PERSONAL FINANCIAL MANAGEMENT MOBILE APPLICATION:

SMART POCKET

KHEW YI XUN

Diploma in Computer Science

UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	KHEW YI XUN
Date of Birth	:	28 MARCH 2006
Title	:	PERSONAL FINANCIAL MANAGEMENT MOBILE APPLICATION: SMART POCKET
Academic Session	:	SEMESTER I ACADEMIC SESSION 2025/2026

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☒	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang Al-Sultan Abdullah reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang Al-Sultan Abdullah
2. The Library of Universiti Malaysia Pahang Al-Sultan Abdullah has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

_____	_____
(Student's Signature)	(Supervisor's Signature)
_____	_____
060328010419	Ts. Dr. Kirahman Bin Ab. Razak
Date:	Date:

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Universiti Malaysia Pahang Al-Sultan Abdullah,
Lebuh Persiaran Tun Khalil Yaakob,
26300, Gambang, Kuantan, Pahang.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

_____
(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor and addressed to the Librarian, *Universiti Malaysia Pahang Al-Sultan Abdullah* with its copy attached to the thesis.

اونيۏرسيتي مليسيا ڤهڠ السلطان عبد الله
UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Science.

_____

(Supervisor's Signature)

Full Name :

Position :

Date :

_____

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang Al-Sultan Abdullah or any other institutions.

_____

(Student's Signature)

Full Name : Khew Yi Xun

ID Number : RC24069

Date : 14 November 2025

PERSONAL FINANCE MANAGEMENT MOBILE APPLICATION: SMART POCKET

KHEW YI XUN

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Computer Science

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

OCTOBER 2025

ACKNOWLEDGEMENTS

In this thesis and project development, I have received a great deal of support from my family, friends, and lectures.

First and foremost, I would like to show my appreciation to my supervisor, Ts. Dr. Kirahman Bin AB. Razak, who gave me an invaluable guidance and feedback to me for developing and completing this project. His encouragement and insightful feedback bought my thesis and project to higher level.

Also, I would like to extend my special thanks wo our Undergraduate Project coordinator, Ts. Dr. Zalili Binti Musa, who explained and tolerated us in the whole thesis writing process and the project development process.

Last but not least, I would be remiss if I am not mentioning my parents and friends specifically for their collaboration and encouragement in this journey. Their ideas and suggestion at every stage of competition is the key of success. Without them, this experience will not be rewarding and memorable.

ABSTRAK

Pengurusan Kewangan Peribadi dalam kalangan golongan belia sering diabaikan kerana kekurangan literasi kewangan dan peningkatan penggunaan kaedah pembayaran tanpa tunai yang mendorong kepada perbelanjaan impulsive dalam kehidupan harian pada masa ini. Aplikasi yang sedia ada di pasaran gagal menangai isu tersebut secara holistik. Contohnya, aplikasi penjejakan manual menyebabkan "jurang ketegaran" yang menyebabkan kelesuan penggua untuk menggunakannya, manakala kaedah pembayaran tanpa tunai mengakibatkan "jurang tingkah laku" di mana pengguna menjadi pasif terhadap tabiat perbelanjaan mereka. Jadi, aplikasi Smart Pocket dicadangkan dengan menggunakan rangak kerja Flutter dan Firebase. Smart Pocket juga bertujuan untuk membangunkan mekanisme kemasukan data dengan pelbagai cara dan bergabung dengan konsep "mod satu sentuhan" untuk penjejakan perbelanjaan pantas dan mod terperinci untuk ketepatan laporan bulanan, serta mengintergrasikan konsep pendidikan kewangan berdasarkan panduan peruntukan aset. Selain itu, metodologi pembangunan yang digunakna adalah agile SDLC untuk mematikan fleksibiliti yang lebih tinggi. Dalam projek tersebut, hasil kerja yang dijangkakan daripada Smart Pocket adalah untuk mengurangkan geseran dalam proses merekod perbelanjaan dan peningkatan kesedaran kewangan dalam kalangan golongan belia.

ABSTRACT

Personal Financial Management (PFM) among young adults is often neglected due to lack of financial literacy and the rise of cashless payment methods, which causes impulsive spending in their daily life. Those existing PFM mobile app in the market fail to solve this issue holistically as manual tracking app cause rigidity gap leading users lazy to use it, while cashless payment methods create a behavioral gap where users become passive regarding their spending habits. This FYP will proposes the development of Smart Pocket, a mobile app developed using Flutter framework and Firebase. Smart Pocket aims to develop multi modal data entry mechanism and combine with one tap mode for rapid expense tracking and detailed mode for accuracy month report, as well as integrating financial education functions based on asset allocation guide. The development methodology used is agile SDLC to ensure more flexibility. The expected result of Smart Pocket is to reduce friction in the expense recording and enhance financial awareness.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

PFM	Personal Financial Management
FYP	Final Year Project
IOS	iPhone Operating System
APP	Application
UI	User Interface
UX	User Experience
GUI	Graphical User Interface
TNG	Touch' n Go
QR	Quick Response
RFID	Radio-Frequency Identification
FPX	Financial Process Exchange
ASNB	Amanah Saham National Berhad
SDK	Software development Kit
SQL	Structured Query Language
SDLC	Software Development Life Cycles

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

In the contemporary digital economy, financial literacy has emerged as an increasingly critical life skill for young adults, who are navigating an economic landscape that is becoming ever more intricate. This phenomenon is primarily attributed to the proliferation of multiple trending online payment methods, making the payment too convenience but simple such as buy-now-pay-later (BNPL) services and digital wallets. Consequently, these kinds of services lead to impulsive spending among young adults and impacting their spending habits negatively (Ajouz et al. 2024).

Figure 1.1 Most Popular Online Payment Methods

Hence, Personal Financial Management (PFM) mobile apps were developed to bridge this gap by offering tools for budgeting, expense tracking, and financial

planning (Cederberg, J. 2013). The goals of these apps normally aim to empower their respective target users by providing accessible data on their spending habits. However, the “one-size-fits-all” philosophy of most existing PFM mobile apps presents a crucial barrier to user adoption and long-term engagement. According to the research (El Zarif et al., 2020), this leads to a high user abandonment rate (churn), used to show users changes from an app to the one offered by the strong competition. This is due to there are various types of PFM mobile apps with similar features exists in the market.

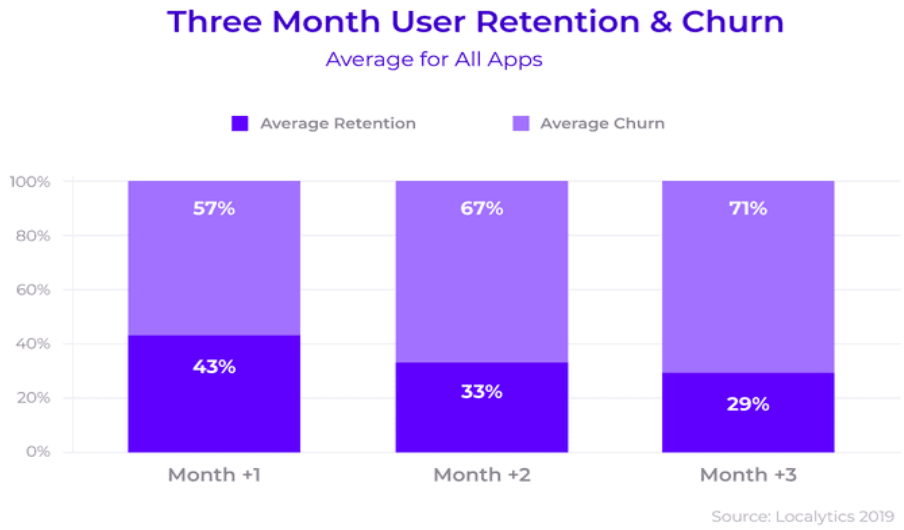

Figure 1.2 Average User Churn in Mobile App

Furthermore, most of the existing PFM mobile apps operate as passive financial recording system which is only like a calculator. This would miss the golden opportunity to train them a good financial management habit in this era of digital economy as not all individuals have perfect financial knowledge especially those young adults. Wang et al. (2025) stated that a significant portion of young adults are facing considerable challenges in financial literacy, with key issues being an inability to set their financial goals and they found difficult to translate theoretical knowledge into practical effectively. For example, the 50/30/20 budget rule offers a simple and balanced framework that helps individuals manage their finances

effectively by covering essential needs, allowing for discretionary spending, and consistently building savings for both emergencies and long-term goals (Whiteside, 2024).

This proposal is structured as follows: Chapter 1 provides an overview of the development background and objectives. Chapter 2 reviews the relevant literature on existing PFM apps. Chapter 3 outlines the methodology used in this study. Chapter 4 presents the finding and analysis. Finally, Chapter 5 discuss the implication of the results and offer recommendations for future research

Therefore, the thesis statement of this project will provide a better solution by integrating flexible tracking modes with passive financial educations, creating a multi modal PFM mobile application that adapts to the user to decrease the churn rate.

1.2 PROBLEM STATEMENT

In recent years, the decline in students' spending habits is one of the rises. According to the research (Andriani and Nugraha, 2021), good spending habits is vital as a bad good spending habits will be creating consumptive behaviour in financial management. Many peoples are spending their money without any recording as they normally focus on "how much they still can spend" instead of "how much saving they still have". Consequently, this kind of bad habit generate a high number of unnecessary losses of their property. The motivation of Smart Pocket and other existing PFM mobile app are developed to address this problem. If the problem is solved, personal saving will be increased when the real personal consumption expenditure is being controlled (Richter, 2025).

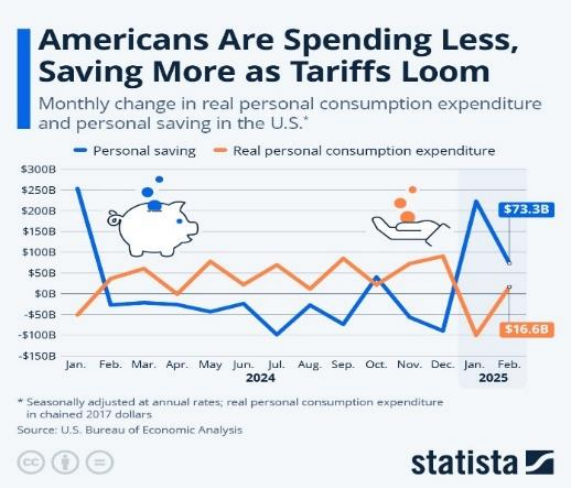

Figure 1.3 Monthly change in real consumer spending and personal saving

The second problem is daily mobile spending contexts such as cafeterias, transit or micro-purchase lack of portable, immediate way to record their expenses within manual recording in phone or immature PFM mobile app. The problem is these methods rely on delayed, manual, multi-step entry or the data analysis of bank statement. Therefore, the high capture friction of the moment of spending cause incomplete data and breaks the personal informatics feedback loop (Li et al., 2010). In the long term, when the burden of record manually remains high, users will eventually stop to record, preventing the formation of financial habits. The reason for this is that the incomplete and delayed transaction capture affect the accuracy of budget tracking, anomaly detection or subsequent goal are reducing perceived usefulness indirectly (Eysenbach, 2005). Hence, Smart Pocket aims to reduce capture fiction by minimising number of input fields and others features to develop an app that more suitable to users.

The third problem is there are many aggregated transaction histories, subscription fee and the most importantly, small high-frequency leak expense often difficult to recognize and remember; concurrently, long term financial behaviors which include building emergency funds and systematic saving are lacked of efficient tracking and reinforcing prompts. A contributing factor is the user's recording method present raw categorical totals without detailed and targeted analytics to surface the anomalous and repeat patterns. According to Weinmann et al. (2016),

the structured progress visualization and behavioural nudges known to support sustained savings actions but the user's action normally not involve this progress. Thus, Smart Pocket is here to enhance user engagement by implementing features such as identify repeat subscription fee, frequent small purchase and others that will be discussed more detailed in the upcoming topic.

1.3 OBJECTIVE

There are 3 objectives are defined in proposed project, Smart Pocket which include:

I. To design and develop a multi-modal PFM mobile app that minimizes data entry fields and incorporates behavioural nudges.
II. To investigate the factors that lead to high capture friction in manual expense recording and the behavioural barriers that avoid users from maintaining consistent financial tracking.
III. To validate and access the usability of Smart Pocket and its effectiveness in improving user engagement and data completeness compared to traditional manual recording and tracking ways.

1.4 SCOPE

From a scope-definition perspective for this FYP, the study will more focus on the context of Malaysian young adults (university students and young professionals) as the primary target users, and the mobile app design and financial education concepts will be tailored to this demographic; the study will cover a period of eight months, from October 2025 to June 2026; Smart Pocket is strictly focused on mobile app development, with the frontend utilizing the Flutter framework to develop Android base application and the backend utilizing Firebase (BaaS), while excluded technologies are native IOS (Swift) or Android (Kotlin) development, and no web-based platform will be built; Smart Pocket will rely on user manually entered data, and excluded data sources are automated data

scraping, API integration with banks, or the use of third-party financial datasets; the functional scope is designed to directly achieve the objectives. The features include recurring expenses, rollover budgeting, and a calendar view, and it will also implement another mode with guided asset allocation like emergency funding and separated reporting (Spending vs Investment Growth), while the project will include core functions such as secure user login, seamless mode switching, and report generation module, and a usability testing plan (detailed in Chapter 3) will be executed. Then, the limitations include single-currency operation where the project will not support multi-currency tracking due to high complexity, no home screen widget as their high complexity that involve native-code integration, without automated bank integration for mitigating security and complexity risks, and without any suggestion or recommendations of financial advice as the project will not include machine learning based financial advice.

1.5 THESIS ORGANIZAITION

This thesis will be divided into 5 Chapters. Chapter 1 will describe and introduce about this project where include problem statements, objective, scope and significance of the project.

Chapter 2 will discuss about the literature review that related of Smart Pocket. It describes the existing system thought a comparation analyst list. There are 3 existing system that will be compared which are Money Tracker, TNG e-Wallet and Fortune City.

Chapter 3 details the methodology of the proposed project, Smart Pocket, encompassing the development process which are use case, storyboard, flowchart, hardware and software specification and Gantt chart.

Chapter 4 describe the implementation of Smart Pocket’s result and discussion.

Chapter 5 discuss the conclusion of Smart Pocket based on the objective, limitations of the project and the future works.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter will consist of five (5) sections, which are introduction, comparison of the advantages and disadvantages of the three existing system with Smart Pocket, and conclusion. Section 2.2 show three (3) existing PFM mobile app's GUI while section 2.3 discusses the comparation analysis. Section 2.4 will discuss about security and integrity of the mobile app and section 2.5 will conclude the whole key findings of Chapter 2.

