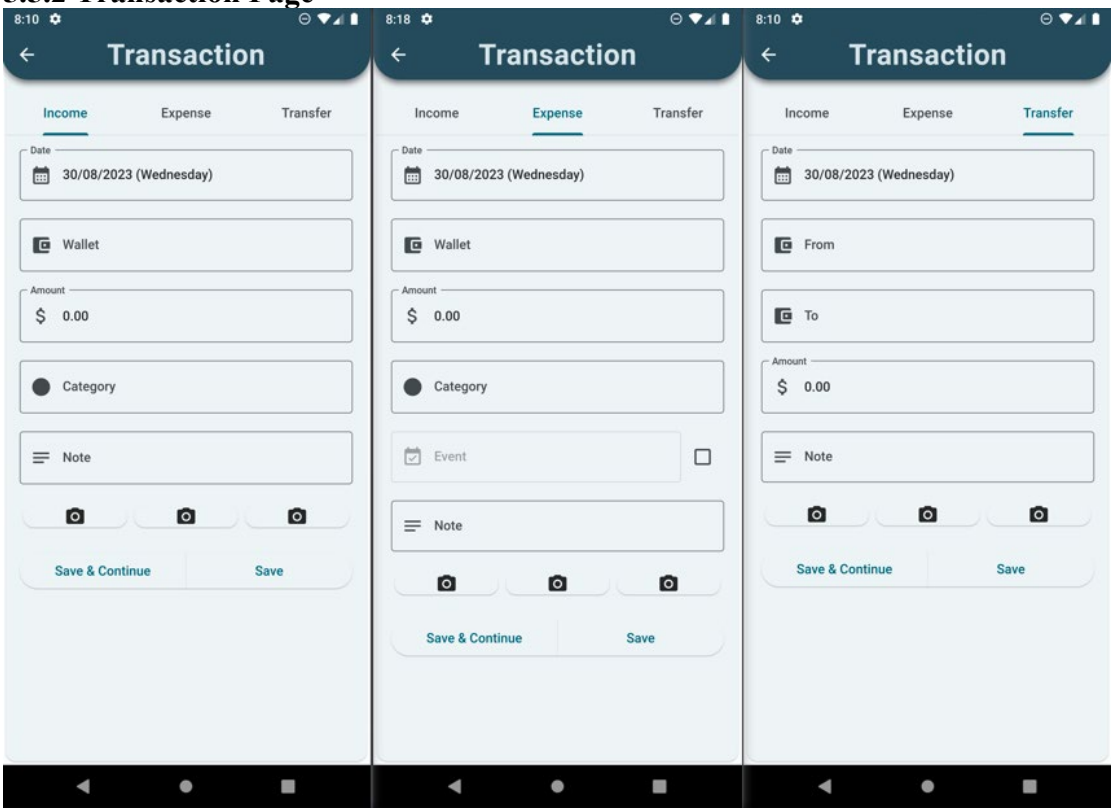


Figure 5.5.2.1

Figure 5.5.2.1 is the pages for creating transaction records and there are three tabs for three different transaction creation which are income, expense, and transfer transactions. All transaction types include crucial details like date, amount, notes, and imported images for the user to record their transaction details. The income transaction tab is used to record the transactions that users as the receiver. The income transaction record will increase the amount of the wallet based on the method through which the users have received money. In contrast, the expense transaction tab is used to record the user's expenditure transaction and the wallet amount will be decreased based on the payment method that users used. Besides, the transfer transaction records enable the transfer of funds between different wallets such as withdrawing RM200 in cash from a bank, which would lead to a deduction of RM200 from the bank wallet and an increase of RM200 in the cash wallet.

5.5.3 Split Transaction

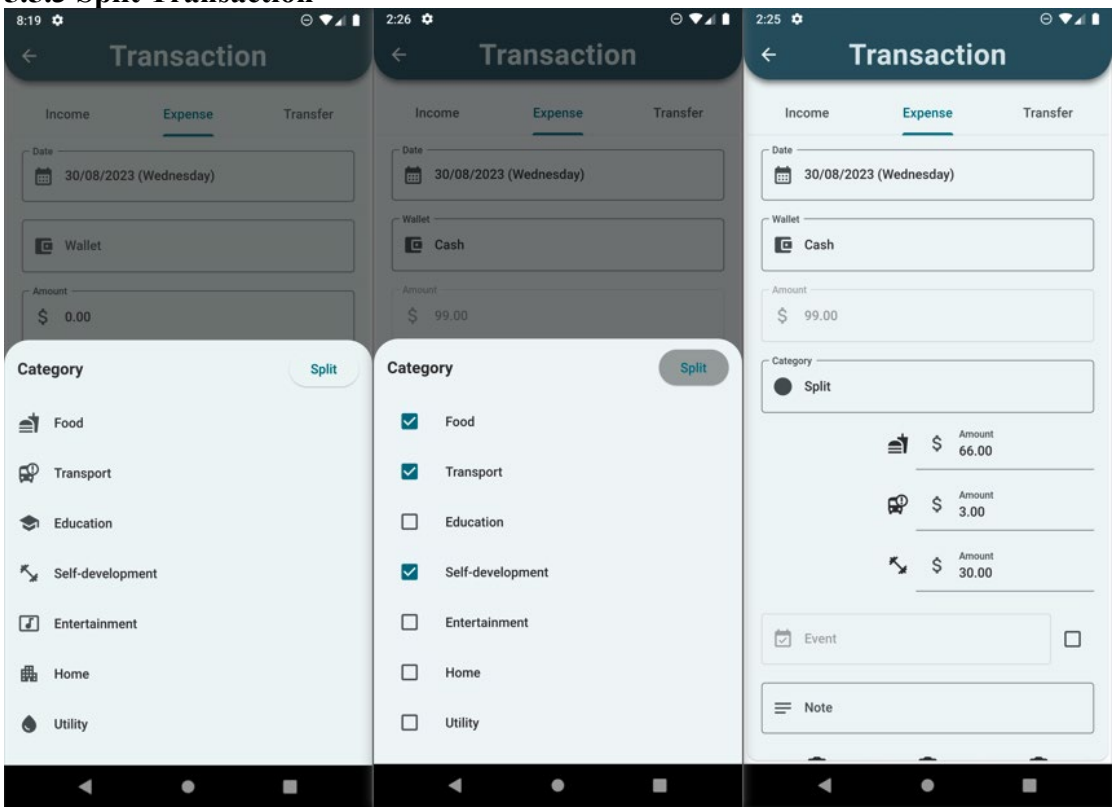


Figure 5.5.3.1

In Figure 5.5.3.1, there is a bottom sheet with all of the expense categories in the expense transaction tab. However, the user can choose only one category to represent the category of the transaction spending; once the 'Split' button is pressed, there will be display all the expense categories with a check box for user the to select multiple categories. After that, the select categories will be displayed on the transaction page with an amount text box for each category. This feature makes providing convenient for users to record multiple categories of expenses in a single transaction instead of creating many transaction records.

5.5.4 Event Transaction

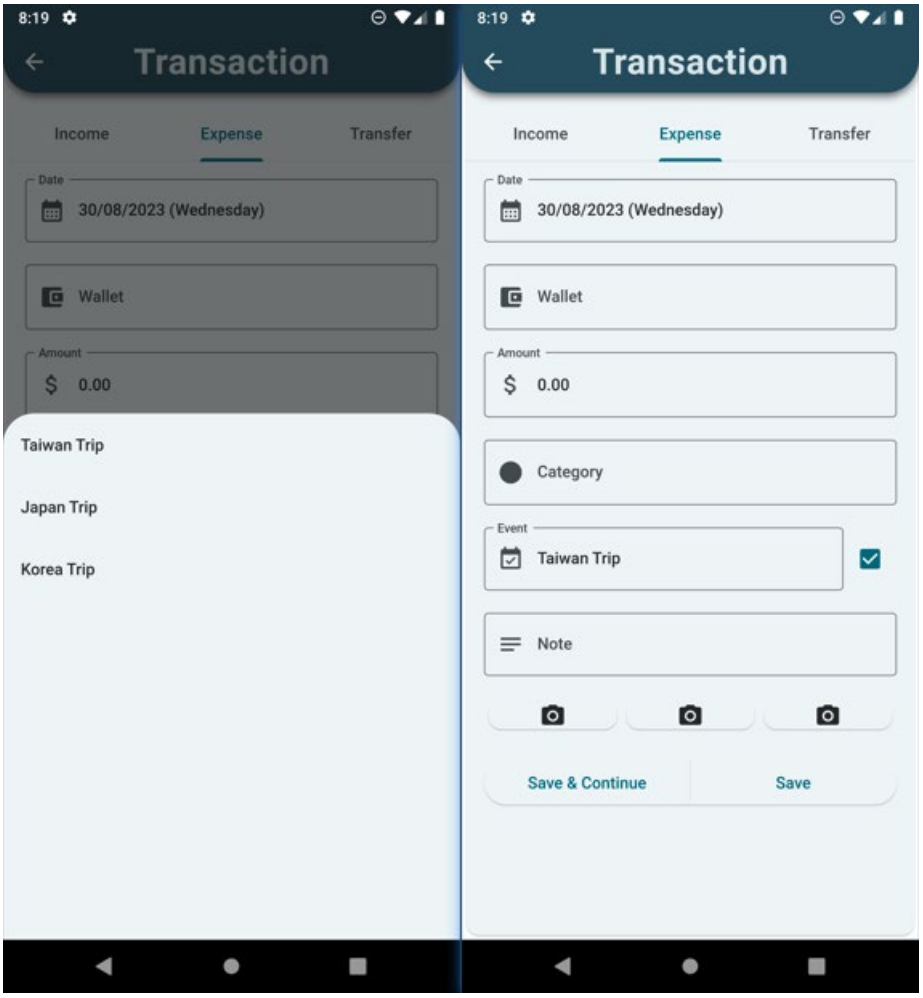


Figure 5.5.4.1

Figure 5.5.4.1 presents the event transaction creation. After checking the check box beside the event text box, the event text will be able to access and select the event that was created in the bottom sheet. This feature provides the ability for users to carefully monitor expenses associated with specific occasions and this transaction would not be recorded in daily expenses. After the transaction creation, the user can overview the event transaction records in the event report of the report module.

