

Joined Account Finance Management System

BY

Wan Xin

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)

JUNE 2024

REPORT STATUS DECLARATION FORM

Title: Joined Account Finance Management System

Academic Session: Y4S1

I WAN XIN
(CAPITAL LETTER)

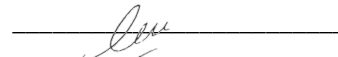
declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)



(Supervisor's signature)

Address:

15, Lorong 34,
Taman Petani Jaya, 08000
Sungai Petani, Kedah

Ku Chin Soon
Supervisor's name

Date: 12 September 2024

Date: 13 September 2024

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 12 September 2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Wan Xin (ID No: 20ACB02288) has completed this final year project/ dissertation/ thesis* entitled “Joined Account Finance Management System” under the supervision of Mr. Ku Chin Soon (Supervisor) from the Department of ACA , Faculty of Information and Communication Technology.

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

Yours truly,

Wan Xin
(Student Name)

DECLARATION OF ORIGINALITY

I declare that this report entitled “**Joined Account Finance Management System**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :  _____

Name : Wan Xin

Date : 12 September 2024

ACKNOWLEDGEMENTS

I would like to express thanks and appreciation to my supervisor, Dr Ku Chin Soon and my moderator, Dr Chan Lee Kwun who has given me this bright opportunity to engage in this final year project. He has contributed his time in giving me precious suggestions. A million thanks to you.

To a very special person in my life, Zhi Jing, for his patience, unconditional support, and love, and for standing by my side during hard times. Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This project is a development-based project for academic purpose. It will provide users with the methodology, concept, and design of a joined account financial management system.

The main purpose of the personal finance management system is to generate a suitable financial budget for users that are starting budgeting plans as a family.

The proposed project for this proposal will be a Joined Finance Management mobile application which is Android-based. Android based application is chosen but not web-based is that smartphones are portable, so it is more suitable for the project, unlike laptops and desktop which is hard to bring by one's side.

Nowadays, smartphones are a necessity for the majority of the people in the globe, and the numbers are still going up. Thanks to technology, people can monitor and control their spending more easily. Finance Technology, also known as FinTech, is dominating users at a rapid pace. By providing financial analysis and planning goals, these applications can simplify people's lives with just a few clicks. By carrying all these actions on user's smartphones, it will be more effective and let users have the motivation to continue recording their daily expenses.

Emphasis is also made on the weight distribution of the budget for users with different salary range. The system will auto generate a starting budget for new users based on the application database, and users can make arrangements to their monthly budget and the system will auto modify other budget categories. This application can also let 2 users merge their budget and expenses, to make a family budget, and let both users know another user spending pattern better.

Lastly, Firebase Cloud Firestore and Android Studio will be used as the development tool and Java will be the main language of the project.

TABLE OF CONTENTS

TITLE PAGE	i
REPORT STATUS DECLARATION FORM	ii
FYP THESIS SUBMISSION FORM	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xi
LIST OF SYMBOLS	xii
LIST OF ABBREVIATIONS	xiii
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement and Motivation	1
1.2 Objectives	3
1.3 Project Scope and Direction	5
1.4 Contributions	6
1.5 Report Organization	7

CHAPTER 2 LITERATURE REVIEW	8
2.1 Review of Technologies	8
2.1.1 Finance Management System	8
2.1.2 Hardware Platform	8
2.1.3 Operating System	8
2.1.4 Database	8
2.1.5 Programming Language	8
2.2 Evaluation of Existing Systems	10
2.2.1 Goodbudget: Budget & Finance	10
2.2.2 Bluecoins Finance: Budget	12
2.2.3 Money Manager & Expenses	14
2.2.4 Belanjawanku	16
2.2.5 Money Manager: Budget & Expense	18
2.3 Comparison Table	20
CHAPTER 3 SYSTEM MODEL	21
3.1 Methodology	21
3.2 System Design Equation	22
3.3 Use Case Diagram and Description	23
3.3.1 Use Case Diagram	23
3.4 Flow Chart	28
CHAPTER 4 SYSTEM DESIGN	38
4.1 System Block Diagram	38
4.2 System Flow Description	39
4.3 Hardware and Software Requirements	42
4.3.1 Software	42
4.3.2 Hardware	42
4.4 System Components Specifications	43
4.5 System Components Interaction Operations	44
4.6 Timeline	46

CHAPTER 5 System Implementation	47
5.1 Hardware Setup	47
5.2 Software Setup	48
5.3 Setting and Configuration	49
5.4 System Operation (with Screenshot)	50
5.4.1 Application	50
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION	64
6.1 System Evaluation Survey Results	64
6.1 System Testing and Performance Metrics	67
6.4 Objectives Evaluation	71
6.5 Concluding Remark	75
CHAPTER 7 CONCLUSION AND RECOMMENDATION	76
7.1 Conclusion	77
7.2 Recommendation	77
REFERENCES	78
APPENDIX	A-1
WEEKLY LOG	A-3
POSTER	A-10
PLAGIARISM CHECK RESULT	A-11
FYP2 CHECKLIST	A-14

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	FMS mobile application global usage in 2021	
Figure 2.1.1	Wallet: Budget Planner Tracker	
Figure 2.1.2	Bluecoins Finance: Budget	
Figure 2.1.3	Money Manager & Expenses	
Figure 2.1.4	Belanjawanku	
Figure 2.1.5	Money Manager: Budget & Expense	
Figure 3.1.1	Agile Methodology Software Development Life Cycle	
Figure 3.2.1	Use Case Diagram	
Figure 3.4.1	Home Page Flow Chart	
Figure 3.4.2	Joining Account Flow Chart	
Figure 3.4.3	Auto Generated Budget Flow Chart	
Figure 3.4.4	Add Spending Records Page Flowchart	
Figure 3.4.5	Profile Activity Flow Chart	
Figure 3.4.6	Show Records History Flow Chart	
Figure 3.4.7	Show Graph Flow Chart	
Figure 3.4.8	Profile Join Account Flow Chart	
Figure 3.4.9	Accept Join Request Flow Chart	
Figure 3.4.10	Join Flow Chart	
Figure 4.1.1	System Block Diagram	
Figure 4.4.1	System Components Specification Diagram	
Figure 4.6.1	Gantt Chart	
Figure 5.4.1	Splash Screen	
Figure 5.4.2	Login and Sign Up Page	
Figure 5.4.3	Join Account Module	
Figure 5.4.4	Register personal details	
Figure 5.4.5	Select Budget Type	
Figure 5.4.6	Show Autogenerated Budget	
Figure 5.4.7	Edit Budget	
Figure 5.4.8	Home Page	

Figure 5.4.9	Add Records
Figure 5.4.10	User Spending Records History
Figure 5.4.11	Profile Activity
Figure 5.4.12	JoinRequestActivity
Figure 5.4.13	Graph Activity
Figure 6.1.1	Evaluation Form Participants
Figure 6.1.2	Evaluation Form Results Part 1
Figure 6.1.3	Evaluation Form Results Part 2
Figure 6.3.1	Budget Plan Table in Cloud Firestore
Figure 6.3.2	Summary of Budget on Report Guide
Figure 6.3.3	Autogenerated budget example for different income
Figure 6.3.4	Income Range Table in Cloud Firestore
Figure 6.3.5	Spending Records for both users

LIST OF TABLES

Table Number	Title	Page
Table 2.2	Comparison Table of Review Systems	
Table 5.1.1	Specifications of desktop	
Table 5.1.2	Specifications of mobile phone	

LIST OF ABBREVIATIONS

<i>iOS</i>	iPhone operating system
<i>JFMS</i>	Joined Financial Management System
<i>FMS</i>	Financial Management System
<i>UI</i>	User Interface

Chapter 1 Introduction

1.1 Problem Statement and Motivation

1) Current financial budget apps in Malaysia only provide generic budget recommendations.