2.2 THREE EXISTING SYSTEM

The three (3) selected existing system are "Money Tracker", "TNG e-Wallet" and "Fortune City". Three of the systems will be explained and analysed in details that include pro and cons.

2.2.1 Money Tracker

Money Tracker is a mobile app that allow users to record their income and expense manually. Money Tracker allow users to choose their expense and income based on category such as shopping, education, pets, investment, salary and other customized category. Money Tracker also allow user to add their regular payment that will be charged every month such as loan payment and subscription fee. The fee will be automatically deducted from the current balance until the end of the payments. Apart from that, Money Tracker provide a clear data analysis regarding to weekly, monthly and yearly expense or income in a pie charge with

percentage. Therefore, users can manage their financial based on the data as they can clearly know which category are expensed more in a week. However, all data needs to enter manually include the balance of last month. Consequently, the analysis of data may be error since the data collection method is not accurate. For example, the profit and loss of investment may be counted as expense or income automatically since all of the expense and income is mix together. A further ramification is user just only know the overall data but still need to provide a large amount of repetitive daily consumption data. Taking a holistic view, Money Tracker present a good UI and UX based on concept of GUI as the layout is simple and the visualization of report is clearly stated. Also, users can record an expense or income less than 10 seconds since the record system is simple but functional. In short, the features of Money Tracker will attract its own target users but the position of market is unclear.

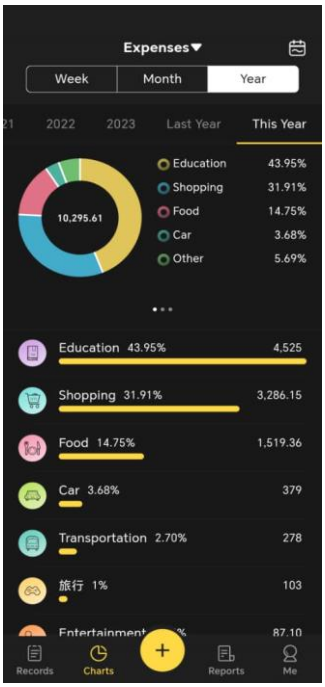

Figure 2.1 Yearly Report of Money Tracker

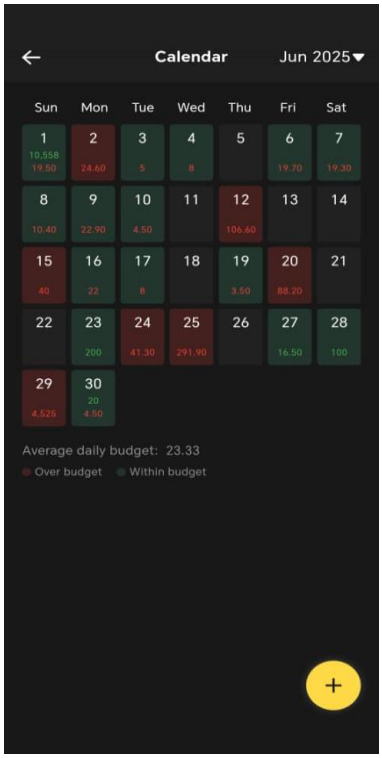

Figure 2.2 Daily Income and Expense

Figure 2.3 Annual Financial Summary of Money Tracker

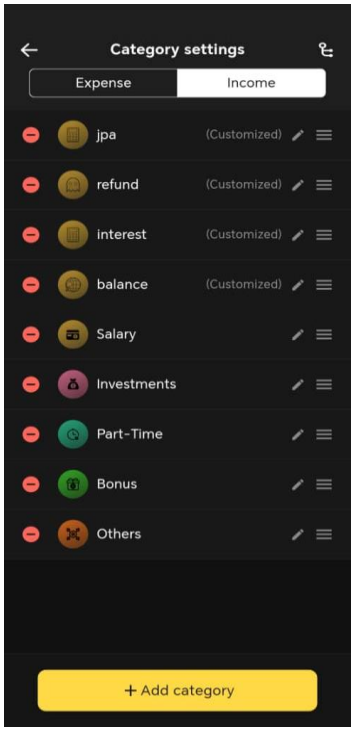

Figure 2.4 Category Settings of Money Tracker

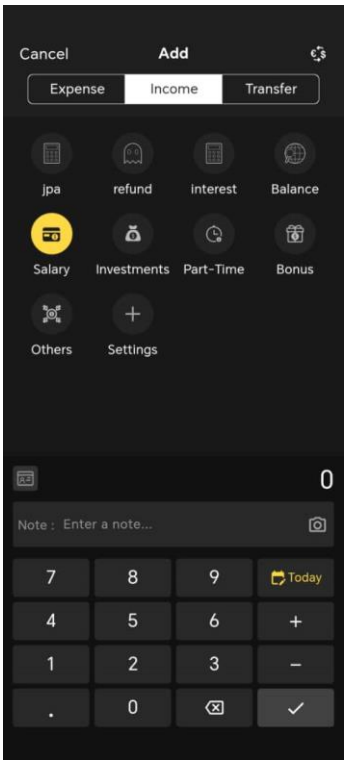

Figure 2.5 Recording Page of Money Tracker

2.2.2 Touch' n Go e-Wallet

TNG e-Wallet is a primary digital payment platform that inadvertently function as an automated financial recorder. Unlike manual trackers such as Money Tracker, TNG will captures transaction details automatically. The transaction details include date, time, merchant name and amount. Whether the payment is made via QR code, RFID or transfer to other account, the transaction details and spending histories in a linear timeline are clearly showed their expense and income. At the same time, TNG e-wallet provide a general view of limited personal investment growth such as ASNB and Principal Unit Trust. This automatic structured capture occurs in a rapidly expanding cashless ecosystem as Malaysia recorded a 24% year on year rise to 11.5 billion e-payment transactions in 2023 while TNG e-Wallet reported over 21 million verified users, and provides 90 days downloadable histories that prevent the problem of manual recording expense and income (Mail, 2024; TNG Digital Reflects on 2023 Performance and Outlines Vision for 2024, n.d.; TNG Digital Sdn Bhd, 2021). In this case, users also not need to perform any post-purchase action to record their expanse and income (transaction from other banks).

From UI/UX perspective, TNG e-Wallet more focused on speed of transaction and payment rather than financial analysis. Nevertheless, TNG e-Wallet only records all the transactions made within the TNG ecosystem, leaving out cash payment or FPX online banking so that users need to achieve a unrealistic precondition which is managing all their property via TNG e-Wallet to unlock a most accurate report that reflect the real expense and income at the respective month. Other than that, expense is often listed by default category such as food or transport. Consequently, TNG e-Wallet function merely as a digital receipt repository but also a potential PFM app if it provides a complete ecosystem that include behavioural insights necessary for users.

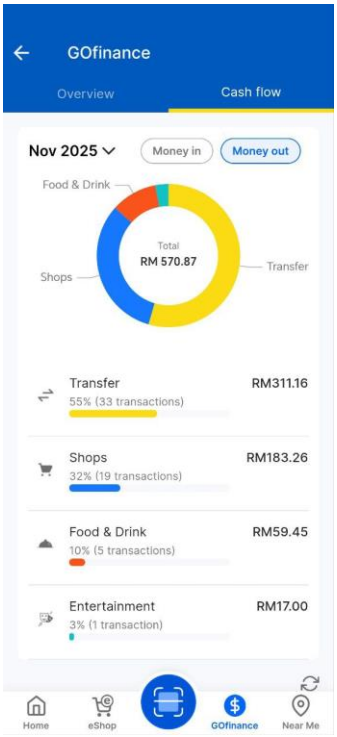

Figure 2.6 Pie Chart of Money out

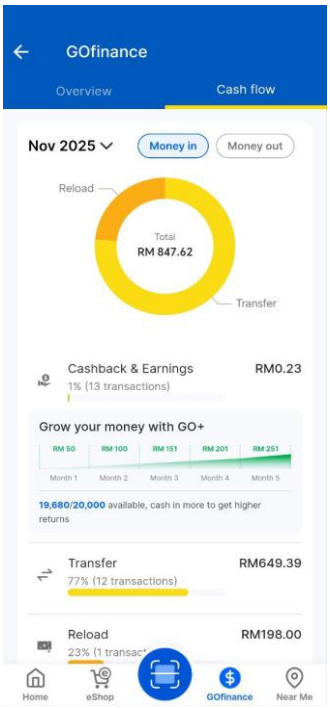

Figure 2.7 Pie Chart of Money in

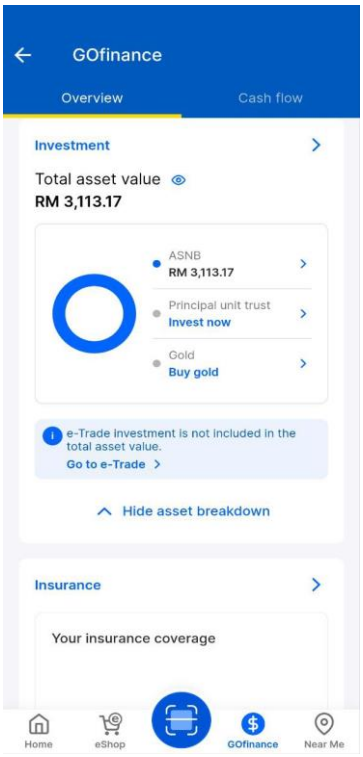

Figure 2.8 Overview of Investment Growth

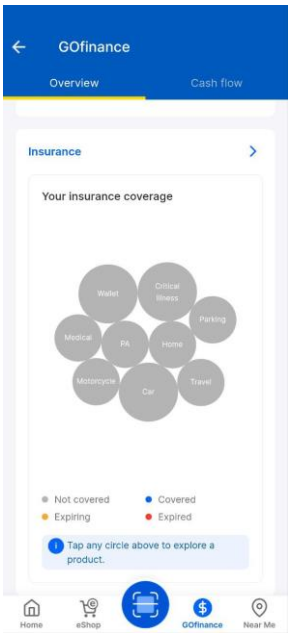

Figure 2.9 Overview of Insurans

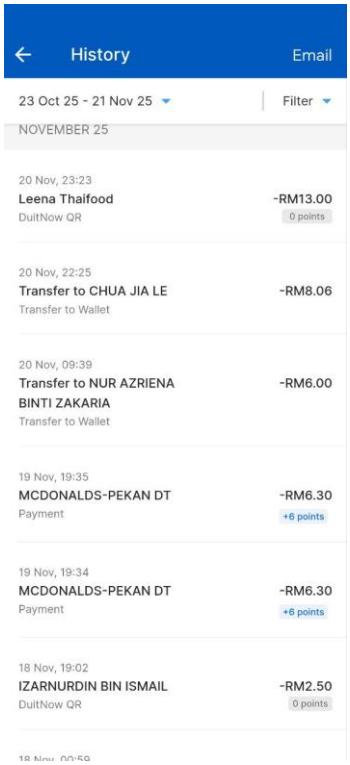

Figure 2.10 Transaction History

2.2.3 Fortune City

Observation and review of existing literature are selected in requirement elicitation of Fortune City. Though observation of my own experience in using Fortune City and reviewing those existing literatures on gamified finance app, Fortune City identified as a PFM and gamified mobile app that combines expense tracking with a simulation of city-building. Its core concept is gamification which involve recording an expense enable the user to construct a building in a virtual city. To illustrate, recording a food expense can build a food stall. This unique PFM ways directly addresses the problem of user's motivation in manual recording. Meanwhile, the UI design is vibrant and playful so that young adults will be attracted easily and start their journey in tracking. In contrast, a playful UI design may reveal a significant flaw in its long-term effectiveness which is the "Gamification Over Finance" paradox. According to the research (Hamari et al., 2014), we can determine that the engagement gains produced by gamified layers

are not guaranteed to persist and it without mechanisms supporting reflection and goal alignment. As a result, the initial appeal declines. Thus, the city-building concept may cause users only log to unlock it instead of embed progress toward real financial goals. In a worse-case scenario, users will prioritize the game aspect instead of financial accuracy since they may record fake transactions in order to upgrade the building or stop recording as they are tired to play a game. Additionally, while Fortune City is encouraging young adults to record in gamification way, it lacks deep and complex financial functionality to educate the young adults.

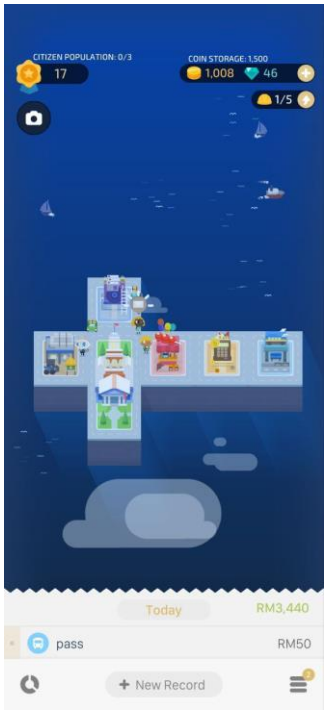

Figure 2.11 Main Page of Fortune City

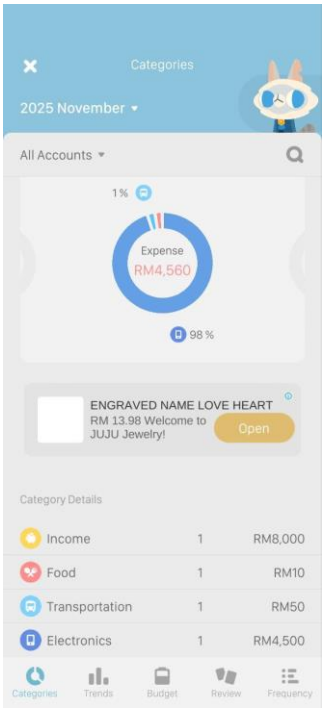

Figure 2.12 Catrgory Lists of Expense and Income

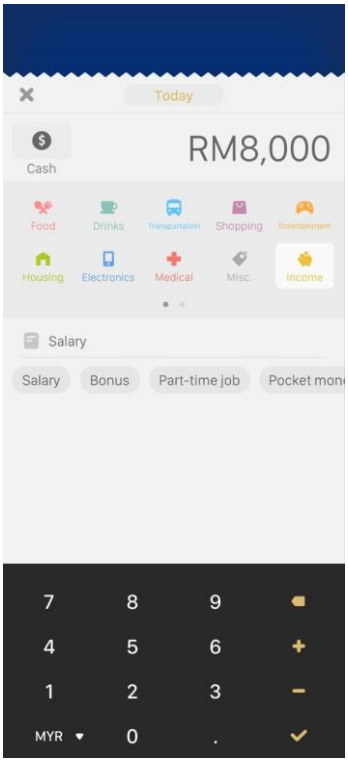

Figure 2.13 Manual Recorder Page

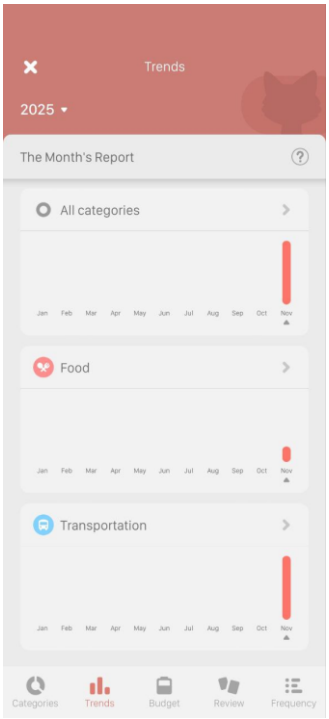

Figure 2.14 Report Trends Page

2.3 REQUIREMENT ELICITAITON

Observation method was selected as requirement elicitation of Money Tracker to evaluate the daily usability via my own using experience. It aims to measure the load required for manual data recording. The key finding identified a critical rigidity gap as the strict requirement to manually input every transaction even thought the UI is clean and reports are generated details. This observation revealed that normal users will be tired after missing just a few days of recording and eventual abandonment.