5.5.5 Receipt Scanning

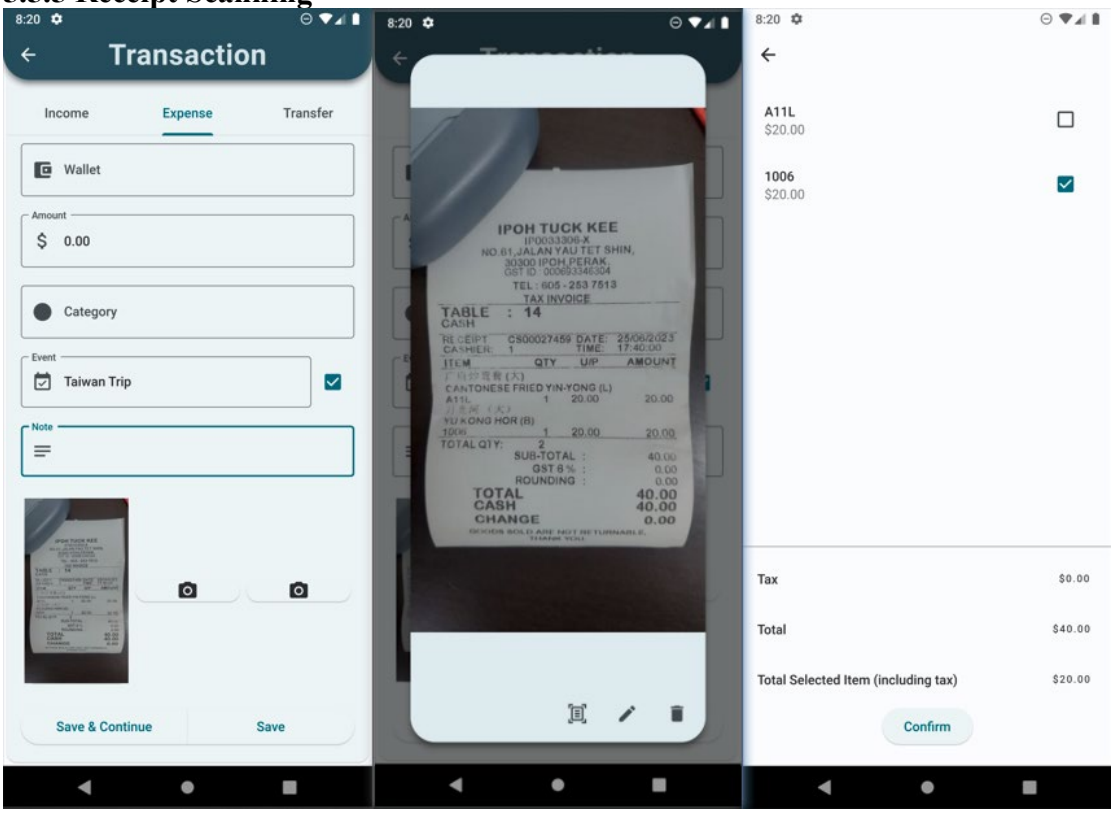


Figure 5.5.5.1

Figure 5.5.5.1 presents the function of receipt scanning. After importing the receipt picture, the user can access the scanning button in the image viewing dialog and navigate to the scanning page. Before navigating to the scanning page, the application will recognize the text from the image and process with the logic to arrange the item name, item amount, tax amount, and total amount on the scanning page. On the scanning page, the user can select one or more items to total up the amount and return to the transaction creation page.

5.5.6 Report (Budget)

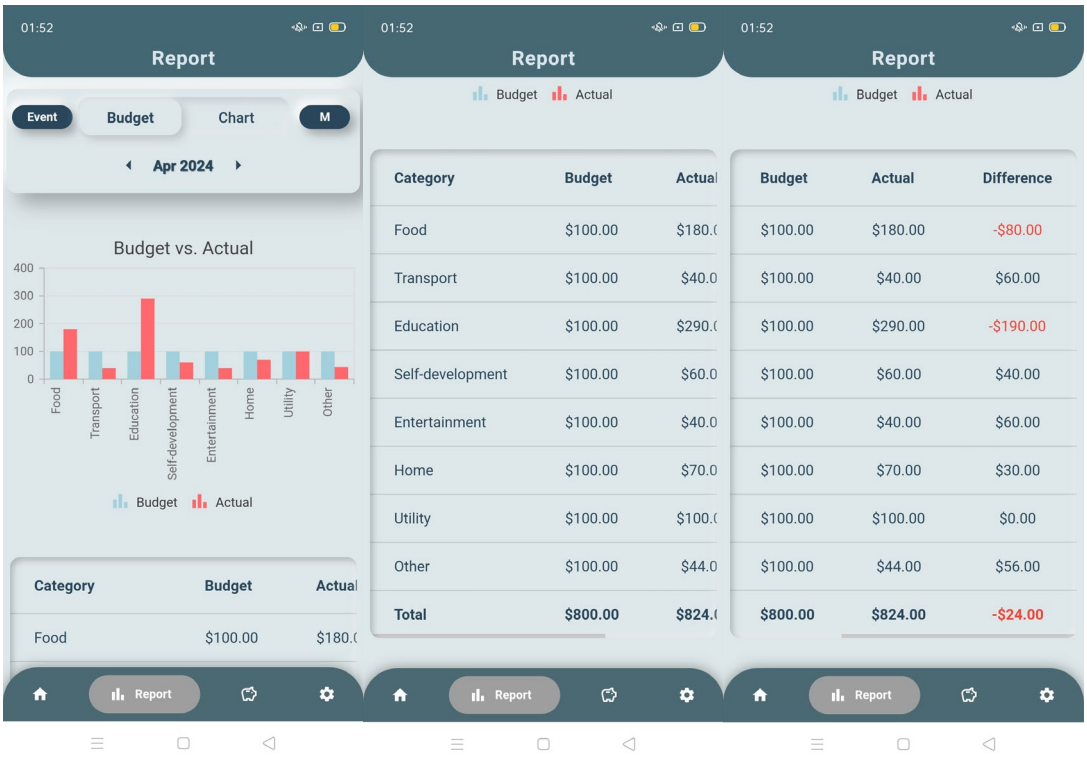


Figure 5.5.6.1 Figure 5.5.6.2 Figure 5.5.6.3

Once users access the 'Report' page from the bottom navigation bar, they will be able to view the Budget report, as depicted in Figure 5.5.6.1. Figure 5.5.6.2 illustrates the table of budget details as users scroll down the page, with the table capable of horizontal scrolling to display all the necessary information, similar to Figure 5.5.6.3. This information contains 'Category', 'Budget', 'Actual', and 'Difference'. At the top of the page, users will find a section facilitating report monitoring, allowing them to select existing events by tapping the 'Event' button, and filter report data by period, including weekly, monthly, and yearly, through buttons located on the top right side. Users are also afforded the flexibility to change between months, enabling them to easily switch to the previous or next month to view reports for different periods.

Furthermore, a bar chart provides a visual representation for users to easily compare their predefined budgets with actual expenses. In this visualization, blue candles represent budget allocations for each category, while red candles represent actual expenditures. This visual helps users to quickly identify categories where they have keep expenses within the predefined budget or exceeded it. Additionally, the table presents each category's name, budget allocation, actual expenditure, and difference

between the two. In the 'Difference' column, amounts exceeding the budget will be highlighted in red, indicating when users have overspent in a particular category.

5.5.7 Report (Chart, Overall)

Figure 5.5.7.1 Figure 5.5.7.2 Figure 5.5.7.3

When users toggle to the 'Chart' view, they will access the overall transaction report, as depicted in Figure 5.5.7.1. At the top of the page, there are distinct features compared to the 'Budget' page which are the period filtering button is disabled, allowing users to switch the period only to yearly. Below the date, three buttons are presented which are 'Overall', 'Incomes', and 'Expenses', those enable users to view different transaction reports.

Within the bar chart, users can observe the incomes and expenses for each month of the year. Blue candles represent the income amounts for each month, while red candles represent the expenses. This visualization enables users to compare their monthly income and expenses to assess whether they have spent more or less than their income. In months where there are no transactions, no candles will be displayed.

At the bottom of the page (Figure 5.5.7.2 and Figure 5.5.7.3), a table similar to the one on the 'Budget' page is presented, but with different column names. These columns include 'Month', 'Total Income', 'Total Expense', and 'Difference'. If there are no transactions in a particular month, the corresponding amounts will be displayed '0.00'. Additionally, if the expenses for a month exceed the income, the amount in the 'Difference' column will be highlighted in red.

5.5.8 Report (Chart, Incomes)

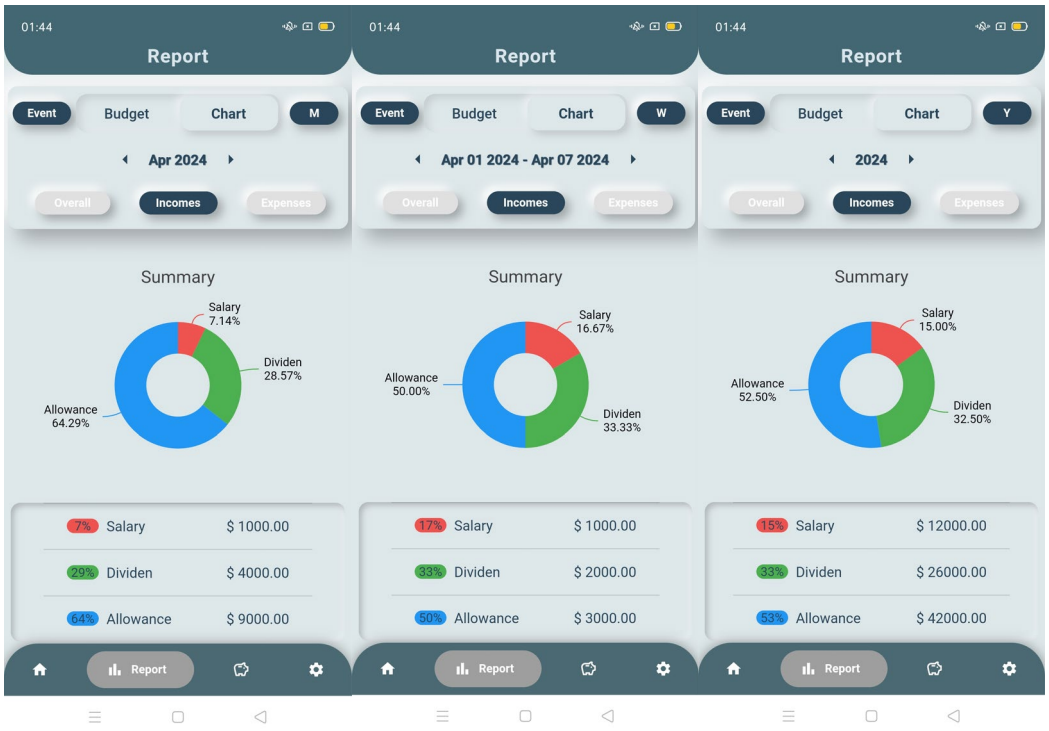
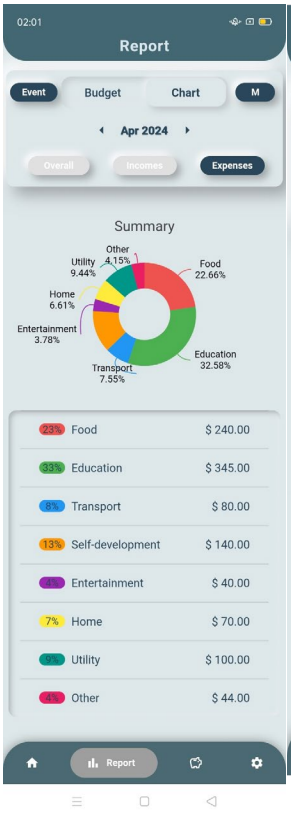
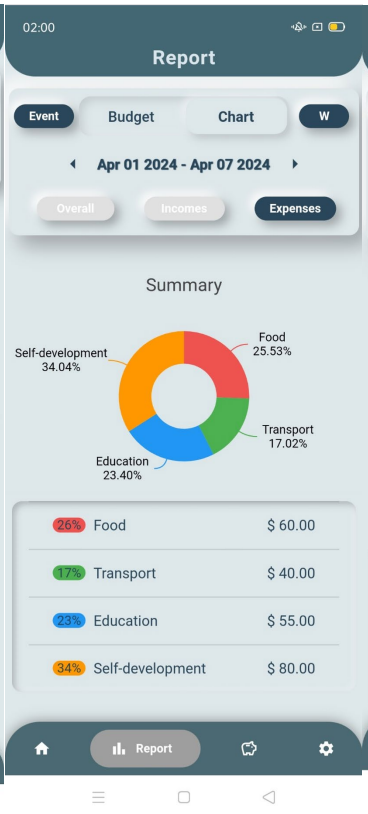