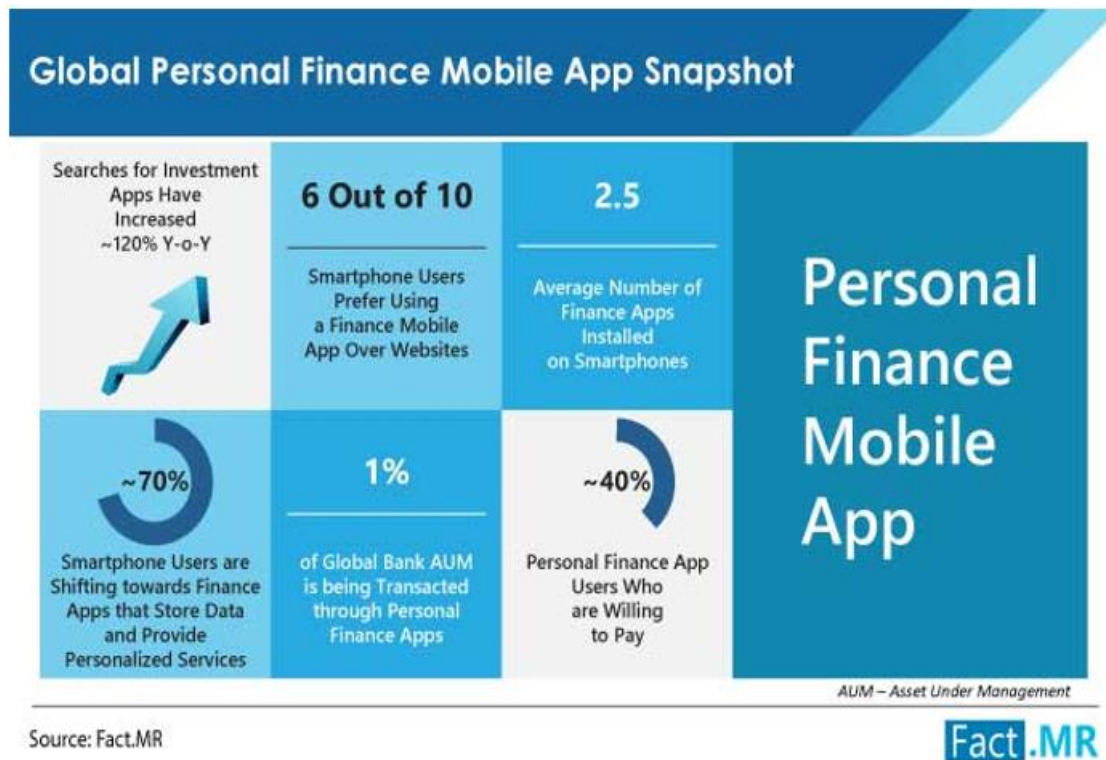


Figure 1.1 FMS mobile application global usage in 2021[1]

According to [1](see Figure 1.1), it shows that 70% of the smartphone users worldwide are using at least 1 FMS. Although the market of FMS is steadily growing, there are still many people who failed at financial planning. This is because without a suitable and personalized budget plan, people can easily overspend. In 2021, Bank Negara Malaysia reports that 47% of Malaysian youth have significant high credit card debt[2]. This is not a healthy phenomenon and should be improvised. Therefore, the purpose of developing this FMS is to help people to generate a suitable budget according to their income tier, to help with their spending and plan their budget well. But normal FMS applications on the market mostly did not have the function to generate a budget for their users, not to mention a personalized budget. This only will let users to know they had overspent or not but does not lead them to the right way to improve their spending. Falling at the first hurdle sets the tone for the entire race. Therefore, some users still failed in financial planning even after using other FMS applications.

2) Lack of budget recommendations of different salary range.

There may be some applications on the market who provide a spending budget, but it gives an one-size-fits-all approach to financial planning, disregarding the diverse income levels. For instance, a budget recommendation designed for someone with a high income may be unattainable and unrealistic for someone earning a lower salary. Conversely, a budget set for a lower income bracket might not adequately accommodate the financial goals and responsibilities of someone with a higher income. This lack of tailored budget recommendations undermines the fundamental purpose of budgeting apps. A financial budget that is not suitable for users will only lead them into failure in financial planning. Thus, there is a need for budget recommendation based on weight distribution in the project to assist users in financial planning.

3) Current financial budget apps do not provide multiple user environment.

Some users in the same family, like married couples both use FMS application on the market, but they end up having inaccuracy in their data as there may be duplicates of data like house rent, water and electric bills, and many more. The inability of 2 users to merge their accounts also makes checking bills difficult for both users. As an example, the husband has to physically access his wife's smartphone in order to check her expenses. This problem could be resolved if users have the capability to create a shared account. It will be a huge advantage if users can have a shared budget and plan financially together.

The aim of the project is to develop a FMS application that is different than the ones which are already available on the market. The project wishes to produce a JFMS application with personalized budgets and shared accounts to help users such as couples and youths to plan financially and reduce the percentage of people overspending in Malaysia.

1.2 Research Objectives

The aim of this thesis is to develop a backend system that is capable of providing a suitable budget recommendation based on user status and integrate shared financial arrangements within budgeting apps.

1)To propose a system that can provide budget recommendations based on location, dependents, and marital status.

The development of the program focus on generating a suitable financial budget by involving their status in the budget. By leveraging these factors, users will receive budget recommendations that is logic and more user friendly. Whether an individual resides in which state, has dependents to support, or is married or single, the system will get all these data and find a most budget that match user's lifestyle. This level of customization ensures that users can effectively manage their finances in a way that suits their specific needs and goals.

2)To integrate salary-specific budgeting recommendations to the diverse income levels and financial needs of users in Malaysia.

Moreover, the project also aims to integrate salary-specific budgeting recommendations to the diverse income levels and financial needs of users in Malaysia. By deploying the salary-specific budgeting recommendations into the system, the app will address the diverse income levels and financial needs of users. Whether the user earns a high income or is living on a modest salary, the system will provide a budget guide that promotes responsible spending, savings, and financial stability. By catering to the specific income levels in Malaysia, the system will empower users to make informed financial decisions that support their long-term financial well-being.

3)To implement collaborative financial planning and management for households or shared financial arrangements within budgeting apps.

Besides that, collaborative financial planning and management is also the essential component of the proposed system. The developed system aims to improve household financial arrangements within budgeting apps. The system will provide features that enable collaboration between couples. Users will have the ability to share financial budget, and track progress together, thus fostering transparency and communication in

CHAPTER 1

financial matters. This collaborative approach ensures that all parties involved have a clear understanding of their financial situation and work together towards a common financial objective, promoting financial success.

1.3 Project Scope and Direction

Several scopes will be set for this project to address the problems identified in the problem statement. The proposed system will act as a JFMS to help users conquer the problems they face in financial planning.

The point to consider in our project scope is that the proposed system is an android mobile application. The proposed system will only focus on android OS development due to time constraints. Moreover, it is known that Android Studio can only develop native Android applications. Although there are many more IDE and language like Flutter that can be used to develop iOS compatible applications, but Android Studio has a rich library that is much more useful for this project.

The next item on our project scope is that the proposed system will only focus on 2 people joined accounts. The target audience for the application is couples in Malaysia. This application target couples because they need a collaborative financial environment at most. Due to the time constraints of this project, it is difficult to develop an application that have multiple users joined account.

Lastly, our project scope is to develop a budget that suits the user status, and will update the budget based on user data when the application have sufficient amount of users. The budget recommendation will be more and more suitable for the users as the recommendation is based on the average user data in the app.

1.4 Contributions

Our project contributes to users by helping them to acquire success in planning financially. The project can enhance overall user financial planning as users have a more suitable financial budget. Users can improve their spending habits and control their expenses better. This could decrease the rate of people that are having debts in Malaysia. User can save time on researching how to create a financial budget from scratch, as the app does the work for them.

JFMS also make financial management in family more efficient and transparent. Every expense can be recorded accurately therefore there will be no miscommunication. All members can have a clear understanding on the expenses, thus any financial decisions could be make to achieve their financial goals.