For TNG e-Wallet, a quantitative survey was conducted among young adults to evaluate the effectiveness of TNG e-Wallet as an automated PFM tool. This survey aimed to validate whether automated PFM tool via transaction history translate to better financial awareness and knowledge. The result (Appendix A) shows a problem of passive data blindness as the majority of respondents rarely review their transaction history due to data fragmentation and various types of payment method. Consequently, the elicitation confirms that TNG e-Wallet

functions as a digital receipt repository and payment method rather than an active PFM tool.

Observation and previous document were utilized to access Fortune City's gamification mechanics. This method aims to determine whether game elements effectively drive long-term financial behavioural change. The findings highlight a concept which is "Gamification Over Finance". Many documents shows that users behavioural will finally be affected by the games although it truly can attract them starting their PFM journey.

2.4 COMPARATION AMONG EXISTING SYSTEM

To provide a structured evaluation, the comparison will be divided into 4 distinct categories with comparison lists which are system profile and context (Table 2.1), operational mechanics (Table 2.2) critical evaluation of advantages and Disadvantages (Table 2.3) and functional features (Table 2.4)

Table 2.1 Comparation of System Profile and Context

Criteria	Money Tracker	TNG e-Wallet	Fortune City	Smart Pocket
Primary Domain	Utility Tool for PFM	FinTech	Gamified Finance	Educational PFM
Target Focus Group	General Users	General Public (Cashless user)	Gen Z	Young Adults
Problem Statement Addressed	Users struggle to organize expensed & income and	Users need a cashless payment method for	Users find traditional expense tracking bored	High churn rate due to the rigidity gap and lack of

	lack visual budget analysis	daily transactions	and lack motivation to sustain the habit	financial literacy in existing apps
Technology Stack	Native Mobile Framework	Enterprise FinTech Stack	Mobile Game Engine	Cross-Platform Framework

Table 2.2 Comparison of Operational Mechanics

Criteria	Money Tracker	TNG e-Wallet	Fortune City	Smart Pocket
Data collection Method	Manual entry for every transaction	Automated capture from transaction record	Gamified Manual Entry for every transaction	Allow manual entry for every transaction or monthly transaction based on balance to minimize user effort
Analysis Technique	Use pie charts and bar graphs to show spending percentages	Present a chronological time line and categorized deep analysis	Visual Metaphor by viewing the prosperity and size of city	Provides asset allocation nudges and use categorized deep analysis

Table 2.3 Critical Evaluation of Advantages and Disadvantaged

Criteria	Advantages	Disadvantages
Money Tracker	Excellent data visualization	Manual entry leads to user fatigue
	Easy to understand ledger logic	Free version relies on advertisement
	Handles budgeting well	No guidance on how to allocate
		No inheritance function of balance
		Hard to manage profit & loss of investment
TNG e-Wallet	100% data accuracy	Misses cash and external bank transactions
	Integrate payment & recording	

	Excellent visualization of spending	Ecosystem-locked (only track spending in TNG)
	Clear distinguish between consumption &investment	
Fortune City	Fun UI	Gamification paradox
	High visual appeal for youth	Lack depth in complex PFM
	Reward the act of recording	Focus turns from finance to game strategy
Smart Pocket	Adapt to any types of users	Still require user input manually
	Teach asset allocation while tracking	No direct API integration with banks
	Allow balance inheritance and fee auto deduction monthly	Single currency support
	Clear distinguish between consumption & investment	
	Excellent visualization of report	

Table 2.4 Comparison of Functional Features

Feature	Money Tracker	TNG e-Wallet	Fortune City	Smart Pocket

Expense Entry Mode	Manual	Automated	Gamified Manual	Manual but minimize effort
Budgeting Tools	Yes	N/A	N/A	Yes
Recurring Payment	Yes	Yes (subscription fee)	N/A	Yes
Visual Analytics	Yes	Yes	Yes	Yes
Behavioural Nudges	N/A	N/A	Yes	Yes
Data Export	Yes	Yes	Yes	Yes
Financial Education	N/A	N/A	N/A	Yes
Free Advertisement Model	N/A	Yes	N/A	Yes

2.5 SUMMARY

In conclusion, the comparative analysis of existing system reveals a important market gap which unsuccess to adders the holistic needs of young adults. While Money Tracker provides necessary analytical depth, it relies on manual input create a rigidity gap that lead to user abandonment. Conversely, TNG e-Wallet eliminated the capture fictions but suffers from data fragmentation, limiting it ineffective as a comprehensive PFM tool. Last but not least, although Fortune City success in solving initial motivation through gamification, it encounters the entertainment value often supersedes financial accuracy. Hence the result of detailed summary will become the basic for developing Smart Pocket and it aims to bridge the divide between rigid utility and passive automation. Meanwhile, Smart Pocket also aims to offer s better solution specifically developed to foster long-term financial literacy among young adults.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter will focus on the methodological framework and systematic procedures adopted to translate the identified theoretical gaps in previous chapter which are rigidity gap and behavioral gap into a function mobile app, Smart Pocket. The development process fulfils Object-Oriented Analysis and Design (OOAD) paradigm, structured within an Agile Software Development Life Cycle (SDLC). These processes are selected in order to ensure user-centric adaptability and iterative refinement. The methodological road map continues with software process, which justifies the selection of the Agile Scrum model as the primary approach. Moreover, a detailed Gantt Chart will be shown to represent the project timeline, ensuring the significant milestones, from the analysis in beginning until final deployment, are executed within the academic timeframe. The following section is project requirement which will translates the user needs into technical specifications. This involves the construction of Use Case Diagram and Sequence Diagrams to model the dynamic interactions between users and system.

Subsequently, the focus shifts to system architecture in proposed Design. This section presents the high-level architectural design connecting the Flutter frontend with Firebase backend, supported by Class Diagram and GUI Wireframes to visualize the simple UI that aimed to minimize user friction. Complementing the interface design in section 3.5 (Data Design) will define the backend storage structured of Smart Pocket. Given the implementation of Firebase, this section will further detail deeply the NoSQL JSON Tree structure, outlining how user profiles, transaction logs, and budget limits are organized hierarchically for efficient retrieval, replacing traditional relational entity relationship models.

The next focussing section is proof of concept, outlining the development environment setup that will includes the configuration of the Flutter SDK and Android Studio, and validates the initial database connectivity. Last but not least, section 3.7 (Testing Plan) establishes the comprehensive validation strategy. The processes are including functional testing to verify feature accuracy and user acceptance testing, which integrates a quantitative survey designed to access user satisfaction and confirm that the application effectively bridges the behavioural gap in PFM.

3.2 SOFTWARE PROCESS

The Adapted Agile Scrum Methodology is selected as the software process to accommodate the innovative features and ensure user centricity. While the traditional scrum framework divides specific roles and rigid ceremonies designed for larger and cross functional teams (Schwaber & Sutherland, 2020), but in a resource-constrained academic environments, strict adherence to traditional agile scrum framework will lead to unnecessary administrative overhead. Consequently, Adapted Agile Scrum Methodology retain the core iterative elements, specifically the product backlog and fixes-length sprints but takes longer time while streamlining communication channels to suit a smaller team structure even one person. Hence, this methodology proved particularly advantages to the development of Smart Pocket, as its modular architecture allows distinct features such as expense tracking and data analytics, to be developed, tested and deployed incrementally. By focusing on iterative delivery, this methodology ensures that the Minimum Viable Product (MYP) while maintaining the flexibility to adapt the evolving technical requirements and feedback throughout the SDLC (PMI, 2017).

Table 3.1 Comparation between Traditional Agile Scrum and Adapted Agile Scrum

Criteria	Traditional Agile Scrum	Adapted Agile Scrum
Team Structure	Designed for multi-functional teams of 5-9 members (Schwaber & Sutherland 2020).	Individual Developer: Smart Pocket is executed by a single student, removing the need for team synchronization.
Role Definition	Distinct roles are assigned: Scrum Master, Product Owner, and Development Team.	Combination Roles: The students assume all roles simultaneously to manage the backlog and executing the code.
Scrum Events	Rigid ceremonies including Daily Stand up (15 minutes), Sprint Review, and Retrospectives.	Streamlined Events: Daily Stand up are replace to Self-Reflection and Sprint Review are conducted periodically with respective supervisor rather than a product team.
Communication	High-bandwidth communication within the team to resolve problems.	Internal & Mentor-Based: Communication focuses on checking the requirements

		based on rubric and seeking guidance from supervisor.
Documentation	Values “Working software over comprehensive documentation”	Hybrid Approach: Requires working software, Smart Pocket and comprehensive academic documentation

3.2.1 GANTT CHART

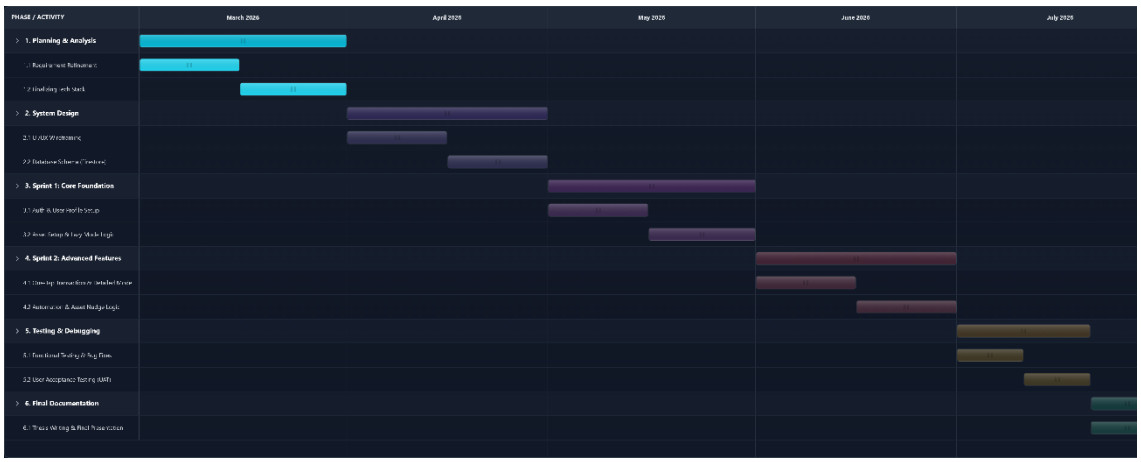

Figure 3.1 Gantt Chart

3.3 PROJECT REQUIREMENT

3.3.1 FUNCTIONAL REQUIREMENT

I. User Authentication & Profile Module

a. When the user initiated the login process, the system shall provide options to authenticate via Google sign in or email/password.

b. When a user logs in for the first time, the system shall display a smart survey to assess the user's financial habits.

c. After the Smart Survey completed, the system shall determine the user's default operating mode (Lazy mode or Detailed Mode) based on the calculated score.

d. The system shall allow users to modify their personal security settings, including username and password updates.

e. If the device supports biometrics, the system shall allow login by using fingerprint or face authentication.

II. Asset & Account Module

a. The system shall allow users to initialize their assets in 2 ways: by entering a total asset sum or by using specific account.

b. If the users initialize asset by using specific account, the system shall provide a localized list of Malaysian financial providers (e.g., Maybank, TNG, ASNB) during account creation.

c. The system shall share data consistency across both modes to ensure changes made in specific account are reflected in total asset sum.

d. When managing investment accounts, the system shall provide an option to exclude specific accounts from “liquid assets” total.

e. If the user updates the balance of an investment account, the system shall record differences as profit/loss instead of income/expense, and prompt a warning to the user regarding this distinction.

f. The system shall allow users to dynamically add, delete and edit financial accounts (Bank, E-Wallet, Investment, etc.) at any time.

g. If an account is deleted, the system shall recalculate the total assets accordingly.

h. When the user views account settings, the system shall suggest specific account tags (e.g., Daily Use Account) based on the user’s asset allocation goals.

III. Transaction Management Module

a. The system shall provide a customizable grid of one-tap shortcut on the dashboard, where each of the shortcut represent a fixed expense amount.

b. When the user taps the shortcut, the system shall check if a default account is configured, if yes, it deducts immediately; if no, the system prompts the user to select a source account before saving.

c. The system shall allow users to modify the shortcut details.

d. When the user clicks the "+" button, the system shall open detailed entry from allowing selection of category (default/custom), amount, optional note and source account.

e. When the users select "transfer" type, the system shall require the selection of source and destination account.

f. The system shall allow users to create both income and expense entries.

g. The system shall allow users to define recurring income or recurring expenses with frequency options (daily/weekly/monthly).

IV. Validation & Error Handling Module

a. If the user attempts a transaction where the amount exceeds the source account's available balance, the system shall display an "insufficient funds" error and prevent the action.

b. If a recorded expense causes the total spending to exceed the user's defined monthly budget, the system shall display a "Budget Exceed" warning message but allow transaction to proceed.

c. The system shall validate all numerical inputs and prompt an error message for any invalid calculations or non-numeric entries.

d. If required fields are missing, the system shall prevent submission and highlight missing field.

V. Automation Module

a. The system shall trigger any pending auto-deduction or auto-increment transaction based on user defined settings.

b. The system shall allow users to define an asset allocation goal.

c. While viewing the account settings, the system shall suggest specific tags for account (e.g., Tag Maybank as daily use) based on the user's allocation goals.

d. If the system detects deviation from the allocation target, the system shall display a suggestion notification.