Figure 5.5.8.1 Figure 5.5.8.2 Figure 5.5.8.3

When users tap the 'Incomes' button, they will access the income report, as depicted in Figure 5.5.8.1. On this page, a pie chart and a table will be displayed, presenting the income transaction report for the specific period selected. The data will summarize all amounts for the specified period and calculate the percentage of total income for each income category. Each income category and its corresponding percentage will be visualized in both the pie chart and the table. The background color of each category's percentage tab in table will match the slice color in the pie chart, providing visual coherence. Users can navigate between dates by tapping the left and right icons and can also adjust the period filtering between weekly (Figure 5.5.8.2), monthly (Figure 5.5.8.1), or yearly (Figure 5.5.8.3) using the period button located on the top right of the page.

5.5.9 Report (Chart, Expenses)





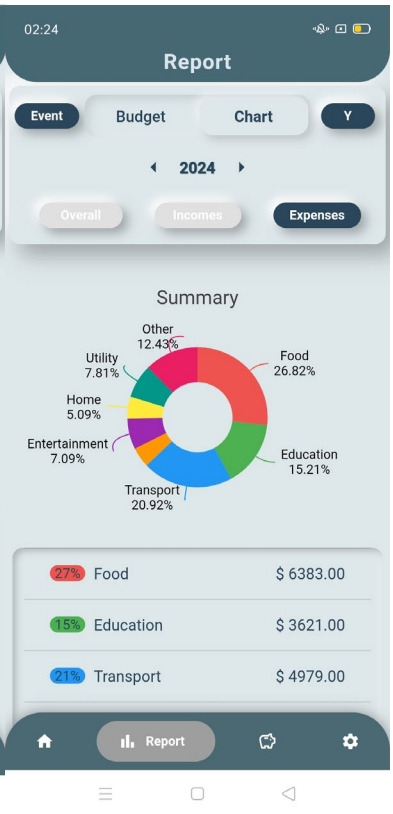


Figure 5.5.9.1 Figure 5.5.9.2 Figure 5.5.9.3

Tapping the 'Expenses' button grants users access to the expense report, which features the same pie chart and table layout as the income report. The application will calculate the percentage of total expenses for each expense category, with the background color of each percentage matching the color of the corresponding slice in the pie chart. Users are also able to adjust dates and filtering periods as needed. For instance, Figure 5.5.9.1 displays the expense report for a monthly view, Figure 5.5.9.2 illustrates the weekly expense report, and Figure 5.5.9.3 showcases the yearly expense report.

5.5.10 Report (Chart, Event)

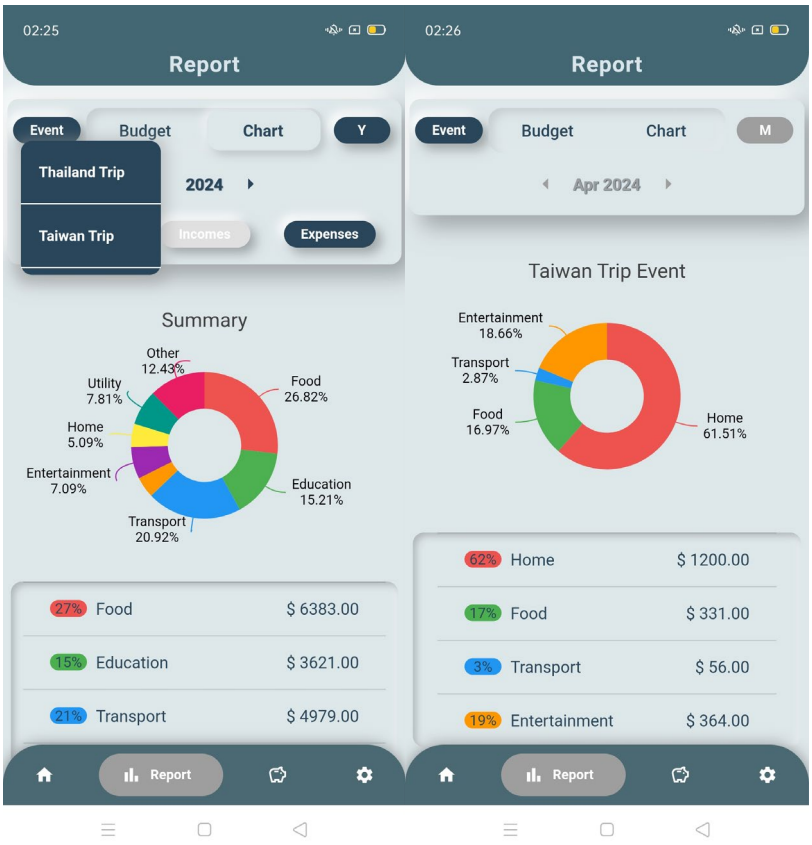


Figure 5.5.10.1 Figure 5.5.10.2

In Figure 5.5.10.1, the illustration depicts the event selection drop-down box that appears upon tapping the 'Event' button. Once an event is selected, the corresponding event transaction report will be presented, as shown in Figure 5.5.10.2. Similar to the incomes and expenses pages, this page also features the same pie chart and table layout to display the transaction report for the selected event. However, both the date and period filtering buttons are disabled for users, who can instead toggle between 'Budget' or 'Chart' to switch back to the other report.

5.5.11 Saving

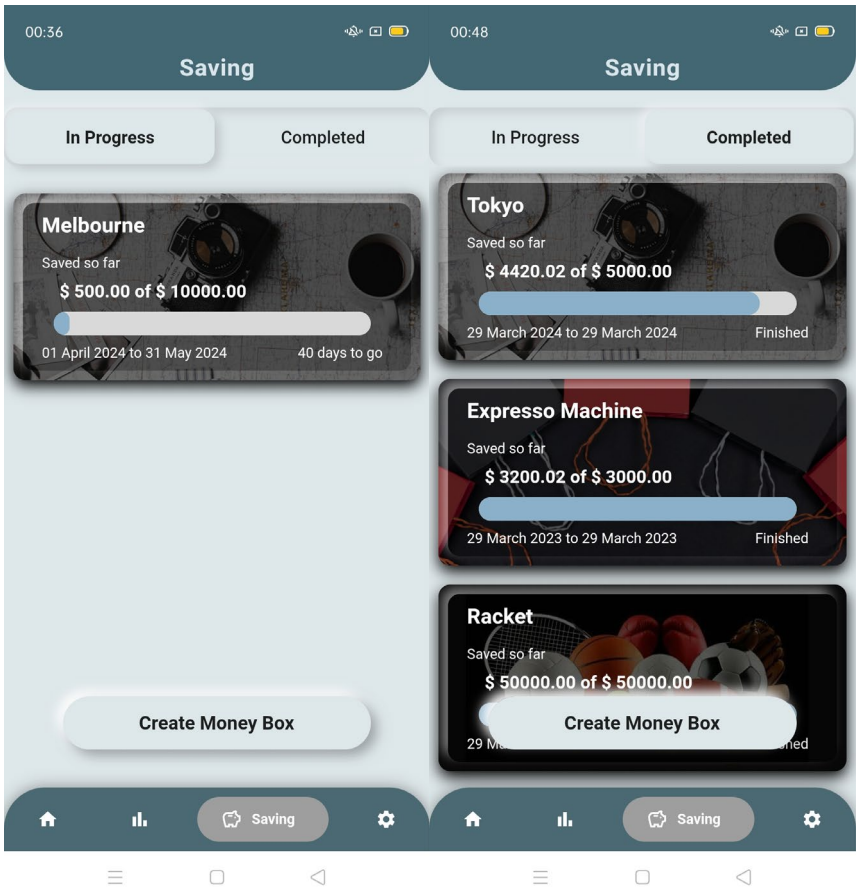


Figure 5.5.11.1 Figure 5.5.11.2

When users navigate to the ‘Saving’ page from the bottom navigation bar, they will see the money boxes, showcased in Figures 5.5.11.1 and 5.5.11.2. In Figure 5.5.11.1, upon toggling the ‘In Progress’ option, users will find the money boxes currently undergoing saving progress. Each money box is labeled with a name, such as ‘Melbourne’, along with its designated saving purpose. The total target saving amount and the current saved amount are displayed prominently in each money box with a progress indicator for visual representation. Additionally, the starting and ending saving dates are provided, spanning from '01 April 2024' to '31 May 2024', alongside the remaining days until completion ('40 days to go'). In contrast, upon toggling the ‘Completed’ option, users will find that the money boxes have passed their ending saving dates, and the remaining days displayed as ‘Finished’. The background picture of the money box is displayed based on the category of money box that users selected. At the bottom of the page, a ‘Create Money Box’ button is conveniently located to allow users to navigate to the page for creating new money boxes.