1.5 Report Organization

The details of this proposal are further elaborated in the following chapters. In Chapter 2, multiple FMS system on the market has been reviewed. It contains the literature review and critical analysis of the review applications. The system requirements, system design, and system architecture diagram are written in Chapter 3. Additionally, Chapter 4 describes the setting up of the project and visualization of the preliminary work of the project. Furthermore, Chapter 5 concludes the project.

Chapter 2 Literature Review

2.1 Review of Technologies

2.1.1 Finance Management System

A finance management system is a software or application designed to help individuals to manage their financial activities. It typically can let users track expenses and generating budgets.

2.1.2 Hardware Platform

A mobile application is a software application designed to run on mobile devices such as smartphones and tablets. Mobile apps can provide a wide range of functionalities and services, often tailored to the specific needs of mobile users.

2.1.3 Operating System

An operating system (OS) in the context of Android is the software that manages the hardware and software resources of Android devices, such as smartphones and tablets. It provides the underlying platform that allows applications to run and interact with the device's hardware. Android is an open-source operating system based on the Linux kernel, designed primarily for mobile devices. It provides a framework for managing hardware resources, running applications, and providing user interface elements. The Android operating system provides the foundational software that manages hardware resources, supports application execution, and delivers a user interface for mobile devices. Its open-source nature, along with its rich set of features and APIs, makes it a versatile platform for both end-users and developers.

2.1.4 Database

A database is a systematic and organized set of data stored electronically. It can be accessed, managed, and updated using database management systems (DBMS). The primary purpose of a database is to efficiently store and retrieve data, support complex queries, and maintain data integrity. Databases are organized into tables, which consist of rows and columns. Each table represents a specific type of data (e.g., customer information, transactions).

2.1.5 Programming Language

A programming language is a formal set of instructions and rules used to communicate with a computer and create software applications. It provides the syntax and semantics

Bachelor of Information Systems (Honours) Information Systems Engineering
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 2

for writing code that the computer can execute to perform specific tasks, process data, or solve problems. The programming language used in this project is Java, and object oriented language.

2.2 Evaluation of Existing Systems

2.2.1 Goodbudget: Budget & Finance

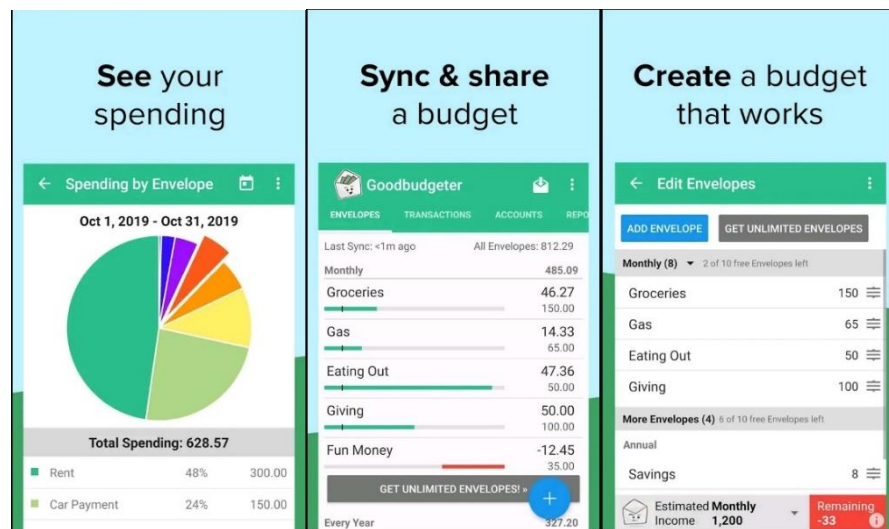


Figure 2.1.1 Wallet: Budget Planner Tracker [3]

Goodbudget is a FMS application that is design to let user manage their income. The main point of Goodbudget is its envelope function. It works like an old-fashioned envelope, where you put money in it and spend it. It acts like a budget for a category, and you can no longer spend after you have reached the limit.

With this function, users can see their quota clearly and try their best not to overspend. They will be more conscious of their spending. Additionally, Goodbudget also can let users sync and share their budget on different devices. This increases the portability of the app. Moreover, user can also set different envelops for every month or add additional envelopes when they have extra income. The flexibility of the envelopes makes the app suitable for users that have no constant income every month.

Lastly, Goodbudget can generate multiple graphs and charts to let users monitor and compare their income and spending. The clear visualization let user know that where their money is spent, and which category can be improved to save more money.

Strength

The clear visuals of their money flow can encourage users to be more mindful in their financial decisions. The envelop functions also discouraged overspending as users will see the numbers on the envelop become negative. The easy concept make it easier for aged users that are not familiar with smartphones to start with.

CHAPTER 2

Goodbudget can let users to sync and share budgets across different devices, making it easier for users to access their data. The portability enables users to stay connected to their budgets, maximizing their financial management capabilities. The analytic data also can make users focus on cutting down unnecessary spendings.

Weakness

The first limitations of Goodbudget is that the envelope function is limited. There are only a few envelopes for the free version, and users will need to pay to use unlimited envelopes. Next, Goodbudget does not have either import or export functions. The summary and transactions could not be created as a pdf file. This makes transfer of files hard as not everyone uses the same FPMS application. Moreover, the weakness of Goodbudget is that it's UI is a bit messy. The text is packed too closed, making it a little bit harder for user to see the data. Goodbudget also does not provide user the option to setup a password to block unauthorized access. Another limitation of Goodbudget is that the amount of graph generated is lesser than other FMS applications. There are only 2 graphs shown, which is the spending and comparison of income and spending. Lastly, the application does not have to function to set reminder for payments. The application is also for personal use only. It does not have the option to join account with another user.

2.1.2 Bluecoins Finance: Budget

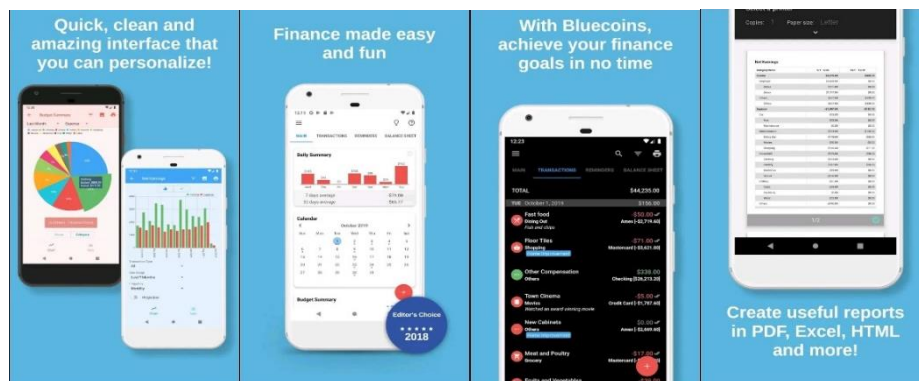


Figure 2.1.2 Bluecoins Finance: Budget [4]

Bluecoins is also an free Android-based [5] PFMS application created by Mabuhay Software. Bluecoins have a clean UI and support dark theme too. The visuals of the application is thoughtfully designed, and it is very useful for users to monitor their expenses. Bluecoins can let users to monitor their data by date, so that users can look back to their data easily. Different graphs are generate automatically too.

Next, Bluecoins also have a transactions tab that records every transaction keyed in, and a balance sheet tab that record user's balance. This app supports multi-currency, therefore it is more flexible for users to record their data based on different currencies. Bluecoins also support users to create their own budget, and it will alert users when they overspend. Filters could also be applied to the records to have a clearer view.

Additionally, Bluecoins allows users to set reminders for expenses. It can remind users to pay their bills after they set the reminder. The reminder could be set as remind the user once, daily, weekly, monthly, or annually. It is useful for users who have a bad memory to remind them before the payment goes overdue.

Bluecoins also provide users the option to setup application security. A passcode and fingerprint could be setup for blocking unauthorized access. This is great for maintaining account security.

Lastly, the records can be export into PDF document. It would be useful for users who want to share their expenses with others or want to keep a report of their expenses into documents. Overall, it is a very useful app.

