Personal Finance Management Mobile Application

BY

Pow Kah Meon

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)

FEBRUARY 2025

COPYRIGHT STATEMENT

© 2024 Pow Kah Meon. All rights reserved.

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Information Systems Engineering at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ts Dr Suthashini a/p Subramaniam, who has given me this bright opportunity to engage in a personal finance management mobile application project. It is my first step to establish a career in the mobile application development field. Thanks a million thanks.

Finally, I would like to express my gratitude to my friends and classmates who offered their insights and encouragement throughout this journey. I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This project is concerned with the design and development of a Personal Finance Management Mobile Application to be utilized for helping users, particularly students and young professionals, manage their financial activities efficiently via a smartphone platform. It tackles some of the key problems, such as inefficient expense tracking, no budgeting discipline, and no access to integrated financial planning tools. This app simplifies record keeping and classifying of income and expenses, makes budgets and monitors saving goals, monitors each saving goal, and reminds one when to pay certain bills. In addition, the application has its own loan calculator that estimates each repayment scheme and there is a real-time currency converter in case one handles multiple currencies. The system was developed using the Agile methodology, enabling iterative refinement through continuous feedback and testing. Android Studio was the primary development environment, while secure and real-time data storage were provided by Firebase Firestore. UI/UX design was implemented using Figma to maximize usability, and graphical data visualization tools were also incorporated within the application to enable users to track spending trends and understand their financial habits. Overall, the project aims to promote financial literacy, enable informed decision-making, and provide a reliable digital solution for personal finance management.

Study: Financial Technology (FinTech), Mobile Application Development

Keywords: Personal Finance Management, Expense Tracking, Mobile Application, Budget Planning, Financial Goal Setting

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Background

In today's fast-paced world, managing personal finances effectively has become essential for achieving financial stability and long-term goals. **According to recent data shown on Figure1.1. below shows that there are 89.29% of Malaysians are using smartphones in 2024, with smartphone usage growing year by year.** The widespread adoption of smartphones provides an ideal platform for individuals to manage their finances conveniently. The Personal Finance Mobile Application is designed to empower users by providing them with a comprehensive tool to track, analyze, and control their financial activities directly from their smartphones. This application serves as a dedicated platform for budgeting, expense tracking, saving goals, and debt management, aiming to foster better financial habits and improved spending awareness.

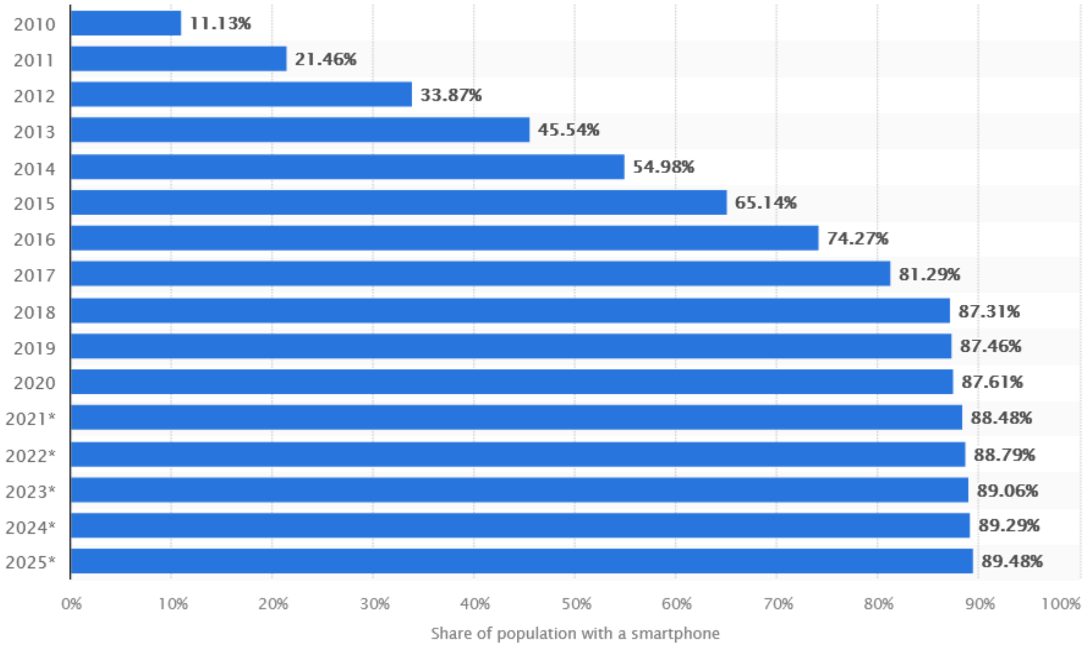


Figure 1. 1 Percentage of Population Using Smartphones

The project integrates essential financial planning features, including income and expense categorization, budget-setting options, and real-time spending insights, tailored to each user's financial profile. Users can set savings goals, monitor their progress, and make informed financial decisions through data-driven insights. In addition, the app incorporates reminders for bill payments, debt repayment trackers, and visualizations of spending patterns, assisting users in staying accountable and financially disciplined.

With security and privacy as top priorities, the application leverages encryption and secure data storage, ensuring users' financial information remains protected. This mobile application is a response to the growing need for accessible financial management tools that are user-friendly, reliable, and designed to support users' financial journeys toward stability and prosperity. Through intuitive design and robust functionality, this personal finance budgeting app will help users achieve greater financial awareness, discipline, and independence.

1.2 Problem Statement and Motivation

1.2.1 Problem Statement

1. Problem of Ineffective Expense Tracking and Budgeting

In our reward-seeking world today, many people struggle to keep track of their money, especially in terms of their spending and budgeting, which is one of the essential tenets of financial literacy. Even the standard methods available for managing finances such as using a notebook for capturing expenditures and calendars to set budgets seem better as, when compared to nothing, they are inefficient and taxing. Most of the end users, however, don’t feel the urge to conduct these tasks regularly which in effect miss a tracking method or an overarching picture of their spending habits. A misunderstanding can result in consumers not being able to assess their spending patterns which would lead them to unreasonable extravagance and poor spending decisions. Without such an overall picture, users are likely to find very few areas of their spending which they can optimize and often will not meet their set financial goals.

2. Lack of Financial Management Motivation and Savings Discipline

Savings are often a topic many people are embracing as their goal however they often try to save money but do not have simple motivators for active self-stimulation of saving money. It is quite customary to surpass the threshold or fail to meet the saving objectives set during their registration, due to lack of notifications or the necessity to monitor the balance. This situation is particularly widespread when a user does visit his account on a regular basis. Most of the time, either people forget to set them in the first place, or these mechanisms are simply not present in the app that would prevent a person from making spendings until they reach the threshold. This allows for poor financial management, which leads to overspending, impulse purchases of high damages and an overall inability to save money which leaves a person in stress and instability of finances in the long term.

In addition, for a range of individuals, it may be difficult to set realistic targets or to evaluate one’s success in relation to the targets set. In the absence of suitable mechanisms which could help the users to effectively track their saving progress or debt levels, users may become discouraged and be unable to uphold a smooth saving routine. Within this mobile application, these problems are addressed by the incorporation of functionalities which allow for automatic reminders when budget limits are exceeded, the notifications for savings encouragement and the functionalities aimed at budget setting and monitoring. Such functionalities will be used by the application in a manner that the users are able to develop positive financial habits and in turn, develop a set saving regime that will foster a sense of security and long-term sustainability.

3. Inability to Optimize Financial method

Besides, many individuals fail to understand the impact of taking out loans or investing on their future finances nowadays. Lacking financial understanding makes it difficult to calculate the amount to be returned, the interest to be paid, and the returns on potential investments. These folks may utilize typical approaches for calculating these variables, such as internet calculators. Most probably, these solutions will be achieved, but their availability and degree are quite elementary and do not meet the person's individual demands nor deviant from the person's total debt situation.

Most individuals lack the desire and financial understanding to sift through this arithmetic daily, leaving them unsure of how such actions may affect them in the future. Lack of awareness of such scenarios may frequently lead to bad financial decisions, resulting in large debt, postponing investment possibilities, or failing to accomplish their primary financial goals. In the absence of a straightforward and integrated loan and investment calculator, individuals are more prone to make poor judgments, limiting their capacity to make good financial decisions and so achieve their financial objective.

1.2.2 Motivation

Due to the oversensitive environment of today, the need for managing one’s finances has increased. Therefore, estimate the ability of the average person to control their spending and stick to a strict budget. Utilizing today’s conventional methods such as manually writing down spending, in addition to budgeting, and anticipating future cash flows to achieve set financial objectives, are tedious, slow, and offer no real-time feedback. Feedback has the potential to prevent various poor financing decisions. The motivation behind this personal finance application is the hope for an easier financial management tool which aims at automating tedious processes, making it easy for all users irrespective of their financial situations

On the other hand, a good number of users show a lack of discipline as well as motivation to control their own finances. Other features include automated reminders to pay bills and tools to help users create and stick to a savings goal so that users can regularly assess their financial status and expenditure. These applications are aimed at improving financial management as well as ensuring that users do not fall behind their engagements and remain focused on their financial objectives.

Finally, since financial knowledge and the usage of the tool is increasing, the ability to calculate loans is an acceptable addition as it will help survive users with loans and future expenses. This project is pursued because it is believed that easy access to a full package of services which includes affordability, education and safety functions will help a user to wisely manage finances in a secure and friendly environment.

1.3 Objectives

1. To develop a comprehensive financial tracking system

This project aims to create a financial tracking system that helps users manage their expenses and income effectively. The system will include features like transaction logs, categorized expense tracking, and visual summaries through graphs and charts. By providing clear insights into spending habits, it will enable users to make better financial decisions and achieve their goals.

2. To implement planning and projection tools

This objective focuses on creating tools that assist users in planning and forecasting their financial activities. Features such as budget planning, goal tracking, and financial projections will empower users to set realistic targets and monitor their progress over time. These tools will provide data-driven insights to help users make informed decisions and stay on track toward achieving their financial objectives.

3. To develop advanced financial calculation features (Loan calculator and real-time currency converter)

The third objective aims to integrate essential calculation tools like a loan calculator and a real-time currency converter into the financial management system. The loan calculator helps users estimate repayment schedules by calculating monthly payments and total loan amounts based on factors like interest rates and repayment terms, aiding informed financial decisions. The real-time currency converter, using APIs for up-to-date exchange rates, provides accurate currency conversions, ideal for users involved in international transactions or travel. These tools enhance user experience by offering practical, efficient solutions for managing finances, aligning with the project’s goal of empowering users with advanced financial tools.

1.4 Project Scope and Direction

1. Develop a Comprehensive Financial Tracking System

To support this objective, the application will include features for manual recording of income and expenses, categorization of transactions, and viewing financial summaries. Users can add transaction details such as date, amount, category (e.g., food, transport, entertainment), and notes. The app will support predefined categories, and users can also create custom ones. All transaction data will be stored in Firebase Firestore, enabling real-time sync across devices. The scope includes local caching for offline access but does not support automated bank data import or OCR receipt scanning. Graphical overviews such as pie charts for expenses by category and bar charts for spending, earning, and all overviews will be included for better visualization.

2. To implement planning and projection tools

This scope includes the development of tools that allow users to set monthly budgets per category and define personal saving goals. Users will be able to input their target amount, track progress, and see a visual indication (e.g., percentage bar) of how close they are to reaching each goal. The budget setting interface will allow users to allocate limits to various categories and receive alerts when they approach or exceed those limits. Budget tracking is manual and rule-based, meaning users must enter expenses to update the budget status. The app will not include forecasting algorithms or historical spending predictions. The system will not automatically suggest budget amounts or goals, nor does it include collaborative goal tracking between users. Everything is individually set and managed within the app. Notifications will be included to remind users about nearing budget thresholds or deadlines to contribute to savings goals.

3. To develop advanced financial calculation features (Loan calculator and real-time currency converter)

The application will include two utility tools: a Loan Calculator and a Real-Time Currency Converter. The loan calculator allows users to input details such as loan amount, interest rate, and term length, then displays monthly repayment, total interest, and an amortization schedule. This helps users understand repayment implications for personal, housing, or car loans. The scope includes various loan types but does not extend to real loan applications, loan offers, or integrations with banks. The Currency Converter fetches live exchange rates via a public API and converts entered amounts between major currencies. It supports real-time calculations but does not include historical rate charts or multi-currency account management. These tools are implemented for individual use only and serve as calculators without saving or linking the results to financial records. Both tools are built directly into the mobile app and do not require additional installations or modules.

1.5 Contributions

This personal finance application employs a variety of functionalities that enhance ease of financial management, promote user participation, and protect user information. One of the features of the application is the bill reminder function that notifies users of pending payment deadlines. This function is useful in enabling users to avert the defaulting of obligations which could have adverse effects on their finances as a result of penalties for late payments.

The app also has a loan calculator which lets users determine the amount they would pay in monthly installments along with the interest amount on loans. This allows the users to make better decisions with their borrowing as they are able to allocate funds for their significant purchases and any outstanding debts. As an enhancement to its users understanding of overall management, the loan calculator highlights aspects of loan repayment and helps in the users coordinates or time frame of repayment.

Finally, the application makes use of the data visualization tools and presents its users in a structured and harmonious manner the analytic of their expenditures, savings performance, and budgeting. The app includes charts and summaries enabling the users to visualize achievement for the financial goal, monitor lagging with targets that imply overspending and correct such budget elements. Through these modern features of the application, the users are enabled to manage their finances and obligations as well as make informed financial decisions in a safe and hassle-free manner.

1.6 Report Organization

This report is organized into six chapters, each of which explains a significant step in developing the Personal Finance Management Mobile Application. Chapter 1 starts with an introduction to the project via the background, problem statement, objectives, scope, and significance of the system. Chapter 2 presents a literature review, including existing research, solutions, and technology related to personal finance applications, and identifies where this project attempts to bridge the gaps. Chapter 3 is concerned with the system design, describing the application architecture, development tools, and design models adhered to, such as use case and interface diagrams. Chapter 4 is about the implementation and testing phase, focusing on the development process, coding techniques, frameworks utilized, and testing techniques adhered to in order to ensure functionality and reliability. Chapter 5 presents the system result and discussion, evaluating the final product in terms of performance, usability, and user feedback. Lastly, Chapter 6 wraps up the report by summarizing the work done, looking back at the overall development experience, and proposing future enhancements to further improve the application.

Chapter 2 Literature Review

2.1 Application Review

People often overspend their budget today because of the lack of financial planning and understanding and spontaneous shopping. Hence, a budget tracking tool is important for people to gain financial knowledge, handle costs, and plan budgets so that they can achieve better financial health and security. Under this part, we will study the related uses in the current market and bring changes and improvements.

2.1.1 Money +

Figure 2. 1 Money+ Logo

The first budgeting app that I have reviewed is Money+. Money+ is a personal finance management application designed to assist users in effectively tracking and organizing their financial activities. The app offers a range of features, including transaction tracking, budget setting, and financial goal management, all presented through a user-friendly interface. Available on both iOS and Android platforms, Money+ aims to simplify financial oversight for individuals seeking to gain better control over their spending and savings habits. One of the standout features of Money+ is its intuitive transaction tracking system, which allows users to record expenses and income effortlessly. The app provides visual representations of spending patterns through graphs and charts, aiding users in identifying areas for potential savings. Additionally, Money+ supports budget creation, enabling users to set spending limits across various categories and monitor adherence to these budgets over time. The inclusion of financial goal-setting tools further enhances its utility, allowing users to define and track progress toward specific financial objectives. Moreover, the app offers the convenience of exporting transaction data into Excel format, facilitating detailed analysis and record-keeping.

One significant advantage of Money+ is its ability to simplify financial management for users. The app offers features like transaction tracking, which is seamlessly integrated with visual graphs, helping users understand their spending habits at a glance. Moreover, the budgeting function is particularly useful, as it allows users to create specific spending limits for various categories, ensuring that they remain financially disciplined. The inclusion of features like daily reminders and bill notifications ensures that users stay informed about their financial commitments, reducing the risk of missed payments. Another notable feature is the ability to export transaction records to Excel, providing flexibility for users who wish to perform in-depth analysis or maintain external financial records. Furthermore, the app's intuitive design ensures that even users with minimal technical knowledge can navigate it comfortably. While it is not overloaded with unnecessary tools, its well-curated features offer practicality and ease of use, making it ideal for users seeking a straightforward budgeting solution.

Despite its advantages, Money+ does have some notable drawbacks. Firstly, the app lacks advanced features that many users might expect from a comprehensive financial management tool, such as real-time currency conversion and recommendations for savings goals. This limitation may make it less appealing to users who require a more robust tool for managing complex finances. Additionally, the absence of a chatbot assistance feature means that users must rely on self-navigation to find solutions to their queries, which could be cumbersome for new users. Another shortcoming is the lack of educational resources, which would otherwise help users improve their financial literacy and make better-informed decisions. Furthermore, the app does not include features like customizable calculation tools, which might limit its flexibility for users with unique budgeting needs. While Money+ is a practical option for basic financial management, its limited scope and lack of advanced functionalities may deter users who require more comprehensive financial tracking and planning capabilities. Nonetheless, for those seeking simplicity and an easy-to-use interface, it remains a suitable choice.

2.1.2 YNAB

Figure 2. 2 YNAB Logo

Next, the budget app that I have reviewed id You Need A Budget (YNAB). YNAB is an application that is designed for budgeting. What makes YNAB interesting is its focus on the psychological aspects of budgeting. Based on the online reviewed, this application has gates and budgeting is simplified into 4 steps [1]. That every dollar should work for you and every dollar is needed to be noted down, proper planning as well as time distribution is essential for great expenditure such as pre-purchase of any item, it is normal to modify your budget and the last thing that the user should do is make his money work for him or her instead of being squandered. YNAB's Cato systems' functionality allows linking with more than 12 000 banks which makes it easy for transactions to be carried out automatically and transactions to be rendered in real time, other systems support desktop and handhelds (iOS and Android). The software also supports multiple budgets which are distinctively different from each other in that there is no necessity to demolish or restart the main aims of all the users and are easily adjustable to the majority of users. Moreover, every new user may use YNAB for 34 days free of charge and therefore use its measures and functions and learn about budgeting through a range of webinars and videos.

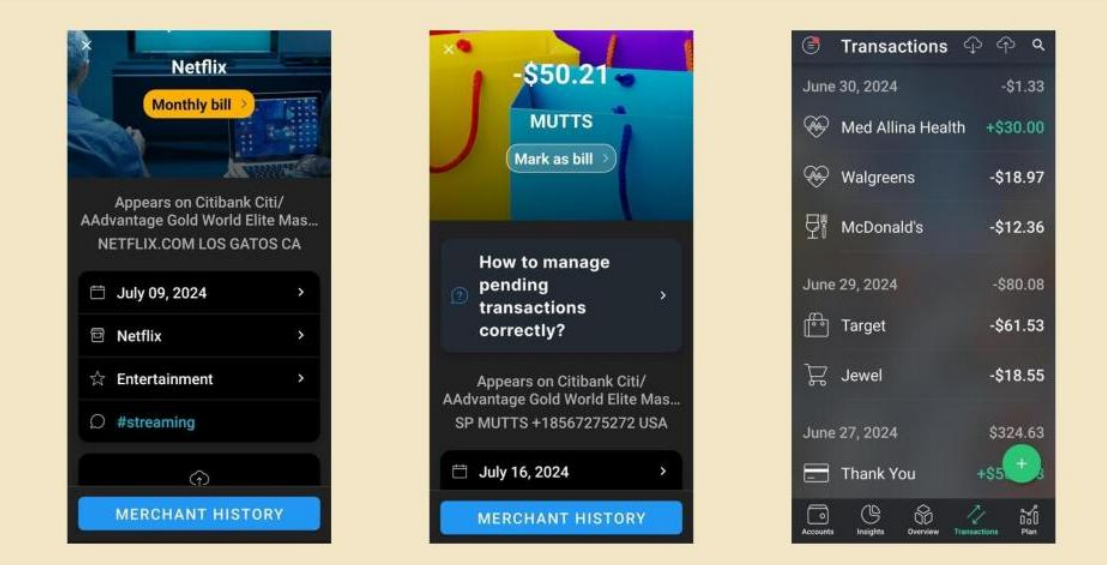
Moreover, there are various advantages that come with the use of YNAB. This application is more competitive in the market as it uses rule-based budgeting which in turn instills a sense of discipline in the use of funds and savings which if followed, can lead to a certain degree of financial security [2]. With this feature, there is no manual budgeting since the budgets are linked to the bank accounts and budgets can be managed with ease. On top of that, the use of multi budget options also combines an individual's family budget and a family budget to ensure that all the needs of the users have been catered for. YNAB users have expressed their admiration for the mobile budgeting app feature, especially when the need to manage the budget arises, which enables YNAB customers to perform budget-related functions while on the move. YNAB's elaborate approach towards its users certainly brings a lot of merits to the said app, for instance, teaching client's various solutions and methods of budget planning and management, which are also recommended for the people who seek financial education.

However, YNAB is also not immune to certain shortcomings. As The College Investor (2023) points out, a monthly or year subscription fee is one of the major cons that may hinder a section of users particularly given that users can at no cost access Mint among other platforms. Its price starts from $11.99 a month or $84 a year [2]. Those who do not want to be members of a subscription list, for instance, would find the feelings unattractive. Furthermore, YNAB does not provide tools for management of investment/bill/payment in the same way as one would expect if the user were to use the application as a complete service. Customer care via email may be a little bothersome to users who would expect to call and receive prompt solutions and avoid fluctuations. In the same way, there are no in-depth reporting tools available on YNAB and the reporting tools that are there are basic which is not meant for serious users with high expectations. Also, the YNAB budgeting's rules may be hard to follow for new users who are used to an advanced approach and more control.

2.1.3 Pocket Guard

Figure 2. 3 Pocket Guard Logo

Thirdly, the application that I have reviewed is Pocket Guard. Pocket Guard is a budgeting application whose primary purpose is to assist users in comprehending their financial behavior so that they can control their expenditures. The users can link different types of accounts, including credit cards and bank accounts, which are tracked automatically by the app to provide them with a comprehensive picture of their expenses [4]. There is no denying the fact that one of the interesting attributes of Pocket Guard which distinguishes it is its transaction classification feature, which sorts transactions on its own and thus addresses the user’s needs. As opposed to one of the competitors, for instance Rocket Money, embedded in the transactions, the Pocket Guard application does not enable its users to tag certain transactions as tax-deductible but does stick to the use of tags such as hashtags (#streamingservices, who #vacation) in the notes section to classify expenses that share a common purpose. And this is how sorting tags appear in the application. This feature is depicted by Figure 5 showed below which presents the Pocket Guard application interface where categorization of transactions and custom hashtags for grouping certain expenses were created. Also, the user’s capability to see long-term trends in spending in certain categories is improved by the presence of a separate mini-reporting feature within the Pocket Guard application that shows trends for hashtags only.



June 30, 2024	-$1.33
Med Allina Health	+$30.00
Walgreens	-$18.97
McDonald's	-$12.36
June 29, 2024	-$80.08
Target	-$61.53
Jewel	-$18.55
June 27, 2024	$324.63
Thank You	+$5

Figure 2. 4 Pocket Guard Managing Expenses

Pocket Guard’s automated categorization which also comes with transactional grouping with hashtags, automating expense management and organization of activities is one of its benefits. This allows for simpler and coherent budgeting regimes. Its integration with the user ’ s financial accounts and its active monitoring of transactions allow users to have a reliable and relevant view of their financial activities, which helps them make better choices. The other unique feature of bill generation forgets payment reminders hence avoiding missed bills penalties [4]. Remaining after bills due and necessary outgoings are subtracted, there is a nice feature in Pocket Guard which shows a balance that is available to the user.

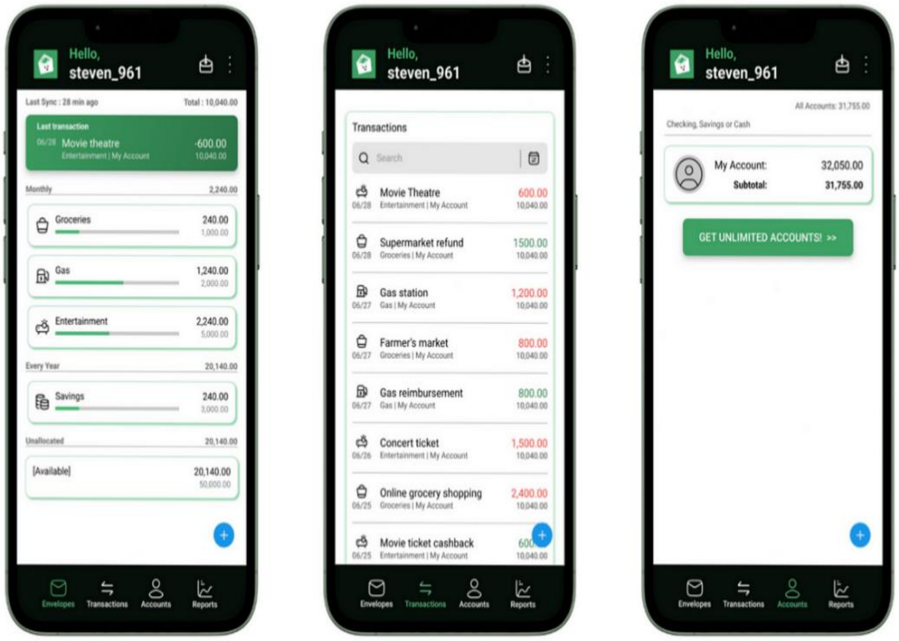
Nonetheless, there are some limitations to the use of PocketGuard. There are no tax tags for people who require tracking tax thoroughly [4]. The feature incorporates some default characteristics of automatic categorization which is good, but some elements of that category might need not be changing to those to fit all individuals. Moreover, the glitch of syncing did at times occur and could lead to fluctuations of real-time spending spans. Even if they are limitations, people looking for the tracking of their financial patterns and people with budgeting problems will find this application handy.

2.1.4 Good budget

Figure 2. 5 Goodbudget Logo

The following application to be reviewed is good budget, which i a budgeting application that relies on the envelope budgeting technique to assist users in sub sourcing their income and monitoring their expenditures in various categories. Good budget empowers users to construct various ‘envelopes’ that are intended for specific expenditures such as products, rent, and entertainment so that they can keep track of their expenditure and spend sparingly [6]. It also offers customers the ability to set their objectives financially, and to record their expenses and expenditures across devices which is a good application for a couple or family members who share management in finance. In addition, these reports of good budget give the users regarding their performed activities information about how their money was spent, assisting the users in wise spending comprehend the reason for this app.

Some of the features of good budget that stand out include the aesthetics and user experience of the envelope budgeting which is also efficient. The other fantastic feature of the application is that it enables companions and family members to share their budgets in real-time which enhances finances collaboration [7]. The minimalistic approach makes sure the boundaries are set on the screen as shown by Figures 9 It uses a pie chart representation to demonstrate the spending limits with the app displaying in real-time that the users’ balance is within the set thresholds.



Last Sync : 28 min ago Total : 10,040.00

Last transaction
06/28 Movie theatre -600.00
Entertainment | My Account 10,040.00

Monthly	2,240.00
Groceries	240.00 1,000.00
Gas	1,240.00 2,000.00
Entertainment	2,240.00 5,000.00
Every Year	20,140.00
Savings	240.00 3,000.00
Unallocated	20,140.00
[Available]	20,140.00 50,000.00

Transactions

Search

06/28	Movie Theatre Entertainment | My Account	600.00 10,040.00
06/28	Supermarket refund Groceries | My Account	1500.00 10,040.00
06/27	Gas station Gas | My Account	1,200.00 10,040.00
06/27	Farmer's market Groceries | My Account	800.00 10,040.00
06/27	Gas reimbursement Gas | My Account	800.00 10,040.00
06/26	Concert ticket Entertainment | My Account	1,500.00 10,040.00
06/25	Online grocery shopping Groceries | My Account	2,400.00 10,040.00
06/25	Movie ticket cashback Entertainment | My Account	600 10,040.00

All Accounts: 31,755.00

Checking, Savings or Cash

My Account: 32,050.00
Subtotal: 31,755.00

GET UNLIMITED ACCOUNTS! >>

Figure 2. 6 Goodbudget App Interface

The drawbacks of good budget application is transaction need input manually. Compared to other applications in the category of budgeting aids typically have an option of bank integration but good budget does not have this feature meaning users have to input transactions manually which is a more laborious task. Furthermore, Users of Good budget also lack features such as bill payment reminders and investment monitoring that some applications have. Some people like good budget for its envelope-based approach to budgeting, which is a valuable method to control one's finances but it's not a full service.

2.1.5 Monefy

Figure 2. 7 Monefy App Logo

The last application that I reviewed is Monefy. Monefy is an app that it developed for the control of personal finance and income among the users in an effective way. The main goal of this app is to split expenses which allow the user to keep track of their spending habits on different things like food, transportation or fun. Monefy allows users to add more than one account so they can easily keep their personal and work spending separated or handle a family account [5]. Another unique feature is that it allows a user to use more than one device with information saved on the cloud with companies such as Dropbox or Google Drive to make it usable with smartphones, laptops and many other devices [5]. Also, Monefy is capable of keeping multiple currencies and thus, can be useful for users from abroad or those that have to work with several currencies.

Monefy has a range of helpful elements which draws to the users. Firstly, the figure below has shown that the interface of the application. The application offers a simple design that is appealing to the eye and allows for easy tracking of how spending distribution takes place by way of a pie-chart interface. This design is obvious and simple to learn, even to those who have not used budget apps before. The general experience is nice due to the mix of cost groups that may be changed as well as the simple data interaction. The design of Monefy allows for usage of several devices, accounts and clouds so that an individual can view resources and funds at the beginning.

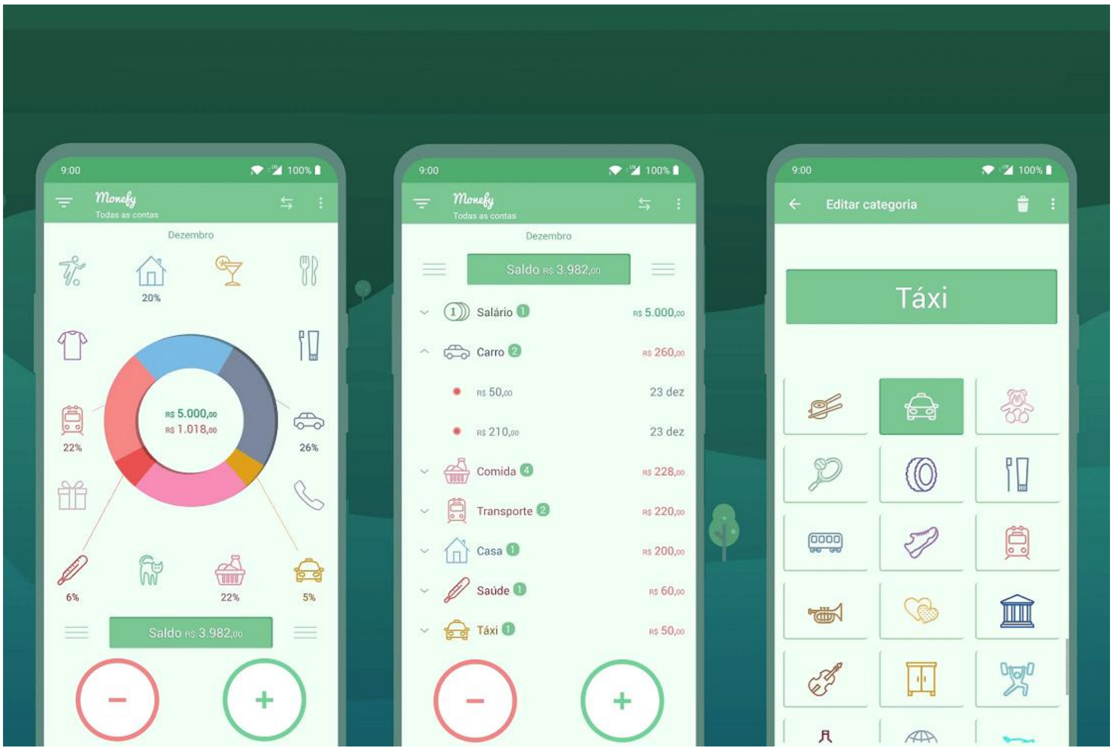


Figure 2. 8 Monefy App Interface

However, Monefy has some flaws as well. On its part, this app seems to fit in the easiest of planning needs such as division of costs, but it lacks some in-depth financial provisions including reporting which could have been useful for some end users. With respect to online transaction handling the program does not provide bank link meaning every transaction needs to be logged which is basic. Further, it does not send alerts for bills or for their management, features that are offered by a huge number of budget application making it to some extent less useful where bills are automatically controlled. In spite of these negative aspects, Monefy is still suggested for use especially by users who seek a simple solution that is visually appealing for spending management.

2.2 Comparison Table

Feature	Money +	Pocket Guard	YNAB	Good budget	Monefy	Proposed System
Registration and Login	✗	✗	✓	✓	✗	✓
Tracking Transactions	✓	✓	✓	✓	✓	✓
View Transactions in Visual Graph	✓	✓	✓	✓	✓	✓
Real-time currency Conversion	✗	✗	✗	✗	✗	✓
Bill Reminder	✓	✓	✗	✗	✗	✓
Set budget and financial goals	✓	✓	✓	✓	✗	✓
Calculation Tools	✗	✗	✗	✗	✗	✓
Categories customization	✓	✓	✓	✓	✓	✓
Search transactions	✓	✓	✓	✓	✓	✓
Hide transaction amount	✓	✗	✗	✗	✗	✓
Feedback	✓	✓	✓	✓	✓	✓

Table 2. 1 Finance App Comparison Table

Chapter 3 System Methodology/Approach OR System Model

3.1 Design Specification

3.1.1 Methodology

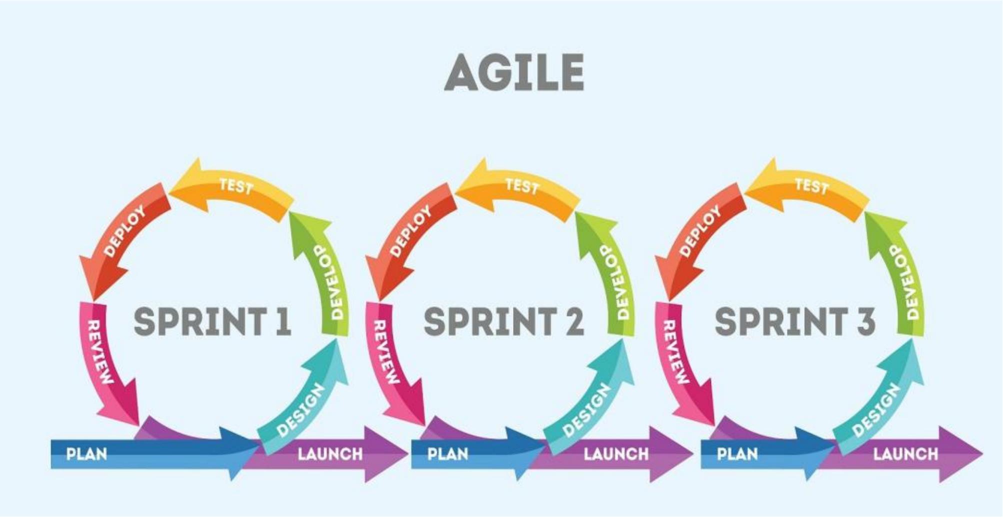


Figure 3. 1 Agile Development Methodology

This section compares three development methodologies which are Agile, Extreme Programming (XP), and prototyping to determine the most suitable approach for developing the Personal Finance Management Mobile Application. Each methodology offers distinct advantages and challenges for software development projects.