5.5.12 Saving (Create Money Box)

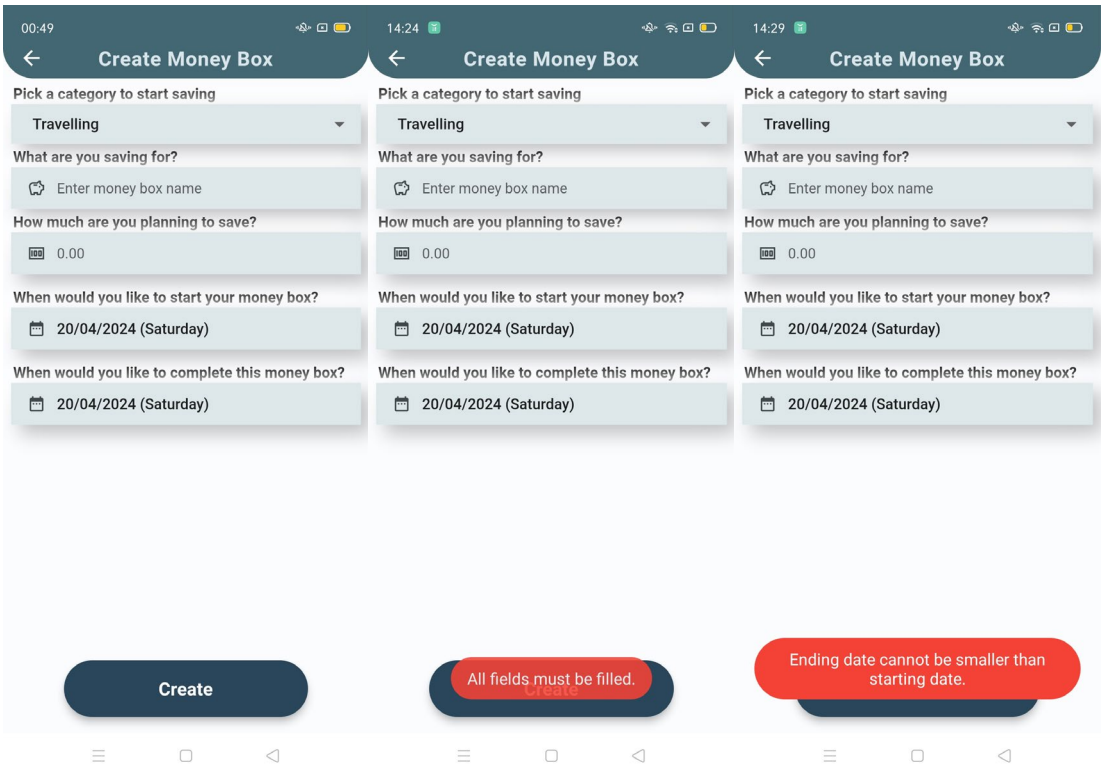


Figure 5.5.12.1 Figure 5.5.12.2 Figure 5.5.12.3

When users access the ‘Create Money Box’ page, they will see five text boxes prompting them to input specific details which are the category of saving, the name of the money box, the target amount to save, as well as the starting and ending saving dates, as depicted in Figure 5.5.12.1. It's mandatory for all fields to be filled in completely; otherwise, tapping the ‘Create’ button will trigger a toast message stating, ‘All fields must be filled.’ In addition, users must ensure that the starting date is smaller than or equal to the ending date; otherwise, selecting an incorrect date will prompt a toast message informing the user of this requirement.

5.5.13 Saving (Money Box Details)

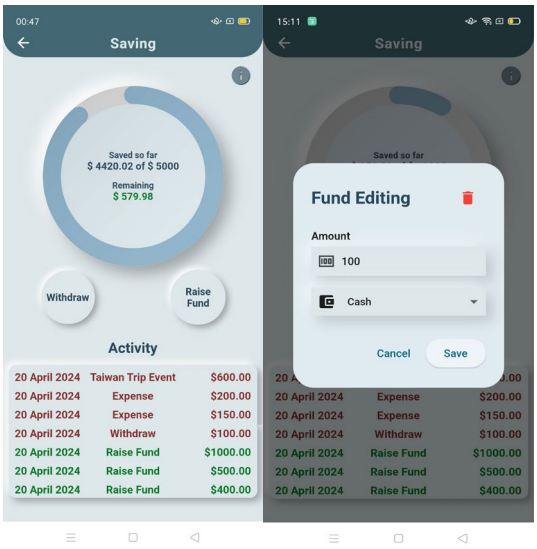


Figure 5.5.13.1 Figure 5.5.13.2

When users tap on their money boxes from Figure 5.5.11.1 and Figure 5.5.11.2, they are directed to the money box detail page to view the selected details, as depicted in Figure 5.5.13.1. Here, a circular progress indicator visually represents the total amount saved in the money box, accompanied by text displaying the saved amount, target total saving amount, and the remaining amount needed to reach the goal. At the bottom of the page, a table displays the funding record activity, including withdraw or raise fund actions, along with expenses transactions utilizing funds from the money box. However, only the withdraw and raise fund activities' records can be tapped to access the fund editing dialog for editing funding details, as shown in Figure 5.5.13.2.

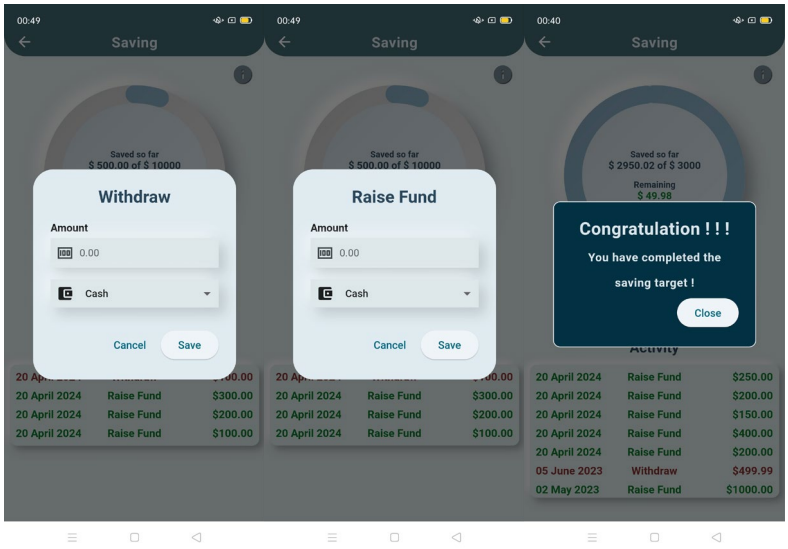


Figure 5.5.13.3 Figure 5.5.13.4 Figure 5.5.13.5

When users tap the ‘Withdraw’ or ‘Raise’ button in Figure 5.5.13.1, they are directed to a dialog for withdrawing funds from the money box or raising funds for it. Users must fill in the amount and select the wallet before saving the transaction. If a user raises funds and successfully reaches the money box's target saving amount, the application will display a congratulatory dialog to acknowledge that the saving target has been achieved, as shown in Figure 5.5.13.5.

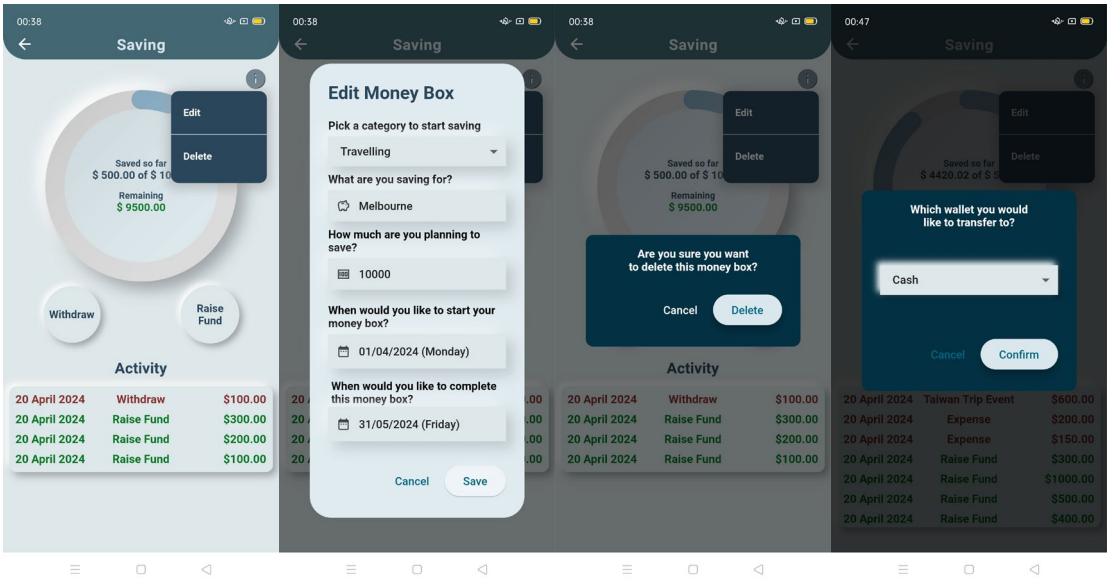


Figure 5.5.13.6 Figure 5.5.13.7 Figure 5.5.13.8 Figure 5.5.13.9

Moreover, users can access an icon button at the top right corner of the page to open a dropdown box, enabling them to select either the edit or delete money box features, as depicted in Figure 5.5.13.6. If users tap the ‘Edit’ button, a ‘Edit Money Box’ dialog will appear, allowing them to modify the money box details and save the changes (Figure 5.5.13.7). On selecting the ‘Delete’ button, the system will prompt users for deletion confirmation (Figure 5.5.13.8). If users confirm deletion, another dialog will appear, requesting users to select a wallet (Figure 5.5.13.9). Users can choose one of the wallets to transfer the amount remaining in the money box to the selected wallet or select ‘Do not transfer’ to delete the money box without transferring the remaining amount.

5.5.14 Profile

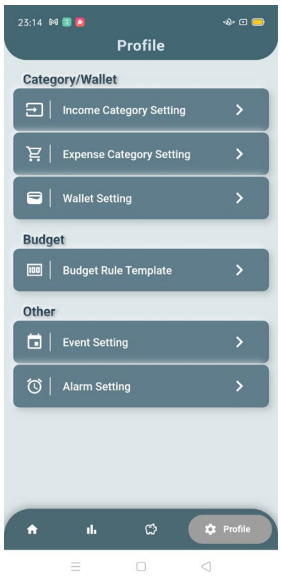
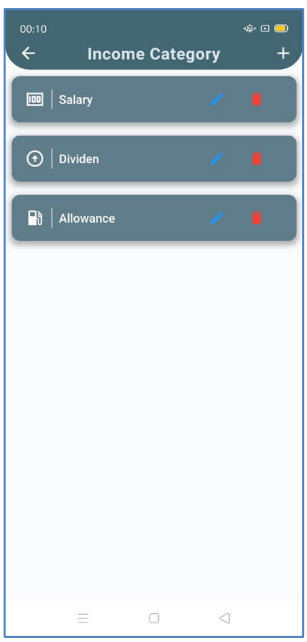
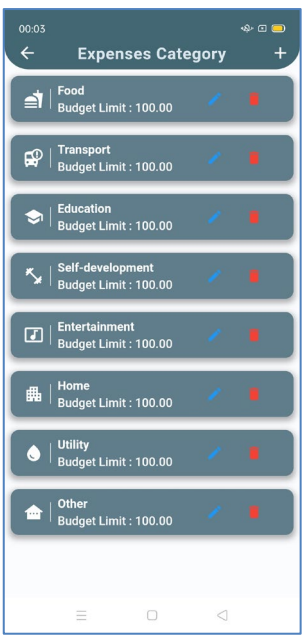


Figure 5.5.14.1

When users navigate to the 'Profile' page from the bottom navigation bar, they will see a list of setting, as show in Figure 5.5.14. Users can access each setting to configure their preferences accordingly. These setting include income category setting, expense category setting, wallet setting, budget rule template, event setting, and alarm setting.