Strength

The strength of Bluecoins is its data visualization. There are more than 9 graphs that can visualize the financial status of the user. The graphs not only consist of monthly summary, it have also yearly summary and daily summary. The application also have the function of setting reminders to remind payment of constant fees like rental fees. The app's interface is well-organized and easy to navigate, with a simple, intuitive layout that makes it easy to find and use different features.

Weakness

Firstly, the limitation of Bluecoins is that it does not have debt management function. Users could not put a debt in the application and input income to repay the debt. Another weakness of Bluecoins is that it cannot import files into the records. Users cannot import the files in the app, and they will need to type it all again as the records could not be sync to another device. This is not user friendly for users who have multiple devices. The application is also for single user only. It does not have the option to join account with another user.

2.1.3 Money Manager & Expenses



Figure 2.1.3 Money Manager & Expenses [6]

Money Manager & Expenses is a FMS application developed by Innim Mobile Exp that can be accessed by both Android and iOS system. The key feature of this PFMS is that it not only supports multicurrency which show real time exchange rate, but it also supports cryptocurrencies such like Bitcoin and Ethereum. Hence, this function let this application stand out from its competitors.

Money Manager & Expenses also provide basic graphs illustrations to show the total expenses and income in a day, a week, a month, a period, and a year. The transaction details could also be kept record of in the transactions tab. Although this is only a basic graph, but it is enough for light users to monitor their expenses and income.

Moreover, Money Manager & Expenses can let users to set reminders for regular payments too to make sure every payment is never overdue. The reminder could be set as remind the user once, daily, weekly, monthly, or annually, and it will be in the form of notification.

The application also provides users the option to export their data as Microsoft Excel File. It also can let users to attach pictures to their income. The data of the application could be sync to another device as it is cloud-based.

Lastly, Money Manager & Expenses provide users the option to setup application security. A passcode could be setup for blocking unauthorized access. This is great for maintaining account security.

Strength

The application had provided real time exchange rates that can sum up all the accounts that the user has, and sum up the money in every account to any currency the user wants.

Money Manager & Expenses also have reminder settings, not only remind user to do fee payment every month, but also record the expenses automatically.

Money Manager & Expenses also allows users to export financial data and allow cloud synchronization to access user data from another device. All records are also kept in the server therefore the data are not lost after user change device to login.

Weakness

Money Manager & Expenses could not let users to import any files in the app. It would be inconvenient for users as they need to type in every record again if they are from another app. Next, the limitation of this application is that it does not have a budget recommendation function. The app only let users to record their transactions but does not teach users how to manage their financial income correctly. Users may not notice they have overspent. Users savings may decrease little by little as time pass by.

2.1.4 Belanjawanku

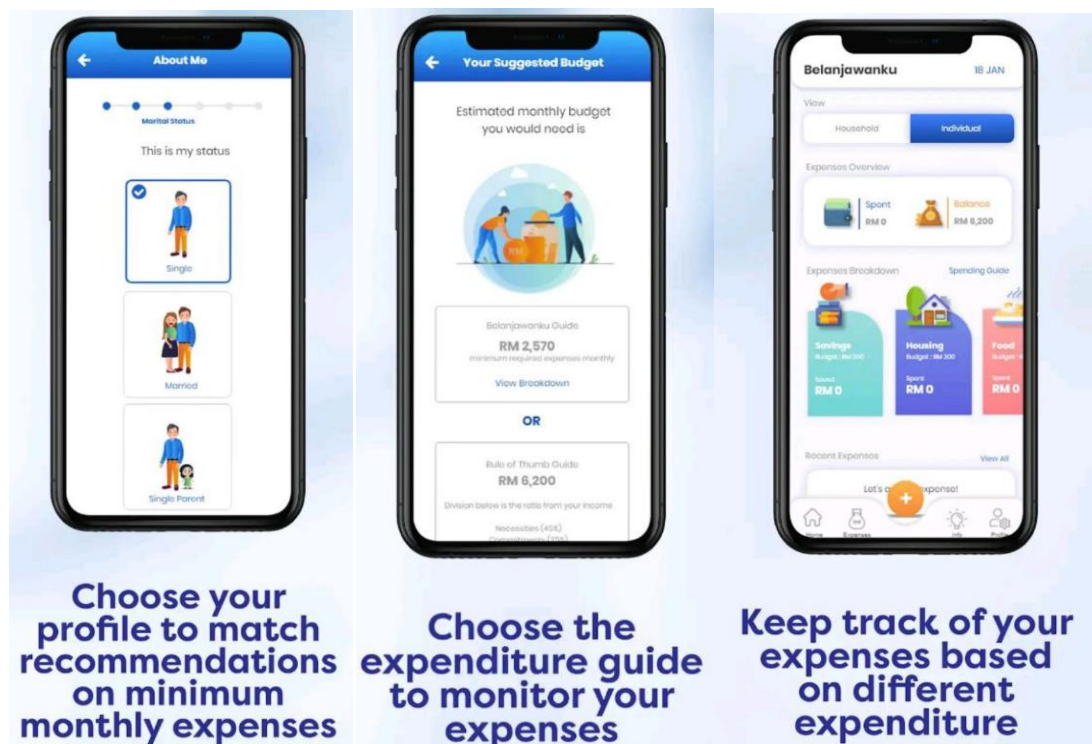


Figure 2.1.4 Belanjawanku [7]

Belanjawanku is a very good FMS application developed by Kumpulan Wang Simpanan Pekerja (KWSP) that can be accessed on both Android and iOS system. The key feature of this FMS is that it can generate a budget based on your personal data like location and marital status. Additionally, it also provide a rule of thumb guide option for those who do not want do use the app budget recommendation. By following the guide, user can save more and spend less. The application budget guide is generated based on Social Wellbeing Research Centre (SWRC), which provides the minimum monthly expenses of different state and personal status.

Belanjawanku also have a clean UI that shows all the transactions clearly. You can know if you are out of budget as the UI will tell you when you overspent that month. The application also provides tips and tricks for users to teach them how to plan financially. Belanjawanku encourages their user to control unnecessary spendings and promotes financial education.

Strength

The application budget guide is tailored to suit the needs of user from different locations and status. The diversity of the budget make it more suitable for users to plan their spendings. The application also goes beyond budgeting by providing users with financial education on how to start budgeting and how to manage financial spending like rent or car loans. The application works hard to teach user how to control unnecessary spending and save more.

Weakness

The Belanjawanku guide only gives the minimum spending and will not change its ratio to a lower or higher amount. This makes the budget not suitable for the users who have higher income and can spend more. Although it have rule of thumb guide, but the guide have a constant ration, thus it is not personalized for every user. The ratio may be too high for some states and too low for urban area like Kuala Lumpur. The budget is non editable therefore it may be not suitable for every user.

Belanjawanku also lacks graphical visualizations. It does not have any graphs that summarize the data. Users can only see the expenses of the month. It will be hard for users to point out unnecessary spendings. The application is also a personal FMS only, and only can be used on a single device. Records in the app could not be exported or imported therefore it will be hard for users who want to switch to this budgeting app.