VI. Analytics & Reporting Module

a. The system shall generate a calendar view visualizing daily cash flow, using green indicators for income and red indicators for expenses.

b. The system shall provide a pie chart comparing the current month's spending against the monthly budget.

c. The system shall display a dedicated part for investment profit and loss which separate from daily living expenses.

d. When the user requests a report, the system shall generate and export a PDF document summarizing the month's income, expense, and asset status.

VII. Category Management Module

a. The system shall provide default category for income and expense.

b. The system shall allow users to add custom categories with any icons.

c. The system shall allow users to edit and delete custom categories.

d. When a category is removed, the system shall prompt whether to reassign or delete affected transactions.

VIII. Backup & Synchronization Module

a. The system shall automatically sync user data to Firestore using the user's unique ID.

b. The system shall allow users to manually export backup files in JSON format.

3.3.2 NON-FUNCTIONAL REQUIREMENT

I. When the user triggers a "One-Tap" transaction, the system shall complete the local data write operation within 1 second.

II. The System shall support Dark Mode and utilize localized icons (e.g., Bank Logos) to enhance user experience.

III. The System shall enforce Firebase Security Rules to ensure that a user can only read and write data belonging to their own unique User ID.

IV. When generating a PDF report, the System shall complete the process within 5 seconds.

3.3.3 CONSTRAINTS

I. The development timeline is strictly bounded between March 2026 to July 2026.

II. The system shall not integrate with real-time Banking APIs as licensing and security constraints so all data inputs rely on user input.

III. The app logic is designed specifically for MYR currency.

3.4 PROPOSED DESIGN

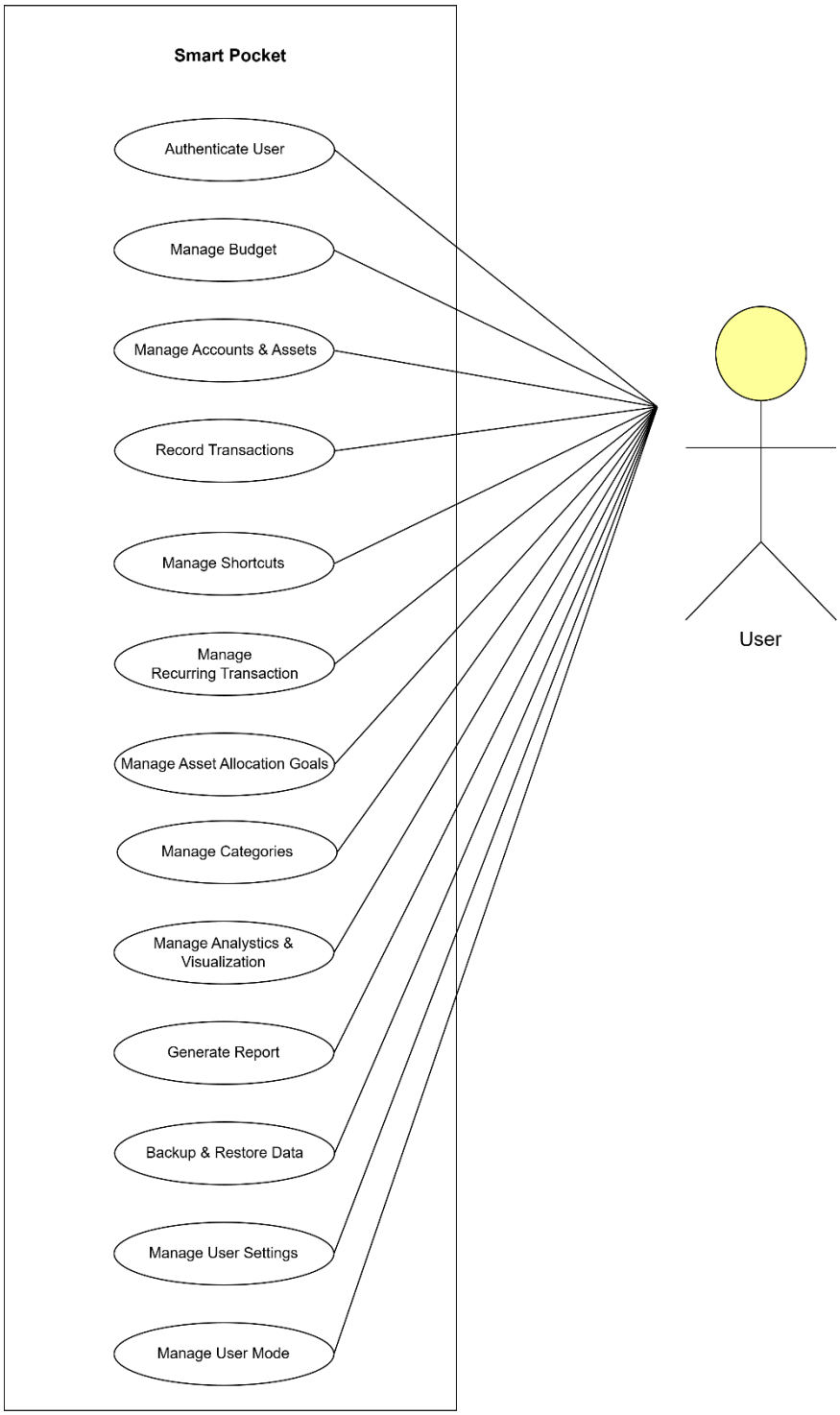

Figure 3.3 Use Case Diagram of Smart Pocket

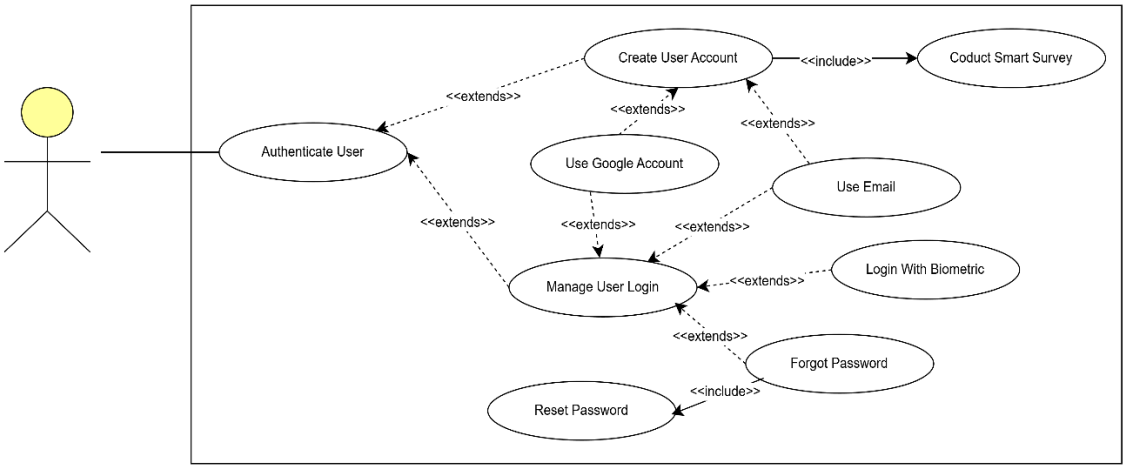

Figure 3.4 Use case diagram of Authenticate User

Table 3.2 Use Case Description of Authenticate User

Use Case ID	UC01
Use Case Name	Authenticate User
Brief Description	This use case explains the process of user authentication using email/password, Google sign-in or biometric login. It aims at allowing users to access Smart Pocket securely. First time login users will be redirected to complete smart survey first before enter dashboard.
Actor	User
Pre-Condition	Smart Pocket has been installed and have a valid email or Google Account.
Basic Flow	1. This use case begins when the users open Smart Pocket Application. 2. System will display the login options (login/register). 3. Users selects preferred login method. 4. System validates login credentials.

	5. System will check if this user is new user or not. 6. System proceeds to load the main dashboard. 7. The use case ends.
Alternative Flow	[A1: Login with Google Account] 1. Users select the “Sign-in with Google” option. 2. System authenticates user via Google API. 3. The use case continues to Step 5 in the basic flow. [A2: Login with email and password] 1. User select the “Sign-in with email” option. 2. System displays the email and password entry screen. 3. User enter their credentials. 4. System performs authentication. 5. The use case continues to Step 5 in the basic flow. [A3: Login with Biometrics] 1. User select the fingerprint or face id icon. 2. System triggers device biometric scanner. 3. System verifies biometric data locally. 4. The use case continues to Step 5 in the basic flow. [A4: First Time Login – Conduct Smart Survey] 1. System detects the user is logging in for the first time. 2. System displays a Smart Survey.

	3. User complete Smart Survey. 4. The use case continues to Step 6 in the basic flow. [A5: Forgot Password] 1. User selects “Forgot Password” from login screen. 2. System displays the password recovery screen. 3. User enter the registered email. 4. System sends a password reset link to the respective email. 5. Users open the reset link. 6. The use case continues to [A6: Reset Password] flow. [A6: Reset Password] 1. User enters a new password and confirm it. 2. System validates password strength. 3. System updates new password. 4. System confirms successful password reset. 5. User returns to the login page. 6. The use case continues to Step 3 in basic flow.
Exception Flow	[E1: Invalid Credentials] 7. System detects incorrect email or password.

	8. System displays an error message. 9. The use case continues to Step 4 in the basic flow. [E2: Google Authenticate Failure] 1. System receives an authentication error from Google. 2. System displays an error message. 3. The use case continues to Step 4 in the basic flow. [E3: Biometric Failed] 1. Device fails to read fingerprint/ face id. 2. System displays an error message. 3. The use case continues to Step 4 in the basic flow. [E4: Email not Registered] 1. System detects that the entered email does not exist. 2. System displays an error message. 3. User return to the password recovery Screen. [E5: Weak Password] 1. System detects a weak password during sign in with email and reset password. 2. System displays an error message. 3. User re-enters a stronger password.

Post Condition	The user is successfully authenticated into system and new users are redirected to Smart Survey

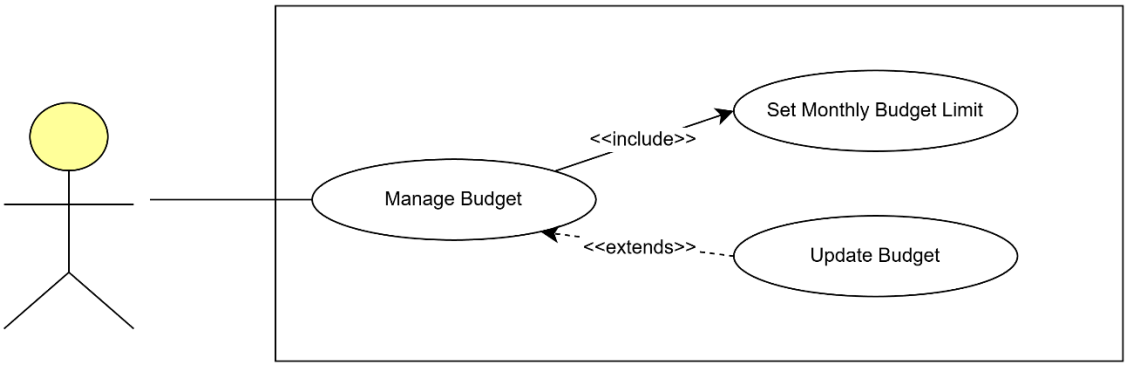

Figure 3.5 Use case diagram of Manage Budget

Table 3.3 Use Case Description of Manage Budget

Use Case ID	UC02
Use Case Name	Manage Budget
Brief Description	This use case explains the process of setting, updating and viewing the monthly budget. It allows users to define their spending limits each month and adjust them as needed. The latest budget amount will be displayed after updates are made.
Actor	User
Pre-Condition	User has logged in successfully.
Basic Flow	1. This use case begins when the user navigates to the budget management settings. 2. System displays the current month's budget amount.

	3. User selects an operation. 4. System updates and displays the latest budget amount. 5. The use case ends.
Alternative Flow	[A1: Set Monthly Budget] 1. User selects the “Set Budget” option. 2. System displays a text entry for budget amount. 3. User enters the desired budget. 4. User selects the Save button. 5. The use case continues to Step 4 of the basic flow. [A2: Update Budget] 1. User selects the “Update Budget” option. 2. System displays the existing budget for editing. 3. User enters the new budget amount. 4. User selects the Save button. 5. The use case continues to Step 4 of the basic flow.
Exception Flow	[E1: Invalid Numeric Input] 1. System will detect non-numeric or invalid values. 2. System displays an error message. 3. User continues to edit the budget.

Post Condition	The updated budget amount is stored and updated.

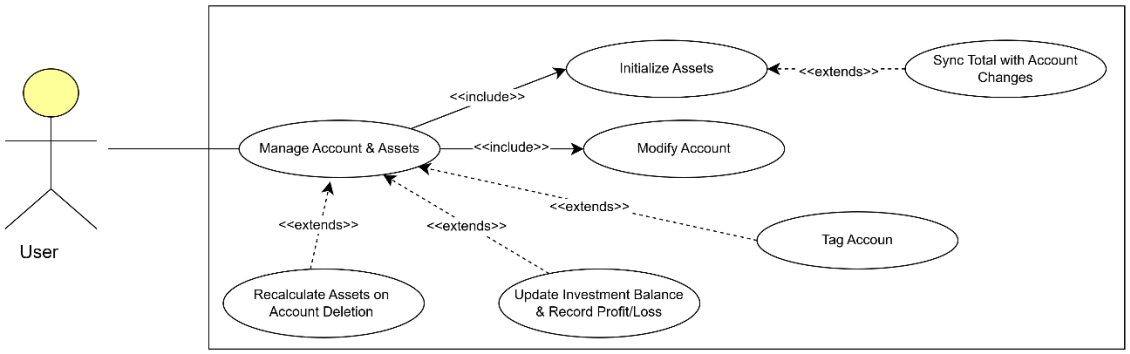

Figure 3.6 Use case diagram of Manage Accounts & Assets

Table 3.4 Use Case Description of Manage Accounts & Assets

Use Case ID	UC03
Use Case Name	Manage Accounts & Assets
Brief Description	This use case will explain about the process of initializing financial accounts and assets, and modifying accounts which including adding, editing and deleting. It aims to manage bank accounts, e-wallets, cash flow, and investment account while ensuring data consistency across total asset and specific accounts mode.
Actor	User
Pre-Condition	User has logged in successfully.
Basic Flow	1. This use case begins when the user navigates to Accounts & Assets page.