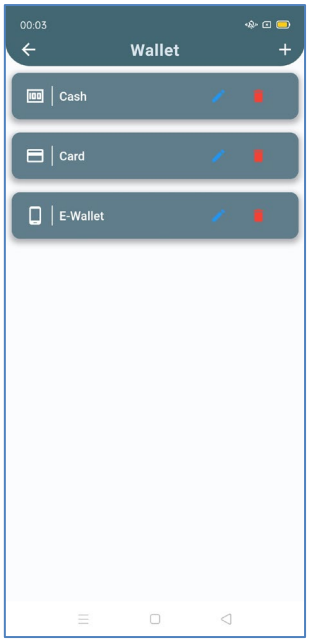


Figure 5.5.14.2 Figure 5.5.14.3 Figure 5.5.14.4

Figure 5.5.14.2 displays the 'Income Category' page, accessed from 'Income Category Setting' in Figure 5.5.14.1. Similarly, Figure 5.5.14.3 and Figure 5.5.14.4 depict pages for 'Expenses Category' and 'Wallet', respectively, accessed from Figure 5.5.14.1. Each page are displaying a list of category or wallet those are created.

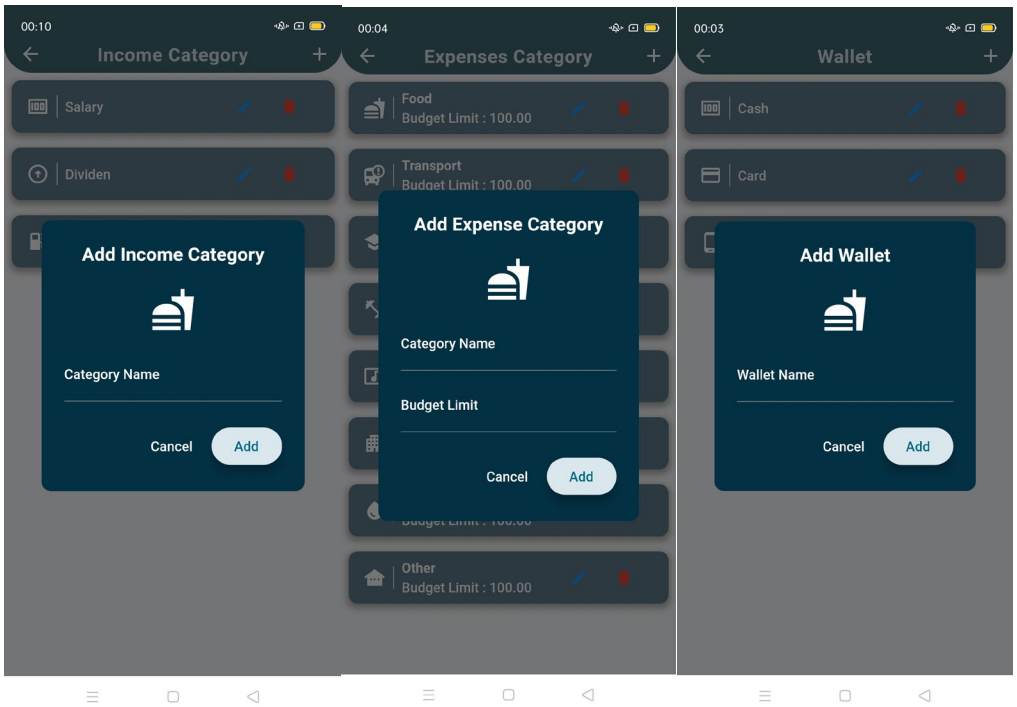


Figure 5.5.14.5 Figure 5.5.14.6 Figure 5.5.14.7

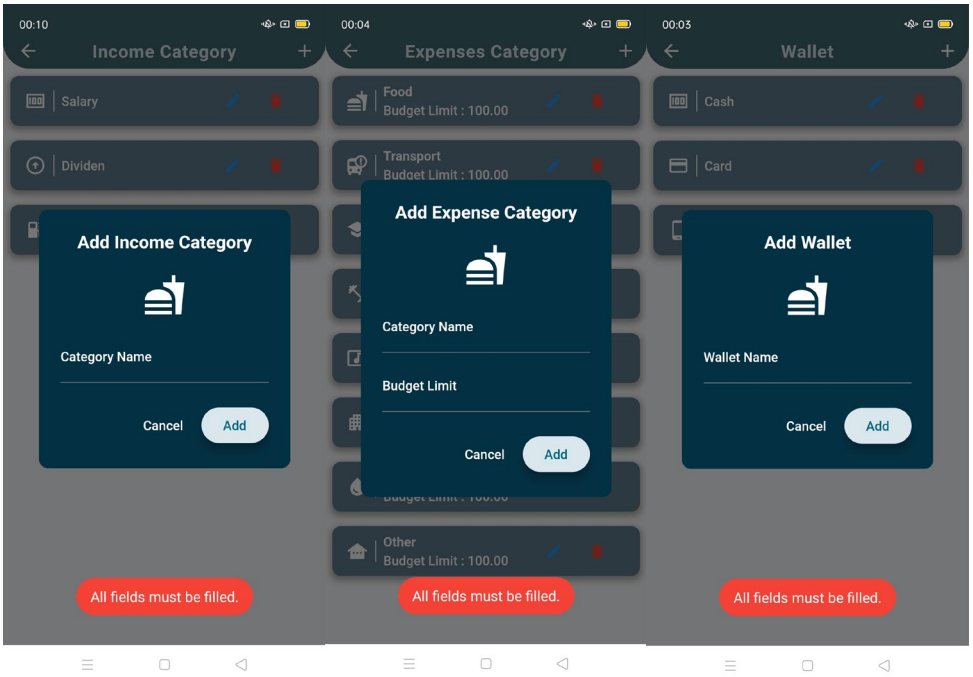


Figure 5.5.14.8 Figure 5.5.14.9 Figure 5.5.14.10

Users can access the add icon button on the right side of the app bar, triggering a dialog for creating a new category or new wallet, as depicted in Figure 5.5.14.5, Figure 5.5.14.6, and Figure 5.5.14.7. However, creating an expense category differs from the other two, as it requires users to enter a budget limit amount for the new category. This feature aims to assist users in managing their expenses and monitoring their spending each month to ensure they stay within their set limits. Additionally, all fields must be filled out completely; otherwise, a toast message will prompt users with the message 'All fields must be filled.', as shown in Figure 5.5.14.8, Figure 5.5.14.9, and Figure 5.5.14.10.

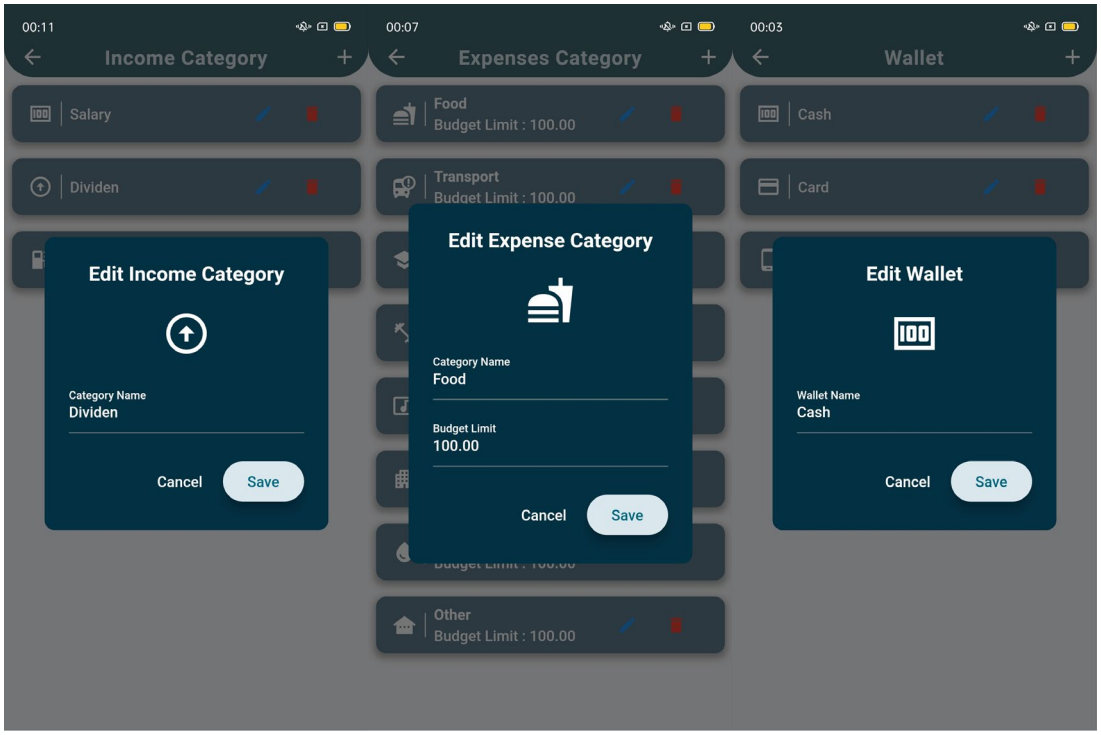


Figure 5.5.14.11

Figure 5.5.14.12

Figure 5.5.14.13

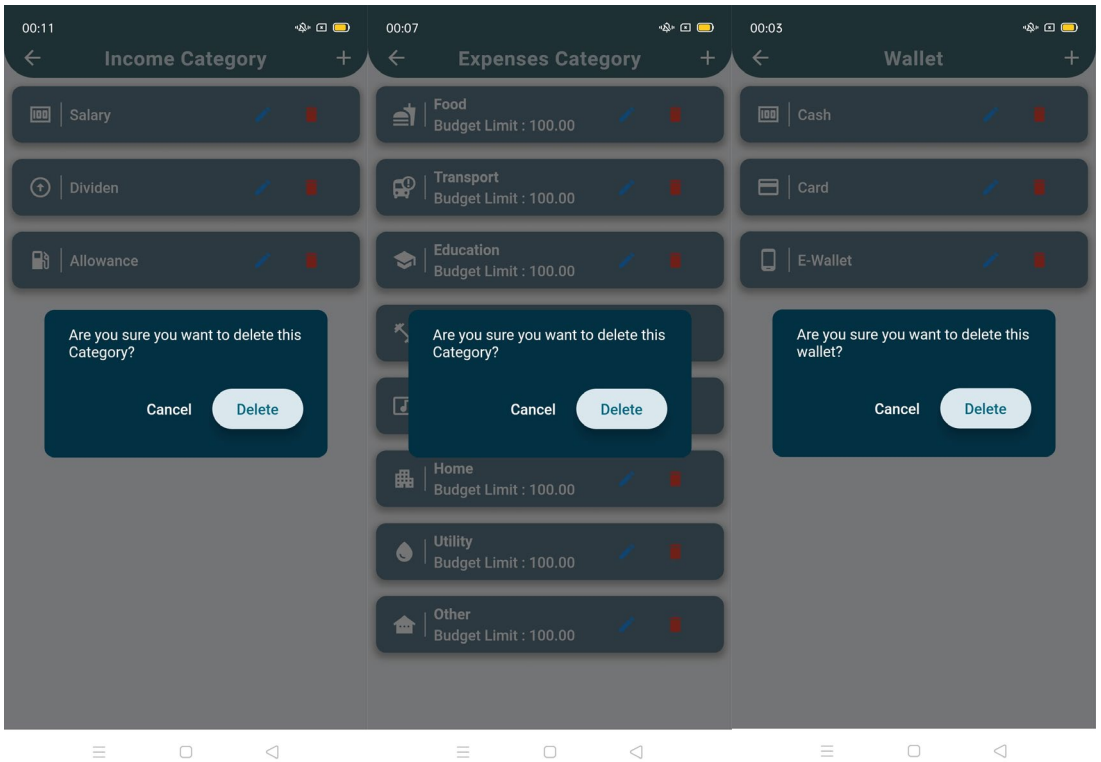


Figure 5.5.14.14

Figure 5.5.14.15

Figure 5.5.14.16

Furthermore, each list item is equipped with both an edit and delete icon, allowing users to modify the items as needed. Tapping the edit button prompts a dialog to appear,

enabling users to edit the details of the selected item (Figure 5.5.14.11, Figure 5.5.14.12, and Figure 5.5.14.13). Alternatively, tapping the delete button triggers another dialog for delete confirmation, ensuring users confirm their intention before proceeding with the delete action (Figure 5.5.14.14, Figure 5.5.14.15, and Figure 5.5.14.16).