Firstly, Agile methodology employs an iterative development approach that breaks down the development process into smaller, manageable sprints, typically ranging from one to four weeks. This approach emphasizes continuous user feedback and adaptive planning, allowing development teams to respond quickly to changing requirements and user needs. One of Agile's key strengths is its flexibility, enabling teams to adjust project priorities and features based on real-time feedback and evolving user requirements. This methodology also maintains a balanced approach to documentation while prioritizing functional software delivery.

Besides, Extreme Programming (XP) represents a more technically intensive methodology that emphasizes practices such as pair programming and test-driven development. This approach focuses on technical excellence through continuous integration, frequent releases, and extensive code review processes. While XP ensures high code quality and robust testing, it typically requires a team of highly skilled developers working in close proximity. The methodology's intensive practices, including constant pair programming and continuous integration, can be particularly challenging for smaller teams or solo developers. Although XP offers significant benefits for code quality and testing, its resource-intensive nature and specific team requirements make it less suitable for projects with limited development resources.

Thirdly, Prototyping methodology involves creating an initial working model of the application to gather user feedback early in the development process. This approach helps visualize the end product and clarify requirements through tangible demonstrations. However, Prototyping often leads to longer development cycles and reduced flexibility once the initial prototype is established. The methodology requires extensive documentation and can result in significant rework if fundamental changes are needed after the prototype phase. While beneficial for understanding user requirements early, this approach may not support the dynamic nature of financial application development where features and requirements often evolve based on user feedback and market demands.

Based on the review above, Agile methodology proves most suitable for this project as it provides the optimal balance between flexibility, user feedback integration, and development efficiency. Its sprint-based approach aligns perfectly with the gradual implementation of features required for a financial management application, while its emphasis on regular feedback ensures the final product meets user needs effectively. The methodology's adaptability allows for continuous refinement of features based on user testing and changing requirements, which is crucial for developing a user-friendly financial management tool.

In this project, we will utilize the Agile methodology through a structured yet flexible development approach. The development process will be organized into two-week sprints, each focused on specific feature implementations. During FYP1, sprints will concentrate on establishing core functionalities including user authentication, transaction management, and data visualization. The FYP2 phase will focus on implementing advanced features such as budget management, financial goal tracking, and currency conversion tools. Each sprint will follow a consistent cycle of planning, development, testing, and review, ensuring regular progress while maintaining flexibility for improvements. User feedback will be continuously integrated throughout the development process, allowing for feature refinement and optimization. This approach ensures systematic development while maintaining the ability to adapt to changing requirements and user needs, ultimately leading to a more robust and user-centered financial management application.

3.1.2 Tools and Technologies Involved

Category	Tool	Description
Interface Design Software	Android Studio	To create a consistent UI for the budgeting app on Android
Database	Firebase Firestore	To manage and sync user financial data in real-time with offline support
UI/UX Design	Figma	To design user friendly interface

Table 3. 1 Tools and Technologies Table

3.2 User Requirement

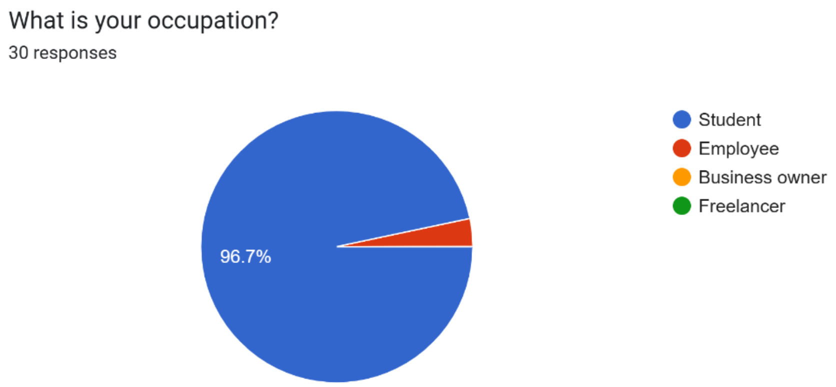


Figure 3. 2 Occupation Distribution among Respondents

Figure 3.2 shows that most of the respondents, 96.7%, are students, with a small percentage being employees. No response was given to freelancers or entrepreneurs. From this, it shows that the primary target audience for the application is students.

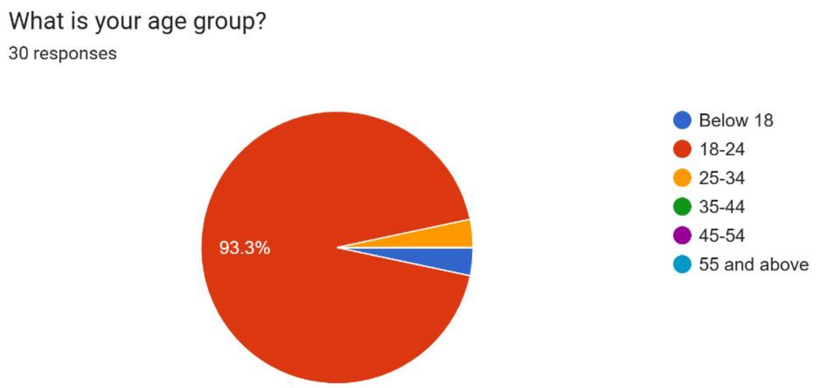


Figure 3. 3 Age Group Distribution among Respondents

In figure 3.3 indicates the age profile, with 93.3% of the sample in the 18–24 age bracket. There is an extremely low figure in the below 18 and 25–34 brackets, and no participants aged over 35. This concurs with the occupation statistics and confirms the young target market demographic.

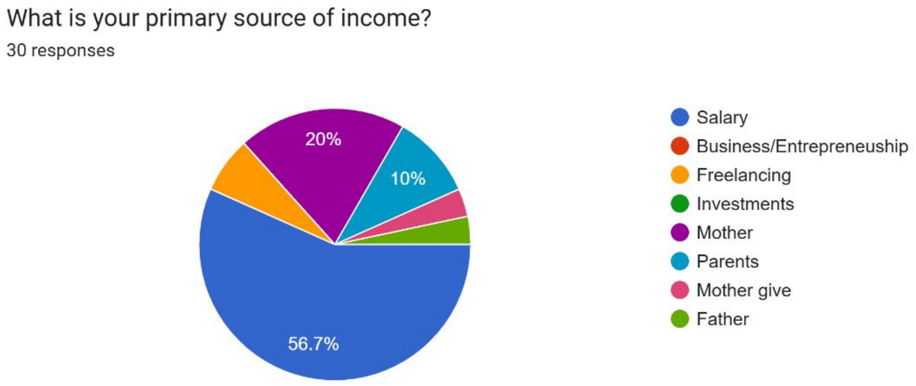


Figure 3. 4 Source of Income Distribution among Respondents

Figure 3.4 illustrates the primary sources of income for respondents. The majority (56.7%) earn a salary, followed by mother (20%) and father (10%) support. The remaining sources include freelancing, business/entrepreneurship, investments, and general references like "mother give" and "father." This therefore indicates that although there are financially independent students, the majority are still financially reliant on their families.

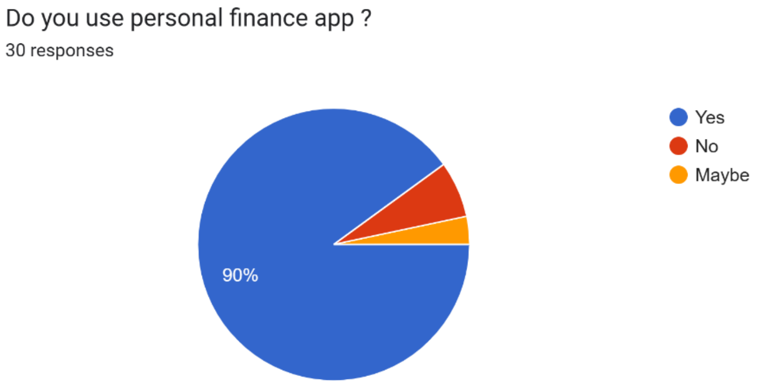


Figure 3. 5 Most used personal finance apps among respondents

Figure 3.5 demonstrates that 90% of respondents already use a personal finance app, evidence of a significant existing interest in online financial aid. Few respondents answered "No" or

"Maybe," indicative of potential further involvement through enhanced features or targeted app design.

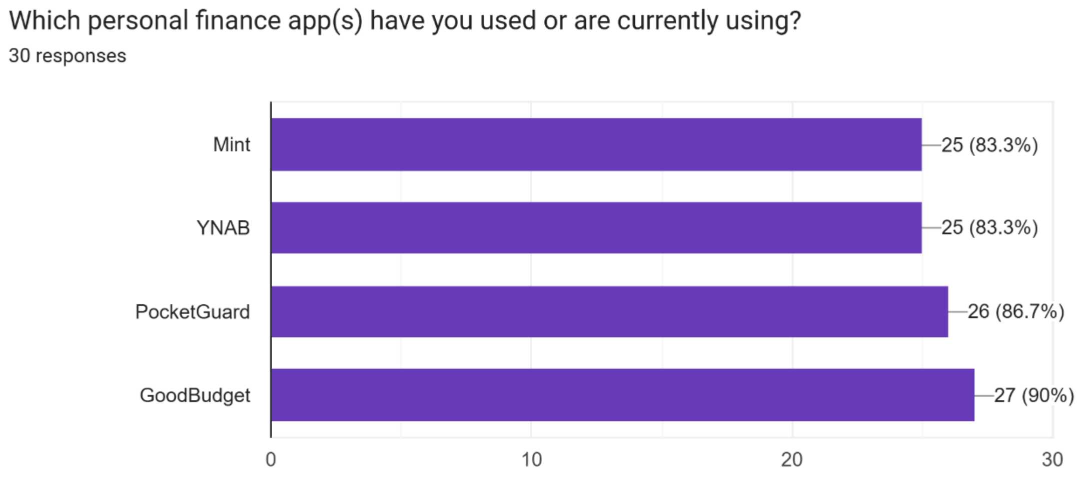


Figure 3. 6 Types of Personal Finance apps the respondents

Figure 3.6 displays the types of personal finance apps the respondents have used or are using. The most used app is GoodBudget, at 90% of the respondents, followed closely by PocketGuard (86.7%), Mint (83.3%), and YNAB (83.3%). The finding indicates that users are exposed to a variety of financial applications, with a slight inclination towards those that specialize in budgeting and expense tracking. The popularity suggests that these applications offer functionality that is perceived as beneficial or convenient to the target group.

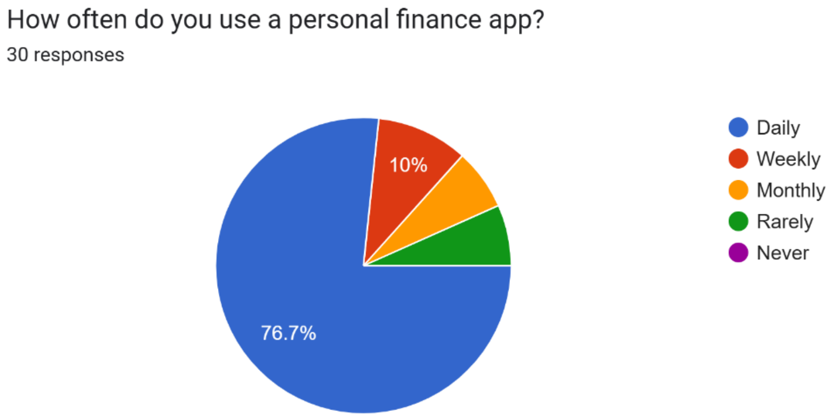


Figure 3. 7 How Often Do Users Use a Personal Finance App

Figure 3.7 shows how often users utilize personal finance applications. A whopping 76.7% use it every day, while the rest split among weekly (10%), monthly, and seldom. Nobody selected "Never," which corroborates the foregoing data of high adoption. This highlighted the requirement for a personal finance application that is secure, fast, and easy to use for everyday checking and management of finances.

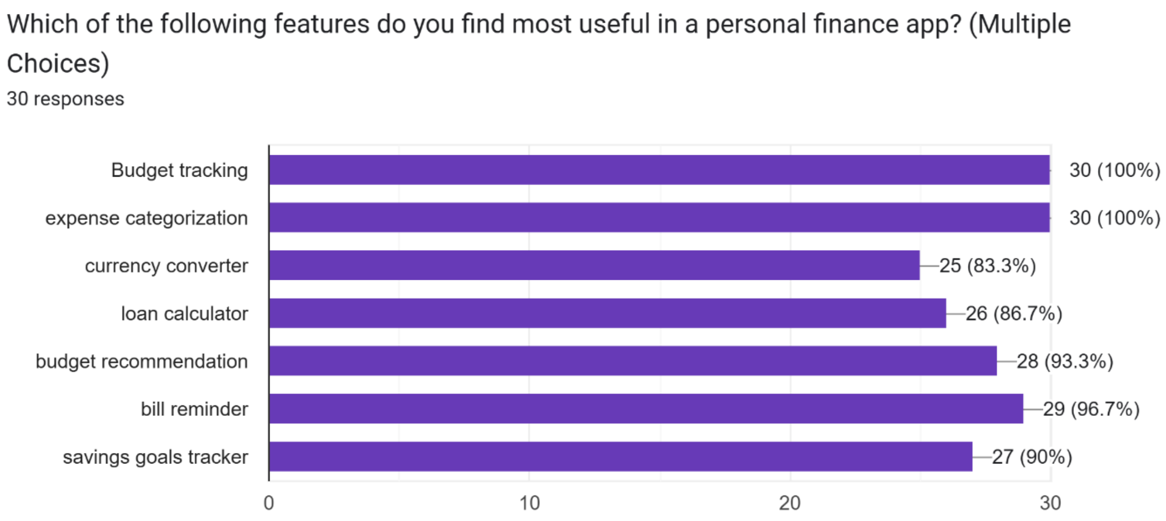


Figure 3. 8 Most valued features in a personal finance app

Figure 3.8 indicates the preferred features for the users in a personal finance application. Budget tracking and expense categorization are considered essential by 100% of the users. Bill reminders (96.7%), savings goals tracker (90%), budget recommendation (93.3%), and loan calculator (86.7%) are the other preferred features. It is strongly evident from this finding that the users require comprehensive tools through which not only can they track but also plan and optimize their financial activities. Those features such as a currency converter (83.3%) are also provided points to user demand by those who may deal with more than a single currency, possibly due to foreign purchases or studies.

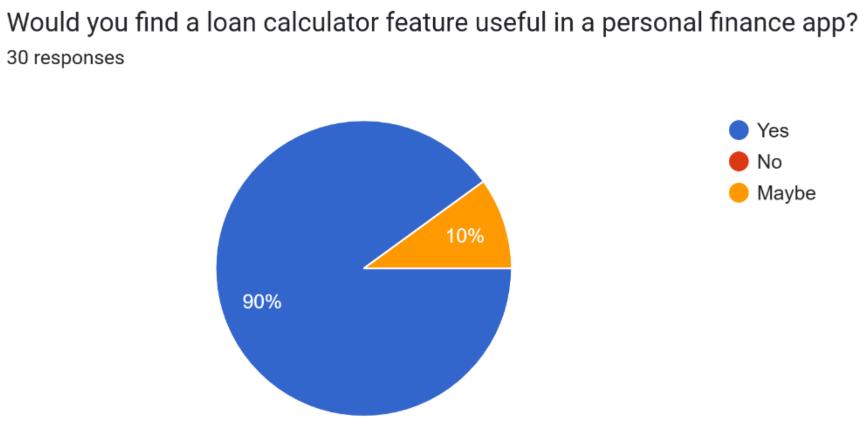


Figure 3. 9 User opinion on the usefulness of a loan calculator

Figure 3.9 verifies the previous result, with 90% of the sample saying that they would find a loan calculator feature useful in a personal finance application. This suggests that there is high demand for those facilities that allow users to practice and discover about loans, which could be personal loans or other loans in the future. It also provides support for the inclusion of planning-oriented features in the application, as opposed to tracking features.

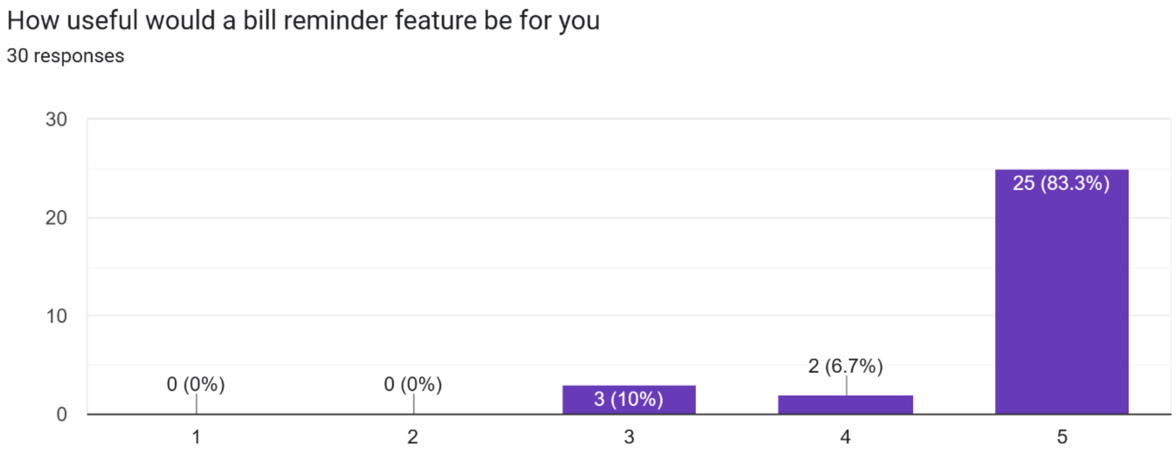


Figure 3. 10 Importance of Bill Reminder Feature

In the figure 3.10 indicates how useful respondents find a bill reminder function within a personal finance application. An enormous majority of users (83.3%) rated it “Very useful", while 6.7% chose 4 and 10% chose 3. Nobody of the interviewees rated it less than 3. This indicates enormous demand for reminder functionality, which is likely due to users needing help with keeping up regular payments and avoiding late fees especially the primarily among students or young users that are still figuring out consistent spending patterns.

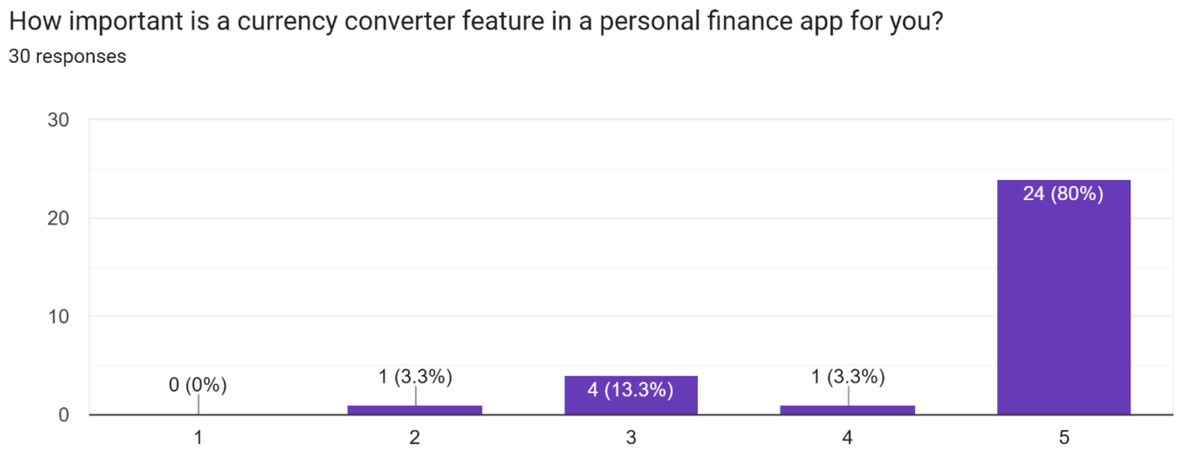
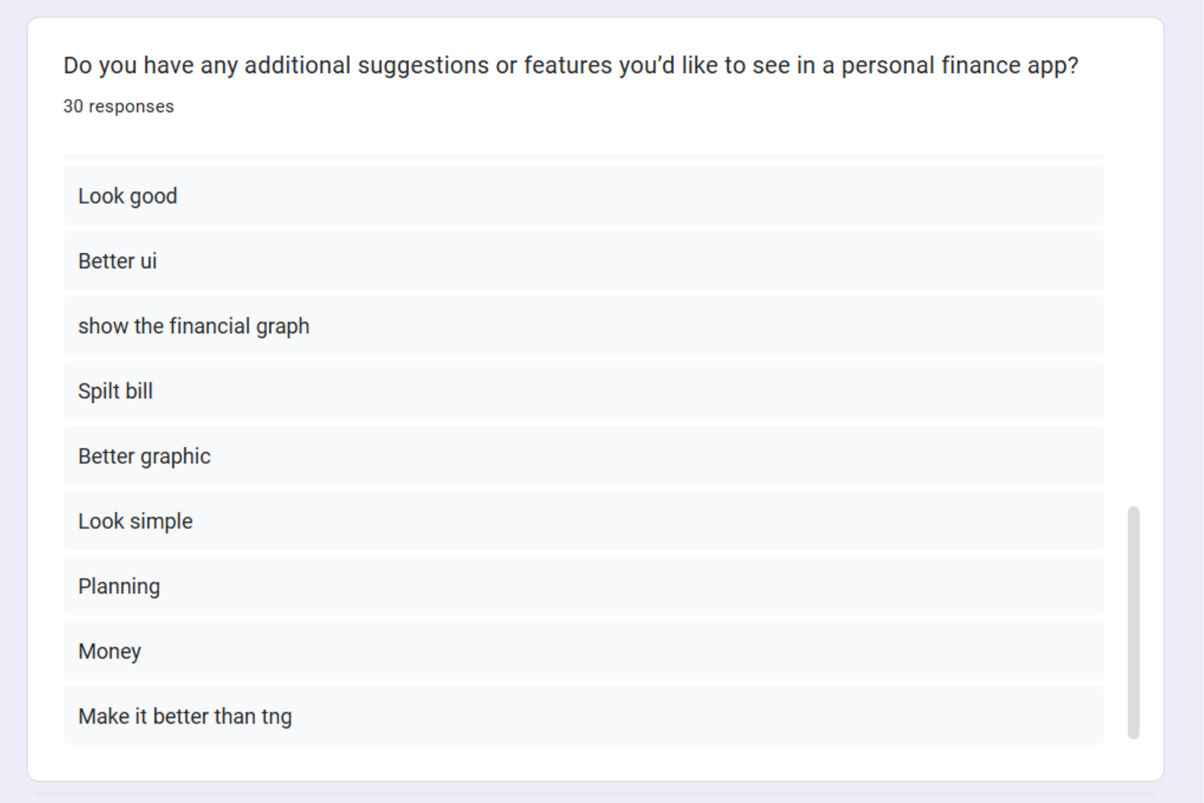


Figure 3. 11 Importance of Currency Converter Feature

Figure 3.11 indicates the importance of a currency converter function. Here, 80% of the respondents gave it an extremely important rating (rating score 5), and very few gave it ratings of middle scores 3 and 4. Only one gave it a rating of 2, and none gave it a rating of 1. This suggests that although the currency converter is critic-proof, nevertheless, it is a very much valued function, especially in the context of those users handling multiple currencies as it is possibly due to international studies, travel, or e-commerce.



Do you have any additional suggestions or features you'd like to see in a personal finance app?

30 responses

Look good

Better ui

show the financial graph

Spilt bill

Better graphic

Look simple

Planning

Money

Make it better than tng

Figure 3. 12 Additional Suggestions of Features from User

Figure 3.12 illustrates open-ended feedback from the respondents regarding other ideas and features they would like to have in a personal finance app. The themes are more UI/UX (Better UI, look simple, better graphic), visualization of data (Show the financial graph), planning features (Planning), and sharing or social (Split bill). Others also emphasized that the app should compete with existing apps like Touch 'n Go (TNG), such as in the suggestion "Make it better than TNG." These responses point to the requirement not only to have functional features but also for the app to be good looking, user-friendly, and updated.

3.3 Functional and non-functional

3.3.1 Functional Requirements

The system shall provide the following functions:

- **User Authentication**: Allow users to register, log in, log out, and reset passwords securely.
- **Transaction Management**: Enable users to add, edit, delete, and categorize income and expense transactions.
- **Budget Management**: Let users set budgets for specific categories and receive alerts when limits are exceeded.
- **Goal Tracking**: Support users in setting and monitoring financial goals, including savings progress.
- **Financial Calculators**: Provide tools like a loan calculator and real-time currency converter for planning and decision-making.
- **Data Visualization**: Display user financial summaries via graphs and pie charts.
- **Notifications**: Send push notifications for bill due dates, and daily reminders.
- **Feedback System**: Allow users to submit app feedback and issues.

3.3.2 Non-Functional Requirements

The system must satisfy the following conditions:

- **Security**: User data must be securely stored and protected using Firebase Authentication and Fire store rules.
- **Availability**: The app should be accessible on Android devices with real-time sync and offline support.
- **Performance**: Ensure smooth app performance with fast data retrieval and minimal loading times.
- **Usability**: User interface must be simple and intuitive for users of all financial knowledge levels.

- **Scalability**: The system should accommodate an increasing number of users and data without degradation.
- **Maintainability**: Codebase should be modular and well-documented to ease future updates and feature additions.

3.4 Gantt chart

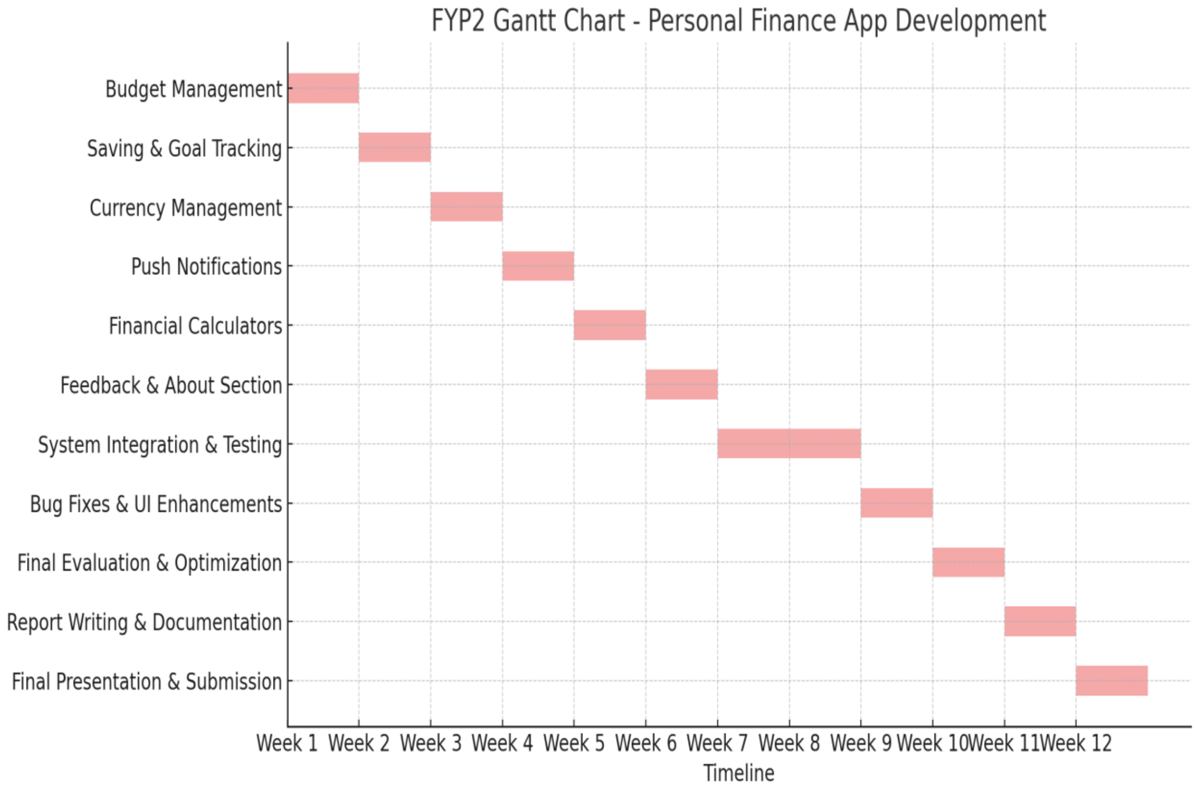


Figure 3. 13 Gantt Chart

CHAPTER 4 SYSTEM DESIGN

4.1 System Block Diagram

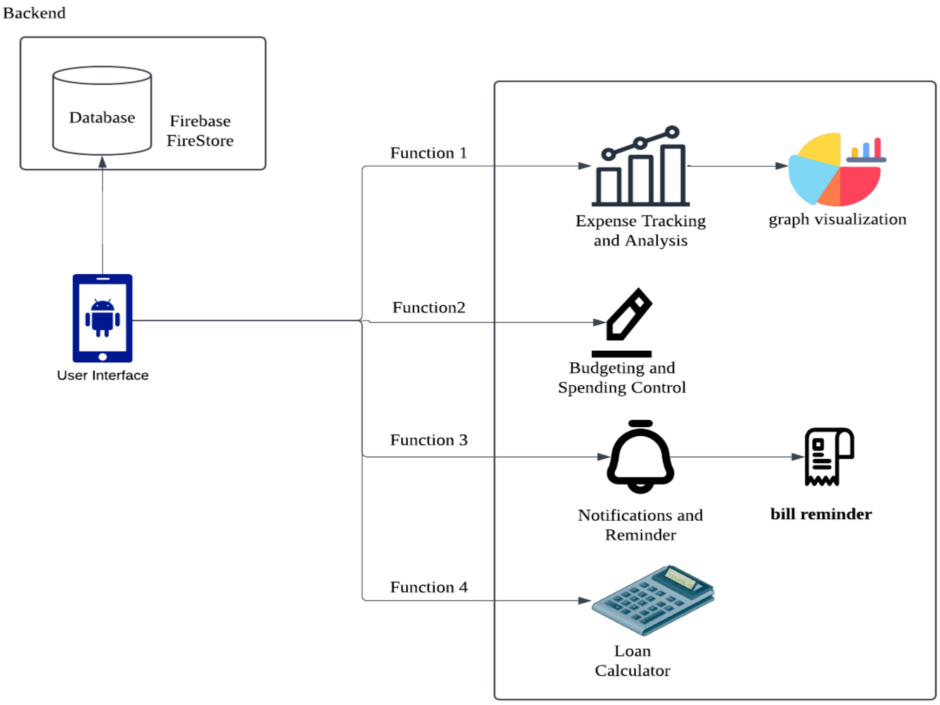


Figure 4. 1 Block Diagram

The block diagram illustrates the architecture of a budgeting application, demonstrating how its various components interact to provide users with essential financial management features. The system is divided into three main components: the backend, the user interface, and the application's functional modules. The backend, comprising a database and Firebase Firestore, serves as the central repository for storing and managing user data, including transaction records, budget limits, reminders, and loan calculations. Firebase Firestore ensures real-time synchronization between the database and the application, enabling seamless data updates and retrieval.

The user interface acts as the intermediary between users and the backend, providing an intuitive platform for users to interact with the application. Through the user interface, users can access four key functions: expense tracking and analysis, budgeting and spending control, notifications and reminders, and a loan calculator. Each of these functions is designed to address specific financial needs. Expense tracking and analysis, for instance, allows users to record their spending and visualize it through graphical representations, such as bar charts and pie graphs, to better understand their financial habits.

The budgeting and spending control module enables users to set spending limits and manage their finances efficiently, while the notifications and reminders module ensures timely alerts for upcoming bills or budget limits to prevent missed payments. Finally, the loan calculator assists users in estimating loan repayments, providing details like monthly installments and amortization schedules. This interconnected design ensures a user-friendly experience while maintaining robust backend functionality to support real-time operations and accurate financial data management.

4.2 System Overview/Design

4.2.1 User Stories

This section will present the user stories that define the functional requirements of the application. Each user story should describe a specific feature the app must provide to meet the user's needs.

1. User Account Creation and Authentication
   - As a user, I want to create an account so that I can securely store and track my financial information.
   - As a user, I want to log in using my email or Google account so that I can easily access my account from different devices.
   - As a user, I want to reset my password in case I forget it, ensuring I can always access my account.
2. Budget Management

- As a user, I want to set a monthly budget so that I can track my spending and avoid overspending.
- As a user, I want to set different budgets for different categories (e.g., groceries, entertainment) so that I can manage my finances more effectively.
- As a user, I want to be notified when I exceed my budget so that I can adjust my spending.

3. Expense Tracking
    - As a user, I want to log my daily expenses so that I can monitor where my money is going.
    - As a user, I want to categorize my expenses (e.g., food, transportation, bills) so that I can see which areas I am spending the most on.
    - As a user, I want to add recurring expenses (e.g., rent, subscription services) so that I don't have to enter them manually each month.
4. Income Tracking
    - As a user, I want to record my sources of income (e.g., salary, freelance work) so that I can track my total earnings.
    - As a user, I want to set income goals (e.g., saving a certain amount each month) so that I can monitor my financial progress.
5. Financial Goals
    - As a user, I want to set financial goals (e.g., saving for a vacation, building an emergency fund) so that I can track my progress over time.
    - As a user, I want to see how much I have saved toward my goals so that I know how close I am to achieving them.
6. Reports and Insights
    - As a user, I want to see a summary of my spending patterns
7. Currency Management
    - As a user, I want to track my expenses in multiple currencies so that I can manage finances across different countries.
8. Notifications and Alerts

- As a user, I want to receive push notifications when I approach or exceed my budget so that I can adjust in real-time.
- As a user, I want to set a daily reminder so that I can have a good habits to always remember track my transactions.
- As a user, I want to be reminded of upcoming bills and payments so that I never miss a due date.