	2. System will display the list of existing accounts and total asset summary. 3. User selects an operation. 4. System performs the requested action and refreshes account data. 5. The use case ends.
Alternative Flow	[A1: Initialize Asset- Total Asset Amount] 1. If the user is in “Lazy Mode”, user can initialize by total asset. 2. System will show input box for total asset. 3. User enters the total amount. 4. User select save option. 5. The use case continues to Step 4 of the basic flow. [A2: Initialize Asset- Specific Account] 1. If the user is in “Detailed Mode", user can initialize by specific account. 2. System shows account creation interface. 3. User selects financial provider based on different categories (e.g., Bank Account and E-Wallets). 4. User enters starting balance. 5. System updates the account. 6. The use case continues to Step 4 of the basic flow. [A3: Update Account] 1. User selects update account based on selected account.

	2. System displays account form. 3. User updates information details or delete the respective account. 4. User select confirm. 5. The use case continues to Step 4 of the basic flow. [A4: Update Investment Account Balance] 1. Users select an investment account. 2. System displays balance update input. 3. User enters the new current balance. 4. System detects profit or loss based on differences between the balances. 5. System records the changes as profit/loss instead of income/expense. 6. The use case continues to Step 4 of the basic flow. [A5: Tag Account] 1. If user have set a allocation goal, system will analyses user's allocation goal and the structure of all existing assets and account. 2. System suggests account tags (e.g., daily usage or savings). 3. The use case ends.
Exception Flow	[E1: Invalid Numeric Input]

	1. System detects invalid balance input. 2. System displays an error message. 3. User returns to the input screen. [E2: Delete Operation Cancelled] 1. User cancels delete confirmation. 2. System returns to Accounts page.
Post Condition	The latest account and asset information is updated and reflected in both modes.

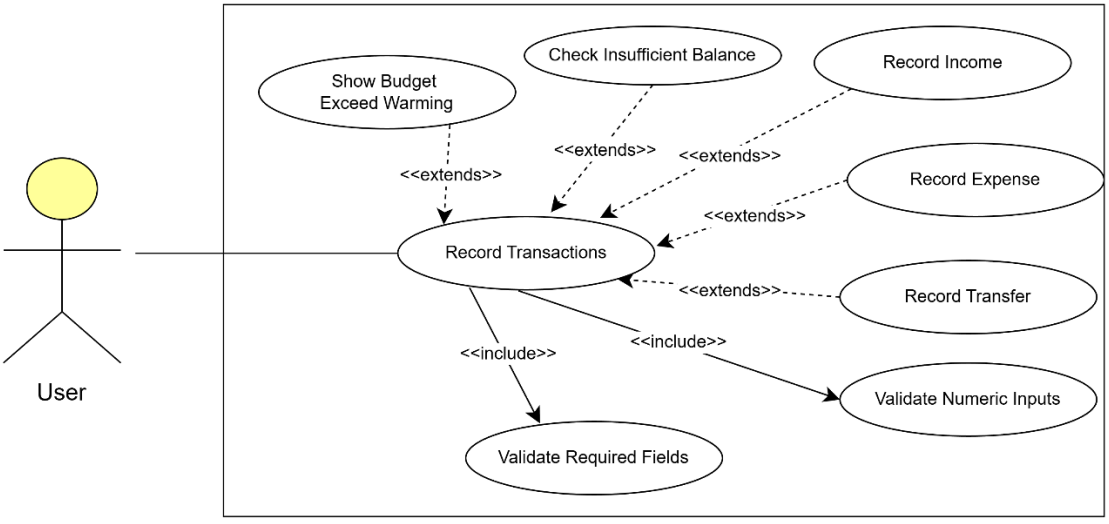

Figure 3.7 Use case diagram of Record Transactions

Table 3.5 Use Case Description of Record Transactions

Use Case ID	UC04
Use Case Name	Record Transactions
Brief Description	This use case explains the process of recording expenses, income, and account transfers. Users may add detailed transactions or record quick expenses using shortcuts.
Actor	User
Pre-Condition	User has at least one active account and the account have enough balance.
Basic Flow	1. This use case begins when the user taps the "+" button on the dashboard. 2. System displays the transaction entry form. 3. User selects transaction type. 4. User fills in all the information. 5. System records the transaction and updated the related accounts. 6. The use case ends.
Alternative Flow	[A1: Select Expense] 1. User selects Expense type. 2. System displays category, amount, account and note fields. 3. User completes the form. 4. The use case continues to Step 5 of the basic flow.

	[A2: Select Income] 1. User selects Income type. 2. System displays category, amount, account and note fields. 3. User completes the form. 4. The use case continues to Step 5 of the basic flow. [A3: Select Transfer] 1. User selects Transfer type. 2. System displays source and destination account fields. 3. User enters transfer amount. 4. The use case continues to Step 5 of the basic flow [A4: Use One-Tap Shortcut] 1. User selects a shortcut button. 2. System checks whether a default account exists. 3. If yes, system deducts amount immediately. 4. If no, user selects the account. 5. The use case continues to Step 5 of the basic flow
Exception Flow	[E1: Insufficient Funds] 1. System detects balance is lower than expense amount.

	2. System displays “Insufficient Funds”. 3. User returns to edit transaction. [E2: Invalid Number] 1. System detects invalid input. 2. System displays error message. 3. User returns to entry form.
Post Condition	Transactions are recorded and the account balances are updated.

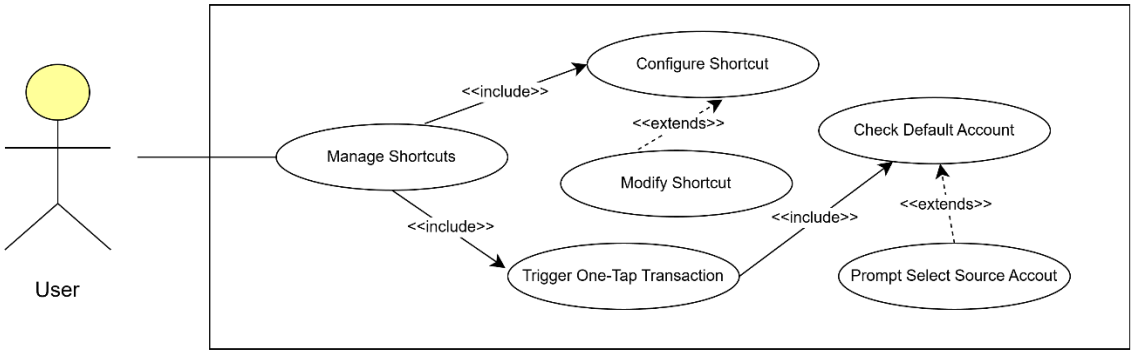

Figure 3.8 Use case diagram of Manage Shortcuts

Table 3.6 Use Case Description of Manage Shortcuts

Use Case ID	UC05
Use Case Name	Manage Shortcuts

Brief Description	This use case explains the process of creating, editing, and deleting one-tap shortcuts, which allow users to record fixed amount expenses faster from the dashboard.
Actor	User
Pre-Condition	User has logged in and has at least one active account.
Basic Flow	1. This use case begins when user navigates to shortcut management page. 2. System displays all the existing shortcuts. 3. User selects an operation which are modifying shortcut. 4. System updates the shortcut list accordingly. 5. The use case ends.
Alternative Flow	[A1: Create Shortcut] 1. User selects “Add Shortcut”. 2. System displays the shortcut creation form. 3. User enters shortcut name, amount, and category. 4. User selects Save. 5. The use case continues to Step 4 of the basic flow. [A2: Edit Shortcut] 1. User selects an existing shortcut. 2. System displays editable fields. 3. User updates shortcut details.

	4. User selects Save. 5. The use case continues to Step 4 of the basic flow. [A3: Delete Shortcut] 1. User selects a shortcut to delete. 2. System prompts for confirmation. 3. User confirms deletion. 4. The use case continues to Step 4 of the basic flow.
Exception Flow	-
Post Condition	Shortcut list updated then reflected on the dashboard for further deduction.

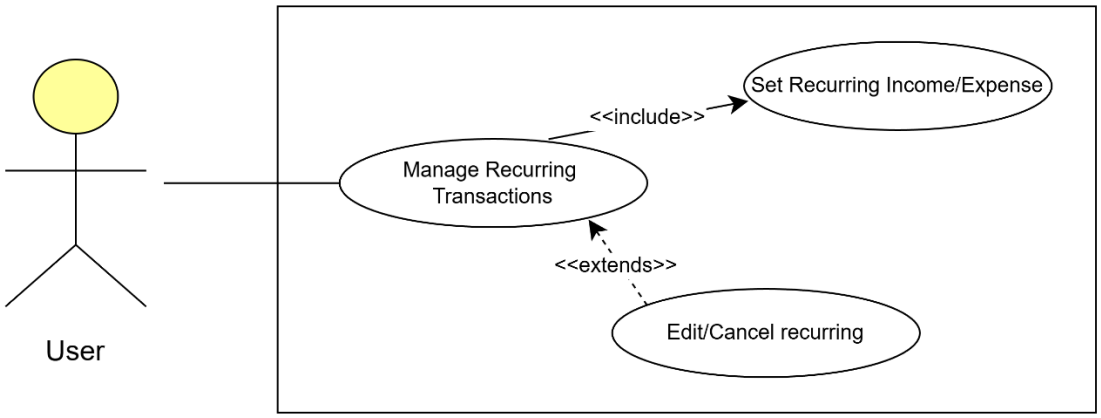

Figure 3.9 Use case diagram of Manage Recurring Transactions

Table 3.7 Use Case Description of Manage Recurring Transactions

Use Case ID	UC07
Use Case Name	Manage Recurring Transactions
Brief Description	This use case explains the process of creating and managing recurring transactions such as subscriptions, allowances, salary, or scheduled transfers.
Actor	User
Pre-Condition	User has logged in and has at least one active account.
Basic Flow	1. This use case begins when the user navigates to the recurring transactions page. 2. System displays the list of recurring items. 3. User selects an operation. 4. System updates the recurring list accordingly. 5. The use case ends.
Alternative Flow	[A1: Create Recurring Transaction] 1. User selects “Add Recurring”. 2. System displays the recurring setup form. 3. User fills in type, amount, date, and account. 4. User selects Save. 5. The use case continues to Step 4 of the basic flow. [A2: Edit Recurring Transaction] 1. User selects a recurring item.

	2. System displays editable fields. 3. User updates information. 4. User selects Save. 5. The use case continues to Step 4. [A3: Delete Recurring Transaction] 1. User selects a recurring item to delete. 2. System prompts for confirmation. 3. User confirms deletion. 4. The use case continues to Step 4. [A4: Automatic Execution of Scheduled Items] 1. Current date matches a recurring transaction schedule. 2. System executes the transaction automatically. 3. System updates account balances. 4. System logs completion in transaction history. 5. The use case ends.
Exception Flow	[E1: Insufficient Funds during Auto-Deduction] 1. System detects insufficient balance. 2. System displays “Auto-Deduction Failed”. 3. System logs the failure.

Post Condition	Recurring transactions are updated or executed according to their schedule.

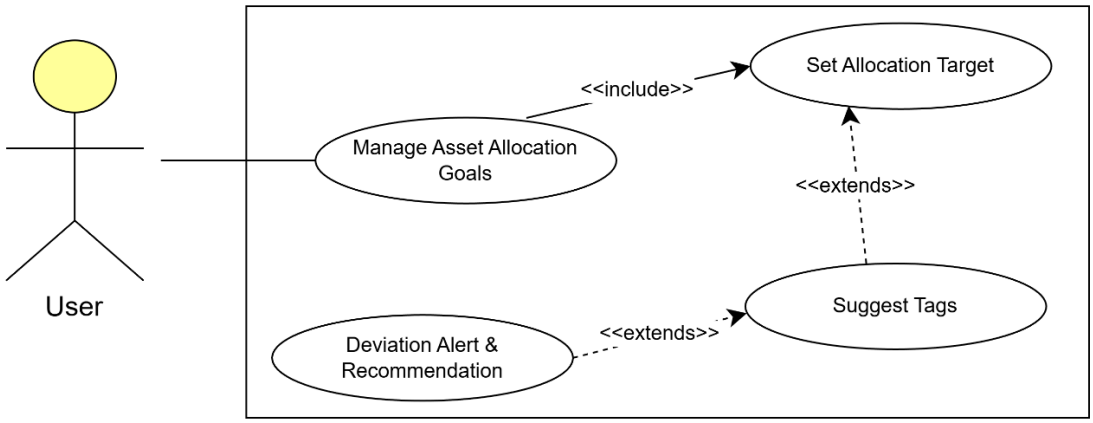

Figure 3.10 Use case diagram of Manage Asset Allocation Goals

Table 3.8 Use Case Description of Manage Asset Allocation Goals

Use Case ID	UC07
Use Case Name	Manage Asset Allocation Goals
Brief Description	This use case explains how a user defines and manage their asset allocation goals such as percentages for savings, investment and cash flow. It allows users to set a target allocation, receive tag suggestions for accounts based on their respective target, and optionally receive deviation alerts and simple recommendations when actual allocation deviates from the target.
Actor	User
Pre-Condition	1. User has logged in and has at least one active account.

	2. The accounts have current balances recorded so the allocations can be computed
Basic Flow	1. This use case begins when the user navigated to the Asset Allocation Goals settings page. 2. System displays current allocation targets and a summary of current allocation by account. 3. User chooses to set or update allocation target. 4. System validates the target percentages sum to 100% or within allowed tolerance. 5. System saves the allocation target and recalculates suggested tags and allocation status. 6. System displays confirmation and updated allocation summary.
Alternative Flow	[A1: Use Suggested Default Template] 1. User selects a pre-defined allocation template (e.g., "Conservative", "Balanced", "Aggressive"). 2. System loads template percentages into the target fields. 3. The use case continues at Step 4 of the Basic Flow. [A2: Suggest Tags for Accounts 1. After saving targets, system analyses account types and balances. 2. System proposes tags for accounts (e.g., "Daily Use", "Emergency Fund", "Long Term Invest"). 3. User accepts or edits suggested tags. 4. The use case continues to Step 6 of the Basic Flow.
Exception Flow	[E1: Invalid Allocation Total] 1. System detects the sum of target percentages ≠ 100% (outside tolerance). 2. System displays "Allocation percentages must total 100%" and highlights offending fields.

	3. User adjusts values and retries. [E2: Insufficient Account Data] 1. System detects accounts lack balances or are empty. 2. System displays “Not enough account data to compute allocation” and suggests adding balances first. 3. Use case returns to Asset Allocation Goals page. [E3: Tag Suggestion Failure] 1. System cannot compute reliable tag suggestions. 2. System displays an error message while saved targets remain intact. 3. User may manually tag accounts.
Post Condition	Allocation target is stored in the user profile and suggested accounts tags are applied to accounts.