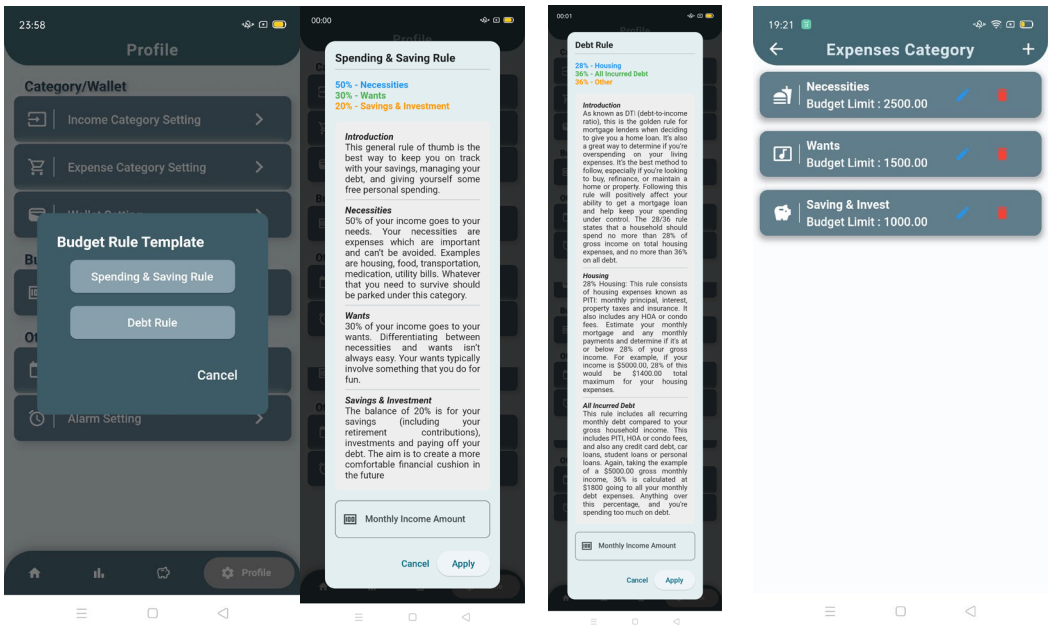


Figure 5.5.14.17 Figure 5.5.14.18 Figure 5.5.14.19 Figure 5.5.14.20

When accessing 'Income Category Setting' in Figure 5.5.14.1, a dialog titled ‘Budget Rule Template’ prompts users with two options: 'Spending & Saving Rule' and 'Debt Rule', as shown in Figure 5.5.14.17. These options represent different kinds of budget rule templates, allowing users to select them to view the details of each rule, depicted in Figure 5.5.14.18 and Figure 5.5.14.19, respectively. Viewing the rule details provides users with an understanding of the budget allocation for each expense category. Users are required to enter their monthly income if they wish to apply the budget rule. The system utilizes the monthly income to calculate the budget limit amount for each expense category. For example, applying the ‘Spending & Saving Rule’ with a monthly income of $5000, the system creates three expense categories: ‘Necessities’, ‘Wants’, and ‘Saving & Invest’, with the appropriate budget limit allocations: $2500 (50%), $1500 (30%), and $1000 (20%), respectively.

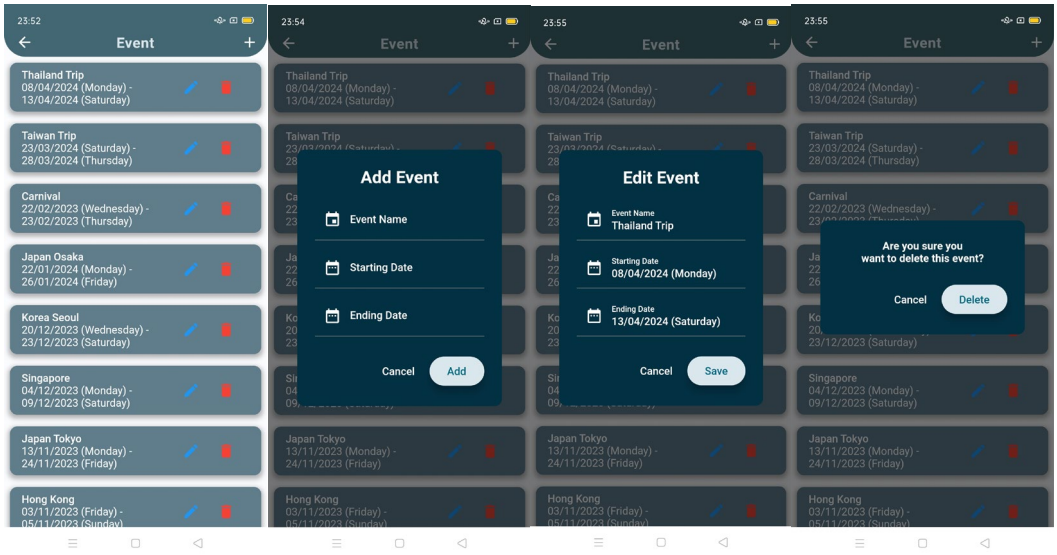


Figure 5.5.14.21 Figure 5.5.14.22 Figure 5.5.14.23 Figure 5.5.14.24

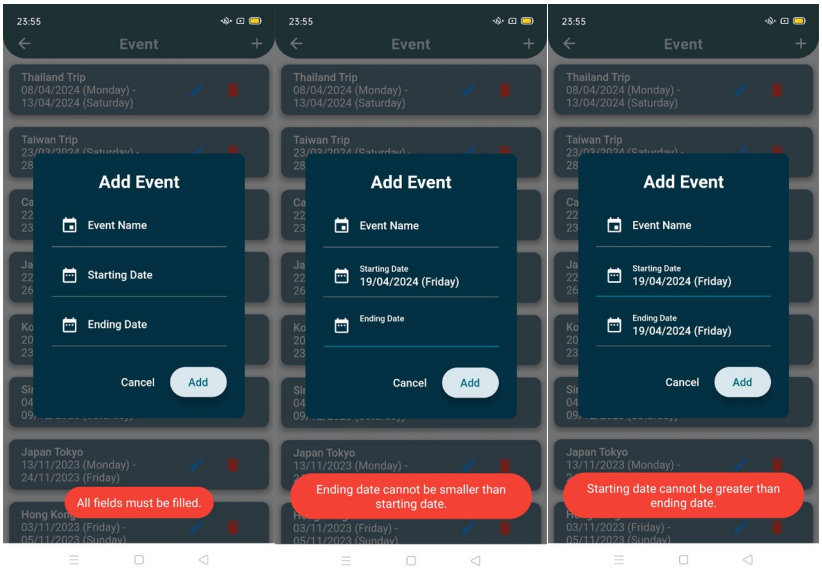


Figure 5.5.14.25 Figure 5.5.14.26 Figure 5.5.14.27

When accessing the ‘Event Setting’ in Figure 5.5.14.1, the application navigates to the ‘Event’ page, displaying a list of created events, as shown in Figure 5.5.14.17. Here, users are also able to access the add icon at the right of the app bar to create a new event (Figure 5.5.14.18) and modify each created event using the edit and delete icon buttons, which open the edit and delete dialogs (Figure 5.5.14.19 and Figure 5.5.14.20). However, users are required to fill out all fields and ensure that the starting date is earlier than the ending date; otherwise, the application will display a toast with an error message to the user, as depicted in Figure 5.5.14.21, Figure 5.5.14.22, and Figure 5.5.14.23.

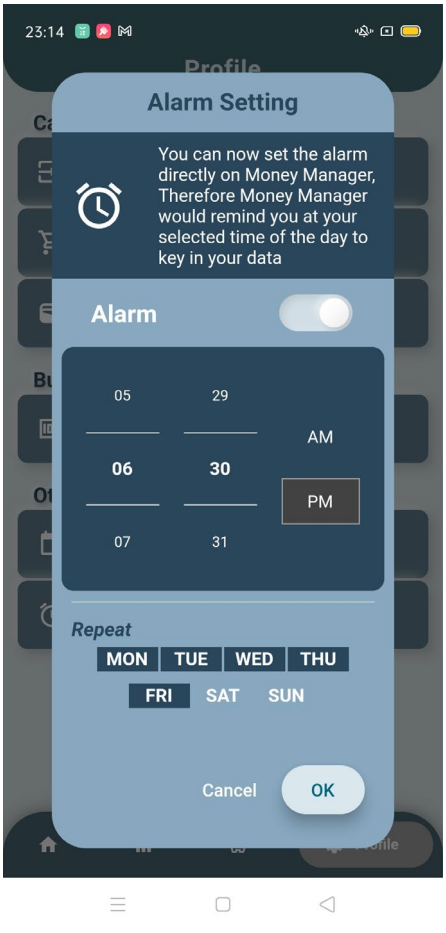


Figure 5.5.14.28

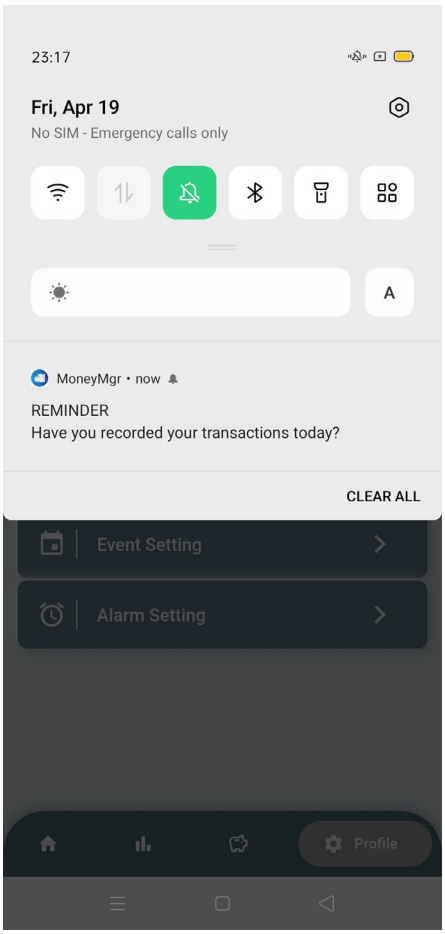


Figure 5.5.14.29

When tapping the ‘Alarm Setting’ option in Figure 5.5.14.1, a ‘Alarm Setting’ dialog is prompted to the user, as shown in Figure 5.5.14.22. Initially, a description about the alarm is provided for users to understand its purpose. Following this, a toggle switch button allows users to enable or disable the alarm notification. If users toggle the switch off, they are required to press the ‘OK’ button to save their action. Conversely, if users toggle the switch on, they must select the time from the time picker and choose the days they wish to receive alarm notifications every week. When the set time and day are reached, a notification will be sent to the user's phone, as depicted in Figure 5.5.14.23.