2.1.5 Money Manager: Budget & Expense

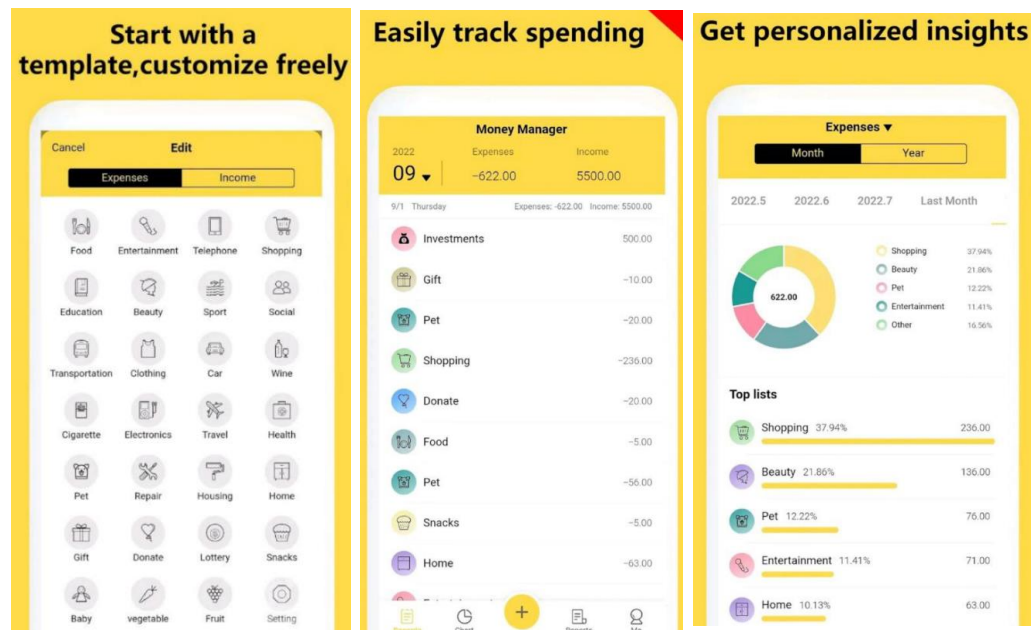


Figure 2.1.5 Money Manager: Budget & Expense [8]

Money Manager is another average FMS application developed by Horoscope365. The application can be accessed both on Android and iOS system. The application focuses on expenses tracking and can store receipts in the record. The special part of the application is that user can search for a tag to display the application. User can also press on the calendar to see the spending on that day.

The premium version of the app also supports features like reminders, planned payments records, multiple currencies and exportation of data. In that way, user can organize and optimize their spendings.

The data visualization is also good, as data is categorized into different categories. User need to select which category their spending is and the simple infographics can easily be understood. Graphs are prepared every month to summarize the financial spending of the user too.

User data could be synchronized too if they login on another device as it has an online database.

Strength

The calendar function can let users find back their spending more easily. This calendar function may be more suitable for users who have income everyday. As an example, vendors of stalls who have different income amount every day. This can let them improve their work efficiency and let users have a clear picture of their income and spending.

The category search function is also very useful. User can select a category and see how much they spend on that category in a month. User can clearly know which category has the most weight distribution.

Weakness

The free version of the app lack many features, budgeting, reminder, planned payments, multi currencies and exportation of data is not available. The features of the free app is too basic, therefore new users may need some time to get the ropes of the application. User may not success financially as there are no budget planning function. The application is also a single user FMS only.

2.2 Comparison Table

Table 2.2 Comparison Table of Review Systems

	Goodbudget	Bluecoins Finance	Money Manager & Expenses	Belanja wanku	Money Manager	Proposed App
Generate budget	×	×	×	√	×	√
Graphs	√	√	√	×	√	√
Data cloud Sync	√	√	√	×	√	√
Export	×	√	√	×	√	√
Reminder	×	√	√	×	√	×
Multi currency	×	√	×	×	√	
Password	√	√	√	×	√	√
Auto generated budget based on user data	×	×	×	×	×	√
Auto generated budget based on income range	×	×	×	×	×	√
Editable budget	√	√	×	×	√	√
Joined account	×	×	×	×	×	√

Chapter 3 System Model

3.1 Methodology

The processes of the project were categorized into different phases in the development, which were Software specification, Software development, Software validation, and . The project includes agile methodology in development. This is because agile methodology can minimize the time of project completion as it embraces changes [9]. Requirements may be improved over time and agile method plan and adapt new strategy on the spot, leading to higher productivity. Agile does not requires extensive upfront analysis during requirement gathering phase. Therefore agile is the most suitable methodology.

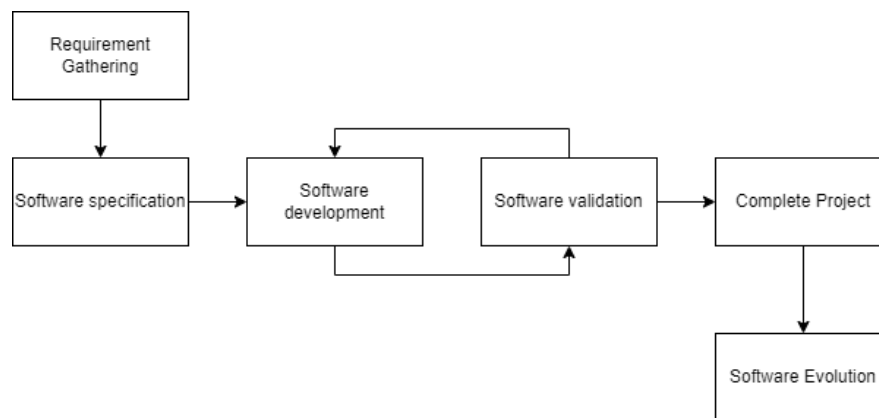


Figure 3.1.1 Agile Methodology Software Development Life Cycle

The agile methodology starts with requirements gathering on the proposed application. The steps include requirements gathering and requirements elicitation.

After the completion of the previous phase, the proposed application will start software specification phase. Planning and analysis of the proposed project is done during this phase, and the user stories, functional and non-functional requirements and software specifications will be determined and recorded.

Next, the proposed application will continue to software development phase. A software architecture pattern will be chosen based on the requirements of the project. The coding process will continue after the architecture pattern is decided, and the proposed application will be tested simultaneously to eliminate bugs and error. System testing and component testing will loop with software development

Software evolution is not involved in the proposed project yet as it is not complete yet.

3.2 System Design Equation

Format for budget ratio calculation:

$$\text{Budget Ratio} = \frac{BCA}{BSum}$$

(1)

where

BSum is Budget Sum

BCA is Budget Category Amount

Format for Recalculated Budget when user's 1-80% of his/her income is higher than basic budget:

$$\begin{aligned}\text{Recalculated Budget} &= 0.8(BR(\text{Income})) \\ \text{Savings} &= 0.2(\text{Income})\end{aligned}$$

(2)

Format for Recalculated Budget when user's 80-95% of his/her income is higher than basic budget:

$$\begin{aligned}\text{Recalculated Budget} &= \text{Basic Budget} \\ \text{Savings} &= \text{Income} - \text{Basic Budget}\end{aligned}$$

(3)

Format for Recalculated Budget when user's 95% of his/her income is higher than basic budget, or income is lower than basic budget:

$$\begin{aligned}\text{Recalculated Budget} &= 0.95(BR(\text{Income})) \\ \text{Savings} &= 0.05(\text{Income})\end{aligned}$$

(4)