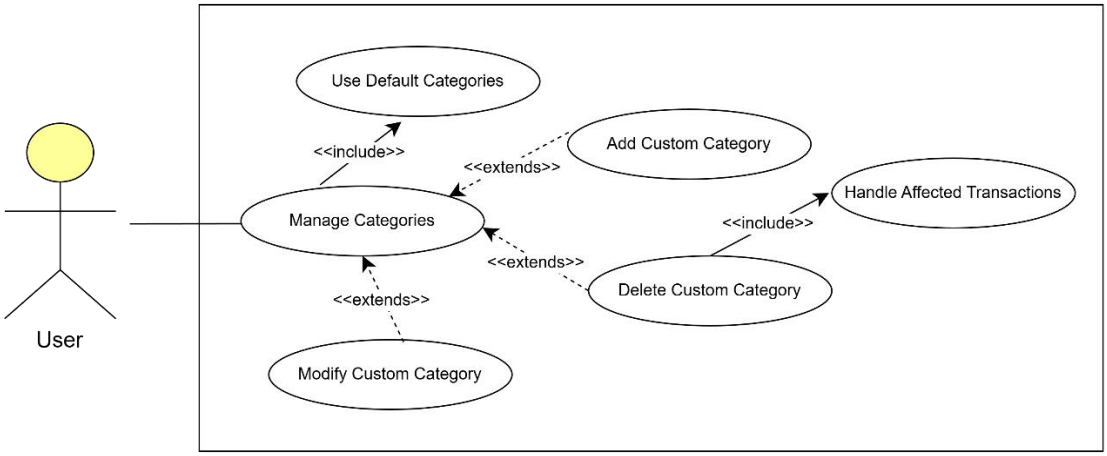

Figure 3.11 Use case diagram of Manage Categories

Table 3.9 Use Case Description of Manage Categories

Use Case ID	UC08
Use Case Name	Manage Categories
Brief Description	This use case explains how users manage transaction categories, including adding custom categories and editing default categories.
Actor	User
Pre-Condition	User has logged in.
Basic Flow	1. This use case begins when the user navigates to the categories management page. 2. System displays the list of existing default categories. 3. User selects an operation. 4. System saved and updates the category list. 5. The use case ends.
Alternative Flow	[A1: Add Custom Category] 1. User selects “Add Category”. 2. System displays the creation form. 3. User enters category name and icon. 4. User selects Save.

	5. The use case continues to Step 4. [A2: Edit Category] 1. User selects an existing category. 2. System shows editable fields. 3. User updates details. 4. User selects Save. 5. The use case continues to Step 4. [A3: Delete Category] 1. User selects a category to delete. 2. System displays a confirmation message. 3. User confirms deletion. 4. The use case continues to Step 4.
Exception Flow	-
Post Condition	The category list is updated accordingly.

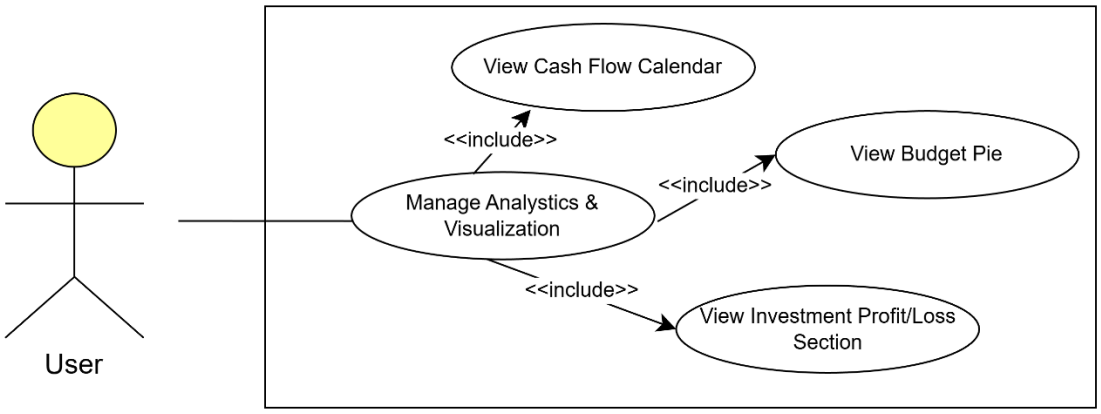

Figure 3.12 Use case diagram of Manage Analystics & Visualization

Table 3.10 Use Case Description of Manage Analytics & Visualization

Use Case ID	UC09
Use Case Name	Manage Analytics & Visualization
Brief Description	This use case explains about how users view their financial insights such as monthly income/expense, calendar cash flow or investment performance.
Actor	User
Pre-Condition	User has recorded at least one transaction.
Basic Flow	1. This use case begins when the user navigates to the analytics page. 2. System generated all required charts and statistics based on user's transaction record. 3. System displays all insights.

	4. The use case ends.
Alternative Flow	[A1: View Calendar Cash Flow] 1. User selects the Calendar View. 2. System highlights income in green and expenses in red. 3. The use case ends. [A2: View Income & Expense Pie Chart] 1. User selects monthly income and expense view. 2. System displays spending vs. budget chart and a monthly income chart. 3. The use case ends. [A3: View Investment Profit/Loss] 1. User selects investment view. 2. System displays dedicated profit/loss section. 3. The use case ends.
Exception Flow	[E1: No Transaction Data] 1. System detects empty data. 2. System displays “No Data Available”. 3. User returns to Analytics menu.
Post Condition	All related analytics visualizations are displayed successfully.

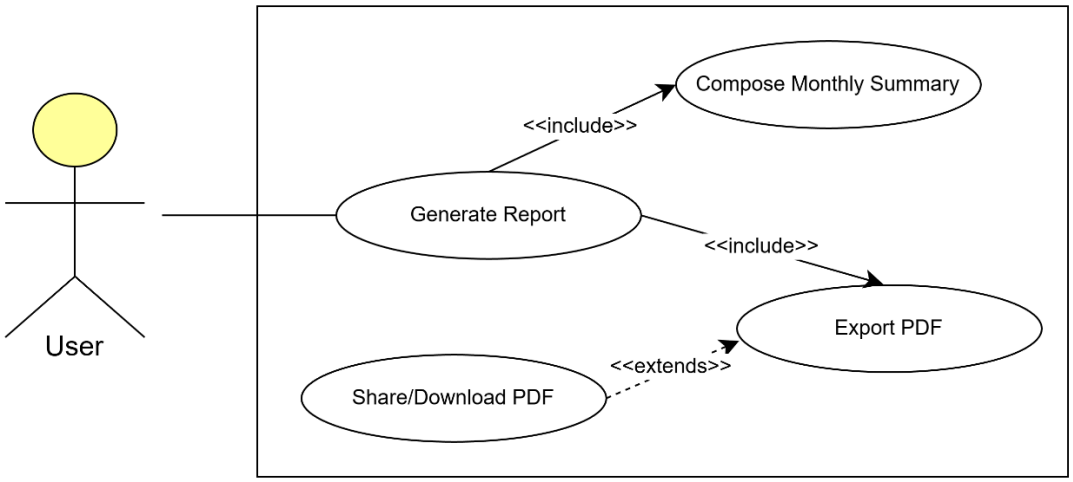

Figure 3.14 Use case diagram of Generate Report

Table 3.11 Use Case Description of Generate Report

Use Case ID	UC10
Use Case Name	Generate Report
Brief Description	This use case explains the process of generating and exporting a PDF report summarizing monthly income, expense, and asset status based on the analytics visualization.
Actor	User
Pre-Condition	User has completed monthly analytics visualization.
Basic Flow	1. This use case begins when the user navigates to the analytics page. 2. System displays the report generation options. 3. User selects a respective month. 4. System compiles all financial data. 5. System generates the PDF report. 6. The use case ends.

Alternative Flow	[A1: Export PDF] 1. User selects the export option. 2. System downloads the PDF to device storage. 3. The use case ends.
Exception Flow	[E1: Report Generation Timeout] 1. System exceeds 10 second generation limit. 2. System displays “Report Failed to Generate”. 3. System prompt users to reexport again.
Post Condition	PDF is generated and exported successfully and available for download.

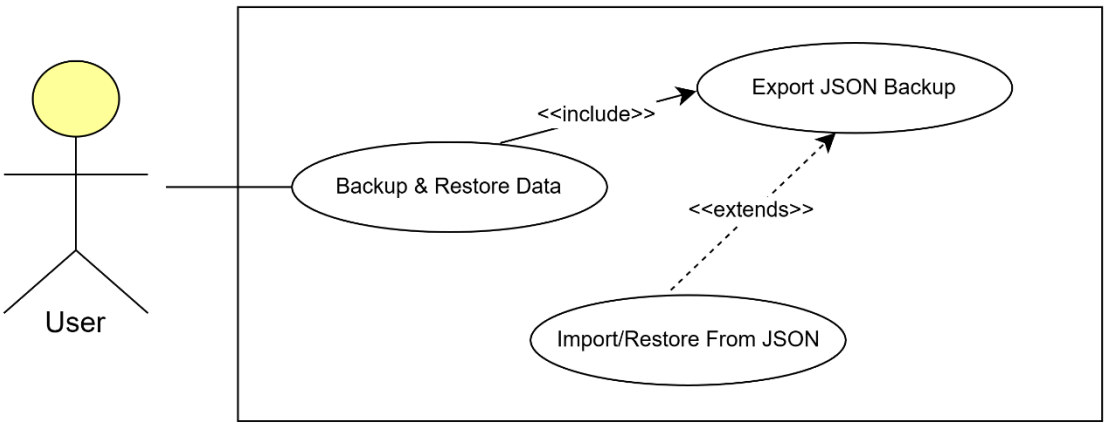

Figure 3.15 Use case diagram of Backup & Restore Data

Table 3.12 Use Case Description of Backup & Restore Data

Use Case ID	UC11
Use Case Name	Backup & Restore Data
Brief Description	This use case explains the process of backing up data to cloud storage and restoring data from backup files.
Actor	User
Pre-Condition	User has logged in.
Basic Flow	1. This use case begins when the user navigates to the Backup & Restore settings. 2. System displays the available options. 3. User selects backup or restore. 4. System processes the selected operation. 5. The use case ends.
Alternative Flow	[A1: Perform Backup] 1. User selects backup option. 2. System packages user data into a structured format. 3. System uploads data to cloud storage. 4. The use case continues to Step 5. [A2: Perform Restore] 1. User selects restore option. 2. System displays available backup files.

	3. User selects a file. 4. System restores data into the application. 5. The use case continues to Step 5.
Exception Flow	[E1: Invalid Backup File Format] 1. System detects corrupted or invalid file. 2. System displays “Restore Failed”. 3. User returns to Restore menu.
Post Condition	Backup and data restored is successfully.

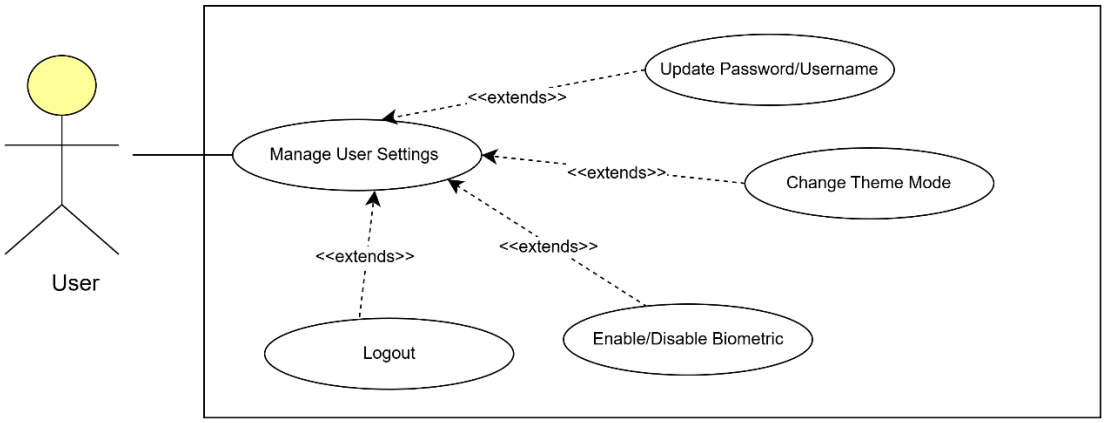

Figure 3.16 Use case diagram of Manage User Settings

Table 3.13 Use Case Description of Manage User Settings

Use Case ID	UC12

Use Case Name	Manage User Settings
Brief Description	This use case explains how the user manages application settings such as username, password, theme mode, biometric preference, and logout. It enables users to personalize their profile and account security preferences.
Actor	User
Pre-Condition	User has logged in and has access to the settings page.
Basic Flow	1. This use case begins when the user navigates to the user settings page. 2. System displays all available settings options. 3. User selects a setting to modify it. 4. System will update and save the latest settings 5. Updated settings are applied immediately. 6. The use case ends.
Alternative Flow	[A1: Update Email or Password] 1. User selects “Update Emil/ Password”. 2. System displays input fields. 3. User enters new email or password. 4. System validates the input. 5. System updates the profile information. 6. The use case continues at Step 5 of the Basic Flow.

	[A2: Change Theme Mode] 1. User selects the theme option (Light/Dark). 2. System applies the selected theme immediately. 3. The use case ends. [A3: Enable/Disable Biometric Login] 1. User selects “Enable/Disable Biometric”. 2. System displays biometric setup/disable screen. 3. User confirms the action. 4. System updates biometric status. 5. The use case continues at Step 5 of the Basic Flow. [A4: Logout] 1. User selects “Logout”. 2. System displays confirmation message. 3. User confirms logout. 4. System logs the user out and returns to Login Page. 5. The use case ends.
Exception Flow	[E1: Weak Password] 1. System detects a weak password. 2. System displays an error message. 3. User re-enters a stronger password.

	[E2: Biometric Setup Failure] 1. Device fails to activate biometric scanner. 2. System displays “Biometric Setup Failed”. 3. User must try again or cancel. [E3: Logout Cancelled] 1. User cancels the logout confirmation. 2. System returns to Settings page.
Post Condition	User settings are updated and applied immediately.