5.6 Implementation Issues and Challenges

A significant challenge was encountered during the development of the receipt scanning functionality for this project. The integration of OCR (Optical Character Recognition) into the Flutter framework presented a substantial obstacle attributable to compatibility concerns between different versions. The OCR software necessitated a newer version of Flutter than the one presently in use. Upgrade to the most recent version of Flutter, however, carried the risk of leading to conflicts with other critical packages that were have been installed into the project.

In order to guarantee the success of the project without depending on OCR, alternative solutions were investigated. A significant advancement was made following weeks of investigation, when the 'google_mlkit_text_recognition' module was identified. This utility provided the functionality to conduct text recognition without requiring an upgrade to the latest version of Flutter. However, in order to optimise the text recognition functionalities, a supplementary package known as "polygon" was necessary. The purpose of these package is to determine lines of text that pertain together, organise the words in a logical sequence according to their relative positions within the bounding polygons, and process the recognised text from an image of a receipt.

Chapter 6

6.1 System Testing

Black box testing represents a method of software evaluation of a system in which the tester does not understand the inner workings of the software or the complexity of the implementation. Instead, the focus is on validating the functionality against the specifications or requirements provided. The method includes various categories such as functional testing, regression testing, and non-functional testing. The project in question intends to use functional testing within the realm of black box testing. Functional testing is characterized in that its goal is to ensure that every function of the software is consistent with predetermined requirements and specifications, regardless of the source code. It needs to test each function by inputting the appropriate test data, predicting the output, and comparing the actual results with the expected results. Notably, the test protocol includes a review of the user interface, APIs, databases, security protocols, client or server applications, and overall software functionality. Functional testing can be performed manually or automatically as a means of determining whether the system meets the functional requirements of the software.

Functional testing, as a subset of black-box testing, offers a lot of advantages for software evaluation. First, it proves that implementing tests within the system is efficient, thus helping to simplify the testing process. In addition, performing tests from the perspective of the end user or customer ensures consistency with real-world scenarios, enhancing the relevance and reliability of test results. Finally, functional testing serves as a valuable tool for finding ambiguities and inconsistencies in functional specifications, promoting clarity and accuracy in software development efforts.

6.2 Testing Setup and Result

Tracking Module

Test Plan	Test Case	Expected Result	Actual Result	Result
Empty text field checking	Case 1: Pressing save button without enter any information	Prompt a toast with error message	Prompt a toast with error message	Success
	Case 2: Pressing save button without select wallet	Prompt a toast with error message	Prompt a toast with error message	Success
	Case 3: Pressing save button without enter amount	Prompt a toast with error message	Prompt a toast with error message	Success
	Case 4: Pressing save button without select category	Prompt a toast with error message	Prompt a toast with error message	Success
Select money box as wallet	Select a created money box	Transaction saved the wallet selection as money box	Transaction saved the wallet selection as money box	Success
Spilt category	Split and select more than one category	Selected categories will feature an amount text box for input and displayed	Selected categories and amount text box are displayed	Success
Event Functionality	Case 1: checked the check box of event	Event text box will be enabled to tap	Event text box is enabled to tap	Success
	Case 2: Unchecked the check box of event	Event text box will be disabled to tap	Event text box is disabled to tap	Success
	Case 3: Select a created event	Transaction will be saved with selected event	Transaction saved with selected event	Success
	Case 4: Unchecked the checked box after select an event	Transaction will not save with selected event	Transaction did not save with selected event	Success
Capture image	Capture receipt image with device camera	Receipt image will be displayed	Receipt image displayed	Success

Import image	Import image from device folders	Receipt image will be displayed	Receipt image displayed	Success
Scan receipt	Scan a receipt	Able to retrieve and display every item name, item amount, total tax, and total		Success

Table 6.2.1 Tracking Module

Report Module

Test Plan	**Test Case**	**Expected Result**	**Actual Result**	**Result**
Budget	Toggle budget	Display bar char and table that contain budget data	Display bar char and table that contain budget data	Success
Chart	Case 1: Toggle chart and press overall button	Display bar chart and table that contain overall data	Display bar chart and table that contain overall data	Success
	Case 2: Toggle chart and press incomes button	Display pie chart and table that contain incomes data	Display pie chart and table that contain incomes data	Success
	Case 3: Toggle chart and press expenses button	Display pie chart and table that contain incomes data	Display pie chart and table that contain incomes data	Success
Flitter period	Case 1: change to annually	Date change to be yearly	Date changed to be yearly	Success
	Case 2: change to annually date and switch the year to previous or next	Date able to change to previous and next year; chart and table also switch to next and previous year data	Date able to change to previous and next year; chart and table also switch to next and previous year data	Success
	Case 3: change to monthly	Date change to be monthly	Date changed to be monthly	Success
	Case 4: change to monthly and	Date able to change to previous and	Date able to change to previous and next monthly; chart and	Success

	switch the monthly date to previous or next	next monthly; chart and table also switch to next and previous monthly data	table also switch to next and previous monthly data	
	Case 5: change to weekly	Date change to be weekly	Date changed to be weekly	Success
	Case 6: change to weekly and switch the weekly date to previous or next	Date able to change to previous and next weekly; chart and table also switch to next and previous weekly data	Date able to change to previous and next weekly; chart and table also switch to next and previous weekly data	Success
Event	Select an event	Display pie char and table that contain event data	Display pie char and table that contain event data	Success

Table 6.2.2 Report Module

Saving Module

Test Plan	**Test Case**	**Expected Result**	**Actual Result**	**Result**
Create new money box	Case 1: Create without enter any data	Prompt a toast with error message	Prompt a toast with error message	Success
	Case 2: Create without enter money box name	Prompt a toast with error message	Prompt a toast with error message	Success
	Case 3: Create without enter amount	Prompt a toast with error message	Prompt a toast with error message	Success
	Case 4: set ending date smaller than starting date; starting date larger than ending date	Prompt a toast with error message	Prompt a toast with error message	Success
	Case 5: Create a new money box	Displayed a new money box	Displayed a new money box	Success

Money box detail	Press any money	Navigate to view money box details	Navigate to view money box details	Success
Edit money box detail	Edit money box detail	Money box details changed	Money box details changed	Success
Delete money box	Delete money box	Money box deleted	Money box deleted	Success
Withdraw fund	Withdraw fund	Withdraw activity recorded and saved amount deducted	Withdraw activity recorded and saved amount deducted	Success
Raise fund	Raise fund	Raise activity recorded and saved amount increased	Raise activity recorded and saved amount increased	Success
Fund activity	Press the activity record	Only withdraw and raise fund activity can be press	Only withdraw and raise fund activity can be press	Success
Edit fund activity	Edit fund activity	Fund activity edited	Fund activity edited	Success
Delete fund activity	Delete fund activity	Fund activity deleted	Fund activity deleted	Success

Table 6.2.3 Saving Module

Account Module

Test Plan	**Test Case**	**Expected Result**	**Actual Result**	**Result**
Income category setting	Create new income category	Successful created and display new income category in income category list	Created successful and display new wallet in wallet list	
	Create without enter any info	Prompt a toast with error message	Prompt a toast with error message	Success
	Edit income category details	Income category edited	Income category edited	Success
	Delete income category	Income category deleted	Income category deleted	Success

Expense category setting	Create new expense category	Successful created and display new expense category in expense category list	Created successful and display new wallet in wallet list	
	Create without enter any info	Prompt a toast with error message	Prompt a toast with error message	Success
	Edit expense category details	Expense category details edited	Expense category details edited	Success
	Delete expense category	Expense category deleted	Expense category deleted	Success
Wallet setting	Create new wallet	Successful created and display new wallet in wallet list	Created successful and display new wallet in wallet list	
	Create without enter any info	Prompt a toast with error message	Prompt a toast with error message	Success
	Edit wallet details	Wallet details edited	Wallet details edited	Success
	Delete wallet	Wallet deleted	Wallet deleted	Success
Budget rule template	Apply a budget rule template with the monthly income amount	successful apply and new expenses category with correct budget limit amount are created	Apply successful and created new expenses category with correct budget limit amount	Success
Event	Create event	Created successful and display new event in event list	Created successful and display new event in event list	Success
	Create without enter any info	Prompt a toast with error message	Prompt a toast with error message	Success
	Set ending date smaller than starting date; starting date larger than ending date	Prompt a toast with error message	Prompt a toast with error message	Success

	Edit event details	Event details edited	Event details edited	Success
	Delete event	Event deleted	Event deleted	Success
Alarm	Switch on alarm and select time and date	Received an alarm notification from application	Received an alarm notification from application	Success

Table 6.2.4 Account Module

6.3 Objective Evaluation

In Chapter 1, five main objective of the project have been delineated:

1. To provide a budget rule template for the user to set their limit spending of each category.
2. To improve the efficiency of tracking expenses by using a split transaction function.
3. To enhance the precision and effectiveness funds by providing account transfer function.
4. To provide an event option for the user to track expenses about a specific event.
5. To optimize time by developing receipt scanning functionality.

At the end of the project, all of the objectives have been successfully achieved. For instance, there are two of different budget rule templates include in the application and can be apply with enter the monthly income amount. Once applied, there will be created the expense categories with proper budget limit based on the selected budget rule template and entered amount. Besides, the split transactions function was provided in the tracking module, allow users to create different expense categories transactions in a time, to improve the efficiency of tracking expenses.

In addition, account transfer function had been offered for users to transfer their fund from one wallet to another wallet and ensure the precision and effectiveness of funds transfer. Creating a transaction for a specific event allowed with the event option. When creating a transaction, users can select the created event for indicating the new transaction during the event. In order to optimize the time of users, receipt scanning functionality allows users to scan the receipt images that imported or captured and select one or more items to total up the amount and return to the transaction creation page.