where

BR is Budget Ratio

3.3 Use Case Diagram and Description

3.3.1 Use Case Diagram

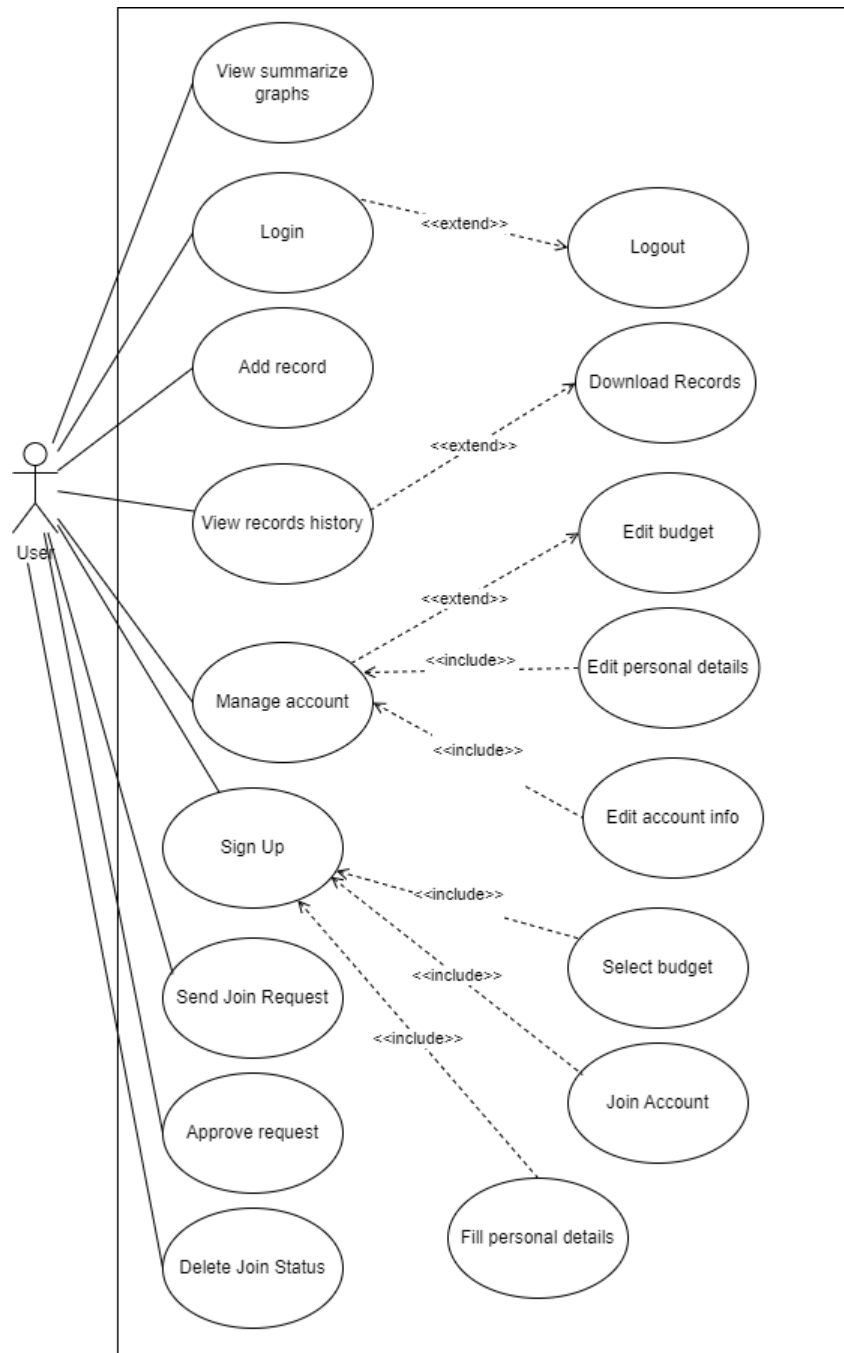


Figure 3.2.1 Use Case Diagram

CHAPTER 3

Actor: User

User act as the primary actor that interacts with the system to perform various task.

Use Case

Use Case Name: View Summarize Graphs		ID: 1
Primary Actor	User	
Description	User can view the summarize graphs generated by their spending records. The graphs will display data like average spending, spending of each category, highest and lowest spending and other useful data.	
Trigger	User click on Graph Button on Main Page	
Flow of events	1. User enter Summary Page	
Sub flows	N/A	
Alternate flows	N/A	

Use Case Name: Login		ID: 2
Primary Actor	User	
Description	User can login to access their account. User can logout after they finish using their account.	
Trigger	User press login button to login	
Flow of events	1. User key in id and password 2. User press login button 3. User enter Main Page	
Sub flows	N/A	
Alternate flows	2a. System prompt invalid Id or password if credentials does not match	

CHAPTER 3

Use Case Name: Add record		ID: 3
Primary Actor	User	
Description	User can add spending record.	
Trigger	User press	
Flow of events	<ol style="list-style-type: none"> 1. User select date 2. User select category 3. User key in description 4. User key in amount 5. User press button to submit 	
Sub flows	N/A	
Alternate flows	<p>5a. System prompt empty values error and require user to fill in all values if empty values detected.</p> <p>5b. System prompt amount 0 error if user key in 0 for amount</p>	

Use Case Name: View records history		ID: 4
Primary Actor	User	
Description	User can view the spending history of theirs. The spending history will include both personal records and joined account records. User can choose which to view.	
Trigger	User click on Record History Button on Main Page	
Flow of events	<ol style="list-style-type: none"> 1. User enter Record History Page 2. System show list of records 3. User select respective buttons to view personal or joined records. 4. User select spinner values to filter records 	
Sub flows	N/A	
Alternate flows	N/A	

CHAPTER 3

Use Case Name: Manage account		ID: 5
Primary Actor	User	
Description	User can edit their personal details, budget details, account details.	
Trigger	User click on Profile Button on Main Page	
Flow of events	<ol style="list-style-type: none"> 1. User enter Profile Page 2. User click Edit Personal Details button <ol style="list-style-type: none"> 2.1. User fill in Personal Details and select Budget again 3. User click Edit Budget button <ol style="list-style-type: none"> 3.1 User select Budget again 4. User click Edit Account button <ol style="list-style-type: none"> 4.1. System prompt edit window 4.2. User fill in new email or password 4.3. User press submit button to submit 	
Sub flows	N/A	
Alternate flows	N/A	

Use Case Name: Sign Up		ID: 6
Primary Actor	User	
Description	User can open new account. User need to fill in personal details to generate budget. User can select budget and key in other user JoinKey to request Join account.	
Trigger	User click on Sign Up button when enter app	
Flow of events	<ol style="list-style-type: none"> 1. User enter sign up page 2. User fill in ID, Email, Password 3. User fill in personal details 4. User select budget 	
Sub flows	N/A	
Alternate flows	<ol style="list-style-type: none"> 2a. System prompt error if there are empty values or duplicates 3a. System prompt error if there are empty values 	

CHAPTER 3

Use Case Name: Send Join Request		ID: 7
Primary Actor	User	
Description	User can send join account request.	
Trigger	A. User clicked on Join Account button on Profile Page B. User choose Join Account option on Sign Up	
Flow of events	1. User enter joinkey 2. User submit joinkey 3. System send request to receiver	
Sub flows	N/A	
Alternate flows	2a. System return joinkey error if joinkey does not exist	

Use Case Name: Cancel Join		ID: 8
Primary Actor	User	
Description	User can send cancel join account.	
Trigger	User clicked on Cancel Join Account button on Profile Page	
Flow of events	1. User clicked on Cancel Join Account button 2. System delete join status for both users 3. System generate new join key for both users	
Sub flows	N/A	
Alternate flows	N/A	

Use Case Name: Accept Request		ID: 9
Primary Actor	User	
Description	User can approve join account request.	
Trigger	User clicked on Pending Request button on Profile Page	
Flow of events	1. System shows sent request 2. User click on Approve button to confirm request 3. System delete other request of user after user accept request.	
Sub flows	2a. System join account if join Request approve	
Alternate flows	N/A	

3.4 Flow Chart

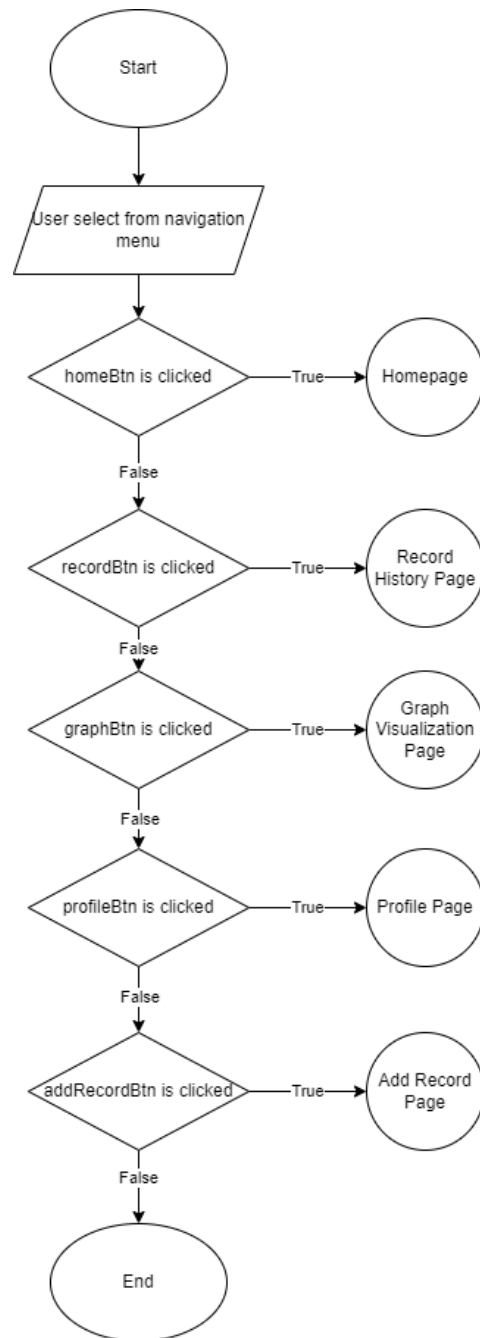


Figure 3.4.1 Home Page Flow Chart

Figure 3.4.1 shows the process of the home page menu. The menu can let users choose to access the page they want. When the page button is selected, it will direct the user to the page by using an OnClick function. If user does not click any button, the flow will end and user will remain on the home page.