4.2.2 FYP1 Sprint Goals

Sprint	Sprint Goal	User Stories	Story Points
Sprint 1 (Week1)	Set up User Authentication (Sign up, log in, Log out)	As a user, I want to create an account" "As a user, I want to log in using my email/Google" "As a user, I want to reset my password"	10
Sprint 2 (Week2)	Implement Transaction Tracking	"As a user, I want to log my daily expenses" "As a user, I want to categorize my expenses"	8
Sprint 3 (Week3)	Implement Expense Recurring and Expense History	"As a user, I want to add recurring expenses" "As a user, I want to see my expense history"	8
Sprint 4 (Week4)	Develop Graph Visualization	"As a user, I want to see a summary of my spending patterns"	8
Sprint 5 (Week5)	Finalize Transaction Tracking and Graph Visualization	"As a user, I want to receive financial insights on my spending" "As a user, I want to generate reports"	10
Sprint 6 (Week6)	Implement Account Management	"As a user, I want to add and manage different accounts" "As a user, I want to see my account balances"	6

Table 4. 1 FYP1 Sprint Goals Table

4.2.3 FYP2 Sprint Goals

Sprint	Sprint Goal	User Stories	Story Points
Sprint 7	Implement Budget Management	"As a user, I want to set a monthly budget so I can track my spending."	10
Sprint 8	Implement Saving and Financial Goal Tracking	As a user, I want to set financial goals "As a user, I want to track my progress toward my financial goals."	12
Sprint 8	Develop Currency Management	"As a user, I want to track my expenses in multiple currencies so that I can manage finances across different countries.	10
Sprint 9	Implement Push Notifications for Budget Alerts	"As a user, I want to receive notifications when I exceed my budget." As a user, I want to receive bill reminder to remind me to pay bill.	8
Sprint 10	Add Calculator for Budgeting and Expense Tracking	"As a user, I want to calculate my monthly interest to help me compare with the interest.	8
Sprint 11	Feedback System & About Section	"As a user, I want to provide feedback on the app so I can suggest improvements." "As a user, I want to view the About section to understand the app's features and goals."	8

Table 4. 2 FYP2 Sprint Goals Table

4.3 Use Case Diagram

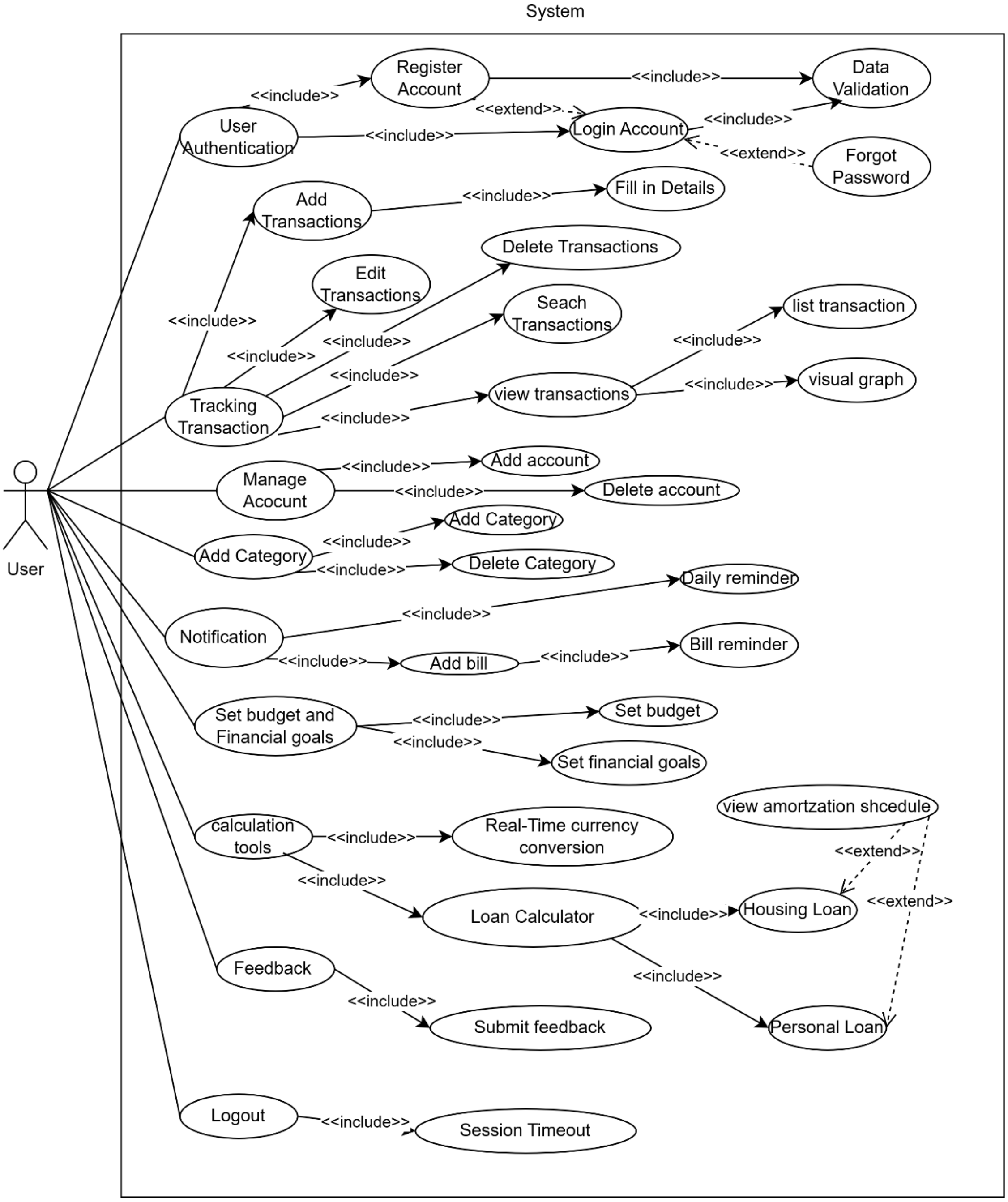


Figure 4. 2 Use Case Diagram

4.4 Activity Diagram

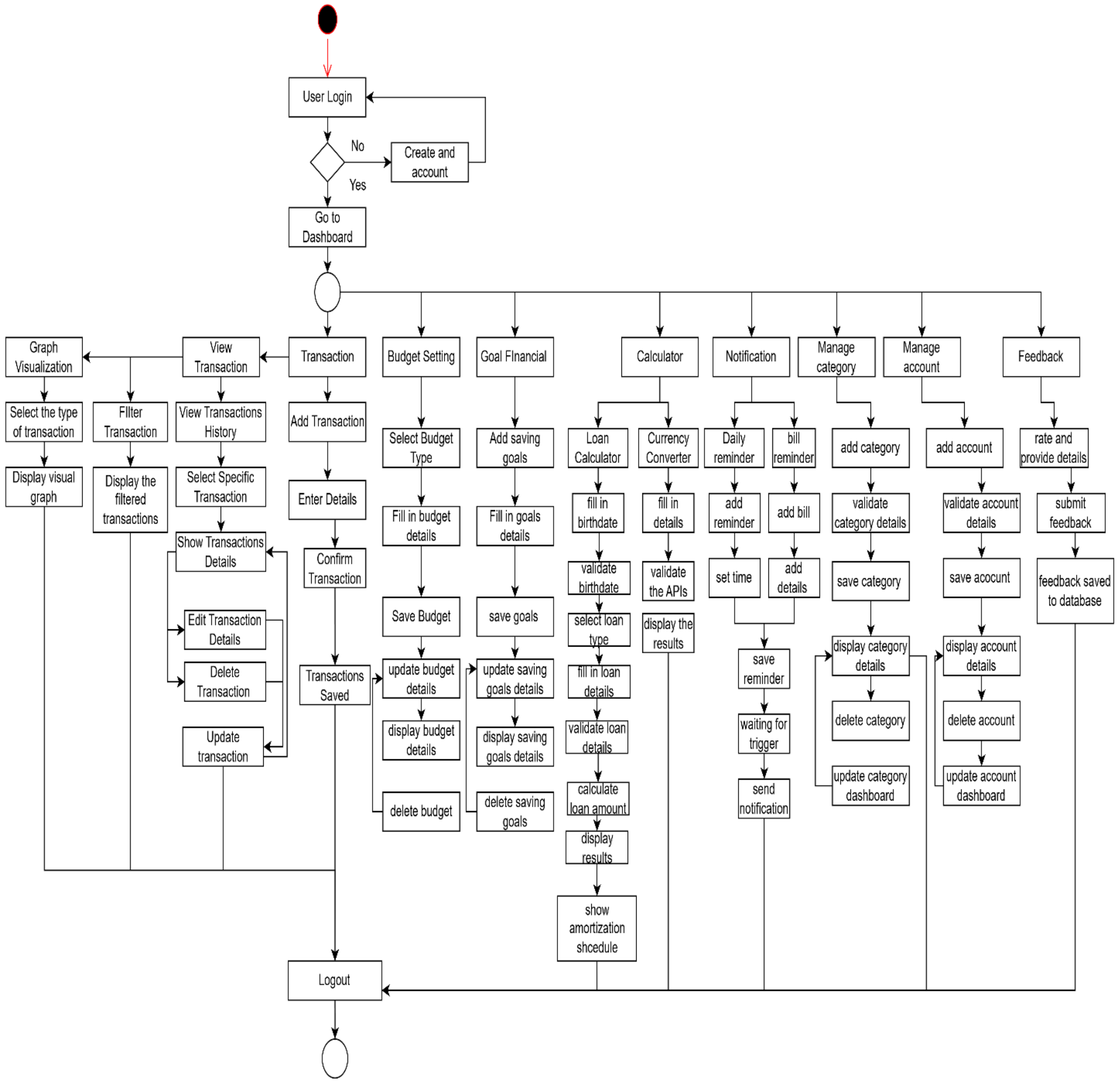


Figure 4. 3 Activity Dagram

4.5 Login Module

4.5.1 Login – Use Case Description

Use Case	Login to an account
Purpose	To allow users to authenticate and access their account
Actor	User
Normal Flow of Event	1) User enters email or username and password correctly 2) System will check the validity of entered user email and password 3) If successfully verified the user information, the system will redirect it into the main page.
Alternative Flow	1) If the user enters invalid credentials, the system will display an error message 2) If user clicks “Forgot Password”, a popup box appears prompting the user to enter their email 4) The system will send a password recovery email with a reset link.

Table 4. 3 Login -Use Case Description

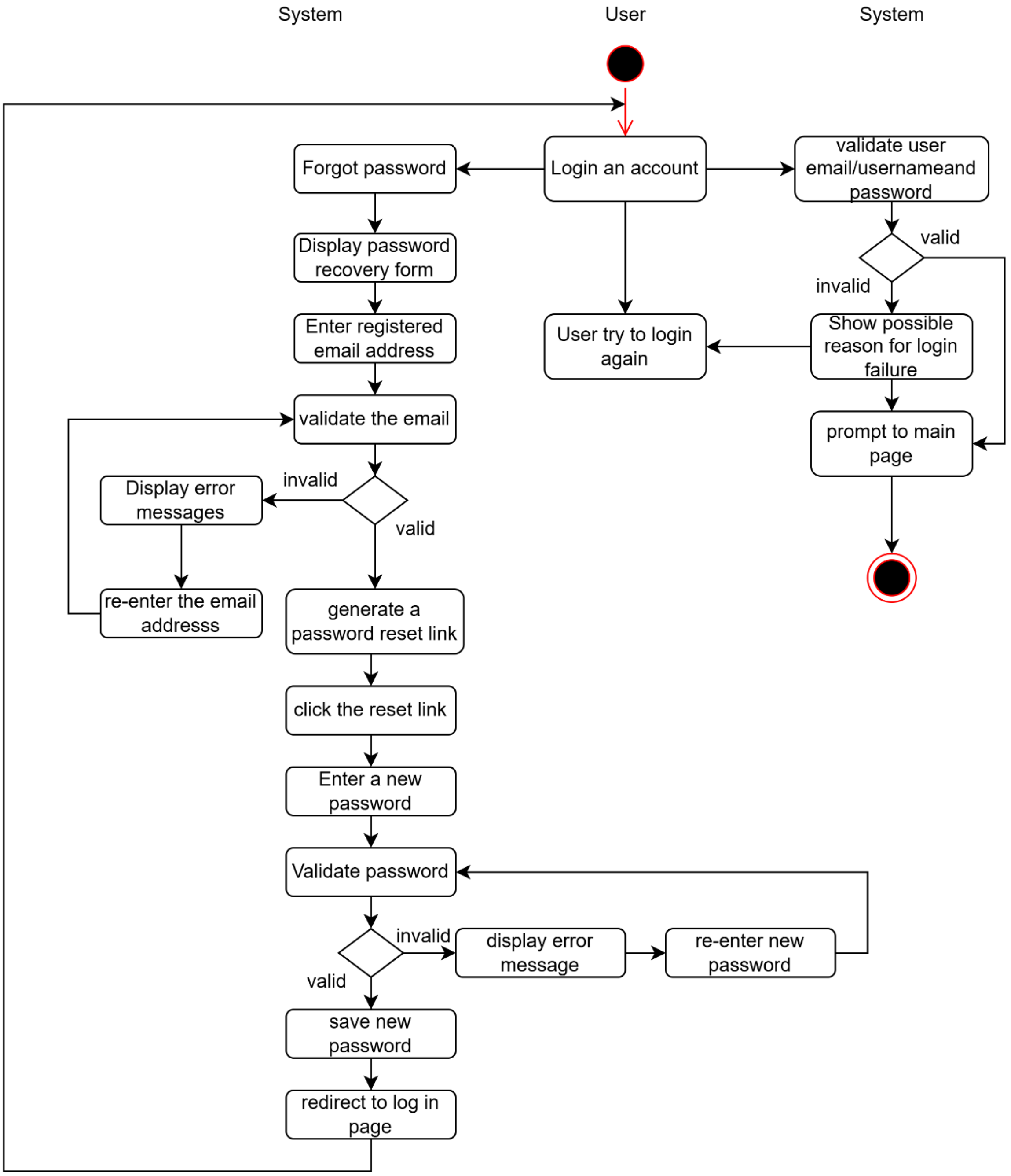


Figure 4. 4 Login - Activity Diagram

The activity diagram illustrates the process flow for both user login and password recovery within the system. It begins with the user attempting to log in by entering their email or username and password. The system then validates these credentials. If the credentials are correct, the user is granted access and redirected to the main page. However, if the login information is invalid, the system displays the reason for failure and prompts the user to try logging in again. This cycle continues until the user successfully logs in or chooses the "Forgot Password" option.

In the event that the user cannot remember their password, they can initiate the password recovery process by selecting the "Forgot Password" option. The system then displays a password recovery form, prompting the user to enter their registered email address. The email is validated by the system—if it's invalid, an error message is shown and the user is asked to re-enter a correct email. If the email is valid, the system generates a password reset link and sends it to the provided email. Upon clicking the reset link, the user is directed to enter a new password. The system then validates this new password. If the password does not meet requirements, an error is displayed, and the user is prompted to re-enter a valid password. Once a valid password is entered, it is saved in the system, and the user is redirected to the login page to complete the process. This flow ensures both secure access and user-friendly recovery options in case of forgotten credentials.

4.6 Register Module

4.6.1 Register – Use Case Description

Use Case	Register for an Account
Purpose	To allow new users to create an account.
Actor	User
Normal Flow of Event	1. When user clicks the "register" button, the system will display a registration form. 2. Users must enter their information such as email, username, and password. 3. The system will validate the input data. 4. If valid, the system creates the account and displays a successful message.
Alternative Flow	1. If the email is already registered, the system will display an error message. 2. If the password does not meet the requirement, the system will display an error message.

Table 4. 4 Register - Use Case Description

4.6.2 Register – Activity Diagram

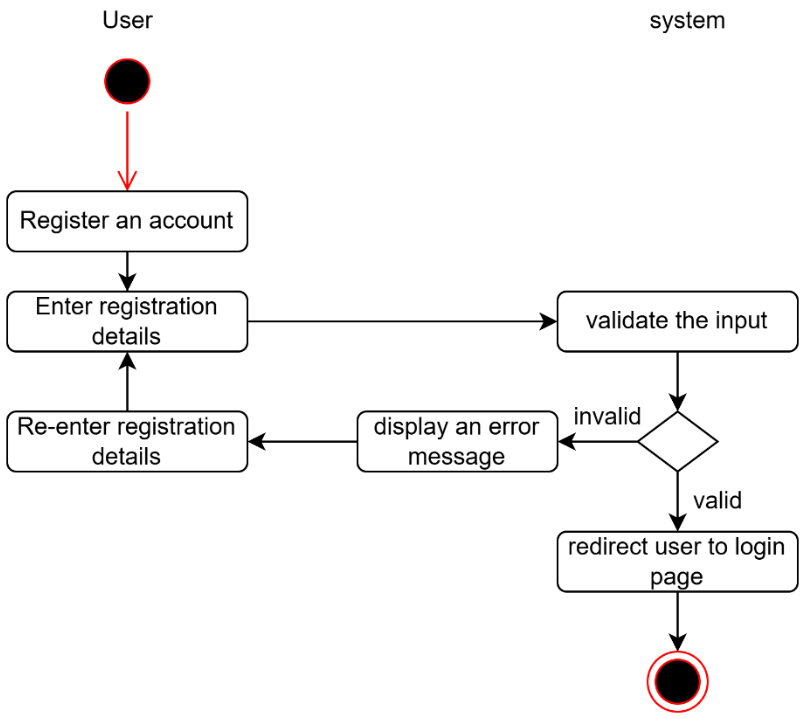


Figure 4. 5 Register - Activity Diagram

The activity diagram outlines the process of user registration within the system. It begins with the user selecting the "Register" option and entering required details such as name, email, password, and contact information. The system then validates the input to ensure that all mandatory fields are completed, and the provided information meets predefined criteria, such as a valid email format or strong password requirement.

If any input is invalid, the system displays an error message (e.g., "Invalid email format" or "Password must contain at least 8 characters"), prompting the user to re-enter the details. This loop allows the user to correct errors and reattempt registration without restarting the entire process. Once the input is validated successfully, the system stores the user's information in the database and redirects it to the login page, completing the registration process.

4.7 Forgot Password Module

4.7.1 Forgot Password – Use Case Description

Use Case	Forgot Password
Purpose	To allow users to reset their password
Actor	User
Normal Flow of Event	1. User clicks the "Forgot password" link 2. The system will display a form to enter the registered email. 3. Users submit the form, and the system will validate email. 4. If valid, the system sends a password recovery email with a reset link.
Alternative Flow	1. If the entered emails is not registered, the system will display an error message "Email not found"

Table 4. 5 Forgot Password -Use Case Description Table

4.7.2 Forgot Password – Activity Diagram

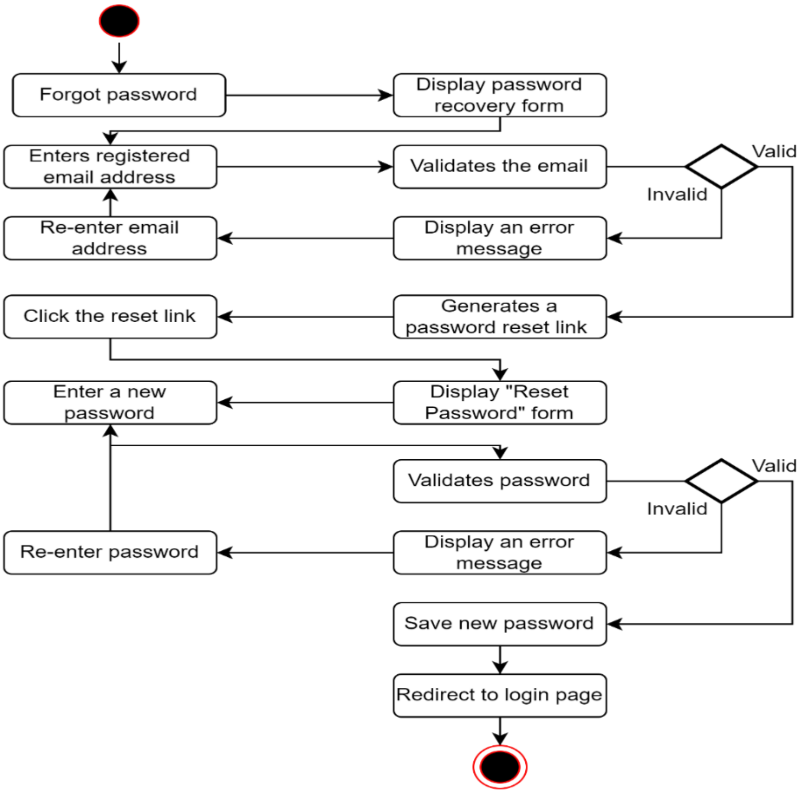


Figure 4. 6 Forgot Password - Activity Diagram

The activity diagram for the "Forgot Password" module illustrates the process of recovering a user's account credentials. The user begins by selecting the "Forgot Password" option and entering their registered email address. The system validates the email provided to confirm it exists in the database. If the email is invalid or not registered, the system displays an error message (e.g., "Email not found"), allowing the user to re-enter a valid email.

For a valid email, the system generates a password reset link and sends it to the user via email. The user clicks the reset link, which redirects them to a password reset form where they can input a new password. The system validates the new password for strength and re-entry confirmation. Upon successful validation, the password is updated in the database, and the user is redirected to the login page to access their account with the new credentials.

4.8 Add Transactions Module

4.8.1 Add transaction – Use Case Description

Use Case	Add transactions
Purpose	To allow users to manually record their income or expenses
Actor	User
Normal Flow of Event	1. When user click the add transaction button, the system will display the form to input the transaction details such as amount, category, date and description. 2. Users enter transaction details, and optionally attaches a photo of the receipts and the location for the transaction. 3. When user click save, the system validates the input data and saves the transaction. 4. The success message is displayed.
Alternative Flow	1. If any input is invalid, the system will display an error message, and the user need to reenter the details again.

Table 4. 6 Add transaction – Use Case Description

4.8.2 Add Transaction – Activity Diagram

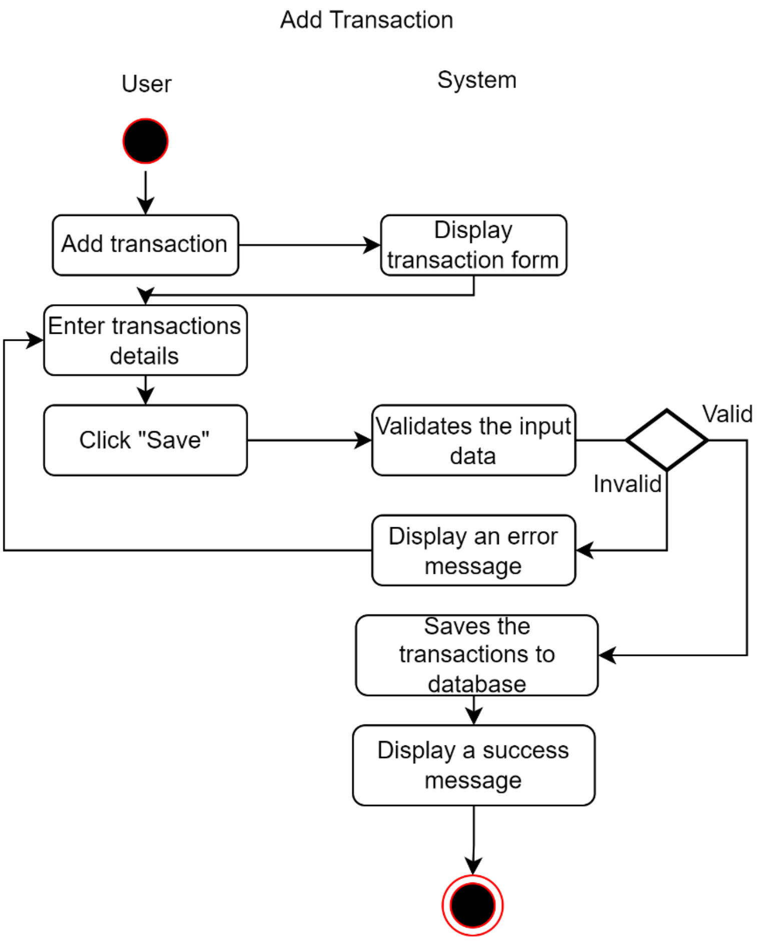


Figure 4. 7 Add Transaction – Activity Diagram

The activity diagram details the process of adding a transaction to the system. The user initiates the process by selecting the "Add Transaction" option, which prompts a form where they input transaction details such as amount, category, date, and description. The system validates the input to ensure that all fields are correctly filled and comply with the expected format.

If the input is invalid (e.g., negative amounts or missing fields), the system displays an error message, prompting the user to re-enter the details. Once all inputs are successfully validated, the system saves the transaction to the database and displays a success message, indicating that the transaction has been recorded. This process ensures accurate data entry while providing users with immediate feedback on their actions.

4.9 View Transactions Module

4.9.1 View Transaction – Use Case Description

Use Case	View transactions
Purpose	To allow users to view and analyze their recorded transactions
Actor	User
Normal Flow of Event	1. Users select the “transactions” menu option, the system will display a list of transactions. 2. Users clicks on a specific transaction to view its details; the system will display the transaction details.
Alternative Flow	1. If the users select the “Report” menu, the system will display a pea chart for summarizing the user’s income or expense. 2. If the user selects.

Table 4. 7 View Transaction – Use Case Description

4.9.2 View Transactions – Activity Diagram

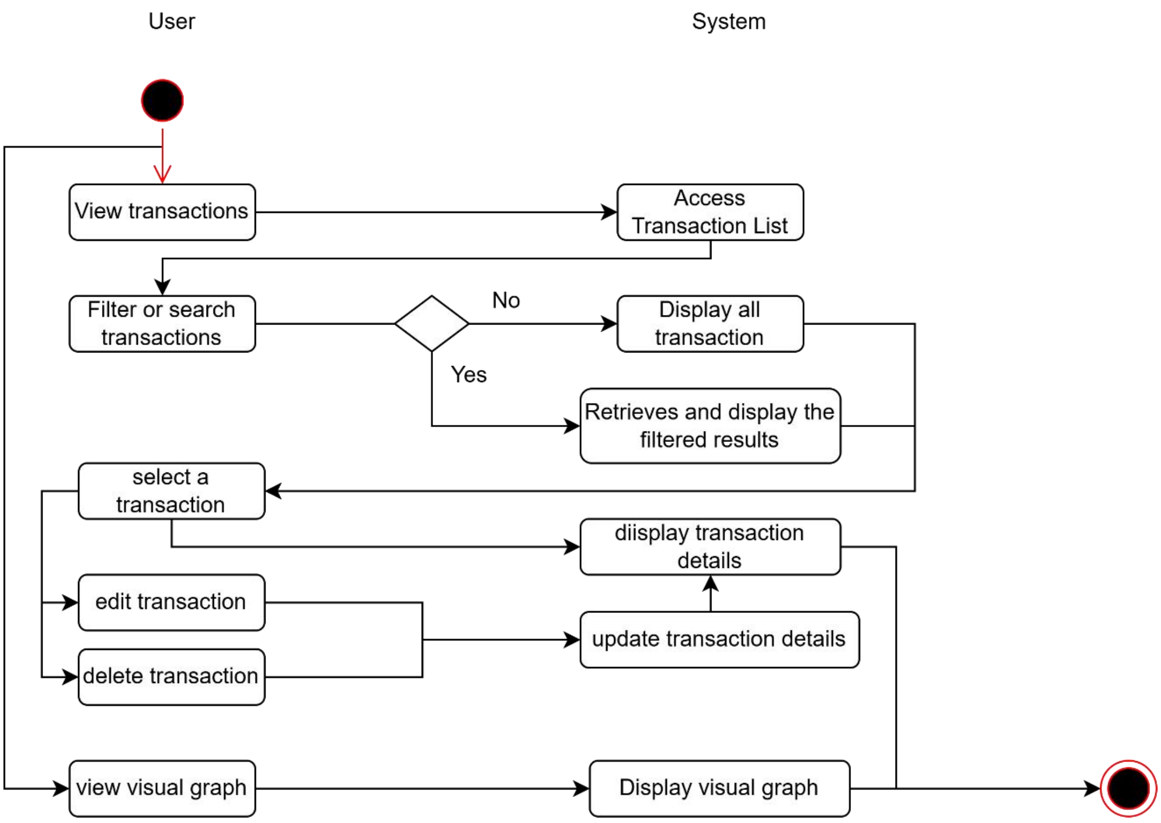


Figure 4. 8 View Transactions – Activity Diagram

This activity diagram illustrates the process of viewing, filtering, editing, deleting, and visualizing transactions within the personal finance system. The process begins when the user selects the option to view transactions. The system responds by accessing the complete transaction list. The user is then given the option to apply filters or perform a search to narrow down the results. If no filters are applied, the system simply displays all available transactions. However, if the user specifies search criteria, the system retrieves and displays only the filtered results that match the given parameters.

Once the transactions are displayed, the user can select a specific transaction to view its detailed information. After viewing, the user has the option to either edit or delete the selected transaction. If editing is chosen, the system allows the user to update the transaction details and saves the changes accordingly. If deletion is chosen, the transaction is removed from the system. Additionally, the user can choose to view a visual graph representing transaction trends or summaries, and the system will generate and display the corresponding graphical data. This flow ensures efficient transaction management, allowing users to not only review but also modify and understand their financial activity visually.

4.10 Notification Module

4.10.1 Notification – Use Case Description

Use Case	Notification
Purpose	To notify users about important financial activities and reminders
Actor	User
Normal Flow of Event	1. User sets notification for bill reminder, daily reminder, or budget limit alerts. 2. The system schedules the notification based on the user's input 3. When the time or condition is met, the system will send a notification to user.
Alternative Flow	4. If the user disables the notification, the system stops sending notifications.

Table 4. 8 Notification – Use Case Description

4.8.2 Notification – Activity Diagram

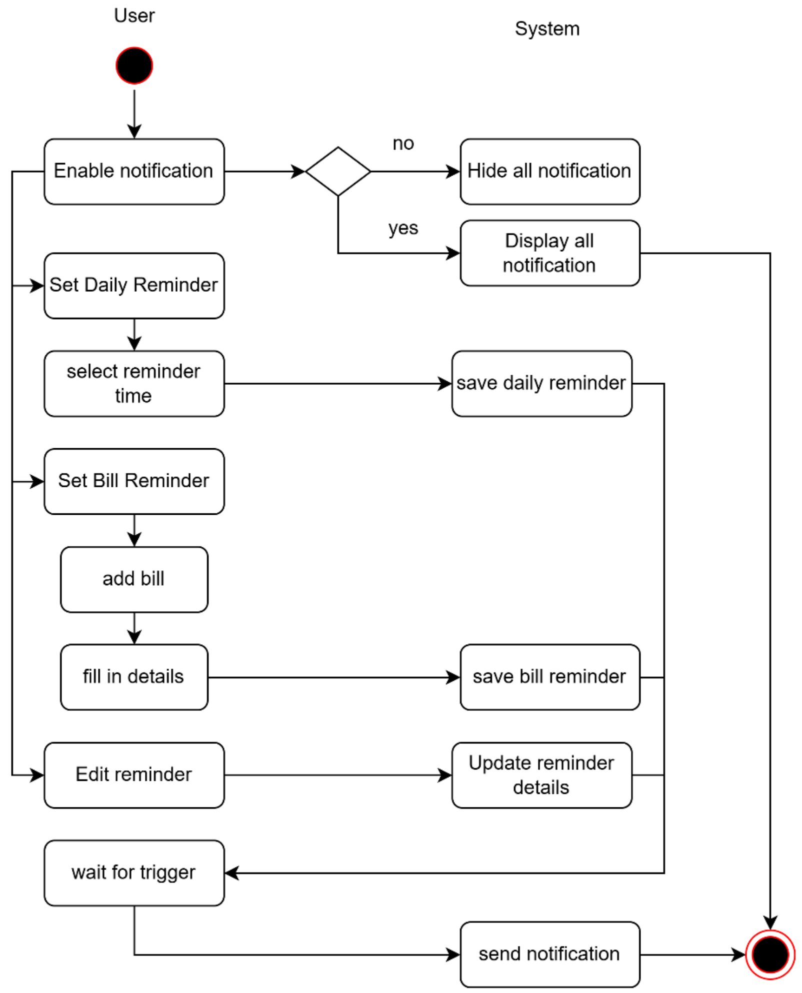


Figure 4. 9 Notification – Activity Diagram

This activity diagram outlines the process of enabling, setting, and managing reminders within the notification feature of the personal finance application. The process begins when the user enables the notification setting. The system then asks whether notifications should be shown or hidden. If the user chooses not to view notifications, the system hides them. If the user opts to allow notifications, the system proceeds to display all enabled notifications.

The user can then proceed to set a **daily reminder** by choosing a specific time. Once the reminder time is selected, the system saves the daily reminder settings. Additionally, the user may also choose to set a **bill reminder**. In this case, the user adds a bill and enters relevant bill details such as the due date, bill name, and amount. The system then saves the bill reminder accordingly.

If the user wishes to make any changes, they can select the **Edit reminder** option. The system updates the reminder details based on the modifications. After all reminders are configured, the system remains in a state of waiting for the scheduled trigger. Once the reminder time is reached, the system sends the notification to the user as configured. This flow ensures timely and personalized alerts for managing daily spending habits and bill payments efficiently.

4.11 Set Budget Module

4.11.1 Set Budget – Use Case Description

Use Case	Set budget
Purpose	To allow users to create spending limits for specific categories
Actor	User
Normal Flow of Event	1. User selects the "Set budget" option from the menu, the system will display a list of categories. 2. Users select a category and inputs the budgets amount. 3. Users confirm the budget by clicking the "Save" button. 4. The system validates the input and saves the budget.
Alternative Flow	1. If the user enters an invalid budget (e.g. negative value), the system will display an error message "Invalid budget amount".

Table 4. 9 Set Budget - Use Case Description

4.11.2 Set Budget – Activity Diagram

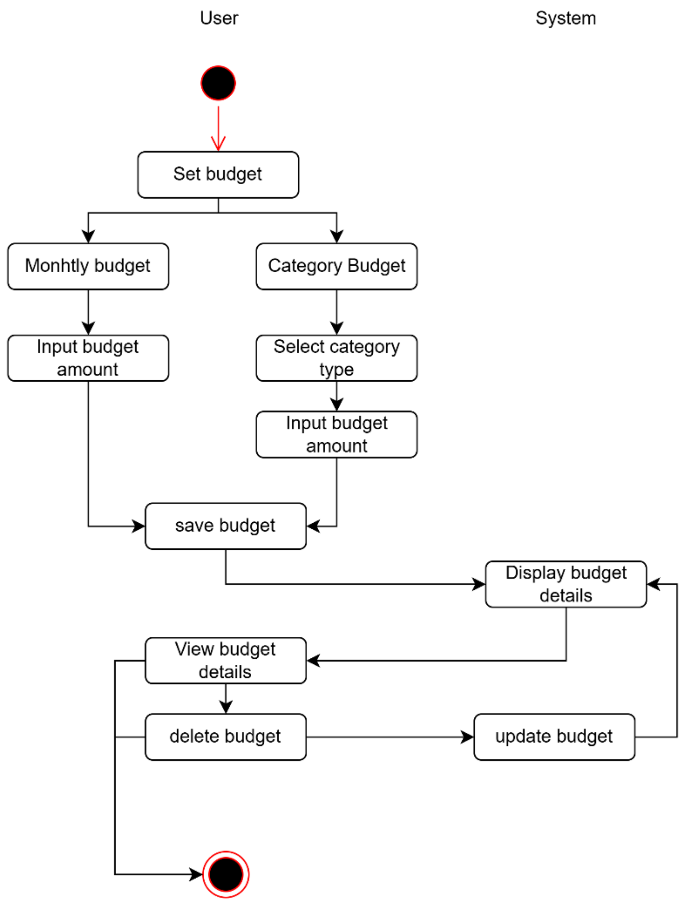


Figure 4. 10 Set Budget - Activity Diagram

This activity diagram illustrates the steps involved in setting and managing budgets within the personal finance management system. The process begins when the user selects the "Set Budget" option. The user is then presented with two choices: to set a Monthly Budget or a Category Budget. If the user chooses a monthly budget, they are prompted to input the total budget amount for the month. Alternatively, if the user opts for a category budget, they must first select a category type (such as food, transport, or entertainment) and then input the budget amount for that specific category.

Once the budget is entered, the user proceeds to save the budget. The system then processes and displays the saved budget details for the user's reference. The user is also able to view previously set budgets and has the option to delete any budget. If deletion is chosen, the system updates the database to remove the selected budget. This structured flow provides users with flexibility in tracking their overall financial limits or managing specific spending categories, promoting more disciplined and goal-oriented financial behavior.