Chapter 7

7.1 Conclusion

In conclusion, the development of the Personal Financial Planning mobile app offers a comprehensive solution to address the challenges and limitations that current users face in effectively managing their finances. The goals of the project have been carefully designed to address these issues and provide innovative features that stand out in the field of financial management applications. The proposed split transaction feature provides users with a convenient way to record multiple fees in different categories in a single transaction, simplifying the tracking process and reducing the need for excessive transaction records. In addition, the introduction of the receipt scanning function has greatly increased efficiency by automatically extracting basic information from receipts, eliminating manual calculations and improving overall transaction accuracy. The inclusion of event-specific trading options introduces a new dimension to the application, enabling users to carefully track fees associated with specific occasions. This addition not only enhances the user experience but also encourages more thorough financial management, especially in cases where users want to allocate resources efficiently for specific events. In addition, the provision of budget rule templates enables users to systematically allocate budgets and align their spending with predefined guidelines, regardless of their financial knowledge. With the help of this app, those who are not well-versed in budget planning may improve their financial literacy and make better decisions.

The culmination of these features and capabilities differentiates the proposed application from existing systems by providing a comprehensive tool that not only simplifies financial tracking but also promotes sound financial practices. By addressing identified limitations and introducing new elements such as receipt scanning, split transactions, event tracking, and budget templates, the application aims to enable users to achieve their financial goals while fostering a proactive approach to financial management. Through this project, we hope to provide people with a useful tool for managing their money effectively, easily and comprehensively.

However, the developed app had limitations in accurately retrieving product names from scanned receipts, especially from institutions other than Costco Wholesale. Although the function effectively tracks the quantities, prices, total taxes, and overall costs of items in Costco receipts, it encounters difficulties in accurately distinguishing the names of items in receipts from sources other than Costco. To address this challenge, there propose integrating artificial intelligence (AI) algorithms with optical character recognition (OCR) technology in order to tackle this limitation. The integration will enhance the application's capability to identify the names of items displayed in a variety of receipt formats. By converting image receipts to digital formats, the implementation of OCR technology can not only enhance the precision of text retrieval but also enable the smooth incorporation of artificial intelligence algorithms. OCR, which transforms image receipts into text, effectively mitigates errors and inconsistencies, consequently providing a dependable foundation for item name recognition produced by artificial intelligence.

7.2 Recommendation

For future works, there is a recommendation involves implementing cross-device data synchronization functionality within the application. This feature will allow users to create accounts and access to their data across multiple devices. This can be achieved by establishing an online database such as Firebase and connecting the local database of each device with the online database to achieve data synchronization. Once a user logs into their account on a new device, they should be able to retrieve their data from the online database, mirroring the information available on their previous device.

REFERENCES

[1] "Financial planning," *JamaPunji*. [Online]. Available: https://jamapunji.pk/financial-planning/planning-process. [Accessed: 13-Nov-2022].

[2] S. Jones, "The top 6 benefits of Financial Planning," *Savology*, 10-Dec-2020. [Online]. Available: https://savology.com/6-benefits-of-financial-planning. [Accessed: 13-Nov-2022].

[3] "What are the benefits of financial planning?," *Canara*. [Online]. Available: https://www.canarahsbclife.com/blog/financial-planning/what-are-the-benefits-of-financial-planning.html. [Accessed: 13-Nov-2022].

[4] Maybank, "Mae by maybank2u," *App Store*, 07-Oct-2020. [Online]. Available: https://apps.apple.com/us/app/mae-by-maybank2u/id1481028763. [Accessed: 20-Nov-2022].

[5] A. C. P. Yin, "Maybank launches New Mae by maybank2u app, offers new features for Better Financial Management," *RinggitPlus*, 09-Aug-2021. [Online]. Available: https://ringgitplus.com/en/blog/e-wallet/maybank-launches-new-mae-by-maybank2u-app-offers-new-features-for-better-financial-management.html. [Accessed: 20-Nov-2022].

[6] T. S. Online, "Maybank sama-sama Lokal Bringing Small Business Online with 0% commission platform," *The Star*, 22-Jun-2021. [Online]. Available: https://www.thestar.com.my/starpicks/2021/06/21/maybank-sama-sama-lokal-bringing-small-business-online-with-0-commission-platform. [Accessed: 20-Nov-2022].

[7] F. Lee, "After 5 years of loyalty to the M2U app, I tried mae to see if it was time to say goodbye," *Vulcan Post*, 25-May-2021. [Online]. Available: https://vulcanpost.com/731136/maybank-mae-vs-m2u-app-comparison-review/. [Accessed: 22-Nov-2022].

[8] I. Expensify, "Expensify: Receipts & Expenses," *App Store*, 22-Oct-2011. [Online]. Available: https://apps.apple.com/us/app/expensify-receipts-expenses/id471713959. [Accessed: 22-Nov-2022].

[9] "Expensify – expense reports – apps on Google Play," *Google*. [Online]. Available: https://play.google.com/store/apps/details?id=org.me.mobiexpensifyg&hl=en&gl=US. [Accessed: 23-Nov-2022].

[10] R. ApS, "Monefy: Money tracker," *App Store*, 17-Apr-2017. [Online]. Available: https://apps.apple.com/us/app/monefy-money-manager/id1212024409. [Accessed: 23-Nov-2022].

[11] "Monefy – Budget & Expenses App – apps on Google Play," *Google*. [Online]. Available: https://play.google.com/store/apps/details?id=com.monefy.app.lite&hl=en_US&gl=US. [Accessed: 23-Nov-2022].

[12] "Money manager expense & budget – apps on Google Play," *Google*. [Online]. Available: https://play.google.com/store/apps/details?id=com.realbyteapps.moneymanagerfree&hl=en_US&gl=US. [Accessed: 23-Nov-2022].

[13] R. Inc., "Money manager expense & budget," *App Store*, 05-Nov-2012. [Online]. Available: https://apps.apple.com/us/app/money-manager-expense-budget/id560481810?see-all=reviews. [Accessed: 23-Nov-2022].

[14] L. Finsify Technology Co., "How to be a financially successful person?: Money lover," *Moneylover*. [Online]. Available: https://moneylover.me/. [Accessed: 23-Nov-2022].

[15] Greamer, "Monny," *App Store*, 05-Jun-2013. [Online]. Available: https://apps.apple.com/my/app/monny/id590327036. [Accessed: 28-Nov-2022].

[16] D. Technologies, "Goodbudget budget planner," *App Store*, 26-Oct-2011. [Online]. Available: https://apps.apple.com/my/app/95oodbudget-budget-planner/id471112395. [Accessed: 28-Nov-2022].

[17] "50/30/20 rule," – *Raiz Malaysia*. [Online]. Available: https://raiz.com.my/blog/503020-rule/. [Accessed: 28-Nov-2022].

[18] Spendee, "3 budgeting rules you need in your life," *Medium*, 13-Sep-2018. [Online]. Available: https://medium.com/spendee/3-budgeting-rules-you-need-in-your-life-a374abdd1ae. [Accessed: 28-Nov-2022].

APPENDIX
FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: T3Y3	**Study week no.: 3**
Student Name & ID: KUAK CHUN PIN, 20ACB02234	
Supervisor: DR CHAI MEEI TYNG	
Project Title: PERSONAL FINANCIAL PLANNING MOBILE APPLICATION	

1. WORK DONE

- Completed the function in the account module including income category setting, expense category setting, wallet setting, budgeting rule template, and alarm feature.

2. WORK TO BE DONE

- Improve the user interface design of account module.
- Starting to develop the saving module.

3. PROBLEMS ENCOUNTERED

- No idea on design the user interface to be more attractive.

4. SELF EVALUATION OF THE PROGRESS

- Moderate of the progress performance.

tyng
Supervisor's signature

ChunPin
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: 6**
Student Name & ID: KUAK CHUN PIN, 20ACB02234	
Supervisor: DR CHAI MEEI TYNG	
Project Title: PERSONAL FINANCIAL PLANNING MOBILE APPLICATION	

1. WORK DONE

- Found a Flutter package for deigning user interface with neumorphic style.
- Completed the saving module and tested all function of this module.

2. WORK TO BE DONE

- Starting to develop the report module.

3. PROBLEMS ENCOUNTERED

- When applying new Flutter package, happened the Flutter version conflict with all the packages that applied.

4. SELF EVALUATION OF THE PROGRESS

The progress is not in the expectation progress due to various issues.

tyng
Supervisor's signature

ChunPin
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: 9**
Student Name & ID: KUAK CHUN PIN, 20ACB02234	
Supervisor: DR CHAI MEEI TYNG	
Project Title: PERSONAL FINANCIAL PLANNING MOBILE APPLICATION	

1. WORK DONE

- Completed the development of report module.
- Solved the Flutter version conflict issues.

2. WORK TO BE DONE

- Conduct testing of all system functions.
- Start the report writing.

3. PROBLEMS ENCOUNTERED

- Encountering issues with displaying bar chart and pie charts while developing the report module.

4. SELF EVALUATION OF THE PROGRESS

- The development progress is under the expectation progress.

tyng
Supervisor's signature

ChunPin
Student's signature

POSTER


FACULTY OF INFORMATION COMMUNICATION AND TECHNOLOGY
MONEY MGR
Personal Finance Planning Mobile Application
02:24
Report
Event
Budget
Chart
Y
2024
Overall
Incomes
Expenses
Summary
Other 12.43%
Utility 7.81%
Food 26.82%
Home 5.09%
Entertainment 7.09%
Education 15.21%
Transport 20.92%
Food $ 6383.00
Education $ 3621.00
Transport $ 4979.00
Report
RAD METHODOLOGY
• REPORT MODULE
• TRACKING MODULE
• SAVING MODULE
• ACCOUNT MODULE
ROBUST FINANCIAL MANAGEMENT TOOLS
Project Developer: Kuak Chun Pin
Project Supervisor: Dr. Chai Meei Tyng

PLAGIARISM CHECK RESULT

PERSONAL FINANCIAL PLANNING APPLICATION

ORIGINALITY REPORT

9%	7%	0%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	19january2021snapshot.epa.gov Internet Source	1%
3	www.comparehero.my Internet Source	1%
4	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
5	blog.back4app.com Internet Source	<1%
6	Submitted to Tower Hamlets College Student Paper	<1%
7	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
8	www.appwereld.nl Internet Source	<1%
9	personal.ph.surrey.ac.uk Internet Source	<1%

PLAGIARISM CHECK RESULT

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	KUAK CHUN PIN
ID Number(s)	20ACB02234
Programme / Course	BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING
Title of Final Year Project	PERSONAL FINANCIAL PLANNING MOBILE APPALICATION

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 9 **%** **Similarity by source** Internet Sources: 7 % Publications: 0 % Student Papers: 4 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

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

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

tyng
Signature of Supervisor

Name: Chai Meei Tyng

Date: 24/04/2024

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB02234
Student Name	KUAK CHUN PIN
Supervisor Name	DR CHAI MEEI TYNG

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.

ChunPin

(Signature of Student)
Date: 18 April 2024