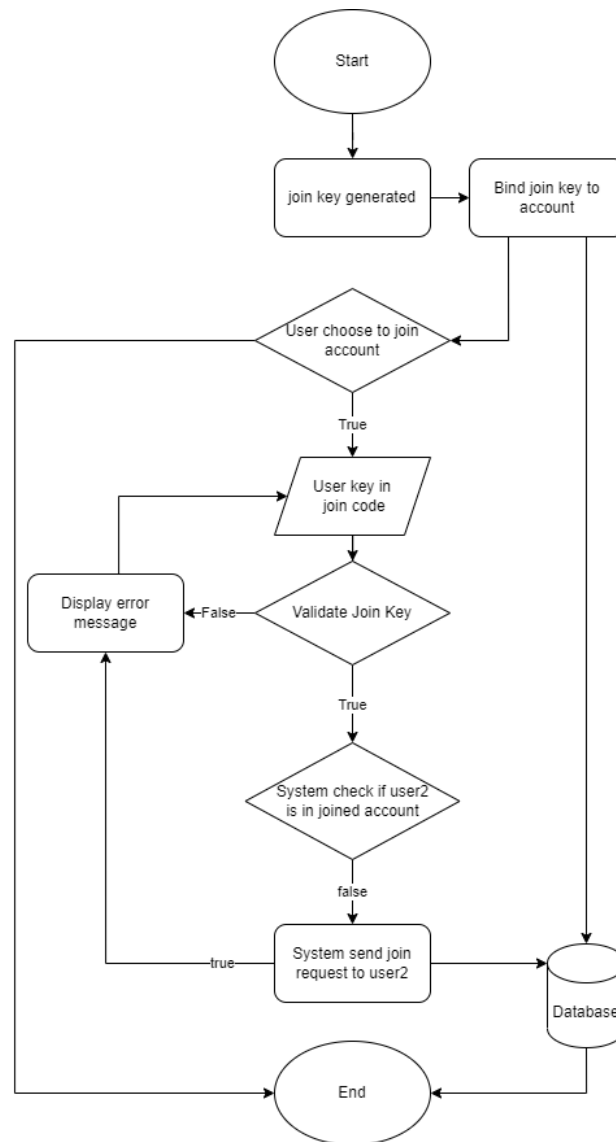


Figure 3.4.2 Joining Account Flow Chart

Figure 3.4.2 shows the process of users joining their account with another user. The task starts when a user chooses to join an account. If a user enters a join key and presses join, the system will check for the join key if it is bonded to another user. If the join key is valid and the 2nd user slot of the join key is empty, the system will send a join request to the user. If the join key is not valid or the 2nd user slot of the join key is not empty, the system will prompt an error message to let the user enter a new join key and try again. If a user chooses not to enter a join key at the start, the system will generate a join key and assign it to the user if the user does not have a join key bonded to the account yet. The join key will be recorded in the database and the user ID will be recorded as user 1.

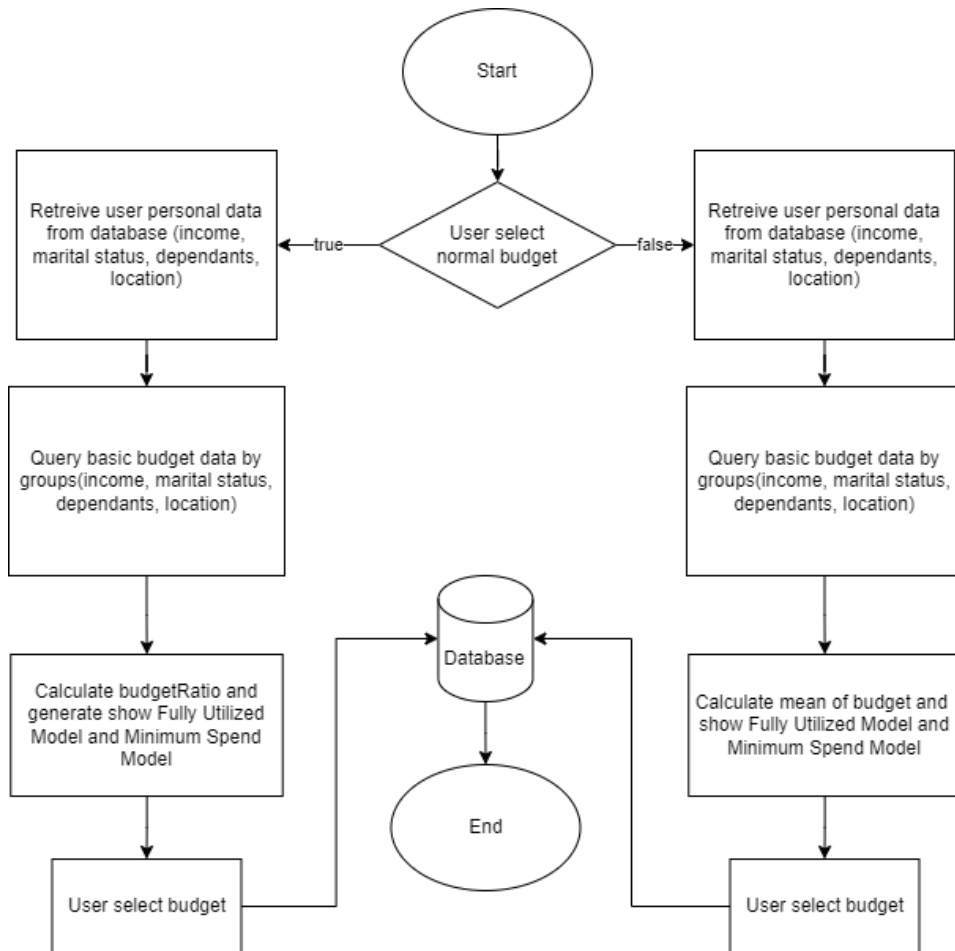


Figure 3.4.3 Auto Generated Budget Flow Chart

Figure 3.4.1 shows the process of the autogenerated budget for proposed application. The server will carry out calculations on the user personal budget that is recorded in the database. User personal budget is first retrieved from the database and categorized into different groups such as location, income, marital status and dependents for both budget types. If user select normal budget, calculations is done to get budget to income ratio and the result is generated to Fully Utilized model budget and Minimum spend model budget. If user does not select normal budget, all user data that are in the same category (income, marital status, state, dependant) will be used to get the mean of the budget. A new budget template will be generated and will be updated for the user.