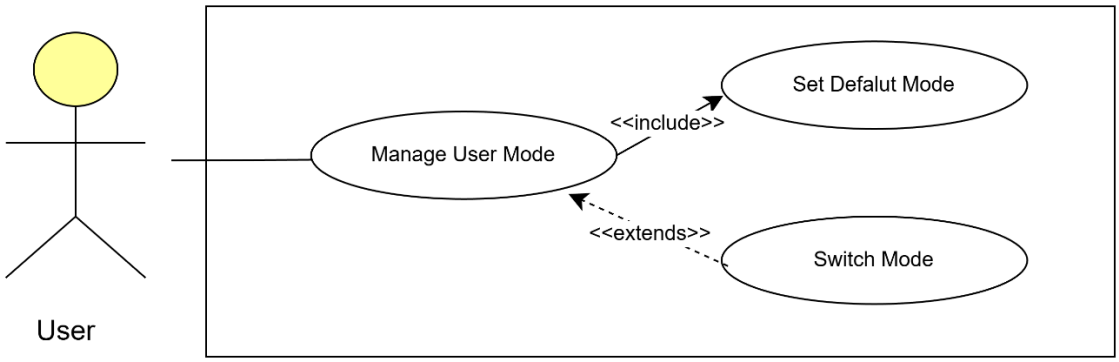

Figure 3.17 Use case diagram of Manage User Mode

Table 3.14 Use Case Description of Manage User Mode

Use Case ID	UC13
Use Case Name	Manage User Mode

Brief Description	This use case explains how users configure their operating mode (Lazy Mode or Detailed Mode). It includes automatically assigned mode after Smart Survey, manually switching modes, and managing privacy related to mode selection.
Actor	User
Pre-Condition	User has completed smart survey or has an existing mode assigned.
Basic Flow	1. The use case begins when the user navigated to the user mode settings. 2. System displays current operating model and related options, 3. User selects an operation. 4. System updates mode or privacy settings. 5. The change is applied throughout the app. 6. The use case ends.
Alternative Flow	[A1: Set Default Mode] 1. System detects first-time mode setup. 2. System prompts user to select initial mode OR auto-assign based on Smart Survey. 3. System sets default mode (Lazy or Detailed). 4. The use case continues at Step 5 of the Basic Flow. [A2: Switch Mode (<<extends>>)]

	1. User selects “Switch Mode”. 2. System displays available modes. 3. User selects the preferred mode. 4. System applies the selected mode. 5. The use case continues at Step 5 of the Basic Flow.
Exception Flow	[E2: Switch Cancelled] 1. User cancels mode switching action. 2. System returns to User Mode page.
Post Condition	Mode is assigned or updated immediately.

3.5 DATA DESIGN

3.6 PROOF OF CONCEPTS

3.7 TESTING PLAN

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

CHAPTER 5

CONCLUSION

5.1 Introduction

REFERENCES

M. Ajouz, F. Abuamria, M. Shehadeh, I. A. Abu-AlSondos, S. H. Aldulaimi and M. Abdeldayem, "Exploring the Cascade: How Buy-Now-Pay-Later Platforms Fuel Impulse Buying Tendency and the Subsequent Impact on Sustainability," 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS), Manama, Bahrain, 2024, pp. 1-5, doi: 10.1109/ICETSIS61505.2024.10459552. keywords: {Correlation;Consumer behavior;Green products;Focusing;Finance;Information age;Credit cards;Buy-Now-Pay-Later;Impulse Buying Tendency;Sustainability},

Cederberg, J. (2013). Changing financial behaviors using mobile PFM tools (Dissertation). Retrieved from https://urn.kb.se/resolve?urn=urn:nbn:se:sh:diva-19735

El Zarif, O., Da Costa, D. A., Hassan, S., & Zou, Y. (2020, June). On the relationship between user churn and software issues. In Proceedings of the 17th International Conference on Mining Software Repositories (pp. 339-349).

S. -M. Wang, V. H. Lan and D. M. Linh, "Promoting Financial Literacy Among Young Adults: A Narrative-Driven Serious Game Approach," 2025 IEEE Gaming, Entertainment, and Media Conference (GEM), Kaohsiung, Taiwan, 2025, pp. 1-6, doi: 10.1109/GEM66882.2025.11155335. keywords: {Correlation;Decision making;Prototypes;Entertainment industry;Media;Quality function deployment;Fatigue;Serious games;Financial Literacy;Serious Game Design;Narrative;Interactive Learning},

Whiteside, E. (2024). *The 50/30/20 Budget Rule Explained With Examples*. https://financialpitstop.com/resources/The_50_30_20_Budget_Investopedia.pdf

Andriani, D., & Nugraha, N. (2018). Spending habits and financial literacy based on gender on employees. IOP Conference Series: Materials Science and Engineering, 407, 012089. https://doi.org/10.1088/1757-899x/407/1/012089

Richter, F. (2025, March 31). Americans are spending less, saving more as tariffs loom. Statista Daily Data. https://www.statista.com/chart/34211/change-in-real-consumer-spending-and-personal-saving/?srsltid=AfmBOoo-00q0xzlicuAGNMhPRolVKf9sYE-xZJVAZrCR60sgDF5yFx_N

Li, I., Dey, A., & Forlizzi, J. (2010). A stage-based model of personal informatics systems. Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI ’10. https://doi.org/10.1145/1753326.1753409

Eysenbach, G. (2005). The Law of Attrition. Journal of Medical Internet Research, 7(1), e11. https://doi.org/10.2196/jmir.7.1.e11

Weinmann, M., Schneider, C., & Brocke, J. vom. (2016). Digital Nudging. Business & Information Systems Engineering, 58(6), 433–436. https://doi.org/10.1007/s12599-016-0453-1

Mail, M. (2024, March 20). Bank Negara: Malaysia recorded 11.5 billion e-payment transactions in 2023. Malay Mail ; Malay Mail. https://www.malaymail.com/news/money/2024/03/20/bank-negara-malaysia-recorded-115-billion-e-payment-transactions-in-2023/124458#google_vignette

TNG Digital reflects on 2023 performance and outlines vision for 2024. (n.d.). Touch ’n Go. https://www.touchngo.com.my/news/tng-digital-reflects-on-2023-performance-and-outlines-vision-for-2024/

TNG Digital Sdn Bhd. (2021, August 21). How can I view or download my transaction history? Touch 'N Go EWallet Help Centre. https://support.tngdigital.com.my/hc/en-my/articles/360035649754-How-can-I-view-or-download-my-transaction-history

J. Hamari, J. Koivisto and H. Sarsa, "Does Gamification Work? -- A Literature Review of Empirical Studies on Gamification," 2014 47th Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 2014, pp. 3025-3034, doi: 10.1109/HICSS.2014.377. keywords: {Context;Psychology;Databases;Educational institutions;Games;Libraries;gamification;motivational affordance;persuasive technology;motivation;HCI},

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

Project Managemetn Insitutue. (2017). Agile practice guide. Project Management Institute

APPENDICES

Appendix A: Google Form- Analysis of Smart Pocket in young adults

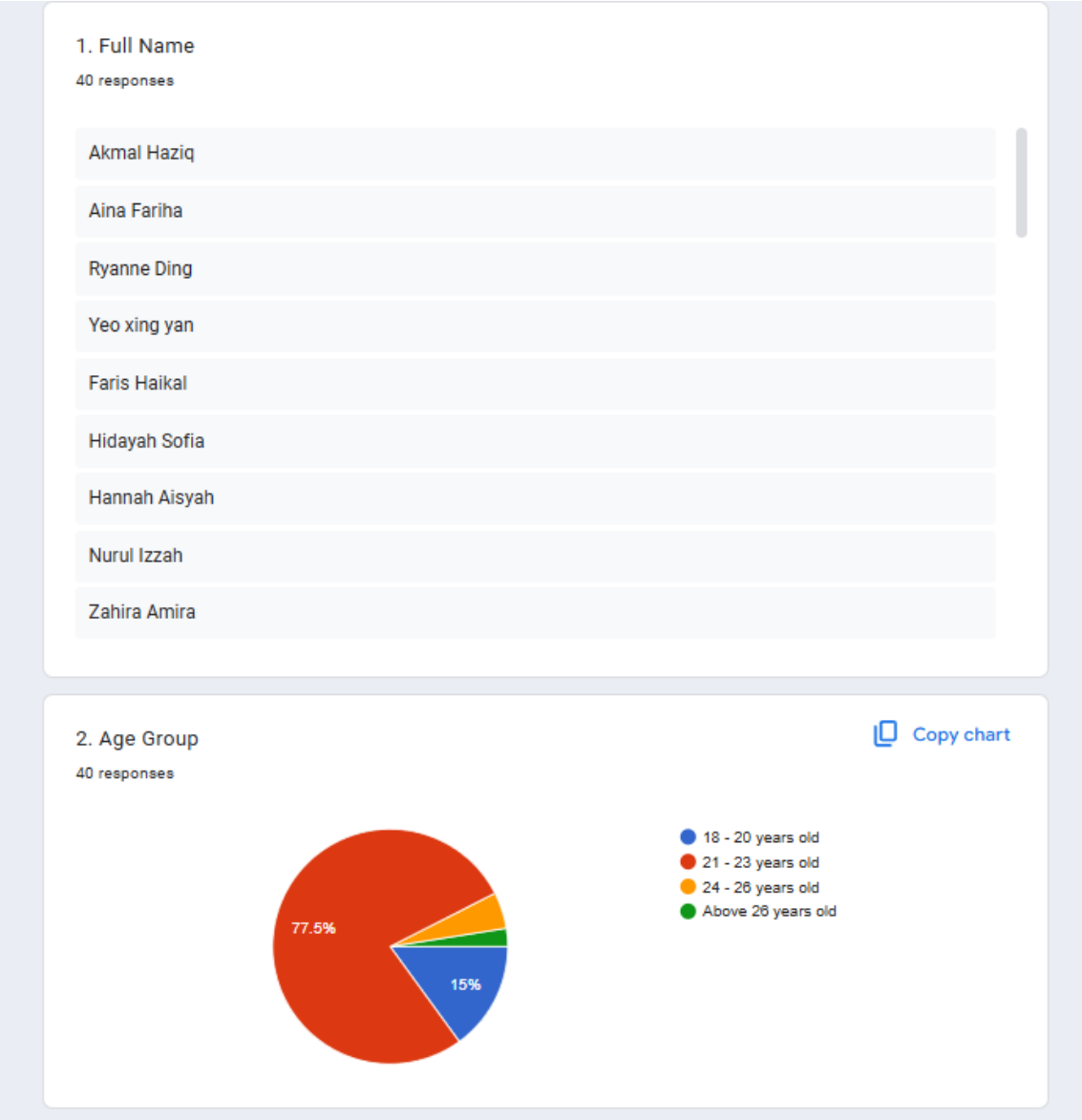

Figure 1: Respondent profile (Part 1)

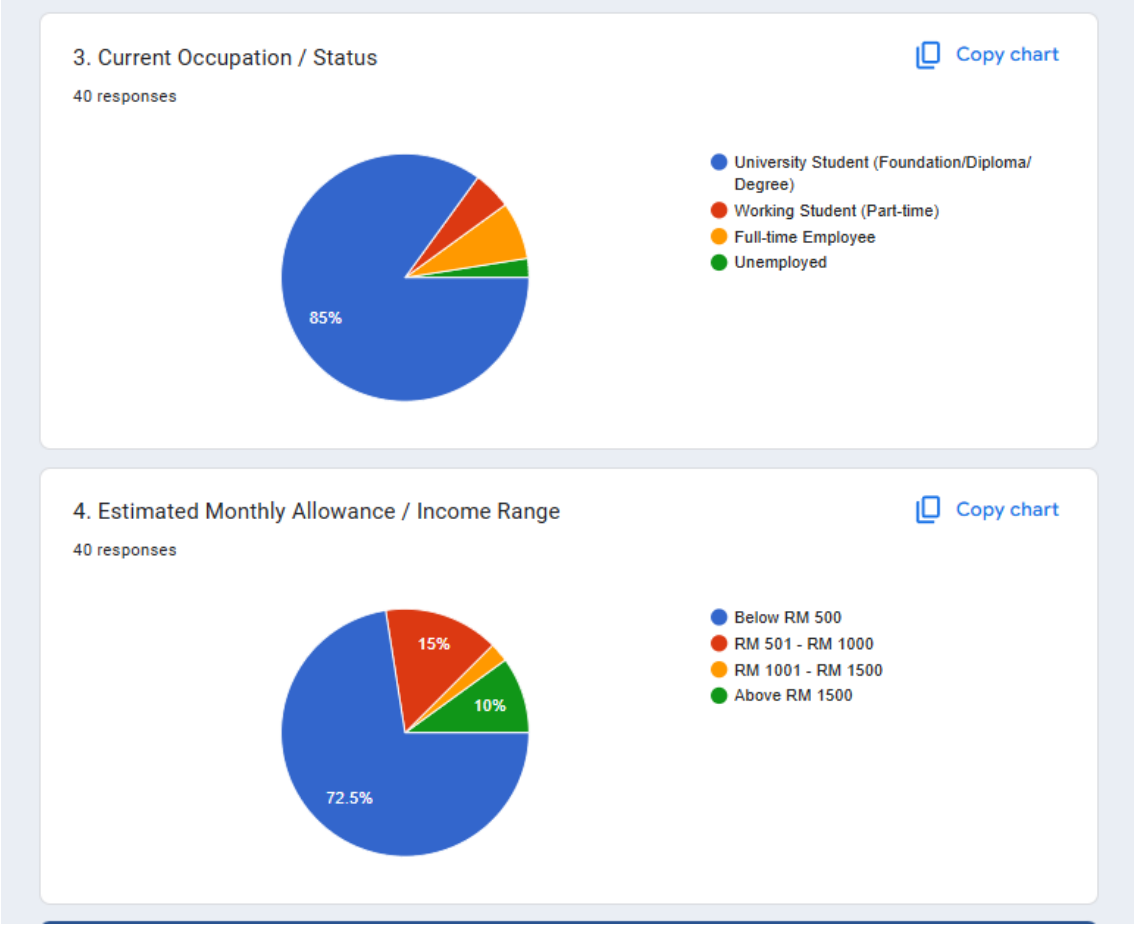

Figure 2: Analysis of respondent profile (Part 2)