4.12 Track Financial Goals Module

4.12.1 Track Financial Goals – Use Case Description

Use Case	Track Financial Goals
Purpose	To monitor the user's progress toward their financial goals
Actor	User
Normal Flow of Event	1. User selects the "Financial Goals" option. 2. The system displays the user's existing financial goals. 3. User selects a goal to view its progress. 4. The system displays a progress bar or percentage completion.
Alternative Flow	1. If no goals are set, the system displays a message "No financial goals set. Create a goal to start tracking."

Table 4. 10 Track Financial Goals - Use Case Description

4.12.2 Track Financial Goals – Activity Diagram

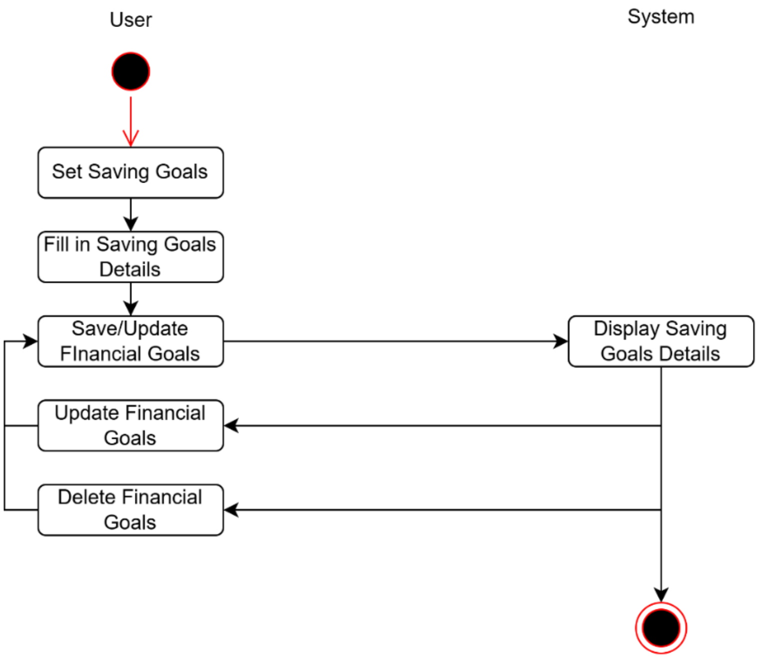

User
System
Set Saving Goals
Fill in Saving Goals Details
Save/Update FInancial Goals
Display Saving Goals Details
Update Financial Goals
Delete Financial Goals

Figure 4. 11 Track Financial Goals – Activity Diagram

This activity diagram illustrates the process by which a user sets, updates, and manages their financial saving goals within the personal finance application. The process begins when the user selects the **"Set Saving Goals"** option. The user is then prompted to **fill in the necessary goal details**, which may include the goal name, target amount, and desired deadline. Once all information is entered, the user proceeds to either **save or update the financial goals**, and the system responds by storing the data and displaying the saved goal details.

The diagram also allows for **goal management actions** such as **updating** existing financial goals when the user wants to modify target values, and **deleting goals** if they are no longer relevant. These actions ensure flexibility and personal control over financial planning. Throughout the process, the system consistently retrieves and displays the most updated goal information, ensuring that users remain informed and engaged with their progress. This feature helps users stay focused and motivated to achieve their financial objectives.

4.13 Real-Time Currency Conversion Module

4.13.1 Real-Time Currency Conversion – Use Case Description

Use Case	Real-time Currency Conversion
Purpose	To allow users to convert amounts between different currencies using real-time exchange rates.
Actor	user
Normal Flow of Event	1. User selects the "currency Conversion" feature from the app, the system displays a form to input the base currency, target currency, and amount to convert. 2. User input the amount to convert. 3. The system fetches the latest exchange rate from an external API. 4. The system calculates the converted amount and display it to the user.
Alternative Flow	5. If the system cannot fetch the exchange rate, the system displays an error message "Unable to fetch real-time exchange rates. Please try again later."

Table 4. 11 Real-Time Currency Conversion – Use Case Description

4.13.2 Real-Time Currency Conversion – Activity Diagram

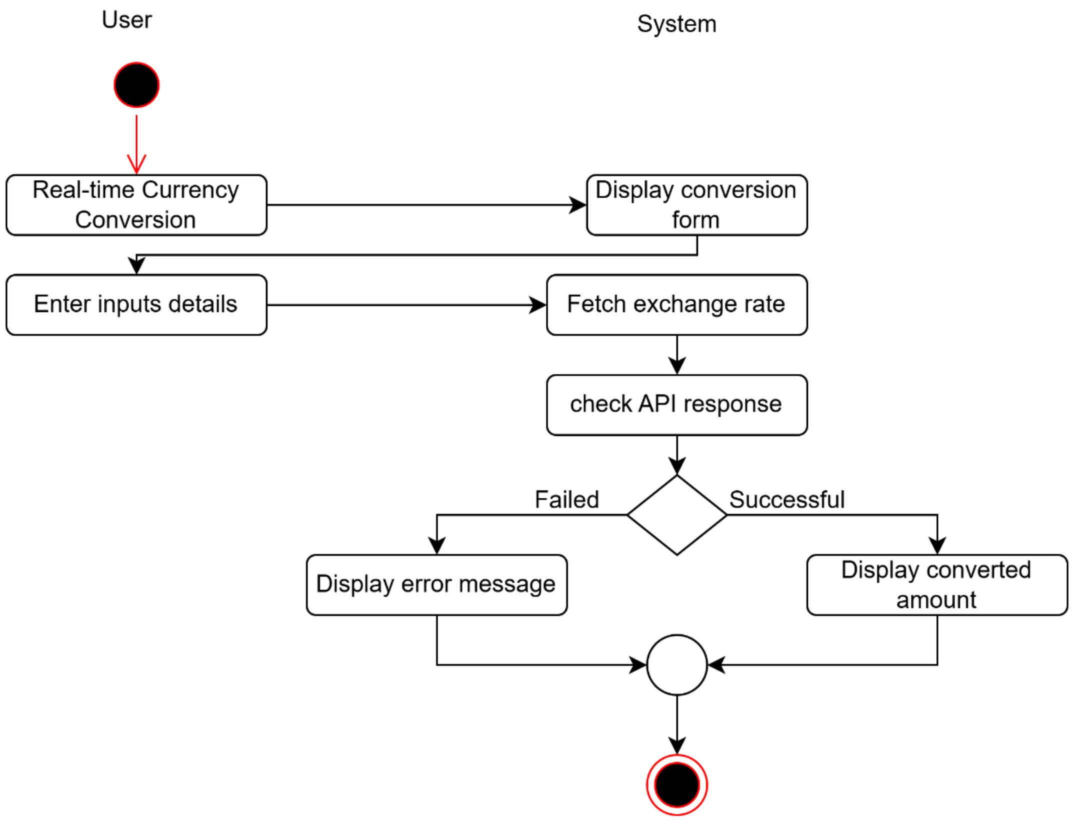


Figure 4. 12 Real-Time Currency Conversion -Activity Diagram

The "Real-Time Currency Conversion" module activity diagram illustrates the process of converting amounts between different currencies. The user selects the "Currency Conversion" option and inputs the base currency, target currency, and amount to convert. The system fetches the latest exchange rates from an external API.

If the API successfully retrieves the data, the system calculates and displays the converted amount. However, if the system cannot fetch the exchange rates (e.g., due to network issues), an error message is displayed, prompting the user to try again later. This feature provides users with real-time financial insights for currency-related transactions.

4.14 Loan Calculator Module

4.14.1 Loan Calculator – Use Case Description

Use Case	Loan calculator
Purpose	To help user estimate loan repayments for personal, housing or car loans.
Actor	User
Normal Flow of Event	1. User selects the "Loan Calculator" feature from the app. 2. Users need to fill their birthdate before selecting the loan type (e.g. personal loan, housing loan) 3. Users select the loan type, and the system will display a form to input the loan details. 4. Users enters the required details 5. The system calculates the estimated monthly repayment amount and total repayment over the loan term. 6. The system displays the results, including an amortization schedule if requested.
Alternative Flow	If the user inputs an invalid data (e.g. negative amount or missing value) 1. The system displays an error message "Please enter valid loan details." If the user didn't fill in their birthdate before selecting the loan type 1. The system displays an error message "Please enter your birthdate"

Table 4. 12 Loan Calculator - Use Case Description

4.14.2 Loan Calculator – Activity Diagram

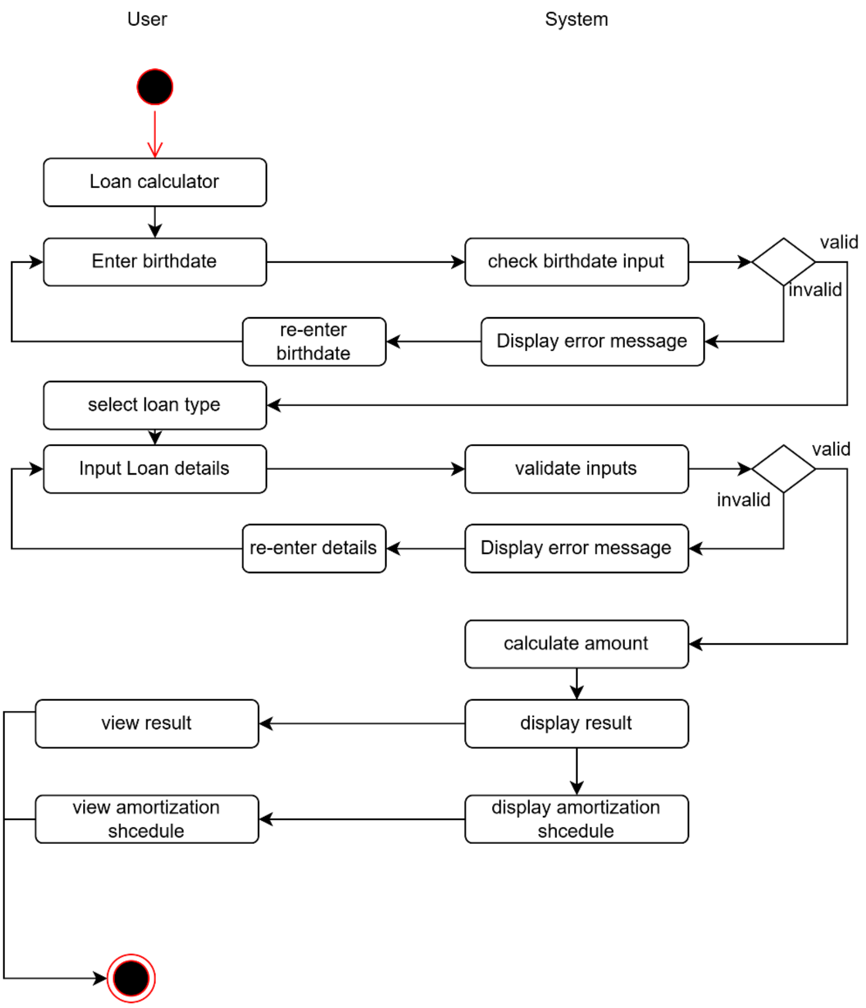


Figure 4. 13 Loan Calculator – Activity Diagram

The activity diagram for the "Loan Calculator" module illustrates how users estimate loan repayments. The user selects the "Loan Calculator" feature and inputs required details, such as their birthdate, loan type (e.g., personal, car, or housing loan), loan amount, interest rate, and loan term.

The system validates the input to ensure all fields are complete and values are valid. If the input is invalid, the system displays an error message. Once validated, the system calculates the estimated monthly repayment amount and total loan repayment. The results are displayed to the user, including an optional amortization schedule for further clarity. This module helps users make informed decisions about loans.

4.15 Feedback Module

4.15.1 Feedback – Use Case Description

Use Case	Feedback
Purpose	To allow users to provide feedback or report issues regarding the app.
Actor	User
Normal Flow of Event	1. Users select the "Feedback" feature from the menu. 2. The system displays a feedback form with fields for user comments and an optional file attachment such as screenshot. 3. Users enter feedback or describes the issues they encountered. 4. User submits the feedback form. 5. The system saves the feedback and send a confirmation message to the user: "Thank you for your feedback!"
Alternative Flow	If the user does not fill in mandatory fields 1. The system displays an error message: "Feedback field cannot be empty."

Table 4. 13 Feedback – Use Case Description

4.15.2 Feedback – Activity Diagram

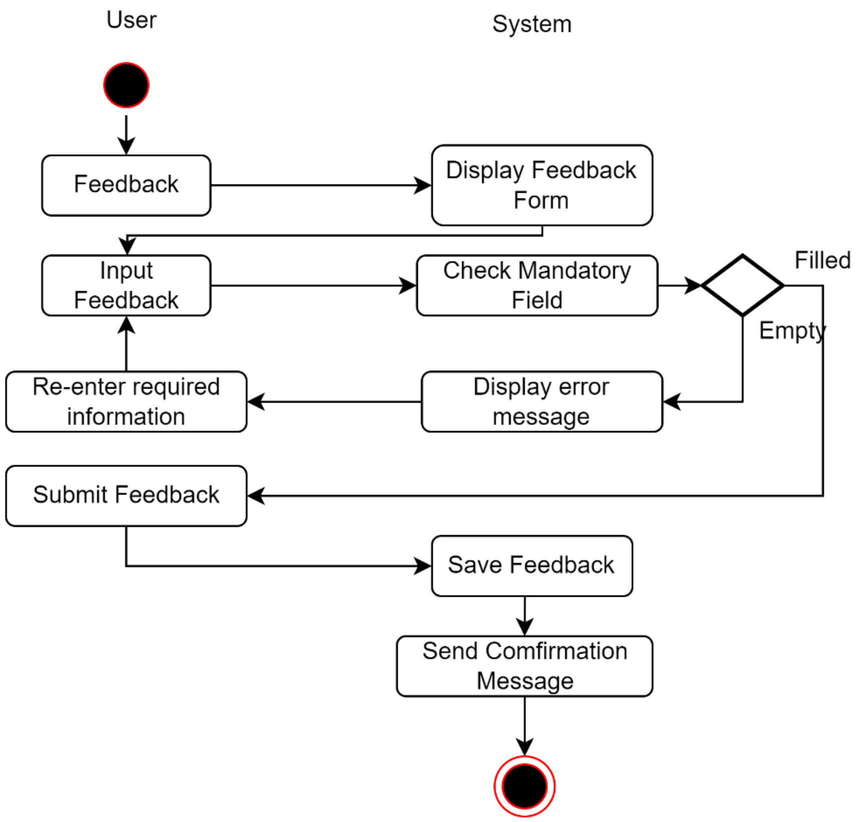


Figure 4. 14 Feedback – Activity Diagram

The "Feedback" module activity diagram shows how users provide feedback or report issues. The process starts when the user selects the "Feedback" option. The system displays a form where users can input their comments, rate their experience, or attach a file (e.g., a screenshot).The system checks that all mandatory fields are filled. If any required information is missing, an error message prompts the user to complete the form. Once the feedback is submitted, the system saves it in the database and displays a confirmation message thanking the user. This feature helps the system administrators identify areas for improvement and maintain user satisfaction.

4.16 Account Management Module

4.16.1 Account Management – Use Case Description

Use Case	Account
Purpose	Allow user to customize their account in transactions, view their account balance.
Actor	User
Normal Flow of Event	1. User navigates to the Account Management section from the settings or dashboard. 2. The system displays a list of existing accounts. 3. User selects to add a new account or delete an account. 4. For adding/editing, user enters details such as account name, type, and balance. 5. User confirms the action. 6. The system saves the changes and updates the account list.
Alternative Flow	– If user cancels the action, the system returns to the previous screen without saving. – If a duplicate category name is entered, the system prompts the user to use a unique name. – If a required field is left empty, the system displays a validation error message.

Table 4. 14 Account Management - Use Case Description

4.16.2 Account Management - Activity Diagram

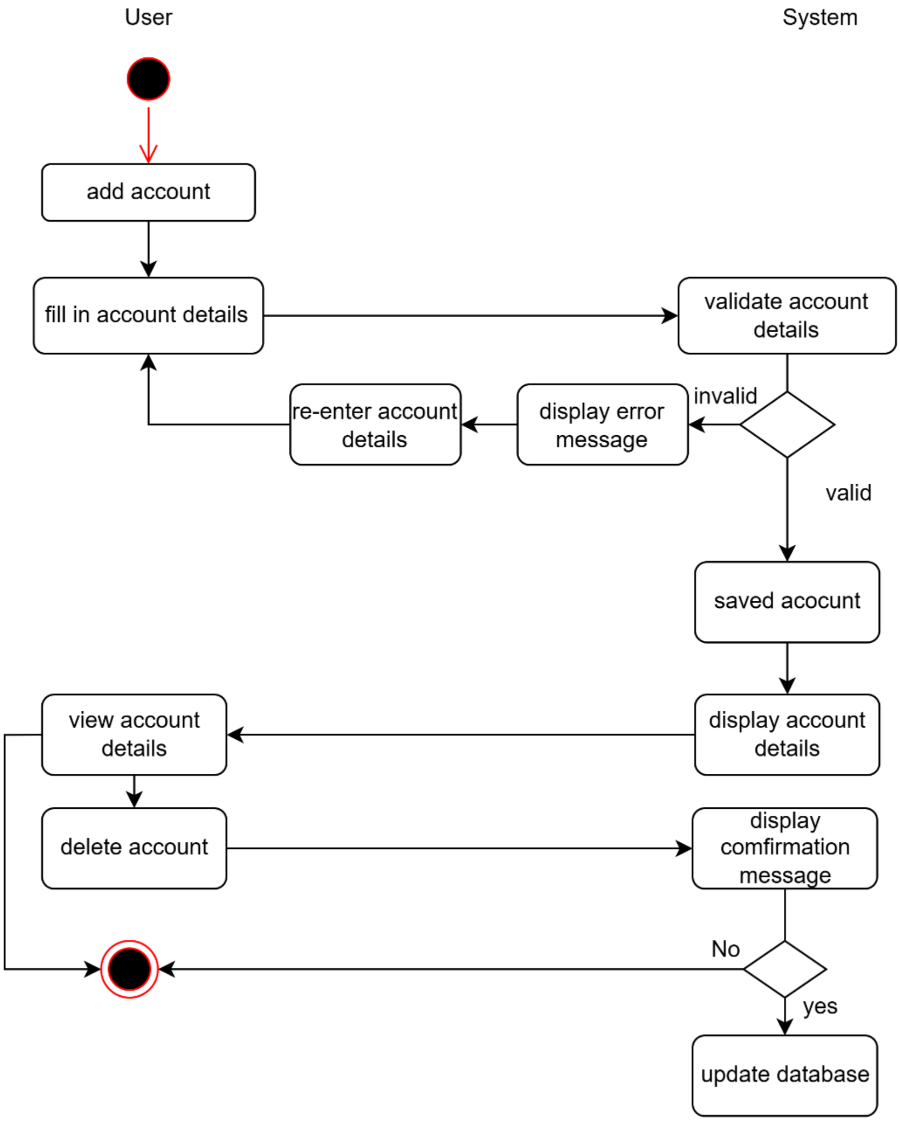


Figure 4. 15 Account Management - Activity Diagram

The activity diagram for **Account Management** describes the structured process a user follows to manage their financial accounts within the system. It starts with the user initiating the action to add a new account, followed by entering account-specific details such as account name, type (e.g., bank, cash, e-wallet), and starting balance. The system then validates these details to check for completeness and uniqueness. If any validation fails, the system responds with an error message and allows the user to correct the input. When the data is valid, the account information is saved, and the system confirms by displaying the newly added account details. The user can then review the list of accounts and optionally delete any account. Upon selecting deletion, a confirmation message is shown, and only with user approval does the system proceed to update the database. This activity flow promotes accurate account setup, prevents accidental data loss, and enhances user flexibility in financial tracking.

4.17 Category Management Module

4.17.1 Category Management – Use case Description

Use Case	Category
Purpose	Allow user to customize their category in transactions, set budget category.
Actor	User
Normal Flow of Event	1. User navigates to the Category Management section from the settings or dashboard. 2. The system displays a list of existing categories. 3. User selects to add a new category or delete a category. 4. User fills in or modifies category details. 5. User confirms the changes. 6. The system saves the updated category list.
Alternative Flow	– If user cancels the action, the system returns to the previous screen without saving. – If a duplicate category name is entered, the system prompts the user to use a unique name. – If a required field is left empty, the system displays a validation error message.

Table 4. 15 Category Management - Use Case Description

4.17.2 Category Management – Activity Diagram

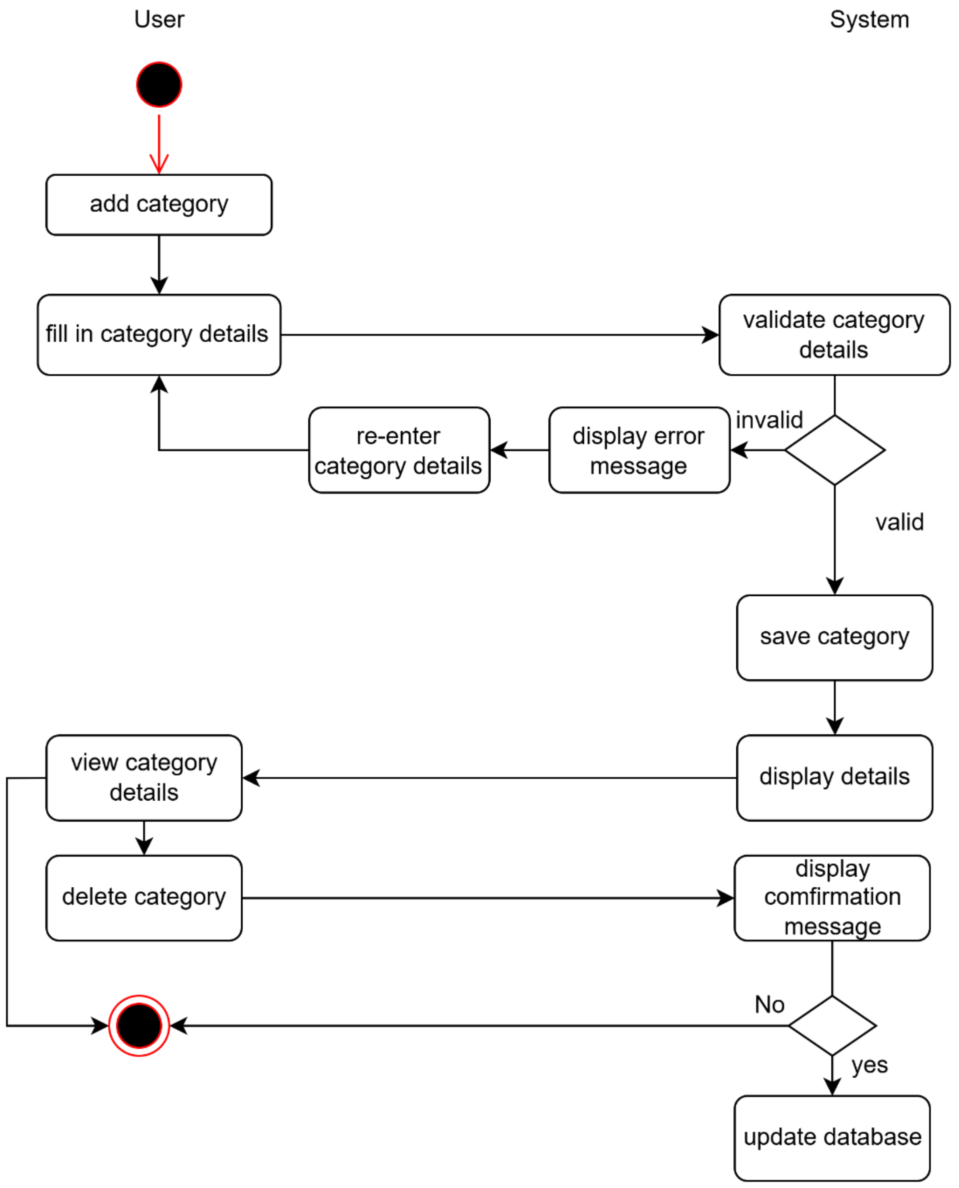


Figure 4. 16 Category Management - Activity Diagram

The activity diagram for **Category Management** illustrates the user-system interaction flow involved in customizing transaction categories within the application. The process begins when the user chooses to add a new category, prompting them to enter relevant details such as the category name or color. Once the input is submitted, the system performs validation to ensure that all required fields are completed correctly and that the category name does not exist. If the input is invalid, the system displays an error message, and the user is guided to re-enter the information. If the input is valid, the system proceeds to save the category and displays the saved details for confirmation. The user is then able to view all category details and optionally delete any category. If deletion is selected, the system presents a confirmation message before updating the database. This flow ensures a smooth and error-handled process for category customization while maintaining data integrity and user control.

4.18 Wireframe /Initial Design

4.18.1 Login and Register Page Wireframe

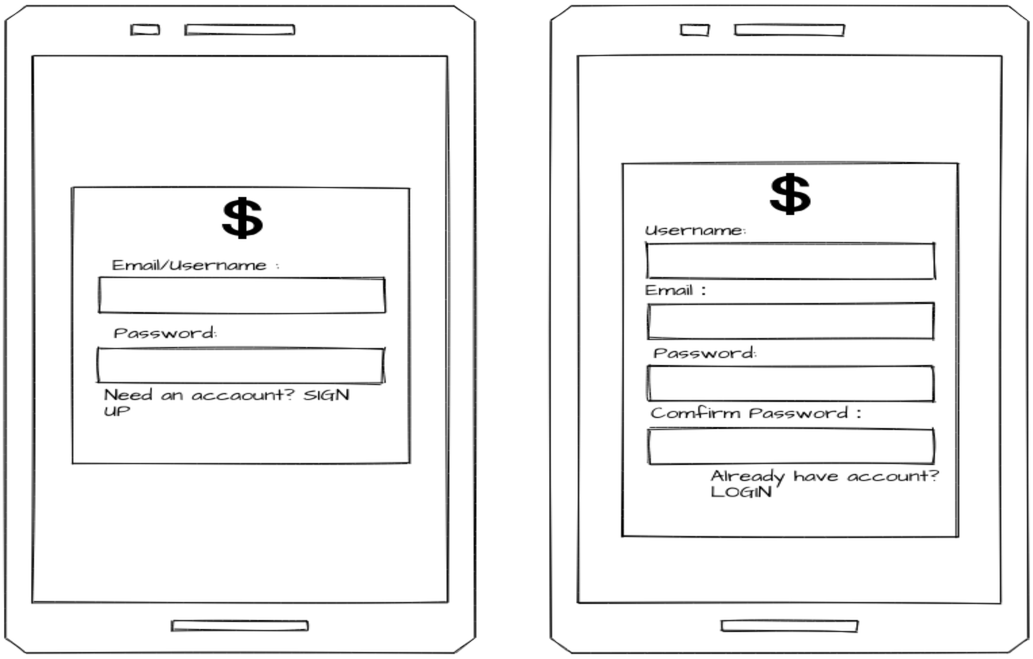


Figure 4. 17 Login & Sign-Up Initial Design

The login and sign-up screens are the preliminary interface designs for user authentication, which is a critical feature of the budgeting app. The login screen is designed to provide a secure and straightforward way for returning users to access their accounts by entering their email or username along with their password. On the other hand, the sign-up screen serves as the gateway for new users to create an account by filling in essential details such as a username, email, and password. To enhance security and minimize errors, the sign-up process also includes a field for confirming the password.

4.18.2 Transaction Tracking and Account Pages Wireframe

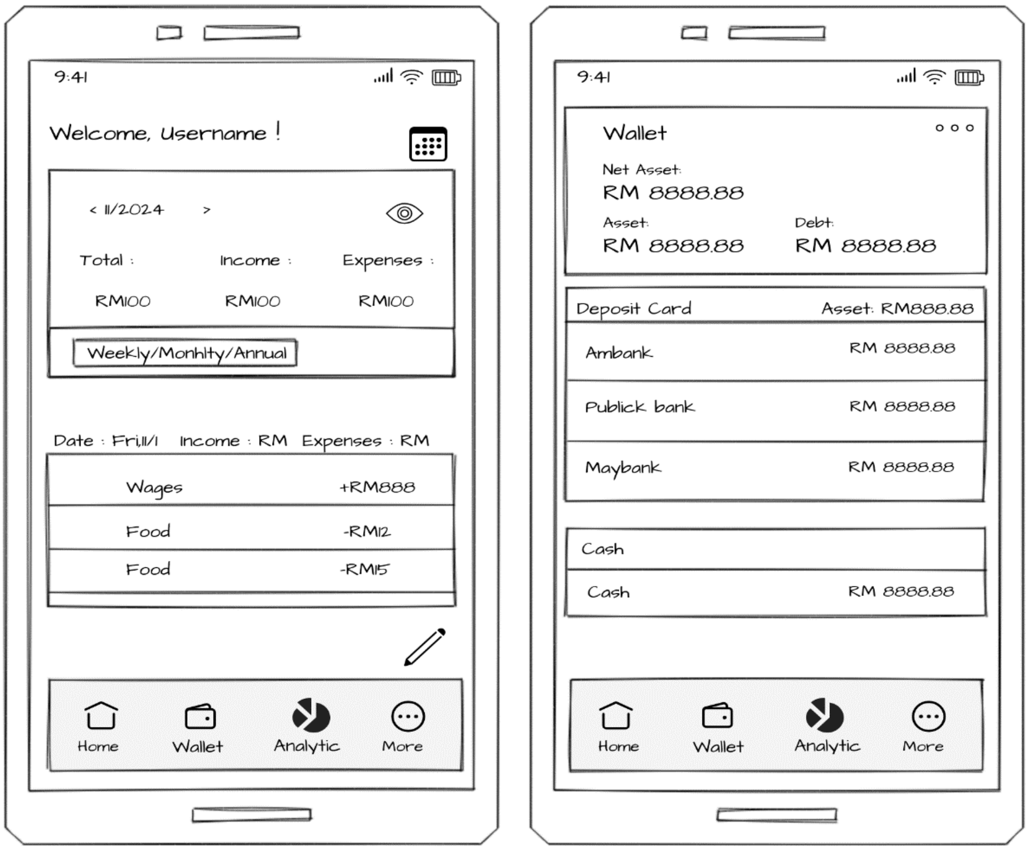


Figure 4. 18 Transactions and Account Dashboard Initial Design

The Home Screen serves as the main dashboard, providing users with an overview of their financial status. It displays key metrics, such as the total income, expenses, and balance, in an easy-to-read format. Users can also toggle between weekly, monthly, and annual views for more tailored insights. Below the summary, a transaction log lists recent transactions, categorized by date, description, and amount. This design ensures users have immediate access to their financial activities and balances upon opening the app. Additionally, a pencil icon allows users to add new transactions seamlessly, promoting consistent tracking.

The Wallet Screen focuses on managing the user's financial accounts and assets. It presents details of deposit cards, assets, and liabilities in a structured manner, including balances for specific accounts (e.g., Ambank, Public Bank, Maybank) and cash holdings. At the top, the screen calculates the net asset value by subtracting liabilities from total assets. This layout simplifies the organization of financial accounts, helping users maintain a clear understanding of their financial distribution.

Both screens include a navigation bar at the bottom, enabling quick access to essential features like Home, Wallet, Analytics, and More, ensuring a user-friendly and intuitive experience.

4.18.3 Graph Visualization and Setting Wireframe

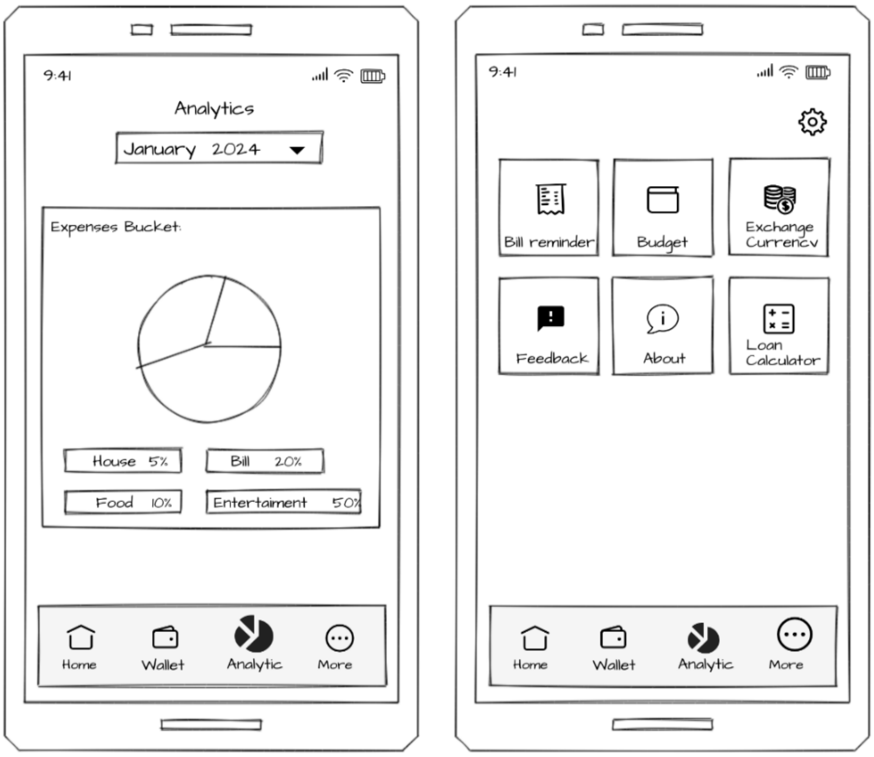


Figure 4. 19 Graph Visualization and Setting Initial Design

The Analytics Screen provides users with a visual representation of their financial data, focusing on expense distribution. A pie chart, labeled as the "Expense Bucket," categorizes expenditures into segments such as housing, bills, food, and entertainment, with percentage breakdowns. Users can select a specific month or date range from the dropdown menu at the top, enabling them to analyze their spending patterns over time. This feature helps users identify areas of overspending and make more informed financial decisions.

The More Screen consolidates additional tools and settings, enhancing the app’s functionality beyond core features. It includes options like Bill Reminder for timely notifications, Budget for setting financial goals, and Exchange Converter for real-time currency conversions. Other tools, such as Loan Calculator, assist users in planning financial commitments, while Feedback and About sections provide avenues for user support and app information. The Settings icon at the top allows for further customization and preferences.