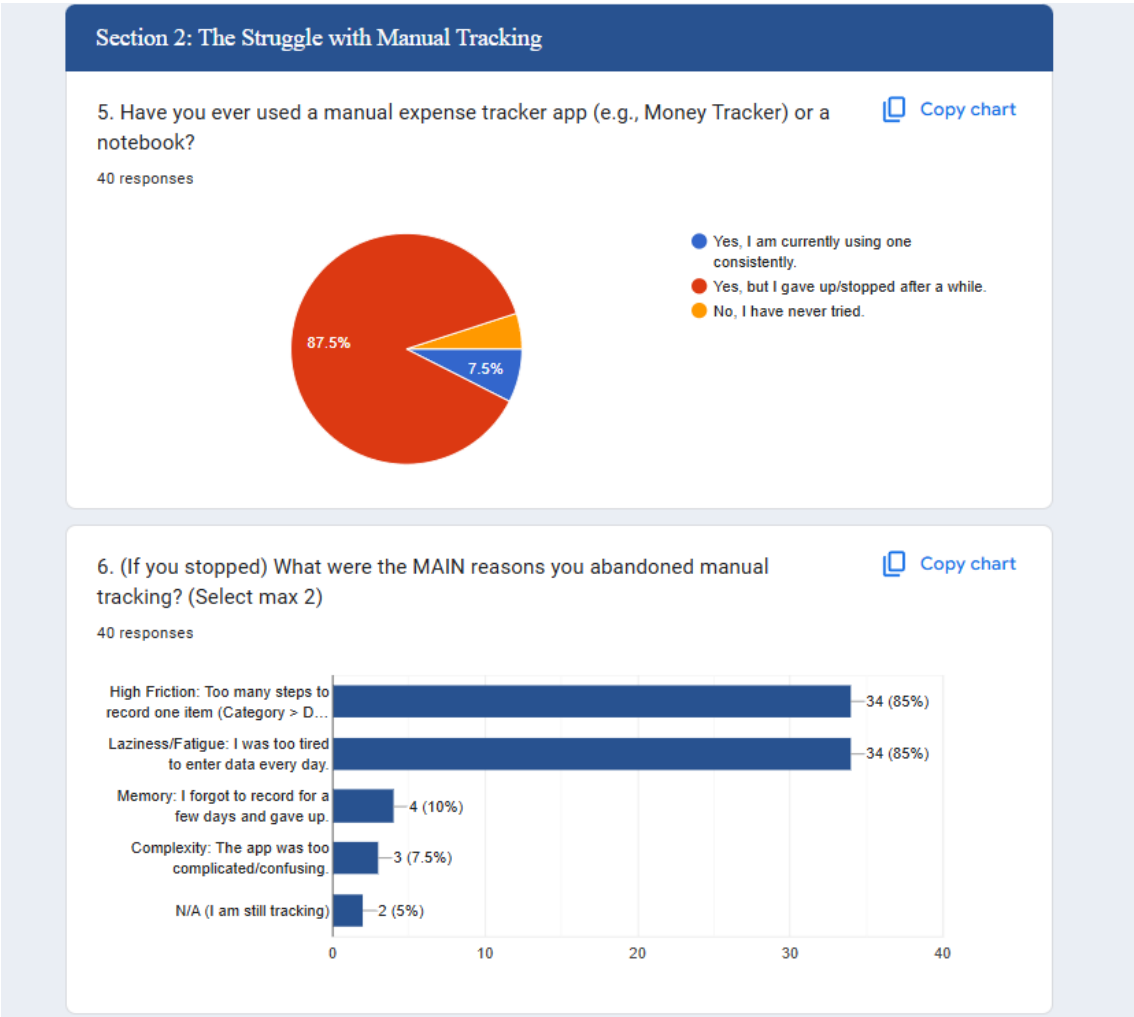

Figure 3: Analysis of the struggles with manual tracking

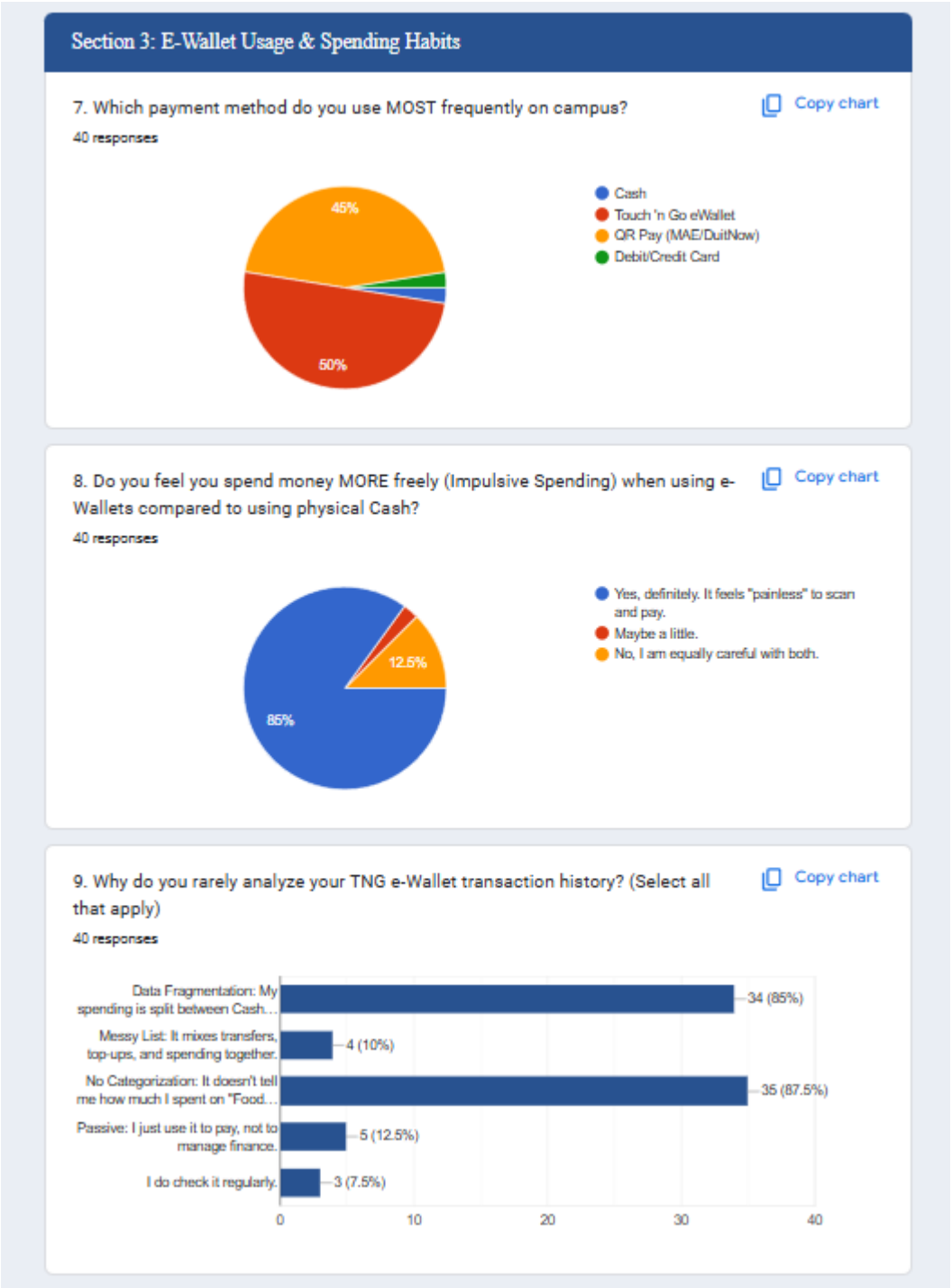

Figure 4: Analysis of e-Wallet usage and spending habits

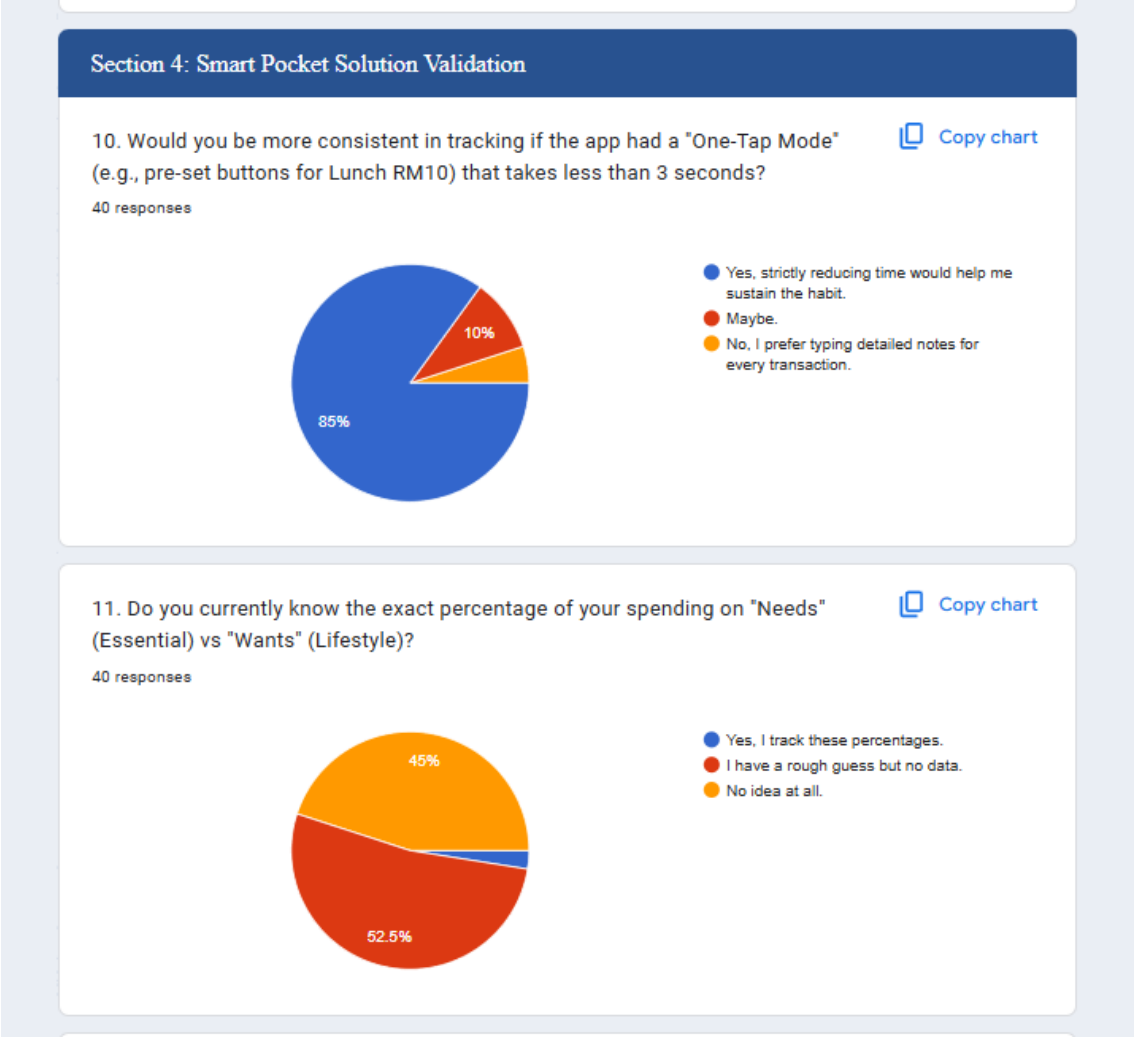

Figure 5: Analysis of Smart Pocket Validation (Part 1)

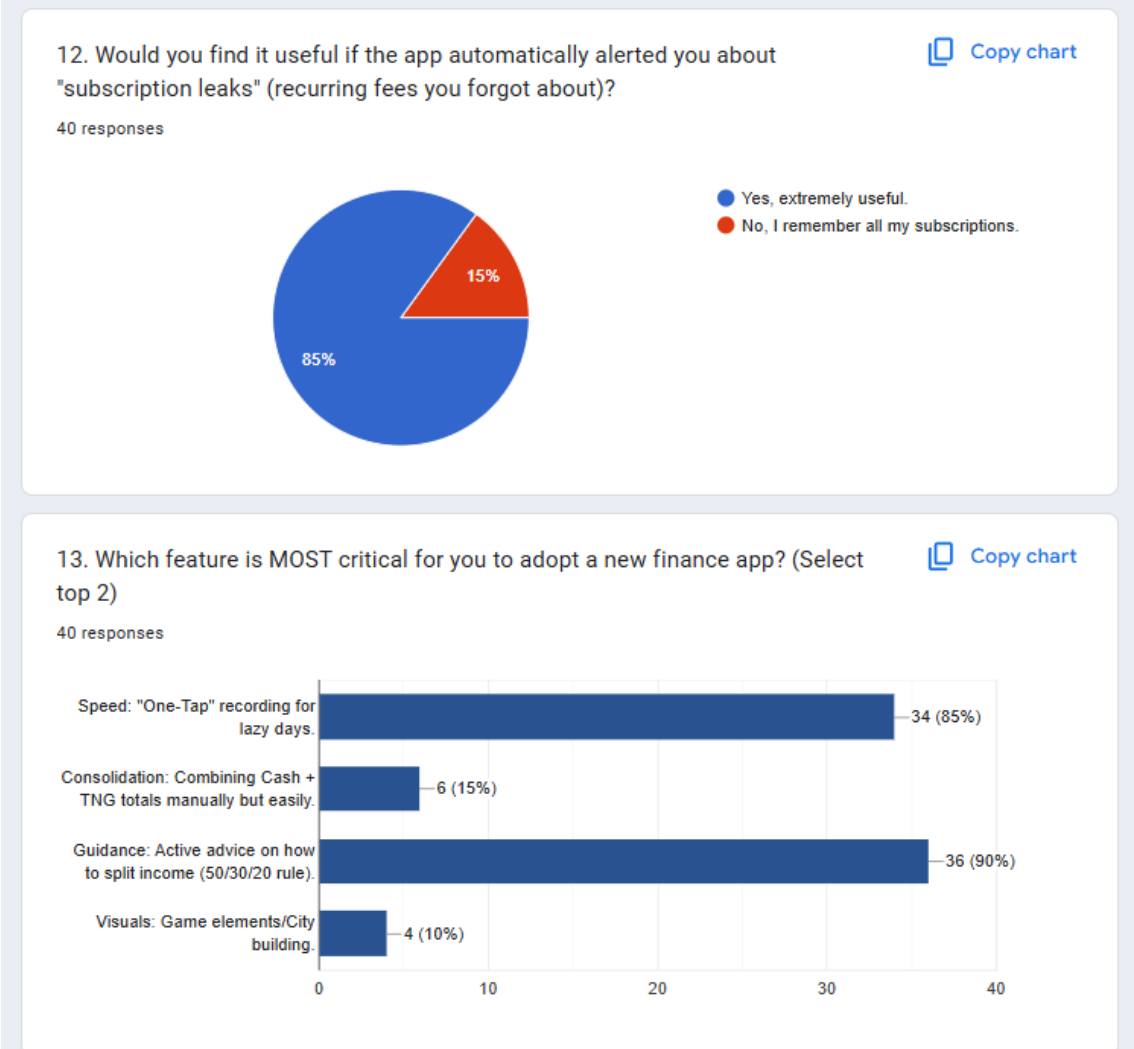

Figure 6: Analysis of Smart Pocket Vvlidation (Part 2)

Appendix B: Title