4.19 ERD (Entity-Relationship Diagram)

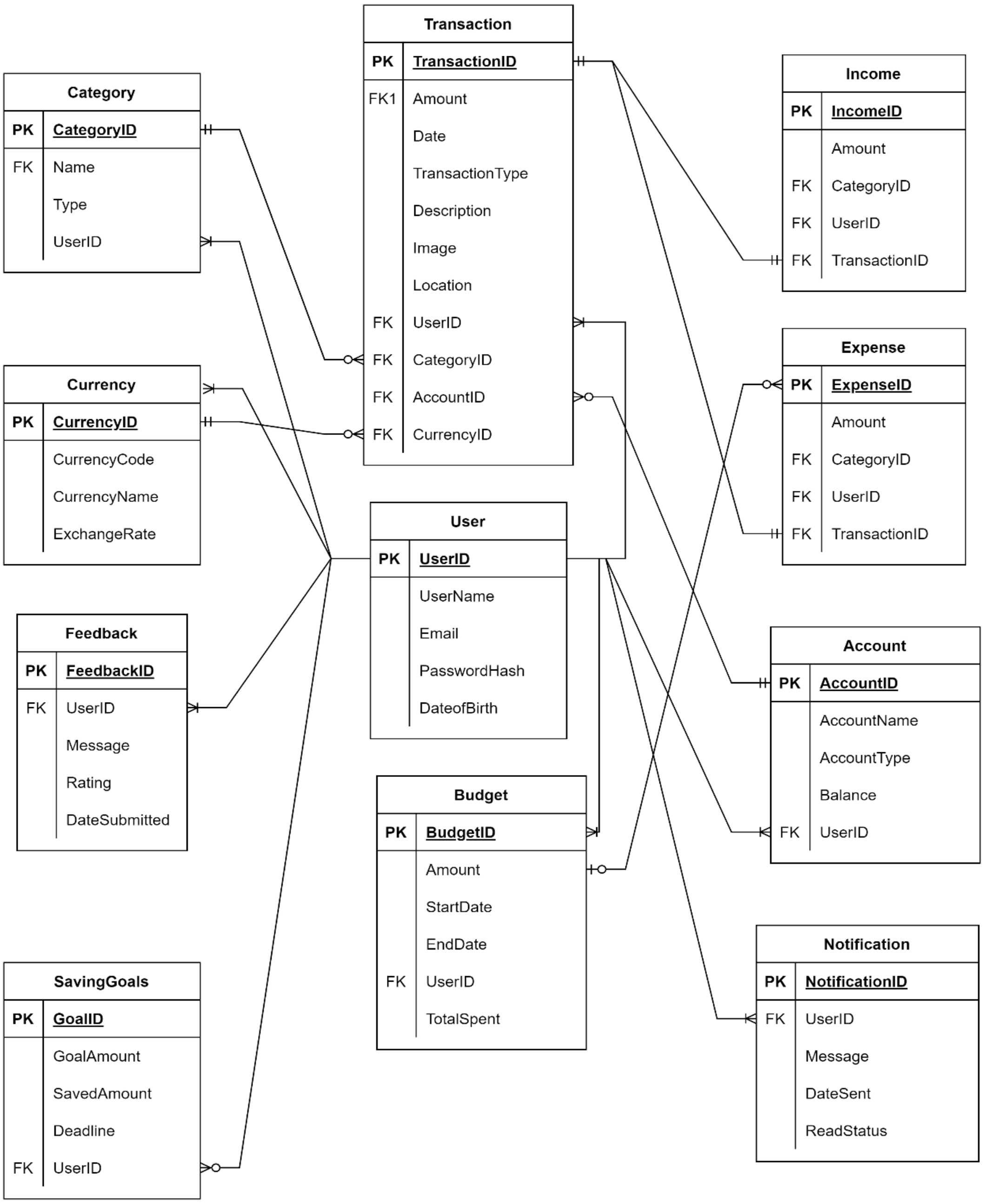


Figure 4. 20 ERD Diagram

The Entity Relationship Diagram (ERD) for the Personal Finance Management Mobile Application models how different entities interact with each other and outlines the structure of the database. At the core of the system is the User entity, which represents the application’s user and holds essential information like email, password, and profile details. Each User can have multiple associated Transactions, such as expenses or income, which record the amount, category, and account used. The Transaction entity is linked to several other components, like Category, Account, and Currency, to provide a comprehensive understanding of a user's financial activity. Specifically, each Transaction belongs to a Category, which can be either an income or an expense, helping to organize and classify transactions.

The Account entity tracks different user accounts such as bank accounts or credit cards, and each account can have multiple Transactions associated with it. Additionally, each Transaction is tied to a specific Currency, ensuring that the system can handle various monetary values across different regions. Users also set Saving Goals, and each goal is linked to a user, allowing them to track their progress toward financial objectives. Budgets, another critical entity, allow users to allocate a set amount for spending in specific categories. A Budget can have many Expenses under it, providing a structure for managing spending limits.

Similarly, the Notification entity ensures that users are notified about key events, like approaching budget limits or upcoming bill payments. Each notification is linked to a single user, but a user may receive multiple notifications over time. Finally, users can leave Feedback, which helps improve the application. Each feedback record is tied to a user, allowing for the collection of user experiences and opinions.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Hardware Setup

Components	**Description**
Development Machine	ASUS VivoBook X512FL (Intel Core i5-8265U CPU @ 1.60GHz, 12GB RAM, Windows 11 Home Single Language 64-bit)
Android Test Device	Realme 12x 5G (Model RMX3997, MediaTek Dimensity 6100+ processor, 8GB RAM, 128GB Storage, Android with realme UI 5.0)
Internet	WiFi Connection and Mobile Hotspot (Tested on both stable broadband WiFi and 5G mobile hotspot connectivity)

Table 5. 1 Hardware Setup

5.2 Software setup

Software/Tool	**Purpose**
Android Studio	Main IDE. development
Firebase Console	Backend services: Auth, Firestore
Figma	UI/UX design and prototyping
MPAndroidChart	Graph and chart visualization library
Volley	API calls for currency exchange features

Table 5. 2 Software Setup

5.3 Setting and Configuration

5.3.1 Firebase Integration

1. **Create Firebase Project**

- Opened the Firebase Console (https://console.firebase.google.com).
- Clicked **"Create a firebase Project"** and entered the project name (e.g. "MyBudgetApp").

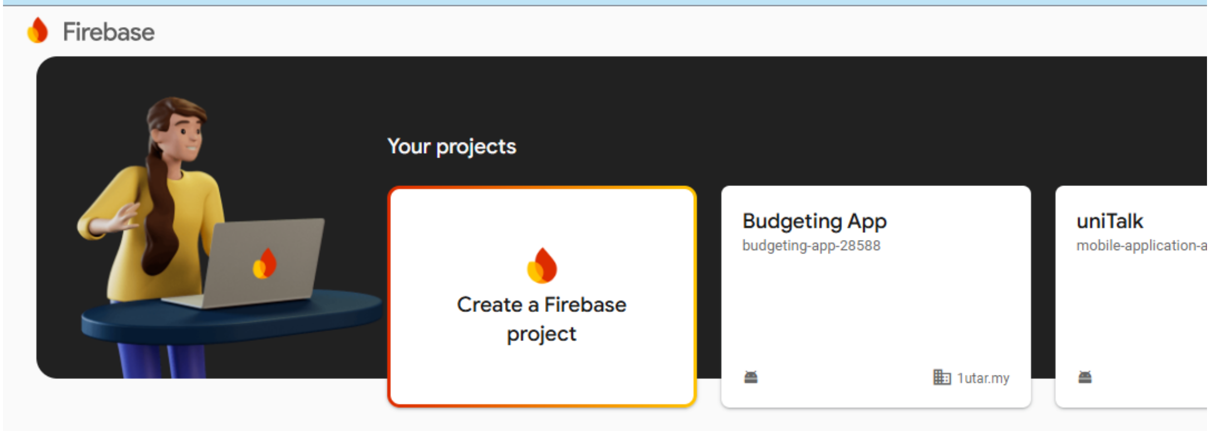


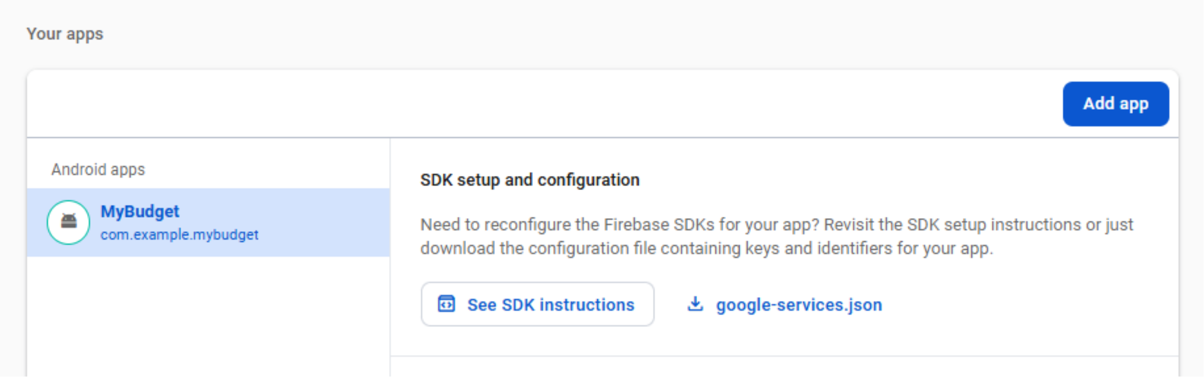
Figure 5. 1 Firebase Project Setup – Create a Firebase Project

2. **Register the Android App**

- Added the Android package name (e.g. com.example.mybudget) and the app nickname.
- (Optional) Added SHA-1 key for authentication and Google Sign-In support.

3. **Downloads google-services.json**

- Downloaded the google-services.json file generated by Firebase.



Your apps

Android apps

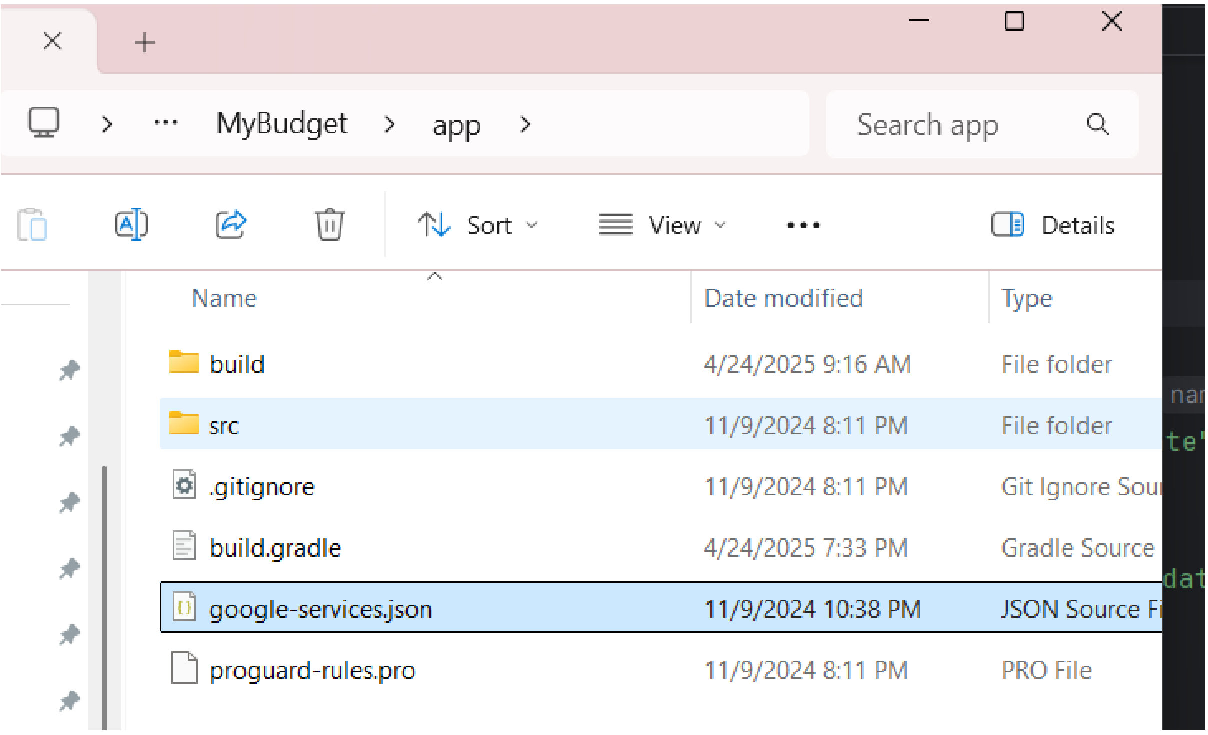
MyBudget
com.example.mybudget

SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

Figure 5. 2 SDK Setup and Configuration screen

- Placed the file inside the app/ directory of the Android project.



Name	Date modified	Type
build	4/24/2025 9:16 AM	File folder
src	11/9/2024 8:11 PM	File folder
.gitignore	11/9/2024 8:11 PM	Git Ignore Sou
build.gradle	4/24/2025 7:33 PM	Gradle Source
google-services.json	11/9/2024 10:38 PM	JSON Source Fi
proguard-rules.pro	11/9/2024 8:11 PM	PRO File

Figure 5. 3 Placement of google-services.json file inside the app directory of Android Studio project

4. **Add Firebase SDK and Plugin**

- In the project-level build.gradle, added the classpath:

```
// Top-level build file where you can add configuration options
common to all sub-projects/modules.
plugins {
    id("com.android.application") version '8.9.1' apply false
    id("com.google.gms.google-services") version "4.4.2" apply false

}
```

- Also added Firebase dependencies

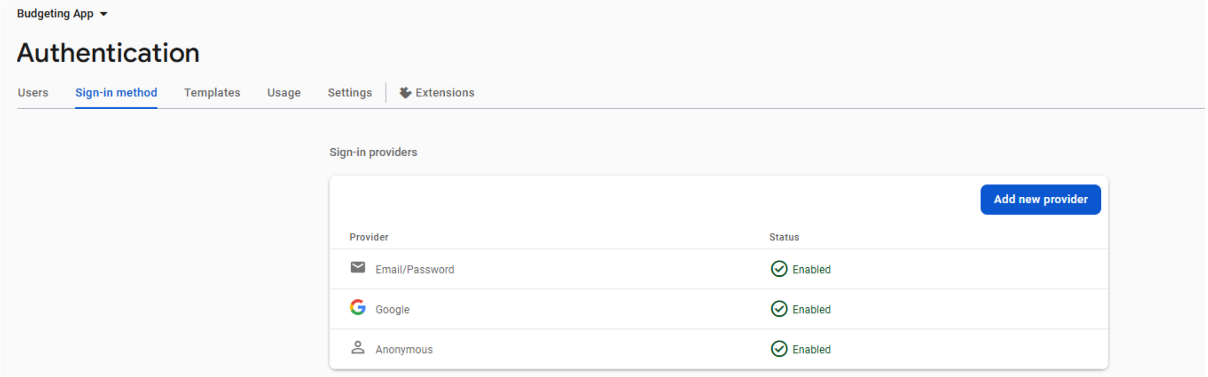
```
•   implementation 'com.google.firebase:firebase-auth:21.0.1'
    implementation 'com.google.firebase:firebase-firestore:24.0.1'
```

5. **Sync the Project**

- Clicked **"Sync Now"** in Android Studio to ensure all Firebase services and libraries were integrated correctly.

6. **Enable Required Firebase Services**

- Enable authentication email/password, google, and anonymous



Sign-in providers

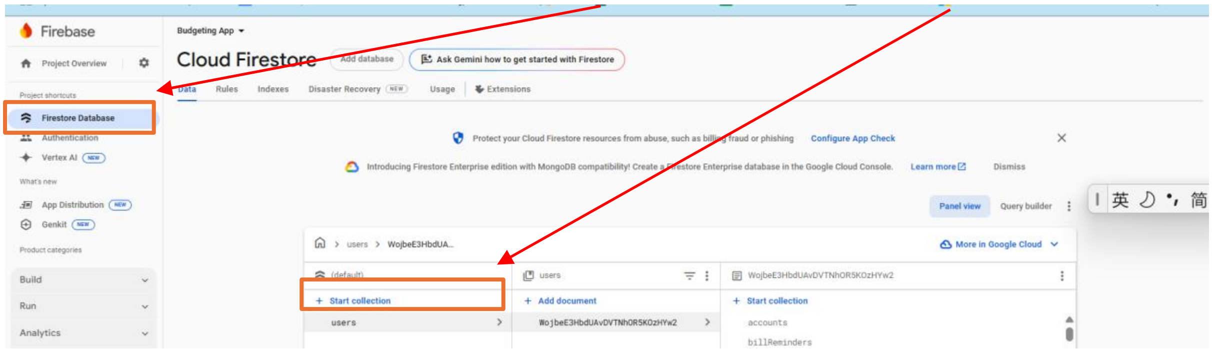
Provider	Status
Email/Password	✓ Enabled
Google	✓ Enabled
Anonymous	✓ Enabled

Figure 5. 4 Firebase Authentication panel showing enabled sign-in providers

5.3.2 Firestore Collections

1. **Open Firestore Database**

- In the Firebase Console, navigated to **Firestore Database** > **Start Collection**.



Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing

Introducing Firestore Enterprise edition with MongoDB compatibility! Create a Firestore Enterprise database in the Google Cloud Console.

Figure 5. 5 Firebase Firestore interface showing navigation to “Start Collection” to create Firestore collections

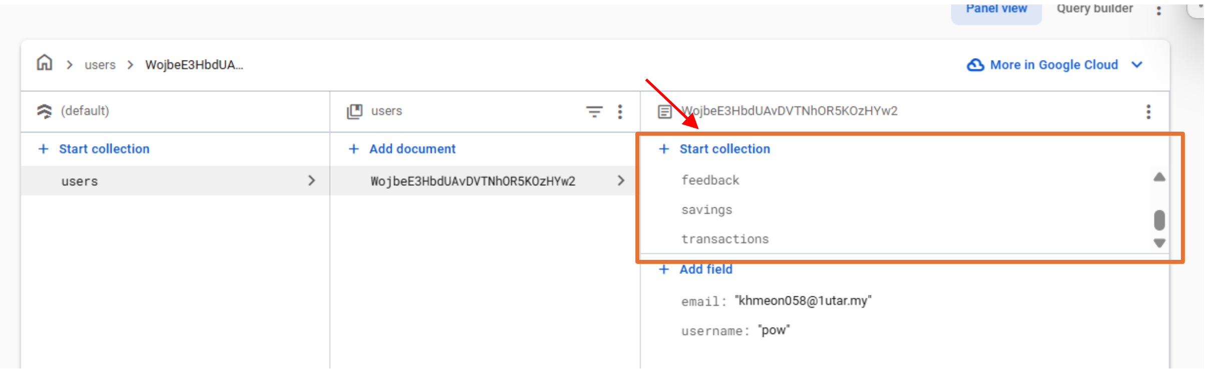
2. **Create users Collection**

- Created a top-level user’s collection.
- Each document inside users represents a unique user (document ID = UID).

3. **Add Subcollections under Each User Document**

- Inside each user document (e.g., /users/{userId}), the following subcollections were created:
    - **transactions**: Stores user income and expense records.
    - **budgets**: Stores monthly budgets categorized by type (with nested budget_history).
    - **goals**: Stores user savings goals with targets and progress.
    - **feedback**: Captures app-related feedback from users.
    - **billReminders**: Stores scheduled monthly bill reminders.

- **categories**: Stores custom income/expense categories created by the user.

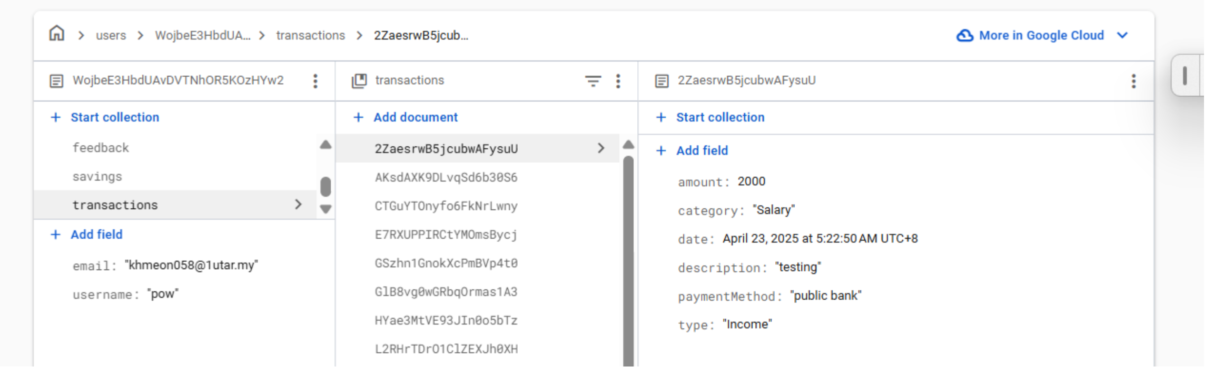


email: "khmeon058@1utar.my"
username: "pow"

Figure 5. 6 Subcollections under a Firestore user document showing feedback, savings, and transactions

4. **Structure Documents with Key Fields**

- Each document in subcollections was structured with relevant fields, for example:
  - **Transactions**: amount, category, date, description, payment method, and type



email: "khmeon058@1utar.my"
username: "pow"

amount: 2000
category: "Salary"
date: April 23, 2025 at 5:22:50 AM UTC+8
description: "testing"
paymentMethod: "public bank"
type: "Income"

Figure 5. 7 Firestore Document Structure for a Transaction with Fields

5.3.4 Gradle Dependencies

1. Implement these dependencies and sync the project

```
plugins {
    id("com.android.application")
    id("com.google.gms.google-services")
}

android {
    namespace 'com.example.mybudget'
    compileSdk 34

    defaultConfig {
        applicationId "com.example.mybudget"
        minSdk 24
        targetSdk 34
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_11
        targetCompatibility JavaVersion.VERSION_11
    }
}

dependencies {

    implementation libs.appcompat
    implementation libs.material
    implementation libs.activity
    implementation libs.constraintlayout
    implementation libs.firebase.firestore
    implementation libs.recyclerview
    testImplementation libs.junit
    androidTestImplementation libs.ext.junit
    androidTestImplementation libs.espresso.core
    implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'
    implementation 'com.google.android.flexbox:flexbox:3.0.0'
    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.activity:activity:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
    implementation 'androidx.recyclerview:recyclerview:1.2.1'
    implementation 'androidx.cardview:cardview:1.0.0'
    implementation 'com.github.bumptech.glide:glide:4.15.1'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.15.1'
    implementation 'com.google.android.gms:play-services-base:18.1.0'
    implementation 'com.google.android.material:material:1.9.0'
    implementation 'com.google.android.material:material:1.11.0'
    implementation "androidx.compose.material:material-icons-extended:1.6.0"
    implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'
    implementation 'com.android.volley:volley:1.2.1'

    implementation(platform("com.google.firebase:firebase-bom:33.5.1"))
    implementation("com.google.firebase:firebase-analytics")
    implementation 'com.google.firebase:firebase-auth:21.0.1'
    implementation 'com.google.firebase:firebase-firestore:24.0.1'
    implementation 'com.google.android.gms:play-services-auth:20.7.0'
    implementation 'com.google.android.material:material:1.6.0'
    implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0'



    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

5.3.5 Security Rules (Firestore)

1. Open Firebase Console to Accessed the Firebase project via https://console.firebase.google.com.

2. Navigate to Firestore Database Rules and Selected Firestore Database from the left menu and opened the Rules tab.

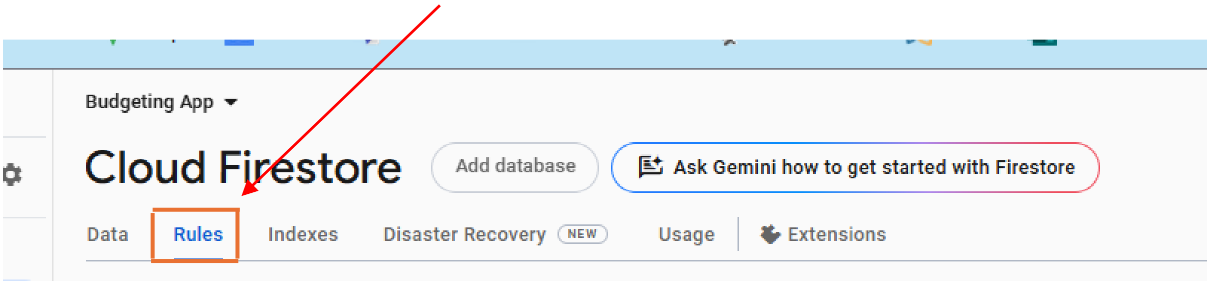
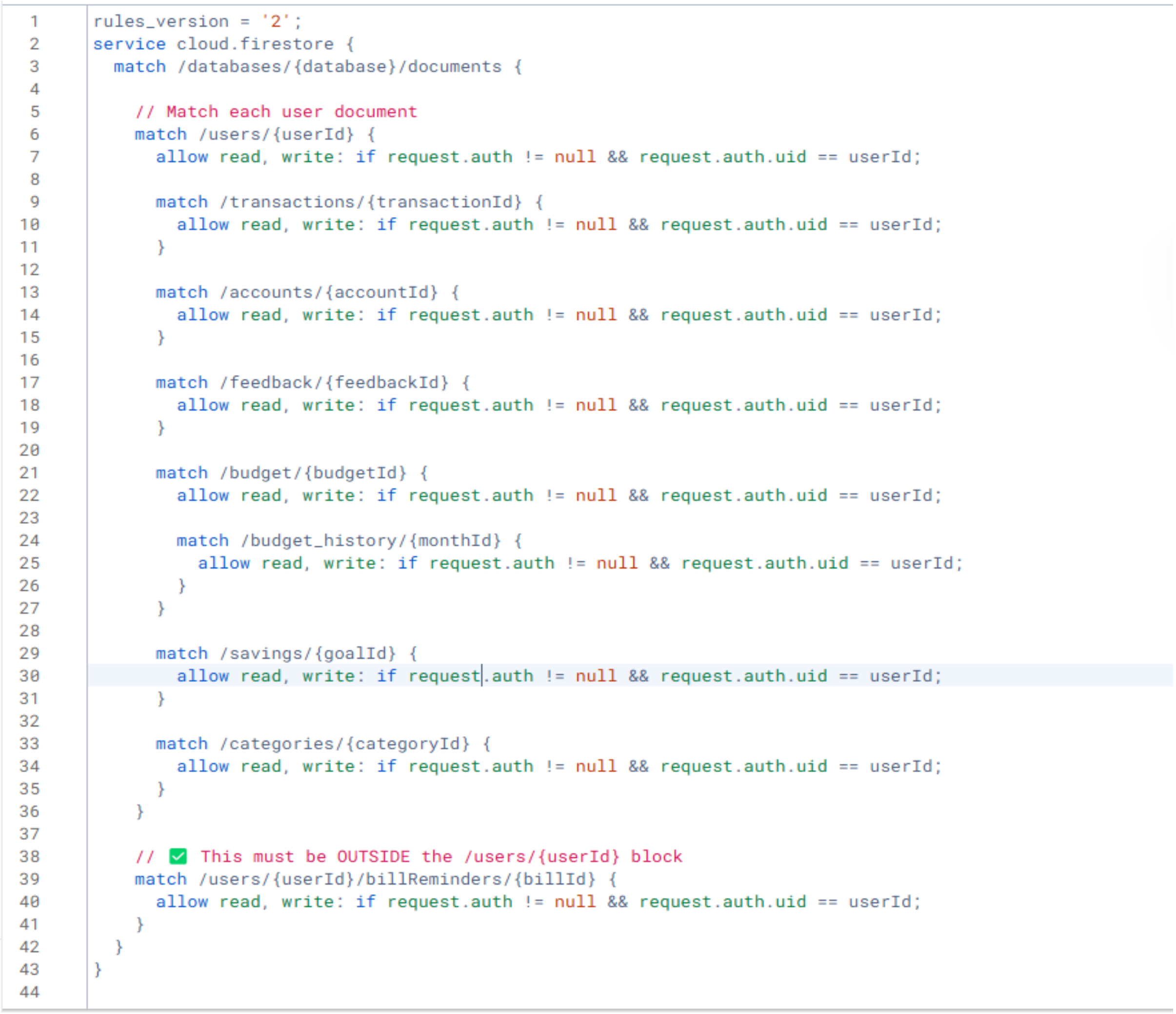


Figure 5. 8 Firestore Database Rules Navigation

3. Edit the Security Rules

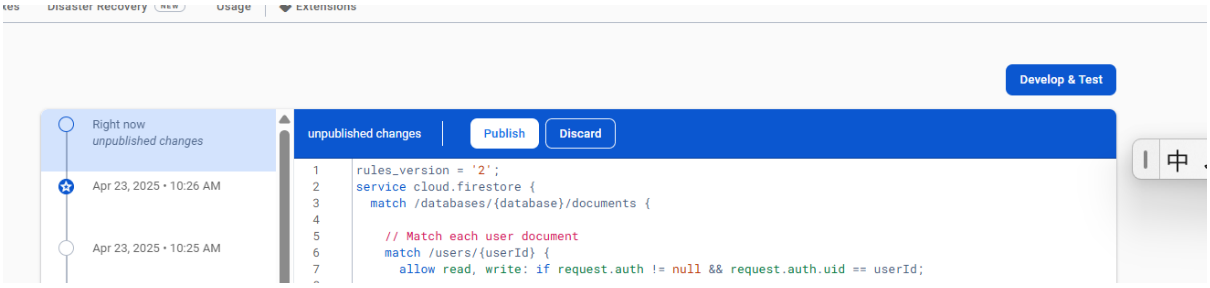
4. Modified the default rules to only allow access to authenticated users using: request.auth != null && request.auth.uid == userId



```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    // Match each user document
    match /users/{userId} {
      allow read, write: if request.auth != null && request.auth.uid == userId;

      match /transactions/{transactionId} {
        allow read, write: if request.auth != null && request.auth.uid == userId;
      }

      match /accounts/{accountId} {
        allow read, write: if request.auth != null && request.auth.uid == userId;
      }

      match /feedback/{feedbackId} {
        allow read, write: if request.auth != null && request.auth.uid == userId;
      }

      match /budget/{budgetId} {
        allow read, write: if request.auth != null && request.auth.uid == userId;

        match /budget_history/{monthId} {
          allow read, write: if request.auth != null && request.auth.uid == userId;
        }
      }

      match /savings/{goalId} {
        allow read, write: if request.auth != null && request.auth.uid == userId;
      }

      match /categories/{categoryId} {
        allow read, write: if request.auth != null && request.auth.uid == userId;
      }
    }

    // ✅ This must be OUTSIDE the /users/{userId} block
    match /users/{userId}/billReminders/{billId} {
      allow read, write: if request.auth != null && request.auth.uid == userId;
    }
  }
}
```

Figure 5. 9 Firestore Security Rules Configuration to Restrict Access

5. Once verified, clicked “Publish” to activate the security settings.



```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    // Match each user document
    match /users/{userId} {
      allow read, write: if request.auth != null && request.auth.uid == userId;
```

Figure 5. 10 Firestore Published Rules Configuration

5.4 System Operation

This part describes how the users engage with the application in day-to-day usage. It describes the salient features of the app, how the users respond to actions such as logging in, processing transactions, setting budgets, and how the system responds by storing data, generating reports, and sending reminders.

5.4.1 User Authentication Section

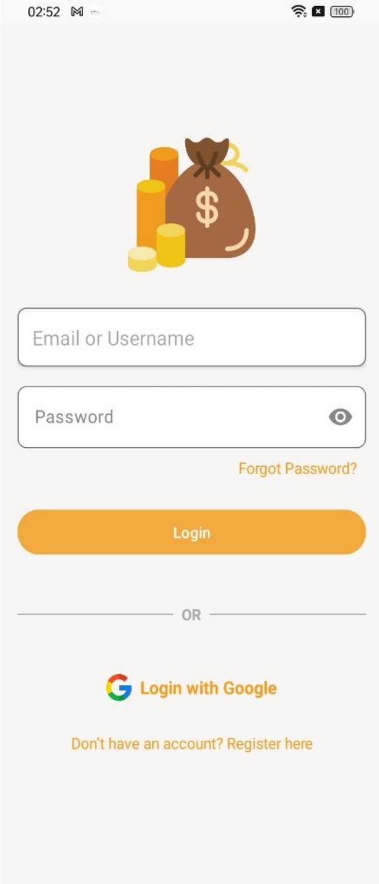
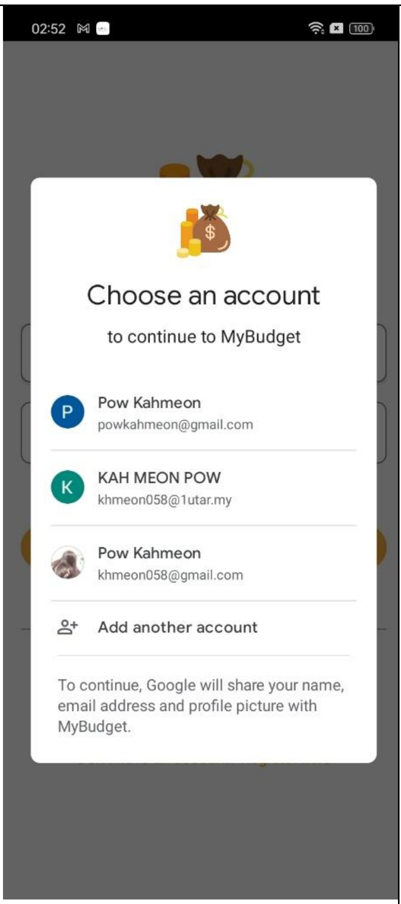


Figure 5. 11 Login page


Choose an account

to continue to MyBudget

Pow Kahmeon
powkahmeon@gmail.com

KAH MEON POW
khmeon058@1utar.my

Pow Kahmeon
khmeon058@gmail.com

Add another account

To continue, Google will share your name, email address and profile picture with MyBudget.

Figure 5. 12 Login with google

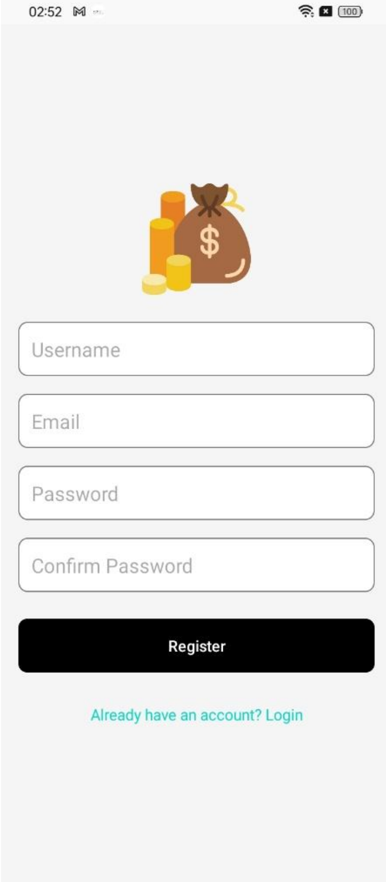
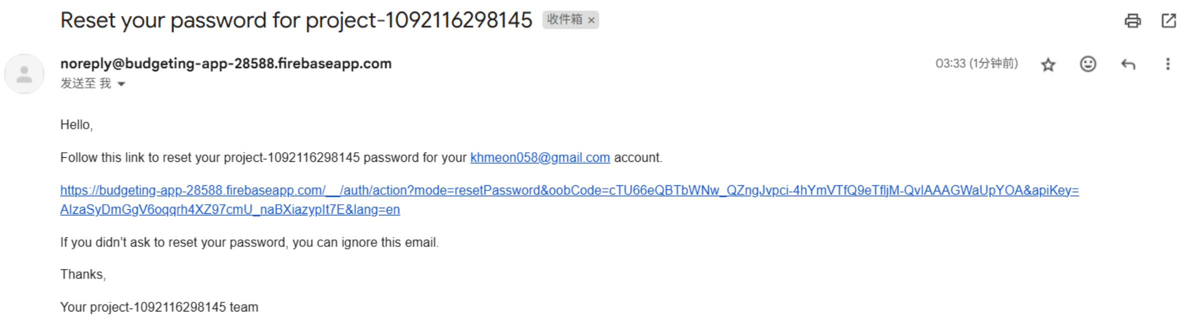


Figure 5. 13 Register Page



Reset your password for project-1092116298145

noreply@budgeting-app-28588.firebaseapp.com
发送至 我

Hello,

Follow this link to reset your project-1092116298145 password for your khmeon058@gmail.com account.

https://budgeting-app-28588.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=cTU66eQBTbWNw_QZngJvpci-4hYmVTfQ9eTfljM-QvIAAAGWaUpYOA&apiKey=AIzaSyDmGgV6oqqrh4XZ97cmU_naBXiazyplt7E&lang=en

If you didn't ask to reset your password, you can ignore this email.

Thanks,

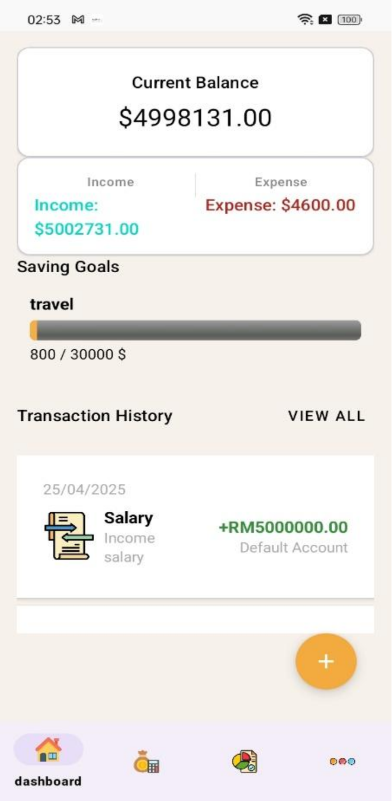
Your project-1092116298145 team

Figure 5. 14 Forgot Password email link

The authentication process begins at the login page (Figure 5.1), where the users are prompted to enter their email or username and password. If the user has forgotten the password, they can click on the "Forgot Password?" link to change the password through their registered email (Figure 5.4). For ease, the users can also choose to sign in using their Google account using the "Login with Google" button, which opens the Google account picker dialog (Figure 5.2). Upon selecting an account, Firebase Authentication handles the sign-in and grants access to the main application.

New users can click on the "Register here" link at the bottom of the login page to be taken to the registration page (Figure 5.3). Here, the users must enter a unique username, valid email, and strong password, which must be reconfirmed before submission. On registration, the user credentials are authenticated and stored securely using Firebase Authentication, and a new user document is created in the Firestore database in the /users/{userId} collection. After logging in or signing up successfully, the users are redirected to the home dashboard where they can begin to manage their finances.

5.4.2 Transaction Management Section



Current Balance
$4998131.00

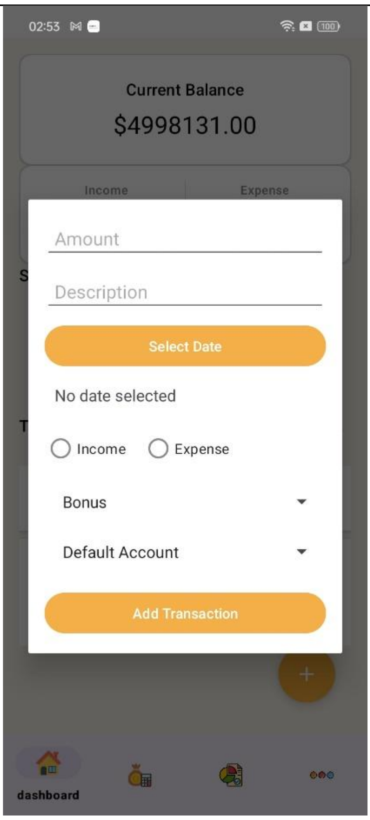
Income: $5002731.00
Expense: $4600.00

Saving Goals
travel
800 / 30000 $

Transaction History

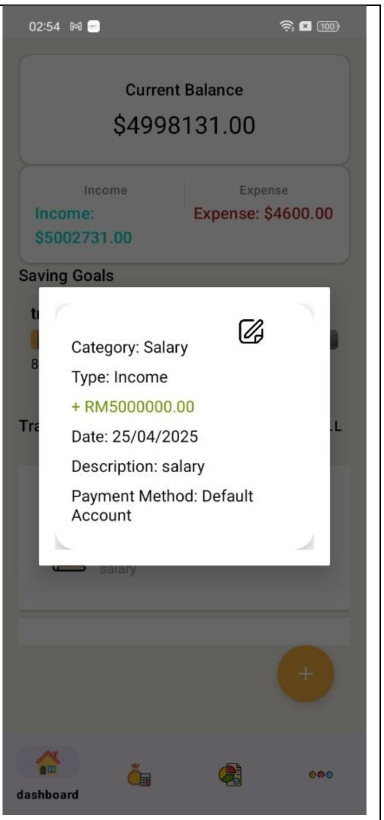
25/04/2025
Salary
Income
salary
+RM5000000.00
Default Account

Figure 5. 15 Transaction Page



Amount
Description
No date selected
○ Income ○ Expense
Bonus
Default Account

Figure 5. 16 Add Transaction Dialog



Category: Salary
Type: Income
+ RM5000000.00
Date: 25/04/2025
Description: salary
Payment Method: Default Account

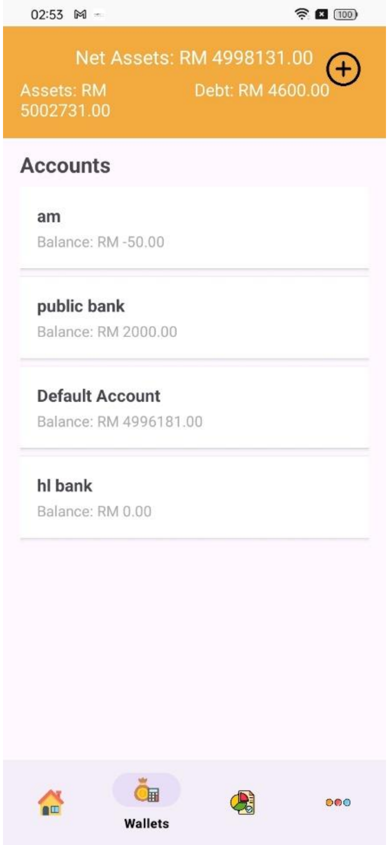
Figure 5. 17 Transaction Details Dialog

The transaction management feature allows individuals to record, view, and update their income and expenditure entries in the application. As Figure 5.5 suggests, the main page of a transaction includes the present balance, overall income, overall expenditure, and an overview of the latest transactions. People can keep tabs on their saving targets and review an overview of their economic activities.

In order to add a new transaction, users click the "+" button, which opens the add transaction dialog (Figure 5.6). Inside this dialog, users have to input the transaction amount, description, date, type of transaction (income or expense), category, and the account concerned.

Upon submission, the transaction is saved to Firestore and is also updated in balance and transaction history. Users can also view detailed details by tapping on any item within a transaction by viewing in a pop-up dialog, as indicated in Figure 5.7. The dialog shows the category, type, amount, date, description, and mode of payment used. This feature helps users monitor their financial accounts better and make better budgeting decisions.

5.4.3 Account Management Section



Net Assets: RM 4998131.00

Assets: RM 5002731.00

Debt: RM 4600.00

Accounts

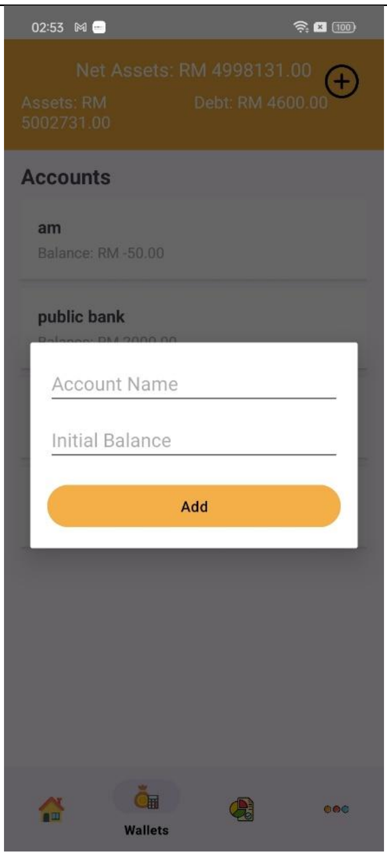
am
Balance: RM -50.00

public bank
Balance: RM 2000.00

Default Account
Balance: RM 4996181.00

hl bank
Balance: RM 0.00

Figure 5. 18 Account Page



Net Assets: RM 4998131.00

Assets: RM 5002731.00

Debt: RM 4600.00

Accounts

am
Balance: RM -50.00

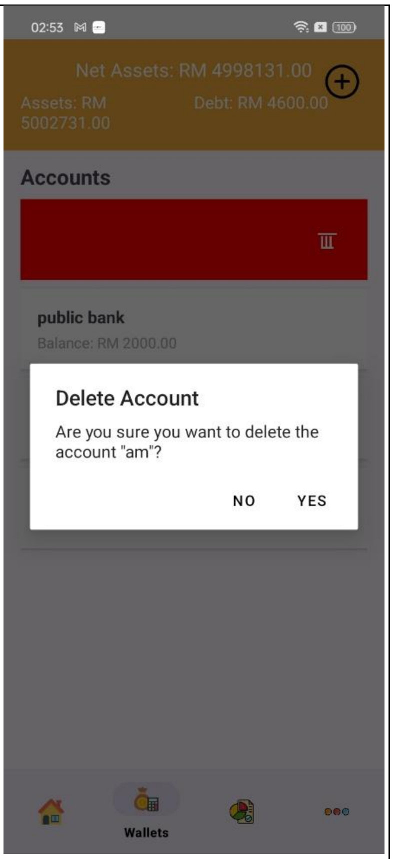
public bank

Account Name

Initial Balance

Add

Figure 5. 19 Add Account Dialog



Net Assets: RM 4998131.00

Assets: RM 5002731.00

Debt: RM 4600.00

Accounts

public bank
Balance: RM 2000.00

Delete Account

Are you sure you want to delete the account "am"?

NO YES

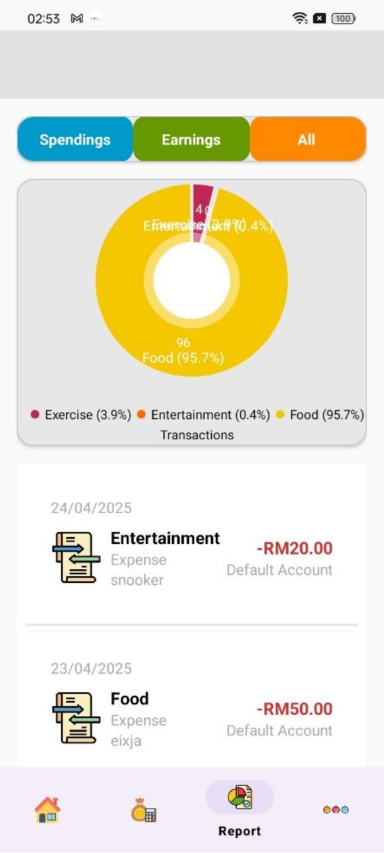
Figure 5. 20 Delete Account Confirmation Message

The account management section allows users to hold multiple bank accounts or wallets as employed in keeping their transactions. As seen in Figure 5.8, the accounts page displays a list of all accounts specified by the user and their balances. It also provides the user's worth by calculating total assets and liabilities from all accounts' balances.

Adding a new account, users can click the "\",″+\" button on the top right, which will open the add account dialog (Figure 5.9). In this dialog, users are required to enter the account name and start balance before clicking "Add" to create the account. Upon creating, the account appears in the list and is included in the total net assets calculation.

Figure 5.10 illustrates the deleted account confirmation message that is displayed when a user attempts to delete an existing account. The system requests confirmation before deletion to prevent accidental loss of data. In short, this feature enables users to have flexibility in arranging, editing, or deleting accounts at will while being able to maintain proper financial records.

5.4.4 Graph Visualization Section

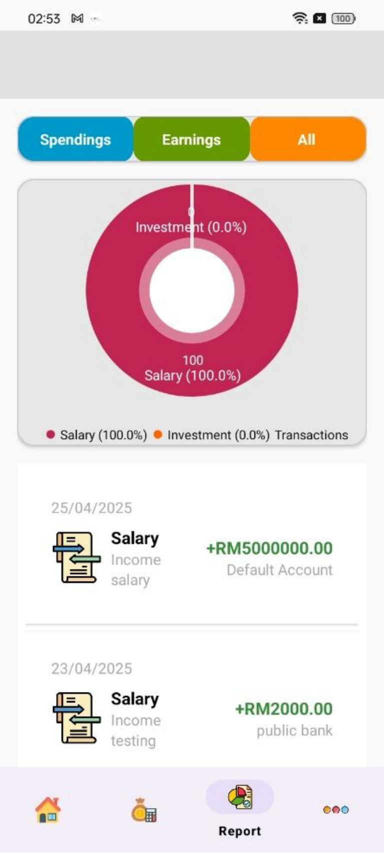


Transactions

24/04/2025
Entertainment -RM20.00
Expense Default Account
snooker

23/04/2025
Food -RM50.00
Expense Default Account
eixja

Figure 5. 21 Graph Visualization Spendings

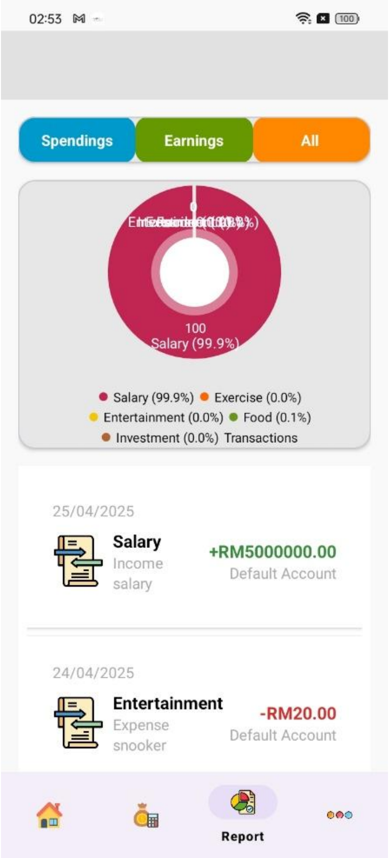


Transactions

25/04/2025
Salary +RM5000000.00
Income Default Account
salary

23/04/2025
Salary +RM2000.00
Income public bank
testing

Figure 5. 22 Graph Visualization Earnings


02:53
Spendings
Earnings
All
0
100
Salary (99.9%)
● Salary (99.9%) ● Exercise (0.0%)
● Entertainment (0.0%) ● Food (0.1%)
● Investment (0.0%)
Report

Transactions

25/04/2025
Salary +RM5000000.00
Income Default Account
salary

24/04/2025
Entertainment -RM20.00
Expense Default Account
snooker

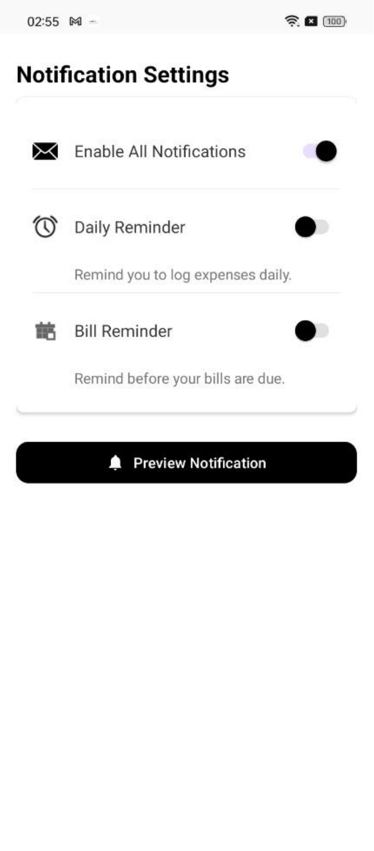
Figure 5. 23 Graph Visualization All

This section of the application provides users with a summary of their financial activity via interactive pie charts split into Spendings, Earnings, and All. As shown in Figure 5.11, the Spendings tab displays a chart graphically portraying user spending by category, such as food, entertainment, and exercise. Every category is colored and proportionally sized based on spending amounts.

Figure 5.12 illustrates the Earnings tab, where one can view income sources like salary or investment. Users can immediately see how much of their income came from every source. Figure 5.13 illustrates combined income and expense transactions to show a complete overview of all the financial activity. This view aids in understanding the balance between the income and the expenses.

Below each chart, the application provides a list of individual transactions in date, category, amount, and account form. The visual aspect provides easy viewing by the users for looking at their money activity, observing trends over a period, and establishing solid budgetary decisions based on graphic illustrations.

5.4.5 Notification Section



02:55

Notification Settings

Enable All Notifications ●

Daily Reminder ○

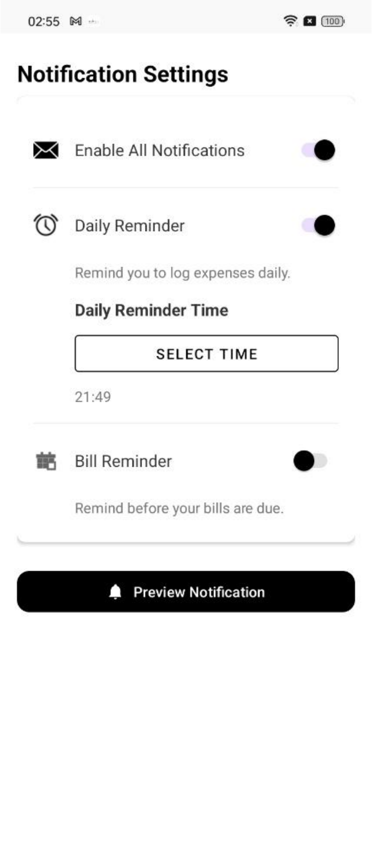
Remind you to log expenses daily.

Bill Reminder ○

Remind before your bills are due.

Preview Notification

Figure 5. 24 Notification Page



02:55

Notification Settings

Enable All Notifications ●

Daily Reminder ●

Remind you to log expenses daily.

Daily Reminder Time

SELECT TIME

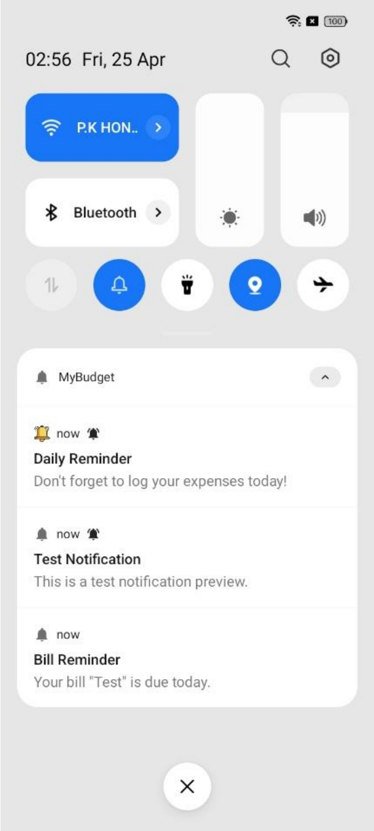
21:49

Bill Reminder ○

Remind before your bills are due.

Preview Notification

Figure 5. 25 Daily Reminder



02:56 Fri, 25 Apr

MyBudget

now

Daily Reminder

Don't forget to log your expenses today!

now

Test Notification

This is a test notification preview.

now

Bill Reminder

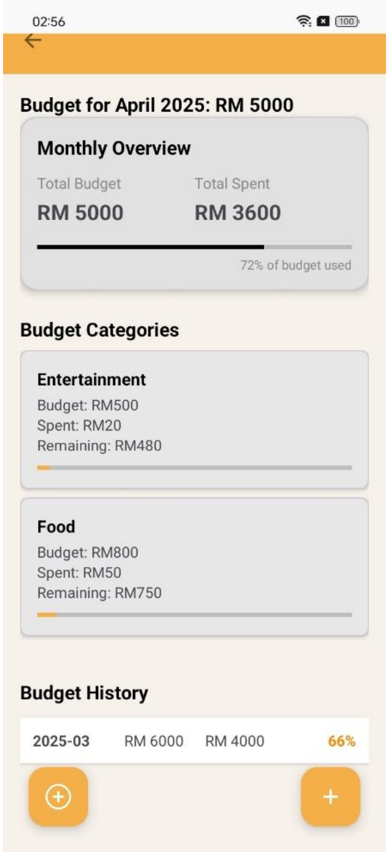
Your bill "Test" is due today.

Figure 5. 26 Output of Notification

The notification feature in the app allows the user to get helpful reminders regarding their financial activities. As can be seen from Figure 5.14, the notification settings page provides toggle switches to enable or disable all notifications with separate switches for enabling or disabling the Daily Reminder and Bill Reminder features. If the daily reminder is enabled, the user is allowed to set a specific time when they want to be notified, as can be seen from Figure 5.15. This reminder asks users to log on their daily spending and maintain frequent financial tracking.

Figure 5.16 shows the output of the notifications as seen on the user device. The system offers a timed Daily Reminder notification, a Bill Reminder when a bill is due, and a Test Notification when users are testing the settings. These notifications are accomplished using Android's AlarmManager and local notification channels so that prompt and regular alerts are delivered even when the application is not active. This serves to make the users habituated to entering their expenses and making payments at the right time.

5.4.6 Set Budget Section



Budget for April 2025: RM 5000

Monthly Overview

Total Budget: RM 5000
Total Spent: RM 3600
72% of budget used

Budget Categories

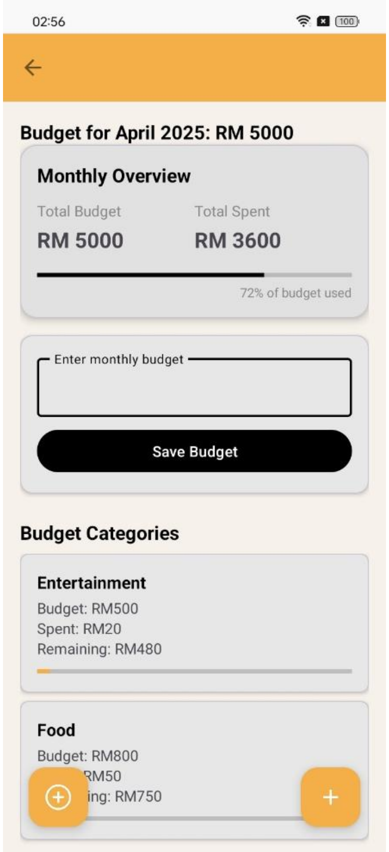
Entertainment
Budget: RM500
Spent: RM20
Remaining: RM480

Food
Budget: RM800
Spent: RM50
Remaining: RM750

Budget History

2025-03	RM 6000	RM 4000	66%

Figure 5. 27 Budget Page



Budget for April 2025: RM 5000

Monthly Overview

Total Budget: RM 5000
Total Spent: RM 3600
72% of budget used

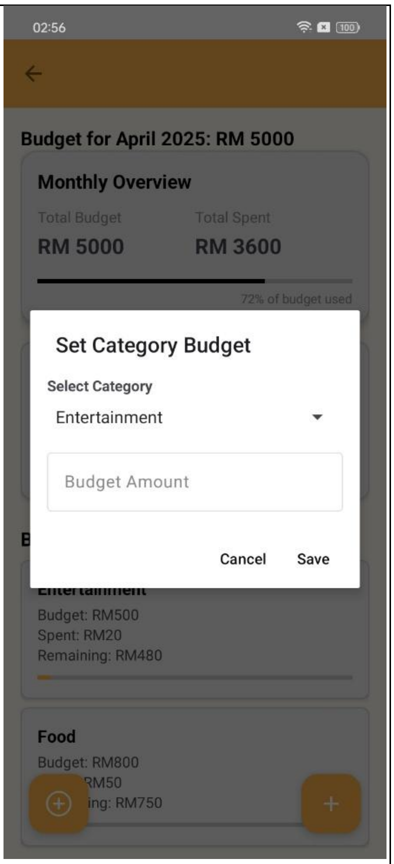
Enter monthly budget

Budget Categories

Entertainment
Budget: RM500
Spent: RM20
Remaining: RM480

Food
Budget: RM800
RM50
ing: RM750

Figure 5. 28 Set budget



Budget for April 2025: RM 5000

Monthly Overview

Total Budget: RM 5000
Total Spent: RM 3600
72% of budget used

Set Category Budget

Select Category: Entertainment

Budget Amount

Budget: RM500
Spent: RM20
Remaining: RM480

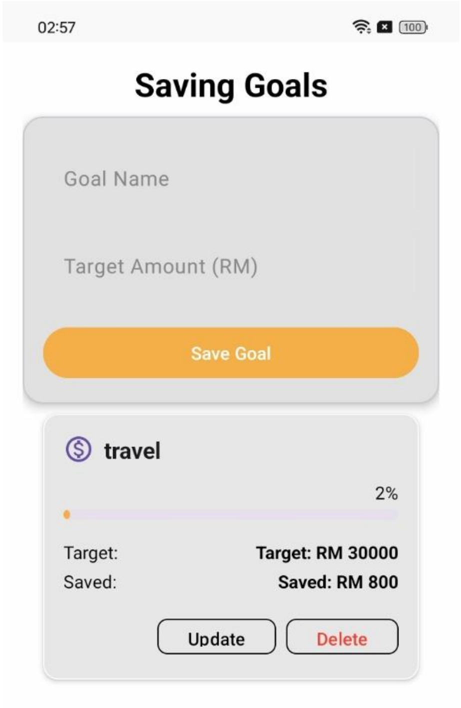
Food
Budget: RM800
RM50
ing: RM750

Figure 5. 29 Set Category Budget Dialog

The budget feature allows the users to effectively set and maintain their monthly spending limits. Figure 5.17 shows that the budget page provides an overview of the user's total monthly budget, total amount spent, and percentage use of the budget. It also displays categorical budgetary allocations and the remaining balance in each.

A new monthly budget can be inserted by pressing the left-hand side plus button at the bottom of the page. This opens the input field in Figure 5.18, where users can enter a total budget figure for the current month and save it. The right-hand side plus button is used to insert a category-specific budget. Clicking on this presents the dialog shown in Figure 5.19, where the user can choose a category (e.g., Entertainment or Food) and enter a budget amount for it.

5.4.7 Saving Goals Section



travel

2%

Target: Target: RM 30000

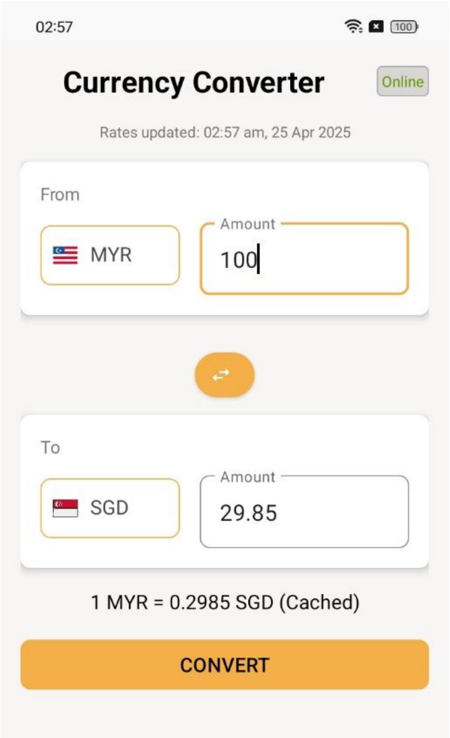
Saved: Saved: RM 800

Figure 5. 30 Saving Goals Page

Next, the Saving Goals functionality allows users to forecast and track long-term financial objectives by creating individual saving objectives. As shown in Figure 5.20, the user can create a goal by typing in the goal name and target amount before clicking the Save Goal button. Once the goals have been added, they are displayed in a list with their names, total target amount, and amount saved thus far. A progress bar and percentage meter graphically show how much of the target has been achieved.

For example, the screenshot displays a goal named "travel" with the target value of RM30,000 and RM800 saved, corresponding to 2% progress. Users can edit the amount saved by using the Update button or delete the goal entirely using the Delete button. This feature encourages users to save appropriately and be dedicated to attaining certain financial goals.

5.4.8 Currency Converter Section



02:57

Currency Converter

Online

Rates updated: 02:57 am, 25 Apr 2025

From

MYR

Amount: 100

To

SGD

Amount: 29.85

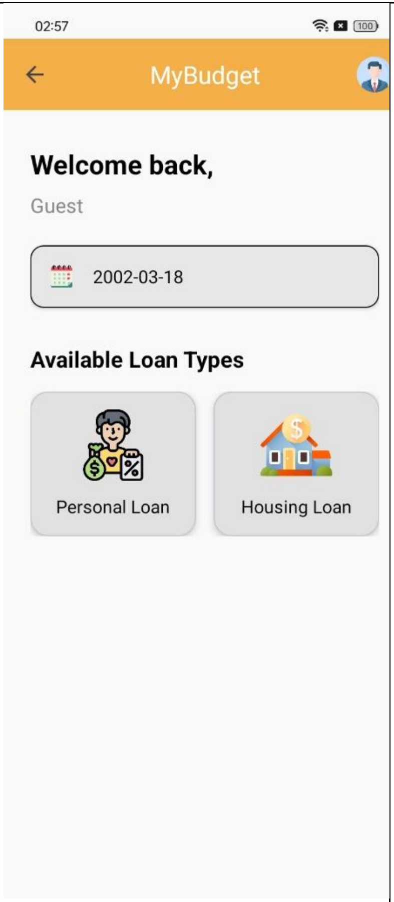
1 MYR = 0.2985 SGD (Cached)

Figure 5. 31 Exchange Currency Converter Page

The Currency Converter feature allows the conversion of one currency to another with up-to-date exchange rates. As shown in Figure 5.21, users can choose the source currency (From), enter the target amount, and choose the target currency (To). When the Convert button is clicked, the equivalent amount is calculated and displayed instantly based on the latest available exchange rate.

In this example, the user is converting 100 MYR into SGD and getting the returned value as 29.85 SGD. The app also shows the exchange rate applicable (i.e., 1 MYR = 0.2985 SGD), as well as a status icon denoting whether rates are live or caught. It is especially handy for regular flyers or global users to quickly convert using the app to the accurate figure.

5.4.9 Calculator Sections



02:57

MyBudget

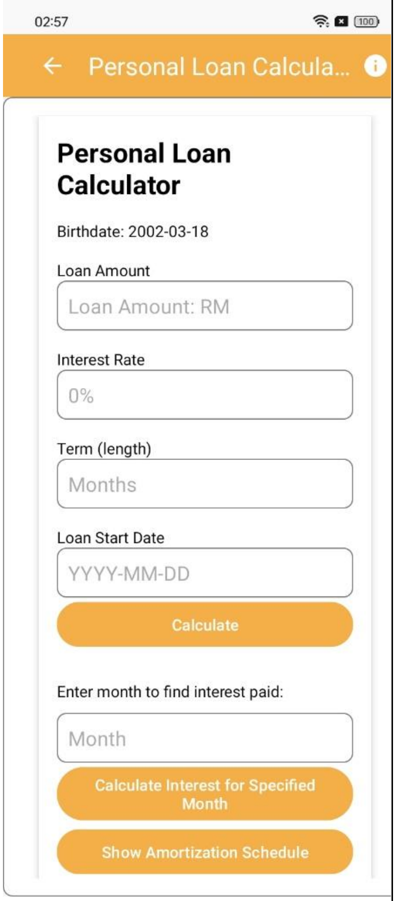
Welcome back,

Guest

2002-03-18

Available Loan Types

Figure 5. 32 Loan Type Selection



02:57

Personal Loan Calcula...

Personal Loan Calculator

Birthdate: 2002-03-18

Loan Amount: Loan Amount: RM

Interest Rate: 0%

Term (length): Months

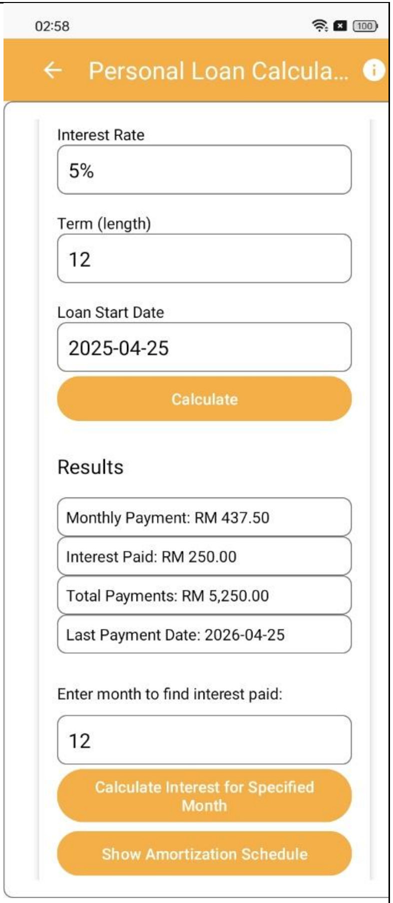
Loan Start Date: YYYY-MM-DD

Enter month to find interest paid:

Month

Calculate Interest for Specified Month
Show Amortization Schedule

Figure 5. 33 Loan Form



02:58

Personal Loan Calcula...

Interest Rate: 5%

Term (length): 12

Loan Start Date: 2025-04-25

Results

Monthly Payment: RM 437.50

Interest Paid: RM 250.00

Total Payments: RM 5,250.00

Last Payment Date: 2026-04-25

Enter month to find interest paid:

12

Calculate Interest for Specified Month
Show Amortization Schedule

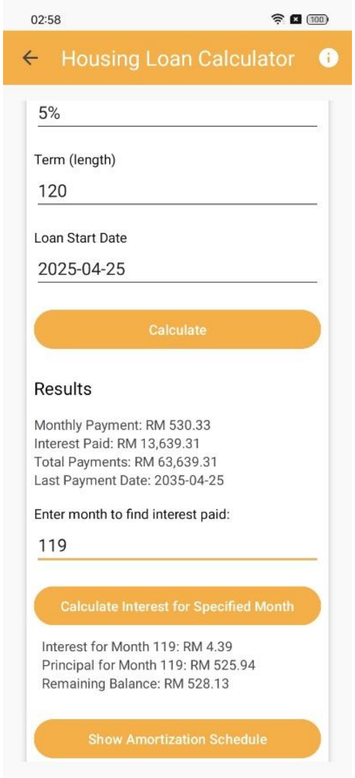
Figure 5. 34 Calculate Output

The calculator feature provides the users with facilities to compute their monthly loan repayments and get a better idea of what their financial commitment is. As can be seen from Figure 5.22, the users begin by selecting the kind of loan they would like to calculate either Housing Loan or Personal Loan. This directs them to the respective input form where detailed loan information can be entered.

Figure 5.23 is the form for calculating a loan, where the users enter the amount of the loan, rate of interest, length of the loan in months, and the date of the loan. After they press the

Calculate button, the system computes the entered values and displays the output as in Figure 5.24. The output provides the calculated monthly payment, interest paid, payment, and the date of last payment.

In addition, the app enables the entry of a specific month for determining the interest portion paid in a month or tapping on Show Amortization Schedule to discover a detailed breakdown of principal and interest payment for the loan period. It's particularly helpful when it relates to financial planning, comparing the two loan scenarios, and making appropriate borrowing decisions for users.



02:58
Housing Loan Calculator

5%

Term (length)

120

Loan Start Date

2025-04-25

Results

Monthly Payment: RM 530.33
Interest Paid: RM 13,639.31
Total Payments: RM 63,639.31
Last Payment Date: 2035-04-25

Enter month to find interest paid:

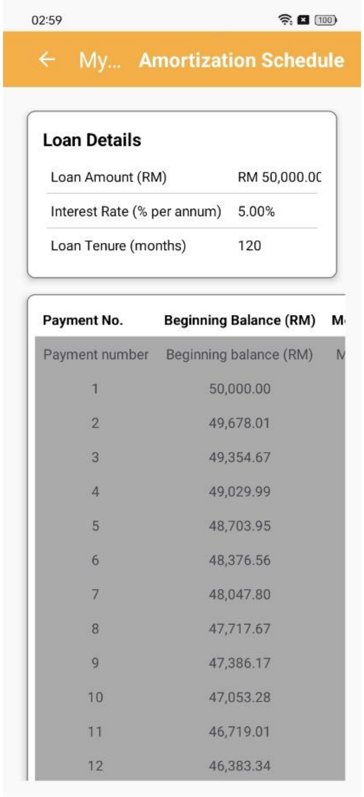
119

Calculate Interest for Specified Month

Interest for Month 119: RM 4.39
Principal for Month 119: RM 525.94
Remaining Balance: RM 528.13

Show Amortization Schedule

Figure 5. 35 Calculate Interest for specific Month output

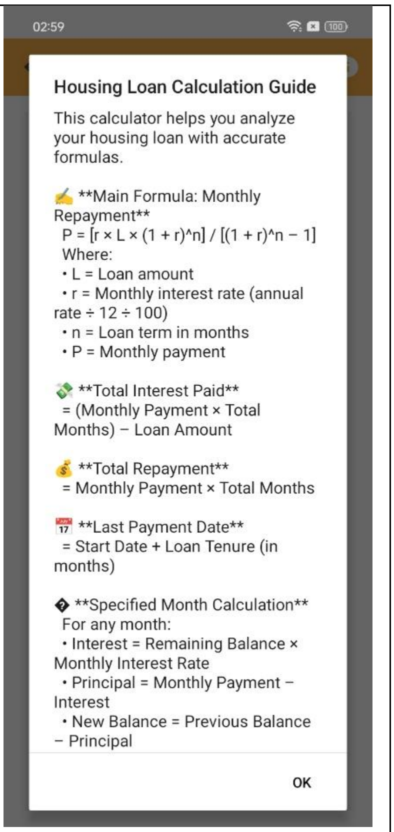


Loan Details

Loan Amount (RM)	RM 50,000.0C
Interest Rate (% per annum)	5.00%
Loan Tenure (months)	120

Payment No.	Beginning Balance (RM)	M
Payment number	Beginning balance (RM)	M
1	50,000.00	
2	49,678.01	
3	49,354.67	
4	49,029.99	
5	48,703.95	
6	48,376.56	
7	48,047.80	
8	47,717.67	
9	47,386.17	
10	47,053.28	
11	46,719.01	
12	46,383.34	

Figure 5. 36 Show Amortization Schedule



Housing Loan Calculation Guide

This calculator helps you analyze your housing loan with accurate formulas.

✍️ **Main Formula: Monthly Repayment**
P = [r × L × (1 + r)^n] / [(1 + r)^n − 1]
Where:
• L = Loan amount
• r = Monthly interest rate (annual rate ÷ 12 ÷ 100)
• n = Loan term in months
• P = Monthly payment

💸 **Total Interest Paid**
= (Monthly Payment × Total Months) − Loan Amount

💰 **Total Repayment**
= Monthly Payment × Total Months

📅 **Last Payment Date**
= Start Date + Loan Tenure (in months)

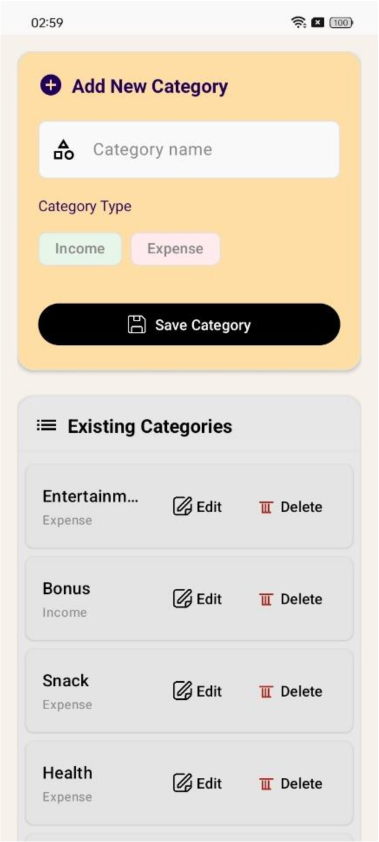
� **Specified Month Calculation**
For any month:
• Interest = Remaining Balance × Monthly Interest Rate
• Principal = Monthly Payment − Interest
• New Balance = Previous Balance − Principal

Figure 5. 37 Loan Details

As Figure 5.25 shows, users enter interest rate, loan duration (in months), and loan start date, then press Calculate to obtain results. The results are monthly payment, interest paid, payment, and last payment date. Users can also calculate interest and principal for a specific month by entering month number and pressing Calculate Interest for Specified Month.

Figure 5.26 illustrates the Amortization Schedule, a detailed table charting each month's payment. It includes the beginning balance, monthly payment, interest paid, principal paid, and ending balance for every loan month. This is especially helpful for users intending to make advance payments or track long-term loan progress.

To enable users to understand how each value is calculated, Figure 5.27 shows a Loan Calculation Guide that outlines the formulas used. It enumerates the principal monthly repayment formula, total interest and repayment calculations, last payment date logic, and how specific month values are derived.



Entertainm...
Expense

Bonus
Income

Snack
Expense

Health
Expense

Figure 5. 38 Customize Category

The Customize Category functionality gives users the power to create and manage their own transaction categories, giving users more flexibility and customization in money tracking. As shown in Figure 5.28, users can add a new category by entering a name and selecting its type as Income or Expense. With the information entered, clicking on the Save Category button will save the new category into the system.

Under the input form, a list of available categories is displayed under Existing Categories. Each category item displays its name and type and Delete button. This allows users to easily change or delete categories as their spending habits evolve. By giving users full control over their system of categorization, this feature makes the app more user-friendly and allows it to support diverse personal budgeting habits.

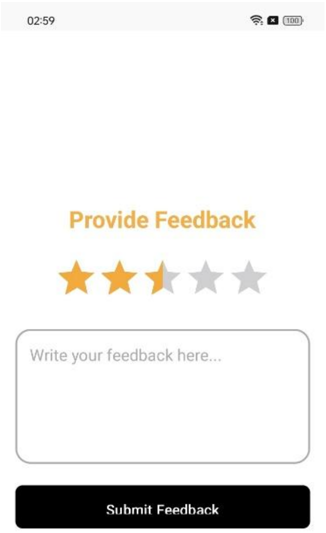
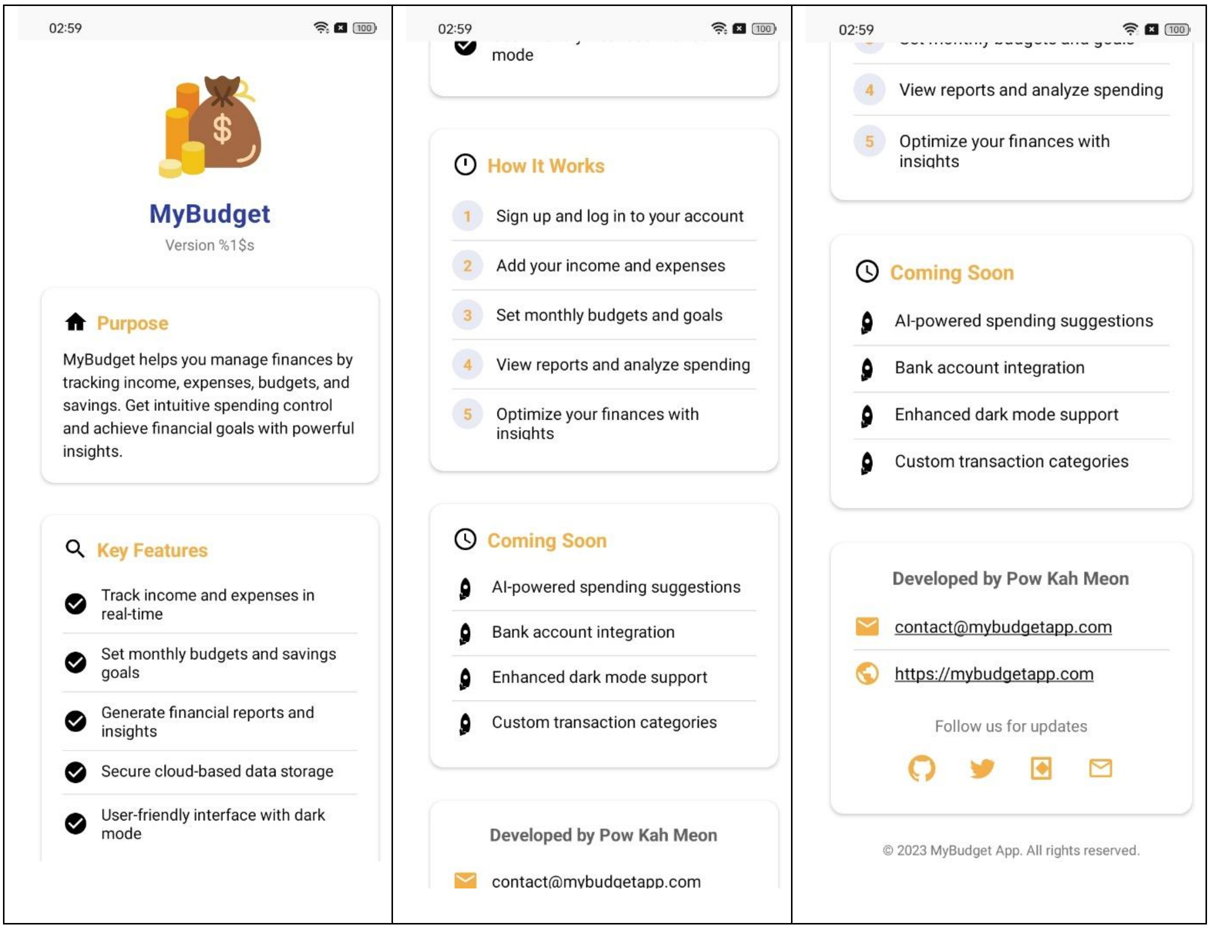


Figure 5. 39 Feedback page

The Feedback page allows users to provide feedback and experience with the app. As shown in Figure 5.29, users are allowed to rate the app in terms of a star rating and provide written feedback in the feedback text field. Once done, they can tap the Submit Feedback button to provide feedback to the system.

This feature is designed to collect suggestions from users, report issues, or highlight features that they enjoy. Feedback is stored in the Firestore database and can be reviewed by the developer to enhance the app's quality, functionality, and user satisfaction in future updates.



02:59

MyBudget

Version %1$s

Purpose

MyBudget helps you manage finances by tracking income, expenses, budgets, and savings. Get intuitive spending control and achieve financial goals with powerful insights.

Key Features

- ✓ Track income and expenses in real-time
- ✓ Set monthly budgets and savings goals
- ✓ Generate financial reports and insights
- ✓ Secure cloud-based data storage
- ✓ User-friendly interface with dark mode

02:59

✓ mode

How It Works

1. Sign up and log in to your account
2. Add your income and expenses
3. Set monthly budgets and goals
4. View reports and analyze spending
5. Optimize your finances with insights

Coming Soon

- AI-powered spending suggestions
- Bank account integration
- Enhanced dark mode support
- Custom transaction categories

Developed by Pow Kah Meon

contact@mybudgetapp.com

02:59

4. View reports and analyze spending
5. Optimize your finances with insights

Coming Soon

- AI-powered spending suggestions
- Bank account integration
- Enhanced dark mode support
- Custom transaction categories

Developed by Pow Kah Meon

contact@mybudgetapp.com

https://mybudgetapp.com

Follow us for updates

© 2023 MyBudget App. All rights reserved.

Figure 5. 40 About us Page

The About Us page provides users with an immediate overview of the purpose, functionalities, and history of development of the MyBudget app. As shown in Figure 5.30, this page is composed of four significant elements: Purpose, Key Features, How It Works, and Coming Soon.

The Purpose highlights the app's mission of helping users manage their finances by tracking income, expenses, budgets, and savings to gain financial stability. Real-time income and expense tracking, goal setting, financial statements, cloud backup of data, and dark mode are features enumerated under Key Features.

The How It Works page guides users through a five-step sign-up process, from log in to optimizing finances with the app's features. Coming Soon details features in development such as AI-based spending suggestions, bank integration, and improved customization options. Developer contact information and social media handles are provided at the bottom for user support and future updates, with an invitation for openness and continuous improvement.

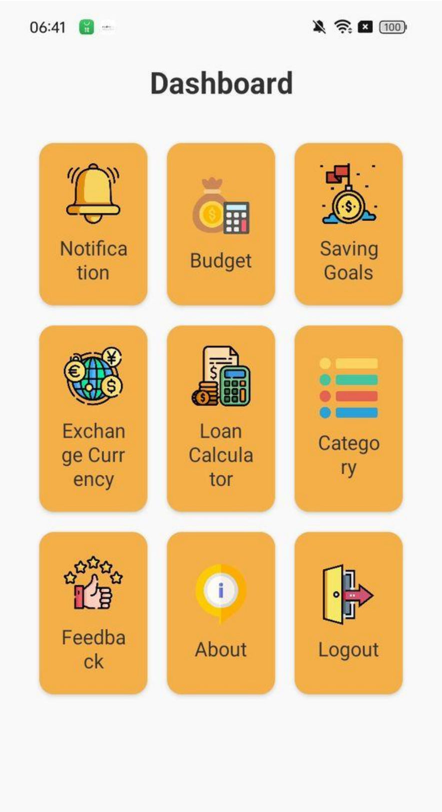


Figure 5. 41 Setting Page

In figure 5.31 showing the setting page layout. The setting page serves as the main navigation hub of the MyBudget application, which providing a quick access to core features such as Notification, Budget, Saving Goals, Exchange Currency, Loan Calculator, Category, Feedback, About, and Logout. It is designed for intuitive use with clear icons and organized layout to enhance user experience

CHAPTER 6 SYSTEM TESTING AND PERFORMANCE METRICS

This chapter gives an overview of the testing methods and performance measures utilized in evaluating the MyBudget application. System testing was performed to ensure that all features work as anticipated, the user interface reacts appropriately, and data is handled appropriately through Firebase services.

6.1 System Testing and Performance Metrics

System testing was conducted to verify the overall functionality, stability, and performance of the **My Budget** mobile application. This testing phase was essential to ensure that all features behave as intended, provide accurate results, and deliver smooth user experience. The testing process focused on both **functional** and **non-functional** aspects of the application.

The application was tested using the Black Box Testing approach. In this method, the internal code of the application is not considered. Instead, the system is tested from the user's perspective, focusing on inputs and the corresponding outputs. Each module was tested by entering valid and invalid data to observe whether the application produced the correct results or displayed appropriate error messages. Black Box Testing was chosen because it closely simulates how actual users will interact with the app. It is especially suitable for user-interface-based mobile applications like MyBudget, where the emphasis is on usability, data accuracy, and reliable output.

6.1.1 Functional Testing

Functional testing was focused on making sure all individual functions of the application execute as needed in the system requirement. All main modules were both tested with positive test cases (correct inputs) and negative test cases (erroneous or edge-case inputs). The objective was to validate the application so that it properly processes correct input and rejects erroneous input gracefully, without generating application crashes or corrupted data.

The modules that were functionally tested are:

Module	**Test Coverage / Description**
User Authentication	Verified logging in, signing up, Google authentication, password reset, and error handling on invalid credentials.
Transaction Management	Verified addition, editing, and removal of income/expense transactions, filter functionality, and showing history records.
Budget Management	Verified functionality for correct setting of monthly and category budgets, budget validity checks, and update routines.
Saving Goals	Verified creation, update, and removal of saving goals, and monitored progress bar functionality for target tracking.
Notifications	Validated for daily verification and billing reminder scheduling, activation, and user interaction.
Loan & Currency Calculators	Validated formula accuracy, field validation, and proper result display.
Visualization & Reports	Validated charts properly represented categorized spendings/earnings and displayed figures with accurate percentages.
Category Customization	Validated users were able to add, edit, and delete custom categories and apply them to transactions.
Feedback Submission	Ensured that feedback and ratings correctly persisted in Firestore for the current user.

6.1.2 Non-Functional Testing

Non-functional testing focuses on how well the system performs in terms of speed, stability, reliability, and usability. This testing ensures the system not only functions correctly but also offer smooth and efficient experience to users in real-world conditions. The following aspects were evaluated:

Metric	**Description / Result**
Startup Time	The application launched in approximately 2–3 seconds.
Screen Responsiveness	Navigation between different screens occurred without noticeable delays.
Data Sync Speed	Data entered into forms synced with Firebase Firestore within 1–2 seconds on average.
Offline Functionality	The app functioned correctly without internet access. Data entered offline was saved locally and automatically synced when connectivity was restored.
Notification Timing	Daily and monthly bill notifications appeared at the exact scheduled time using the AlarmManager.
Crash and Stability Testing	The app was used continuously for 20 minutes with various input scenarios and showed no crashes or ANRs (Application Not Responding errors).
Input Validation	Fields throughout the app handled incorrect, empty, or mismatched input gracefully and showed appropriate validation messages.

6.1.4 Devices and Tools Used

Tool / Device	**Purpose**
Android Studio	Development, manual testing, debugging via emulator and logs
Firebase Console	Monitoring Firestore data, authentication, and cloud sync behavior
Android Smartphone (10+)	Real device testing for real-user experience
Stopwatch / Timer	Measuring screen load time and notification timing
Android Profiler	Observing memory usage and performance metrics

6.2 Testing Setup and Result

6.2.1 Authentication Module

TestCase ID	**Feature/ Module**	**Test Scenario**	**Input**	**Expected Output**	**Actual Output**	**Status**
TC01	Login	Valid login credentials	Email: khmeon058@1utar.my Password: 020508pow	Dashboard screen is displayed	Dashboard screen is displayed	Pass
TC02	Login	Empty fields	Email: " Password: ""	Show error "Fields cannot be empty"	Show error "Fields cannot be empty"	Pass
TC03	Login	Incorrect credentials	Email: wrong@gmail.com Password: 123456	Show error message: "Invalid email or password"	Show error message: "Invalid email or password"	Pass
TC04	Login	First-time login without profile	Email: khmeon058@1utar.my Password:020508pow	Redirect to Setup Profile screen after login	Redirect to Setup Profile screen after login	Pass
TC05	Register	Valid registration	Email: khmeon058@1utar.my Password:020508pow	Account created and redirected to Setup Profile	Account created and redirected to Setup Profile	Pass
TC06	Register	Invalid email format	Email: user1.com Password: Abc@1234	Show error: "Invalid email format"	Show error: "Invalid email format"	Pass
TC07	Register	Password too short	Email: test@gmail.com Password: 123	Show error: "Password must be at least 6 characters"	Show error: "Password must be at	Pass

					least 6 characters"	
TC08	Register	Email already registered	Email: khmeon058@1utar.my Password:020508pow	Show error: "Email already in use"	Show error: "Email already in use"	Pass
TC09	Google Sign-In	Sign in with existing Google account	Valid Google account	Redirect to Home screen / Setup Profile	Redirect to Home screen / Setup Profile	Pass
TC10	Google Sign-In	Sign in with no existing account in app	New Google account	Account created prompt to complete profile	Account created prompt to complete profile	Pass
TC11	Google Sign-In	No Internet during sign-in	No connection	Show error message	Show error message	Pass
TC12	Password Reset	Valid email input	khmeon058@1utar.my	Password reset email sent	Password reset email sent	Pass
TC13	Password Reset	Empty email field	(empty)	Show error: "Email is required"	Error message shown	Pass
TC14	Password Reset	Invalid email format	user@@gmail	Show error: "Invalid email format"	Error message shown	Pass
TC15	Password reset	Non-registered email address	notfound@example.com	Show: "No user found with this email	Error message shown	Pass

Table 6. 1 Authentication Module Testing Setup And Result

6.2.2 Transaction Management

Test Case ID	Feature/ Module	Test Scenario	Input	Expected Output	Actual Output	Status
TC16	Transaction	Add valid income	Amount: 500, Desc: Salary	Transaction saved successfully	Transaction saved successfully	Pass
TC17	Transaction	Add blank amount	Amount: "", Desc: Rent	Show validation error	Show validation error	Pass
TC18	Transaction	Add blank description	Amount: 300, Desc: “”	Show validation error	Show validation error	Pass
TC19	Transaction	Add large amount	Amount: 999999999, Desc: Car	Transaction saved successfully	Transaction saved successfully	Pass
TC20	Transaction	Edit amount of existing transaction	Change amount from 50 to 80	Updated transaction saved successfully	Updated transaction saved successfully	Pass
TC21	Transaction	Edit category	Change category from "Food" to "Travel"	Updated transaction saved successfully	Updated transaction saved successfully	Pass
TC22	Transaction	Edit with blank description	Leave description empty	Show validation error	Show validation error	Pass
TC23	Transaction	Delete an existing transaction	Swipe to delete a valid transaction	Transaction removed from list	Transaction removed from list	Pass
TC24	Transaction	Cancel delete confirmation	Tap delete, then tap "No" on confirmation	Transaction remains unchanged	Transaction remains unchanged	Pass
TC25	Transaction	Delete transaction while offline	Delete transaction with no internet	Marked for deletion, syncs when online	Marked for deletion, syncs when online	Pass
TC26	Transaction List	Display all transactions	App launched	Transactions are shown in	Transactions are shown in	Pass

			with transactions	descending order	descending order	
TC27	Transaction List	Display empty state when no transactions exist	App launched with no transactions	Show "No transactions found" message	Show "No transactions found" message	Pass
TC28	Transaction List	Display transaction details	Tap on transaction card	Show transaction category, amount, account	Show transaction category, amount, account	Pass
TC29	Transaction List	Offline transaction list loading	Launch app without internet	Transactions loaded from local cache	Transactions loaded from local cache	Pass
TC30	Category and date selection	Select a category from dropdown	Choose "Food" from category spinner	Category "Food" is selected and saved correctly	Category saved correctly	Pass
TC31	Category and date selection	Select a valid transaction date	Select 24/04/2025 using date picker	Selected date displayed in input and saved	Selected date displayed in input and saved	Pass
TC32	Category and date selection	Select a future date (not allowed)	Select 31/12/2030	User not allowed to select future date	User not allowed to select future date	Pass
TC33	Category and date selection	Select category with no option available	Spinner empty (no categories exist)	Show "No categories available" or keep spinner disabled	Proper message or disabled state shown	Pass

Table 6. 2 Tansaction Management Moduule Testing Setup and Result

Test Case ID	Feature/ Module	Test Scenario	Input	Expected Output	Actual Output	Status
TC34	Set budget	Set a valid budget amount	1000	Budget saved successfully and displayed	Budget saved successfully and displayed	Pass
TC35	Set budget	Leave the budget field empty	Empty input	Show error message "Please enter a valid amount"	Error shown	Pass
TC36	Set budget	Enter an invalid value	1000.99	Show error "Invalid amount"	Error shown	Pass
TC37	set category budget	Set valid budget for a category	Category: Entertainment Amount: 3000	Budget saved and reflected under category	Budget saved successfully	Pass
TC38	set category budget	Leave amount field empty	Category: Transport, Amount: (empty)	Show error "Please enter a valid amount"	Error shown	Pass
TC39	set category budget	Set Existing Category Budget	Category: Entertainment Amount: 6000	Budget saved and reflected under category	Budget updated successfully	Pass

Table 6. 3 Budget Management Module Testing Setup and Result

6.2.4 Saving Goals

Test Case ID		Test Scenario	Input	Expected Output	Actual Output	Status
TC40	Add new goal	-validate progress bar	Goal Name: "New Car", Target: RM 20,000	Goal added successfully to the list	Goal added successfully to the list	Pass
TC41	Add new goal	Add goal with empty name	Goal Name: (empty), Target: RM 10,000	Show error "Please fill in all fields"	Error shown	Pass
TC42	Add new goal	Add goal with invalid target amount	Goal Name: "Vacation", Target: "1000.20"	Show error "Invalid amount format"	Error shown	Pass
TC43	Add new goal	Add goal with empty amount	Goal Name: New car Target: “”	Show error "Please fill in all fields"	Error shown	Pass
TC44	validate progress bar	Add partial savings	Goal Target: RM 2,000, Saved: RM 500	Progress bar updates to 25%	Progress shown correctly	Pass
TC45	validate progress bar	Fully achieve goal	Goal Target: RM 1,000, Saved: RM 1,000	Progress bar shows 100%	Progress 100% displayed	Pass
TC46	Delete goal	Delete an existing saving goal	Select goal "New Car" → Delete	Goal removed from the list immediately	Goal successfully deleted	Pass

Table 6. 4 Saving Goals Management Module Testing Setup and Result

6.2.5 Bill reminder/Notification

Test Case ID	Feature/ Module	Test Scenario	Input	Expected Output	Actual Output	Status
TC47	Add Bill Reminder	Add a bill with correct name/date/amount	Name: "WiFi", Amount: RM100, Date: 1st May	Reminder added and listed under bill reminders	Reminder added successfully	Pass
TC48	Bill Notification	Verify notification triggers on due date	Date set to today, time: 9:00AM	Notification is shown at scheduled time	Notification received	Pass
TC49	Monthly Repeat	Ensure reminder recurs monthly	Name: "Rent", Date: 5th, Repeats Monthly	Notification appears monthly on the same date	Reminder repeated correctly	Pass
TC50	Enable/Disable Reminder	Disable a specific reminder	Toggle bill reminder OFF	Reminder will not show when disabled	Notification didn’t show	Pass
TC51	Daily Reminder	Check daily reminder notification	Daily Reminder set at 8:00AM	Notification triggers daily at 8:00AM	Daily notification triggered	Pass

Table 6. 5 Notification Management Module Testing Setup and Result

6.2.6 Charts and reports

Test Case ID	Feature/ Module	Test Scenario	Input	Expected Output	Actual Output	Status

TC52	View Expense Chart	Open report and view expense pie chart	User navigates to "Reports" > "Expenses"	Pie chart showing expense categories appears	Chart rendered correctly	Pass
TC53	View Income Chart	Open report and view income pie chart	User navigates to "Reports" > "Income"	Pie chart showing income sources appears	Chart rendered correctly	Pass
TC54	View All Transactions	Open full chart view of all transactions	User selects "All" filter	Combined view of expenses and income is shown	Chart rendered correctly	Pass
TC55	Data Consistency	Compare chart totals with transaction list	The chart shows RM400 spent, transaction list also RM400 spent	Chart total matches transaction total	Data matches	Pass
TC56	No Data Case	Open chart with no transactions logged	New user opens report	"No data to display" message shown	Message shown	Pass

Table 6. 6 Charts and Reports Management Module Testing Setup and Result

6.2.7 Currency converter

Test Case ID	Feature/ Module	Test Scenario	Input	Expected Output	Actual Output	Status
TC57	Convert Valid Currencies	Convert between valid currencies (e.g., USD → MYR)	Enter 100 USD and select MYR	Correct converted amount displayed	Output corrects	Pass
TC58	Decimal Amount	Enter a decimal amount	Enter 500.1	Correct converted amount displayed	Output corrects	Pass
TC59	Validate Output Correctness	Cross-check currency conversion with API rates	Enter 1 USD to MYR and manually check the rate	Output matches real exchange rate within tolerance	Output corrects	Pass
TC60	Different Currency Pairs	Convert between multiple different currencies	Try EUR → USD, MYR → JPY, GBP → INR	All conversions complete without crashing	Smooth conversion	Pass

Table 6. 7 Currency Convertor Management Module Testing Setup and Result

6.2.8 Loan calculator

Test Case ID	Feature/ Module	Test Scenario	Input	Expected Output	Actual Output	Status
TC61	Basic Loan Calculation	Calculate loan normally	Principal = 200,000, Tenure = 30 years, Rate = 4%	Monthly payment calculated correctly	Correct calculation	Pass
TC62	Edge Case: Long Tenure	Input long tenure (e.g., 70 years)	Principal = 100,000, Tenure = 70 years, Rate = 3%	Error Message	Eror Message	Pass
TC63	Age Limit Exceeded	Simulate applicant age over 70 years	Age = 75, Principal = 150,000	Show warning "Age exceeds limit"	Warning displayed	Pass
TC64	View Interest Breakdown	View breakdown for a specific month	Select month 12	Correct interest/principal breakdown shown	Breakdowns correct	Pass
TC65	View Amortization Schedule	Open full amortization schedule	Open full schedule	Show all months, interest and principal portion	Table displayed	Pass
TC66	Blank Field Validation	Submit without filling fields	Leave fields blank and click calculate	Error message "Please fill in all fields"	Error handled	Pass

Table 6. 8 Calculator Management Module Testing Setup and Result6.2.9 Customize categories

Test Case ID	Feature/ Module	Test Scenario	Input	Expected Output	Actual Output	Status
TC67	Add New Category	Add a new category	Category Name: "Investment", Type: "Expense"	Category saved and listed	Category added successfully	Pass
TC68	Duplicate Category Check	Try adding the same category twice	Add "Investment" again	Error: "Category already exists"	Duplicate prevented	Pass
TC69	Delete Existing Category	Delete a previously created category	Delete "Investment" from list	Category removed successfully	Category deleted	Pass
TC70	Category Type Selection	Choose Income or Expense correctly	Category Name: "Gift", Type: "Income" selected	Saved under correct type	Correctly categorized	Pass

Table 6. 9 Category Management Module Testing Setup and Result

6.2.10 Non-functional

Id	Category	Test Scenario	Test Tool	Expected Result	Actual Result	Status
TC71	Performance	App loads dashboard within 3 seconds	Manual / Timer	Dashboard loads ≤ 3 sec	Dashboard loads in 2.5 sec	Pass
TC72	Usability	App layout follows Material Design 3 guidelines	UI Review Checklist	UI is clean, intuitive, and responsive	UI matches MD3 standards	Pass
TC73	Reliability	App handles offline transaction saving without crashing	Manual	Transactions saved offline and synced when online	App crashed when adding 3+ offline items	Partial
TC74	Compatibility	App works on Android 10, 11, 12, 13	Emulator / Real devices	No crashes or layout issues across versions	Works on all but small layout issues on Android 10	Minor
TC75	Scalability	App handles 1000+ transactions without slowdown	Insert test data manually	Smooth scrolling and filtering	Noticeable lag after 80+ records	Partial
TC76	Security	User passwords are securely stored (not in plain text)	Firebase Authentication	Encrypted/authenticated access only	Encrypted/authenticated access only	Pass
TC77	Maintainability	Codebase is modular and uses proper naming conventions	Code review	Easy to maintain and update	Code is clean and well-commented	Pass
TC78	Accessibility	App supports screen readers and large fonts	Device settings	All major screens are accessible	Some minor label issues on buttons	Need Fix

Table 6. 10 Non-Functional Testing Setup and Result6.3 Project Challenges

6.3 User Acceptance Testing (UAT) / User Feedback Analysis

User Acceptance Testing (UAT) for the Personal Finance Management Mobile Application was conducted to check the usability, functionality, and performance of the application. The aim was to get honest feedback from real users, primarily students and young professionals, to determine how well the application meets their needs and expectations in personal finance management.

6.3.1 Age Distribution of Respondents

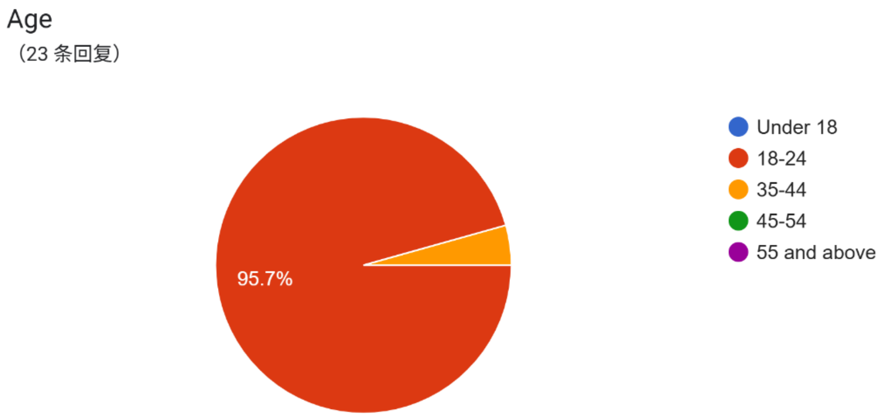


Figure 6. 1 Age Distribution of Respondents

This section presents the age distribution of the 20 respondents who participated in user testing. Most participants were between the ages of 18–24, which aligns with the primary target audience for the MyBudget app students and young adults beginning to manage personal finances. This demographic is ideal for testing usability and relevance, as they often seek simple, intuitive tools to build financial habits. The presence of a few older users also helps validate that the app's usability extends across a wider audience.

6.3.2 Occupation Distribution of Respondents

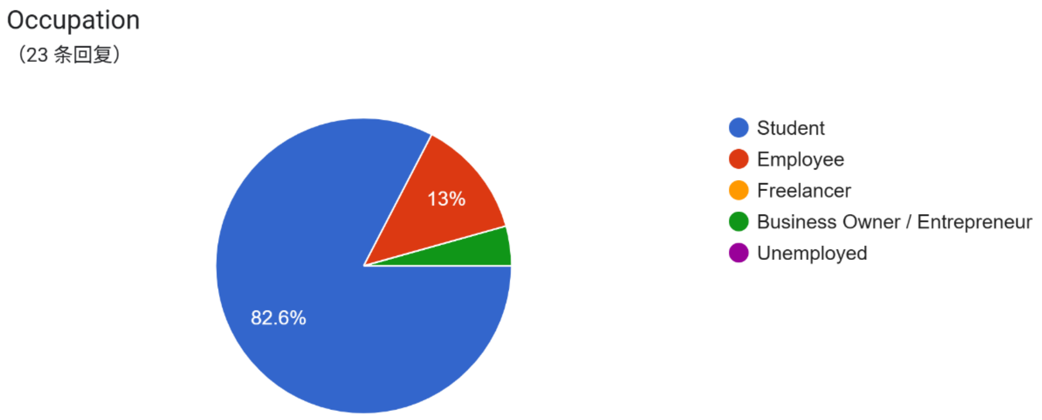


Figure 6. 2 Occupation Distribution of Respondents

In figure 6.2 has shown that most of the users involved in testing were students, followed by early-career professionals. This reflects the intended user base of the app and confirms that it resonates with individuals managing smaller, irregular incomes or learning to budget independently. The variety in occupation also helped assess whether the app is flexible and accessible to a wider audience, including freelancers or part-time workers.

6.3.3 Experience with Finance/Budgeting Apps

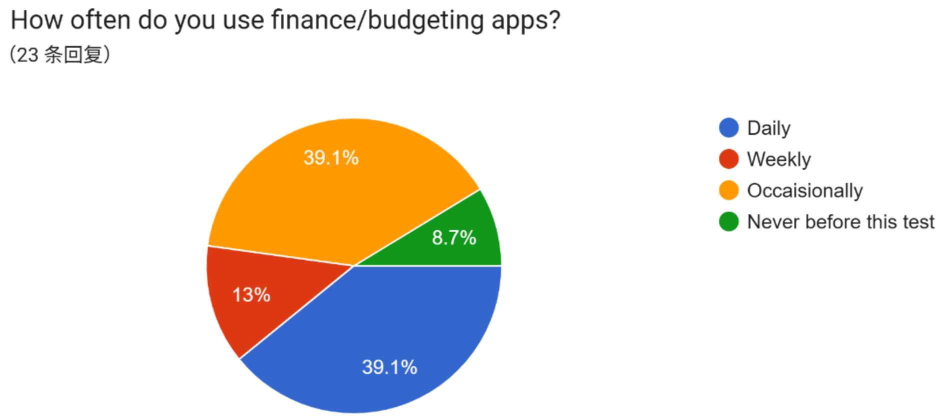


Figure 6. 3 Experience with Finance/Budgeting Apps

Figure 6.3 illustrates user responses to the question: *"How often do you use finance/budgeting apps?"* The data reveals that 39.1% of respondents use these applications daily, while an equal 39.1% use them occasionally. Additionally, 13% indicated weekly usage, and only 8.7% had never used a finance app before this test. This distribution indicates that most participants are already familiar with financial tools, which enhances the relevance and reliability of their feedback. The presence of frequent users implies that expectations for performance and usability were realistic and based on prior experience, making their evaluations valuable for assessing the effectiveness of the MyBudget application.

6.3.4 Navigation Usability Rating

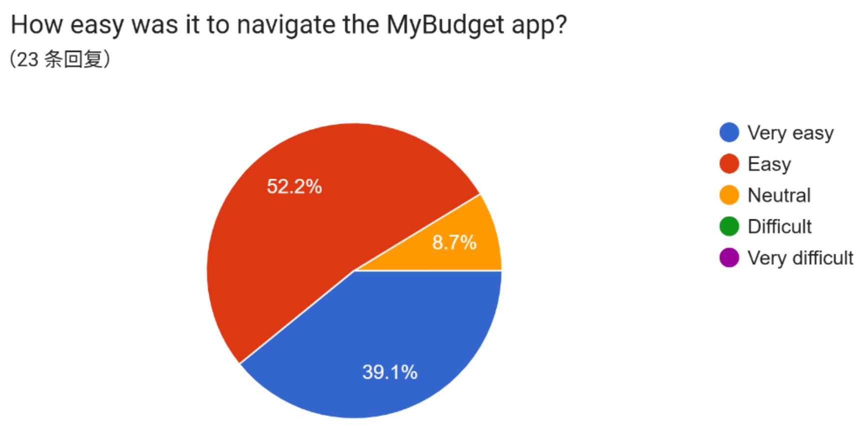


Figure 6. 4 Navigation Usability Rating

Figure 6.4 explores the ease of navigating the MyBudget application. The results show that 52.2% of users found the app easy to use, while another 39.1% considered it very easy. Only 8.7% selected neutral, and none of the respondents reported any difficulty navigating the app. This positive outcome highlights the effectiveness of the app's user interface (UI) and layout, indicating that users could intuitively access its features with minimal guidance. The lack of negative responses suggests that the design is well-aligned with user expectations and supports seamless interaction.

6.3.5 Understanding of App Features

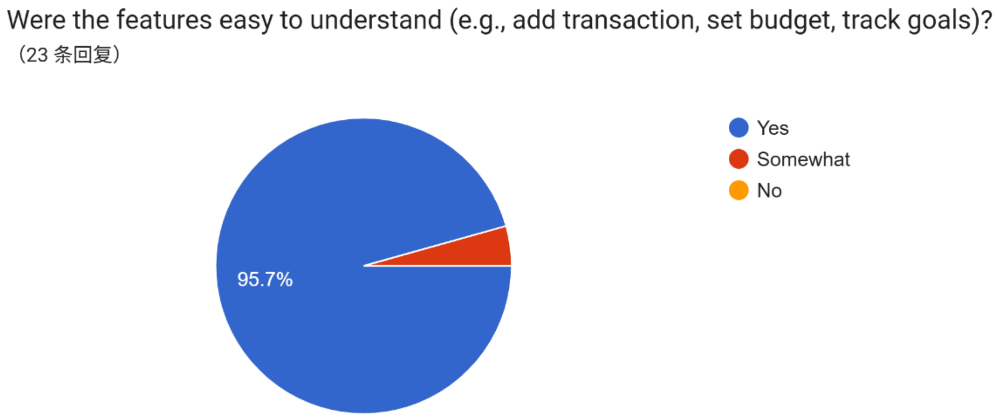


Figure 6. 5 Understanding of App Features

Figure 6.5 presents the clarity and understandability of the application's core features, such as adding transactions, setting budgets, and tracking financial goals. A remarkable 95.7% of users stated that the features were easy to understand, while 4.3% reported they were somewhat understandable. Importantly, no users found the features difficult to grasp. This indicates that the application successfully delivers clear and accessible user experience, especially for young users or those new to financial management. The design and language used within the app effectively guide users through its functionalities, minimizing the learning curve and maximizing usability.

6.3.6 User Interface (UI) Design Evaluation

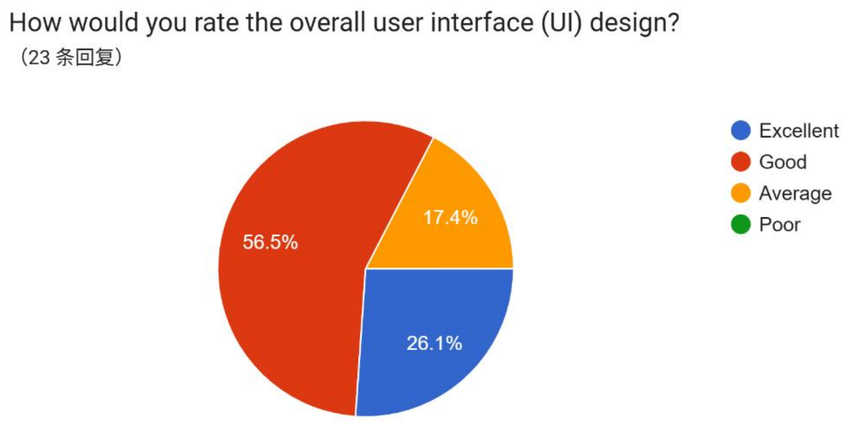


Figure 6. 6 User Interface (UI) Design Evaluation

Figure 6.6 assesses the users' perception of the overall user interface (UI) design of the MyBudget application. Many respondents (56.5%) rated the UI as Good, while 26.1% considered it Excellent. Another 17.4% found the design to be Average, and notably, none rated it as poor. This overall positive feedback indicates that the application's design effectively meets aesthetic and usability expectations. The favorable UI ratings reflect a layout that is visually clear and easy to navigate, contributing significantly to the user experience. However, the presence of “Average” ratings suggests there may still be opportunities for refinement, such as improving visual hierarchy or enhancing theme customization for more engaging interaction.

6.3.7 Usefulness of Data Visualization (Charts & Reports)

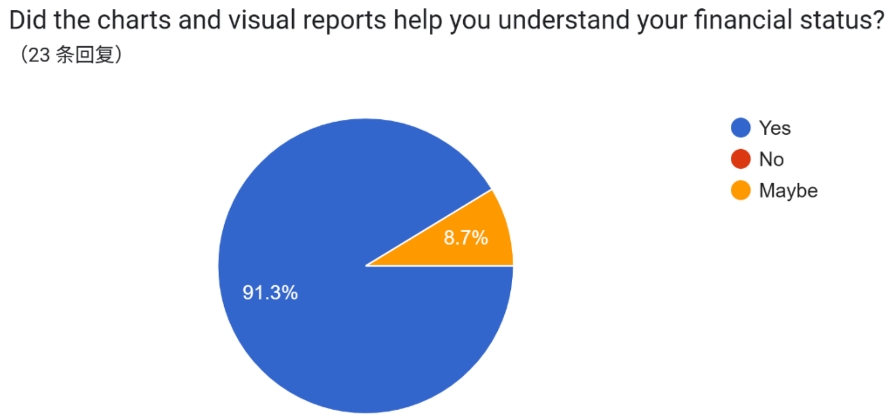


Figure 6. 7 Usefulness of Data Visualization (Charts & Reports)

Figure 6.7 evaluates the impact of charts and visual reports on helping users understand their financial status. An overwhelming 91.3% of respondents confirmed that the visual aids helped them better interpret their spending and savings patterns, while 8.7% chose “Maybe,” indicating partial effectiveness. No user selected “No,” which confirms that the app’s data visualization features such as bar graphs and pie charts are valuable in communicating financial information in a meaningful and user-friendly manner. This reinforces the importance of visual analytics in personal finance tools, particularly for younger users or those without advanced financial literacy.

6.3.8 Most Useful Features Identified by Users

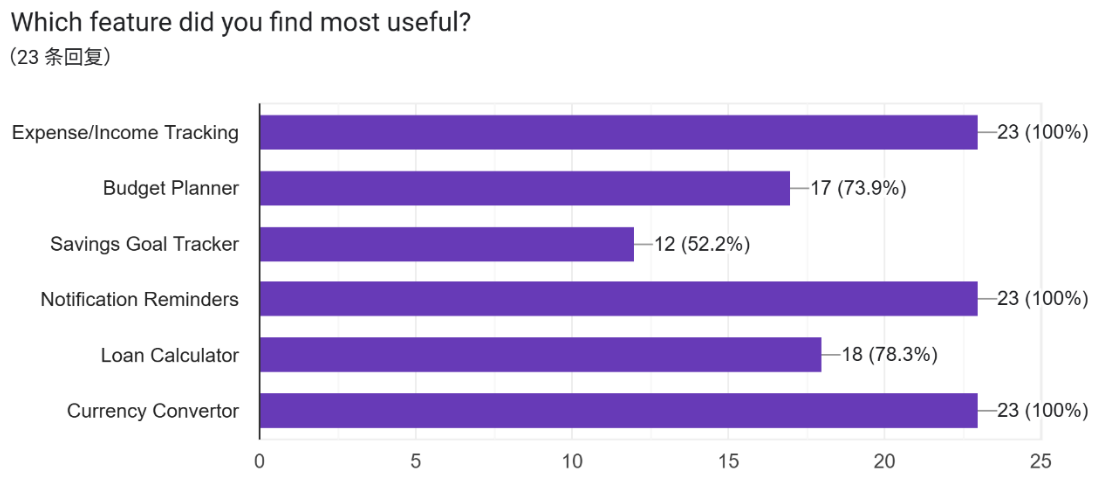


Figure 6. 8 Most Useful Features Identified by Users

Figure 6.8 illustrates the most useful features rated by the users. From the results, 100% of the users found Expense/Income Tracking, Notification Reminders, and Currency Converter to be extremely useful. The Loan Calculator was valued by 78.3%, 73.9% valued the Budget Planner, and 52.2% found the Savings Goal Tracker to be useful. These findings emphasize that users appreciate features that offer functional daily convenience and real-time monetary data, such as transaction tracking and currency conversion. Meanwhile, although relatively lower preference was accorded to the savings tracker, it remains an essential feature for a significant number of users who are keen on building long-term monetary discipline. This information gives explicit directions on what features to improve, and which might need more promotion or streamlining in the app interface.

6.3.9 Least Useful or Confusing Features

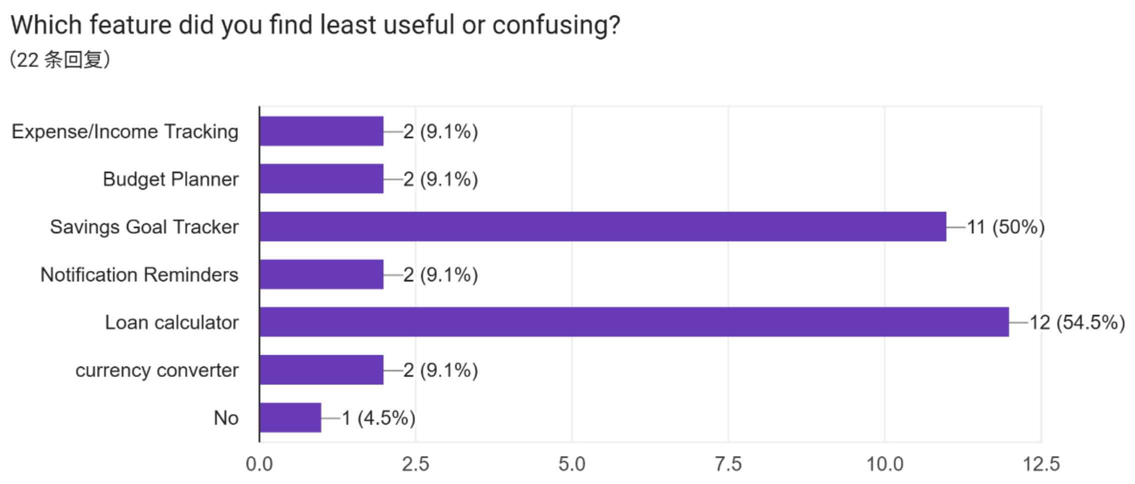


Figure 6. 9 Least Useful or Confusing Features

Figure 6.9 illustrates the most observed, most least useful or confusing features. Two features which were most often cited were the Loan Calculator (54.5%) and the Savings Goal Tracker (50%). Other features such as Expense/Income Tracking, Budget Planner, Notification Reminders, and Currency Converter each received 9.1% votes. Only 4.5% of those surveyed responded that they did not have any confusing features. This indicates that although central features such as tracking and reminders were largely well received, more advanced tools such as the loan calculator and savings tracker may need clearer instructions, better integration, or more UX-friendly design to better support user understanding and use.

6.3.10 Technical Issues Experienced (Bugs, Crashes)

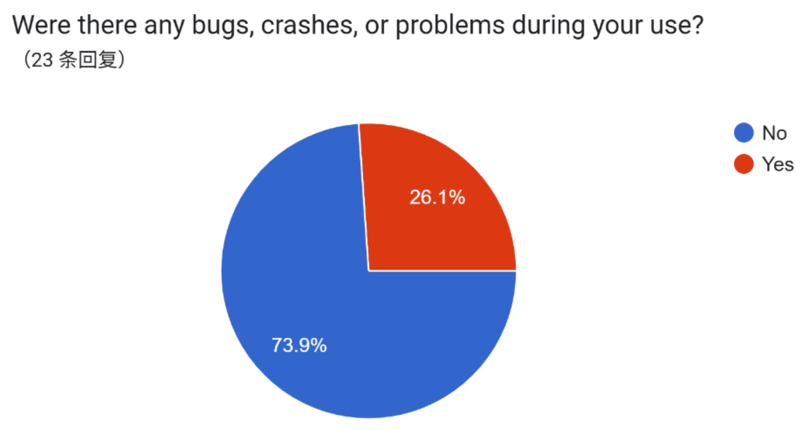


Figure 6. 10 Technical Issues Experienced (Bugs, Crashes)

Figure 6.10 relates to technical performance by inquiring if the users experienced any bugs, crashes, or problems when using the application. The majority (73.9%) had no technical issues, a highly encouraging indicator of system stability. 26.1% experienced some problems, reflecting some room for improvement in application strength and error control. Additional testing and refinement are recommended to reduce these occurrences and enhance overall application reliability.

6.3.11 Overall Satisfaction Rating

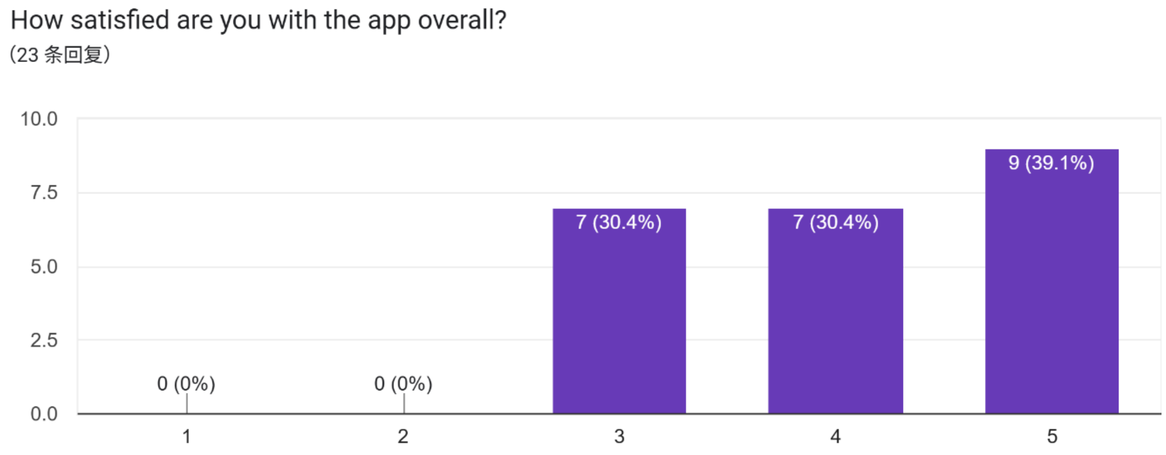


Figure 6. 11 Overall Satisfaction Rating

Figure 6.11 captures overall user satisfaction with the app. Most users gave high ratings, with 39.1% giving a 5, the highest rating, and 30.4% each giving 4 and 3. There were no ratings below 3. This high rating suggests a high level of user approval and acceptance. While there is still some space to move more users into the most satisfied category, the findings as they are present indicate that the app is widely popular and meets user expectations as far as usability and aesthetics go.

6.3.12 Willingness to Use the App Regularly

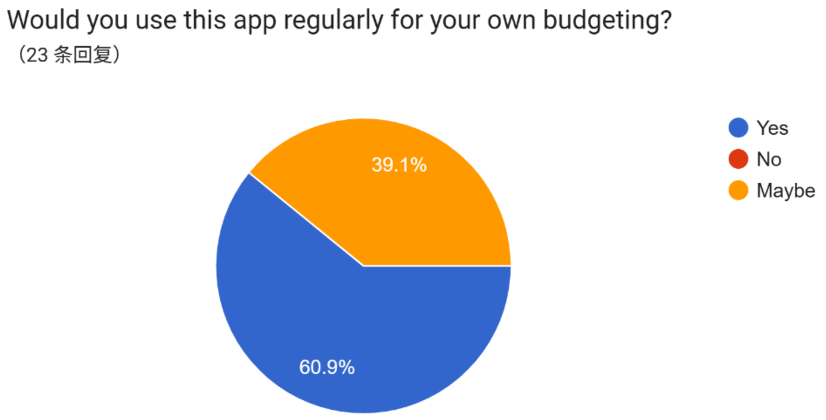


Figure 6. 12 Willingness to Use the App Regularly

Figure 6.12 questions whether the users would utilize the app on a daily basis for their personal budgeting. A decisive 60.9% answered "Yes," with intent for ongoing use. Still, 39.1% selected "Maybe," demonstrating conditional interest depending on additional updates or personal need. No one selected "No," which is a very positive sign for potential long-term usage. This also supports the fact that user feedback-driven features can only boost the levels of use and daily interaction.

6.4 Project Challenges

1. Gradle Build Failure

Because of File Permissions Among the most basic problems that were faced during the development stage was a Gradle building failure caused by restricted file access or a Java Development Kit (JDK) which wasn't properly set. The error would typically arise when Gradle fails to relocate or open certain cached transform files during a build operation execution.

Solution: The issue was resolved by reinstalling the correct version of the JDK to enable compatibility with the Gradle and Android Studio environment. A clean project Rebuild was also performed, and the IDE cache was cleared through the utilization of "Invalidate Caches / Restart" in Android Studio. These steps successfully corrected the build functionality and allowed development to continue without further Gradle-related disruptions.

2. Currency Converter API Integration Issue

During use of the currency converter feature, there was an issue while using a third-party API to receive real-time exchange rates. The app failed to receive the information or couldn't display it accurately and therefore ended up with inexact or missing conversion data.

Solution: Although the exact solution cannot be fully recalled, the issue was resolved after a series of debugging attempts were made, including checking API keys, endpoint URLs, and parsing JSON responses. After the testing and alteration of network request and response handling, the feature effectively fetched live currency exchange rates and showed the converted result as expected.

3. Set budget function failed to work as expected.

When implementing the feature of budget management, there existed an issue wherein the system did not save and display the values of month-wise and category-wise budgets correctly.

At times, the application used to show empty data, or the entered budget values were not reflected on the screen. It was misleading and prevented proper budget tracking while testing.

Solution: The issue was resolved by debugging the Firestore data structure and ensuring that the data was being stored under the correct user and month. The structure was also changed to correctly refresh and display the current values from Firestore whenever a budget was added or updated. Checks were also included to prevent empty or duplicate entries. These changes ensured that budget data was being saved properly and displayed in real-time.

4. Monthly Reminders and Notification Permissions Across Android Versions

During implementation of the reminder feature, the application had issues scheduling notifications and handling permission prompts correctly across various Android versions. Some devices failed to display notifications or required additional runtime permission handling, especially on newer Android systems where notification policies are stricter.

Solution: This issue was resolved through extensive research and testing on different devices and OS versions. The app was updated to check and request runtime notification permissions where required, particularly for Android 13 and above. Additionally, improvements were made in how reminders were scheduled using the appropriate APIs for alarm and notification handling. These changes ensured that monthly bill reminders worked consistently and reliably across a range of Android devices.

5. Fragment Crash Due to IllegalStateException (HomeFragment Switching Issue)

A critical issue occurred when users quickly navigated between pages in the app, particularly while Firestore was still loading data. This triggered an IllegalStateException due to the app attempting to update a UI fragment that had already been detached from the activity.

Solution: The crash was resolved by implementing lifecycle safety checks. Specifically, isAdded() was used to ensure the fragment was still attached, and getActivity() != null confirmed the activity was available before attempting any UI updates to prevent UI updates when the screen was no longer active.

6.5 Objectives Evaluation

1. To develop a comprehensive financial tracking system

Objective:

This project aims to create a financial tracking system that helps users manage their expenses and income effectively. The system will include features like transaction logs, categorized expense tracking, and visual summaries through graphs and charts. By providing clear insights into spending habits, it will enable users to make better financial decisions and achieve their goals.

Achievement/Evaluation:

This objective was successfully achieved. The MyBudget application enables users to log both income and expenses, assign transactions to user-defined categories, and view summaries of their financial activity through interactive charts. The system supports visual representation of monthly spending and earnings, helping users identify patterns and areas to optimize. Real-time synchronization through Firebase ensures seamless tracking across devices, while offline functionality enhances reliability. These features collectively support better financial awareness and informed decision-making, fulfilling the intent of this objective.

2. To implement planning and projection tools

Objective:

This objective focuses on creating tools that assist users in planning and forecasting their financial activities. Features such as budget planning, goal tracking, and financial projections will empower users to set realistic targets and monitor their progress over time. These tools will provide data-driven insights to help users make informed decisions and stay on track toward achieving their financial objectives.

Achievement/Evaluation:

The application meets this objective by incorporating a robust budget management system that allows users to define monthly spending limits overall and by category. A dedicated savings goal module lets users set long-term objectives and monitor their progress using percentage-based visual indicators. Users also receive timely reminders for budgeting and bill payments, which helps reinforce positive financial habits. These tools work together to provide users with planning capabilities and meaningful financial projections, directly supporting users' ability to manage and achieve financial goals.

3. To develop advanced financial calculation features (Loan calculator and real-time currency converter)

Objective:

The third objective aims to integrate essential calculation tools like a loan calculator and a real-time currency converter into the financial management system. The loan calculator helps users estimate repayment schedules by calculating monthly payments and total loan amounts based on factors like interest rates and repayment terms, aiding informed financial decisions. The real-time currency converter, using APIs for up-to-date exchange rates, provides accurate currency conversions, ideal for users involved in international transactions or travel. These tools enhance user experience by offering practical, efficient solutions for managing finances, aligning with the project's goal of empowering users with advanced financial tools.

Achievement/Evaluation:

Both the loan calculator and the real-time currency converter were successfully implemented. The loan calculator provides users with accurate EMI and repayment breakdowns based on inputs like interest rate and loan term, assisting in financial planning and borrowing decisions. The currency converter uses live exchange rate APIs to deliver precise conversions, useful for users managing international transactions or travel expenses. These tools enhance the app’s practicality and align closely with the project’s goal of offering advanced financial support features.

6.6 Concluding Remark

In short, the system test and evaluation step of the MyBudget mobile app verified that the project objectives were successfully achieved, and the application runs smoothly under functional as well as non-functional situations. The testing process, which was carried out through the Black Box Testing technique, verified that each feature like recording transaction, managing budget, savings goal setting, currency exchange, loan calculation, and notifications acts as expected under regular and edge-case conditions.

The functional testing confirmed the correctness and stability of core modules, while the non-functional testing confirmed system performance, responsiveness, notification correctness, and offline capability. Real-time data synchronization via Firebase Firestore, effective user authentication, and adequate error handling contributed to overall application stability and usability. User testing feedback indicated good user experience, with features being easy to use and navigate.

Difficulties faced while in the implementation process, such as API integration issues, Gradle build failures, and Firebase data structure errors, were overcome through research, trial-and-error, and config refining. Such impediments not only made the app more technologically resilient but also increased the developer's problem-solving capability and learning capabilities.

In general, the MyBudget application has been thoroughly tested and proven to be stable, functional, and user-friendly financial management software. The system is deployable and can be further enhanced with additional features such as machine learning-based financial advice, auto-bank linking, and sophisticated data analytics in future development phases.

CHAPTER 7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

In conclusion, the development of the MyBudget mobile application has successfully filled its position as a general-purpose personal financial management system. The project aimed to help its users track their earnings and expenses, keep track of their monthly budgets, set savings goals, send bill reminders, and make intelligent financial decisions. all through a convenient and intuitive mobile interface. Over the course of this project, a number of modules were designed, built, and tested to ensure the system runs with optimum efficiency under real-world conditions.

The app includes principal financial features like transaction categories, customizable pie charts, reminders both daily and monthly, a housing loan calculator, and an external API-based real-time currency converter. Firebase was used effectively to manage authentication, data storage, and synchronization, resulting in a secure and scalable backend. Black Box Testing has been performed on all basic modules, such that the system will run correctly given valid as well as invalid inputs.

Despite several problems faced during the development stage, such as Gradle build failure, API integration failure, and data structure complexity, these were successfully addressed with proper debugging and continuous enhancements. As a result, the MyBudget app is executed successfully, meets all initial objectives, and provides trouble-free experience for the end-users.

Finally, the project is a successful creation of a mobile financial application that is effective, functional, and ready for real deployment. Not only does it accomplish the set objectives but also provides a platform for future growth and innovation in personal finance technology.

7.2 Recommendation

Although the MyBudget mobile application has accomplished its core objectives and runs smoothly, there are several areas where the system can be refined and improved upon in subsequent releases. The largest recommendation would be adding secure bank account linking to allow users to automatically transfer transactions and reduce manual entry. This would improve usability and accuracy of data. Also, including machine learning algorithms to provide the user with individualized financial data and suggestions can empower the user to make improved budgeting choices based on their unique expenditures.

Another potential enhancement is to include data export features by which users can generate PDF or Excel-based reports for offline use or personal record-keeping. Furthermore, modifications to the user interface, such as theme support, dark mode, and improved accessibility features, could enhance user satisfaction and increase the attractiveness of the app.

Lastly, incorporating multilingual functionality would increase the app's user base, particularly in regions with multiple commonly spoken languages. With these upgrades, the MyBudget application can become a more powerful, intelligent, and user-centered personal finance application.

Recommendations:

- Implement bank account integration
- Integrate machine learning for personalized budgeting insights
- Enable data export (PDF, Excel) for offline access and reporting
- Enhance UI/UX with dark mode, themes, and accessibility
- Provide multilingual support to reach a wider user base

REFERENCES

[1] Experian, "You Need a Budget app review," ***Experian.*** [Online]. Available: https://www.experian.com/blogs/ask-experian/you-need-a-budget-app-review/. [Accessed: Nov. 5, 2024].

[2] Savology, "YNAB vs. Mint: Which budgeting app is better?," ***Savology.*** [Online]. Available: https://savology.com/ynab-vs-mint. [Accessed: Nov. 5, 2024].

[3] Technology Evaluation Centers, "Mint," ***Technology Evaluation Centers.*** [Online]. Available: https://www3.technologyevaluation.com/solutions/53449/mint?srsltid=AfmBOookrRKBZMyjoqS1Xxv9-kobrzi7uvJmOren_ZWgQ604bo0hYhtS. [Accessed: Nov. 6, 2024].

[4] B. Rubenking, "PocketGuard Review," ***PCMag,*** Sep. 28, 2023. [Online]. Available: https://www.pcmag.com/reviews/pocketguard. [Accessed: Nov. 6, 2024].

[5] Incubating Wallet, "The Checkbook App: Monefy Product Review," ***Incubating Wallet.*** [Online]. Available: https://incubatingwallet.com/the-checkbook-app-monefy-product-review/. [Accessed: Nov. 6, 2024].

[6] The Motley Fool, "Goodbudget Review: Pros, Cons, and Who Should Set Up a Budget With This App," ***The Motley Fool***. [Online]. Available: https://www.fool.com/money/personal-finance/good-budget-review/. [Accessed: Nov. 6, 2024].

[7] Experian, "Goodbudget Budgeting App Review," ***Experian***. [Online]. Available: https://www.experian.com/blogs/ask-experian/goodbudget-budgeting-app-review/. [Accessed: Nov. 6, 2024].

[8] Mint, "Mint App Review: The Pros and Cons of Mint's Budgeting Tool," Mint.com. [Online]. Available: https://mint.intuit.com/blog/what-is-mint/. [Accessed: Nov. 6, 2024].

[9] NerdWallet, "The 9 Best Budget Apps for 2024," NerdWallet. [Online]. Available: https://www.nerdwallet.com/best-budget-apps. [Accessed: Nov. 6, 2024].

[10] MoneyUnder30, "YNAB (You Need a Budget) Review," MoneyUnder30. [Online]. Available: https://www.moneyunder30.com/ynab-review. [Accessed: Nov. 6, 2024].

[11] PocketGuard, "Manage Your Finances with PocketGuard," PocketGuard.com. [Online]. Available: https://pocketguard.com. [Accessed: Nov. 6, 2024].

[12] Forbes, "The Best Financial Apps to Manage Your Money," Forbes. [Online]. Available: https://www.forbes.com/best-financial-apps. [Accessed: Nov. 6, 2024].

[13] Bankrate, "Top Budgeting Apps to Help Manage Your Finances," Bankrate.com. [Online]. Available: https://www.bankrate.com/budgeting-apps. [Accessed: Nov. 6, 2024].

[14] The Balance, "Best Personal Finance Software for 2024," TheBalance.com. [Online]. Available: https://www.thebalance.com/best-personal-finance-software. [Accessed: Nov. 6, 2024].

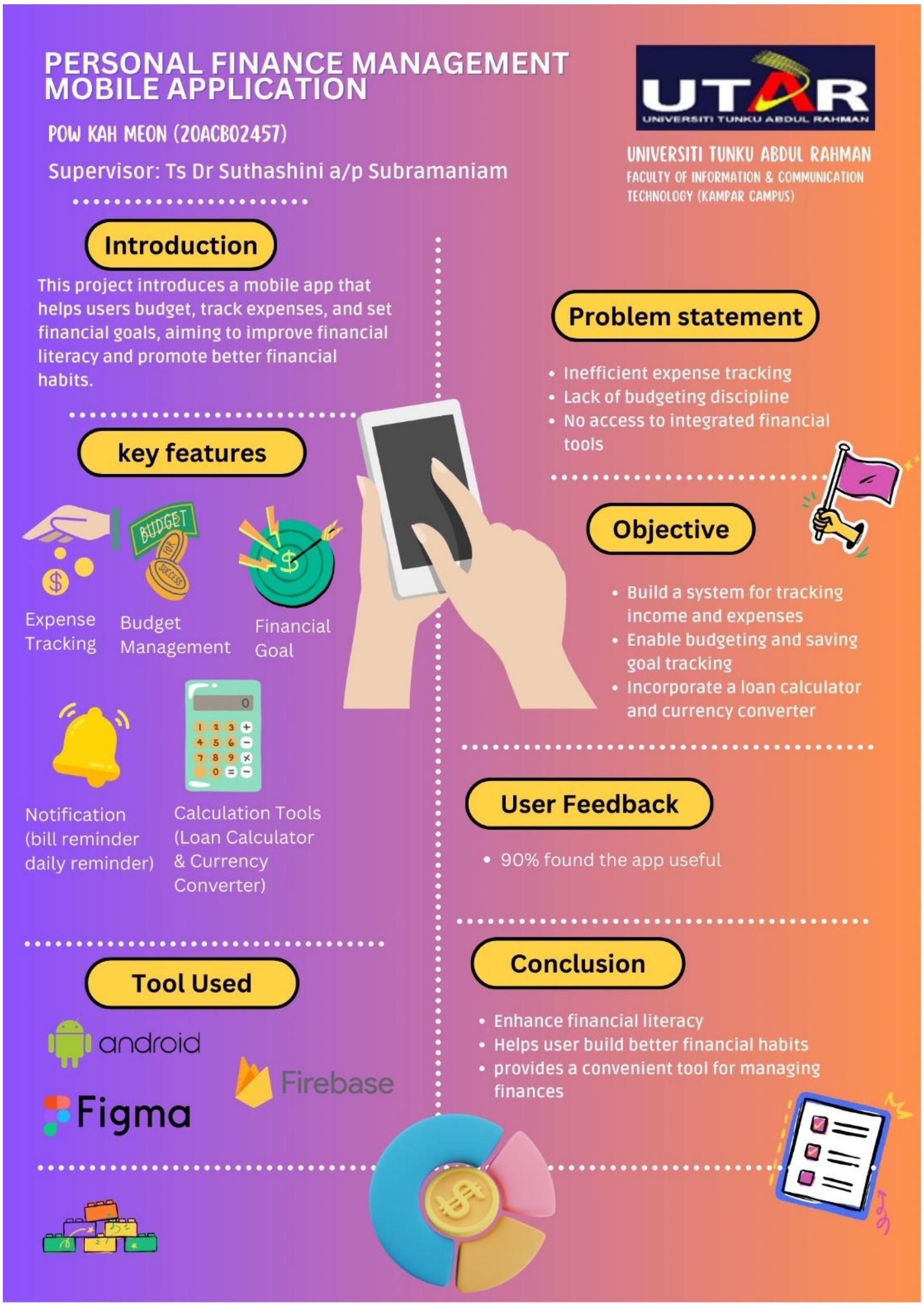
[15] Investopedia, "Top 5 Budgeting Tools to Simplify Your Financial Life," Investopedia. [Online]. Available: https://www.investopedia.com/best-budgeting-tools. [Accessed: Nov. 6, 2024].

[16] TechCrunch, "How FinTech Apps Are Revolutionizing Budgeting," TechCrunch.com. [Online]. Available: https://techcrunch.com/fintech-budgeting-apps. [Accessed: Nov. 6, 2024].

[17] AppInstitute, "Designing Mobile Apps for Budgeting and Personal Finance," AppInstitute.com. [Online]. Available: https://www.appinstitute.com/design-budget-apps. [Accessed: Nov. 6, 2024].

APPENDIX

POSTER



PERSONAL FINANCE MANAGEMENT MOBILE APPLICATION

POW KAH MEON (20ACB02457)

Supervisor: Ts Dr Suthashini a/p Subramaniam

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

Introduction

This project introduces a mobile app that helps users budget, track expenses, and set financial goals, aiming to improve financial literacy and promote better financial habits.

key features

Expense Tracking

Budget Management

Financial Goal

Notification (bill reminder daily reminder)

Calculation Tools (Loan Calculator & Currency Converter)

Tool Used

android

Figma

Firebase

Problem statement

- Inefficient expense tracking
- Lack of budgeting discipline
- No access to integrated financial tools

Objective

- Build a system for tracking income and expenses
- Enable budgeting and saving goal tracking
- Incorporate a loan calculator and currency converter

User Feedback

- 90% found the app useful

Conclusion

- Enhance financial literacy
- Helps user build better financial habits
- provides a convenient tool for managing finances