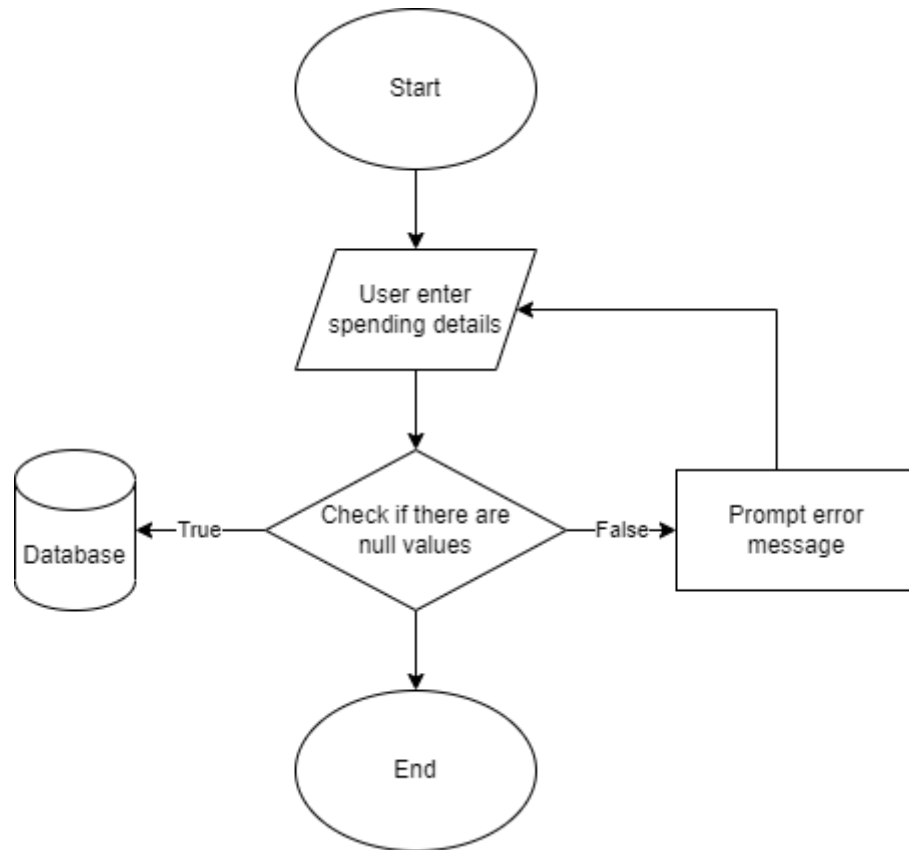


Figure 3.4.4 Add Spending Records Page Flowchart

Figure 3.4.4 shows the process of the add spending records. First, user enter the add records page and fill the spending details. This includes choosing a category, enter spending description and spending amount. The system will check if there are any empty values after user submit. If there are no empty values, the record will be stored in the database. If there are empty values, system will prompt user with an error message and request user to fill in all the details. The flow will end after user successfully submit the record.

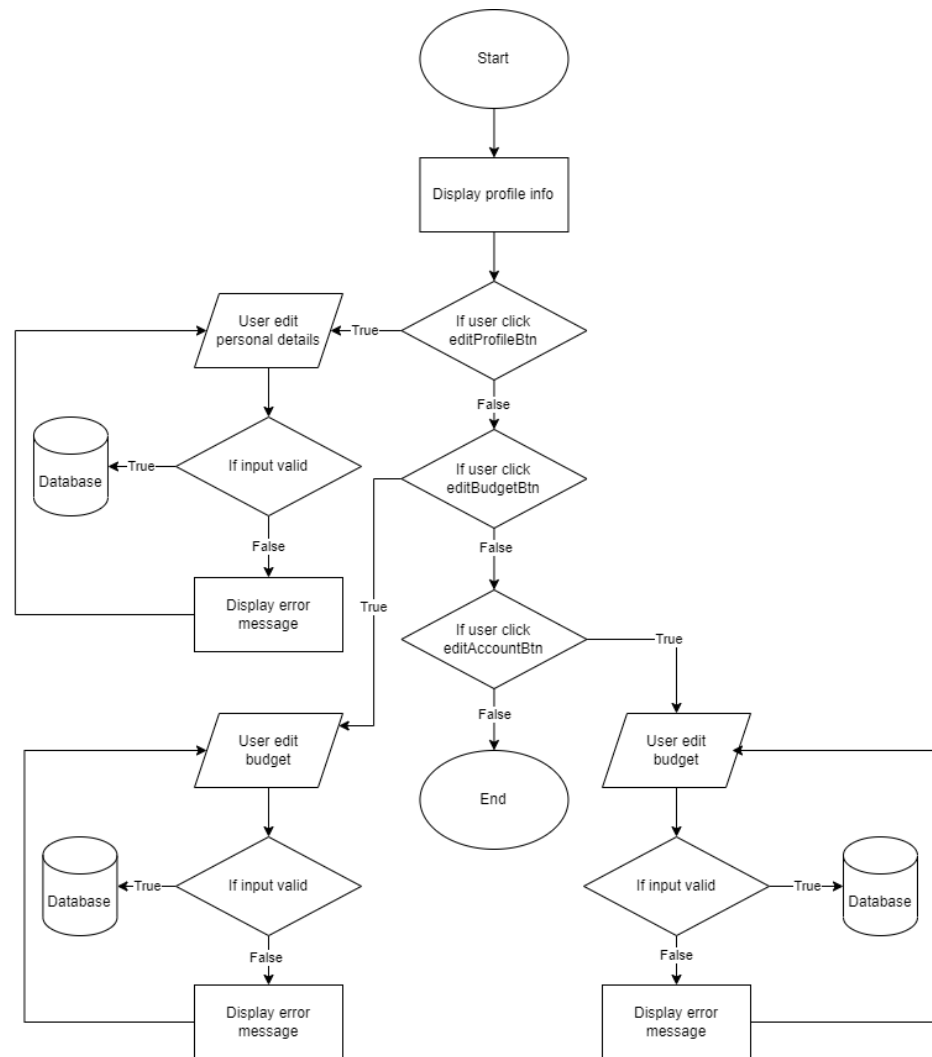


Figure 3.4.5 Profile Activity Flow Chart

Figure 3.4.5 shows the process of the user editing their profile personal data. System will display user personal data like income, marital status, age, living location, dependents, and the budget plan they choose. If the user choose to edit the details, they will have to key in the values. The values will be validated so there are no empty values. If there are empty values, error message will be displayed and user will be prompt to fill in correct values. If successfully validated, the info will be submitted to the database table. User can also choose if they want to edit their information. If user choose to edit their info, system will check if their information is valid or not then upload to database. If invalid, system will prompt error for user to reenter their info.

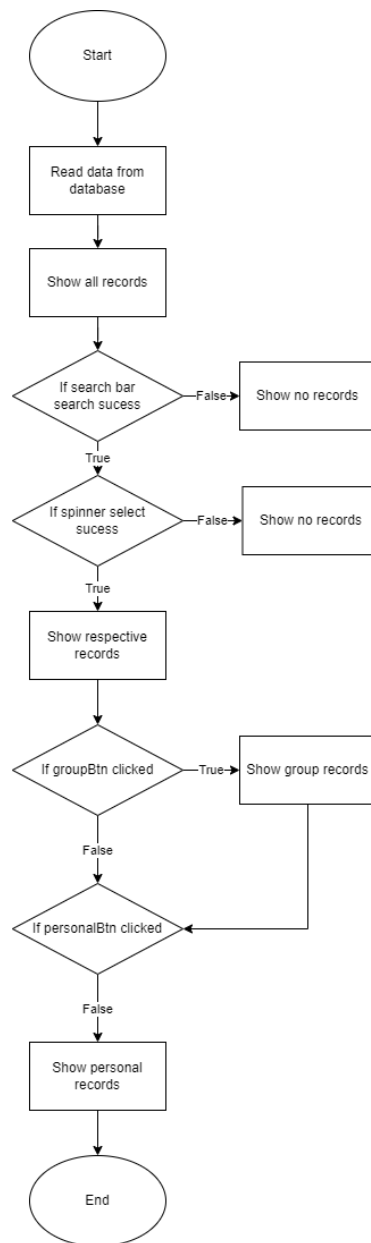


Figure 3.4.6 Show Records History Flow Chart

Figure 3.4.6 shows the process of the display of records history. The system first read data of the user records from the database and display them in the page. If joined button is click, the recycler view will display group spending records. If the button is not clicked, the recycler view will continue to display personal spending records. The spinner of the page when selected will filter the records based on selection and display selected results. This apply to search bar too.

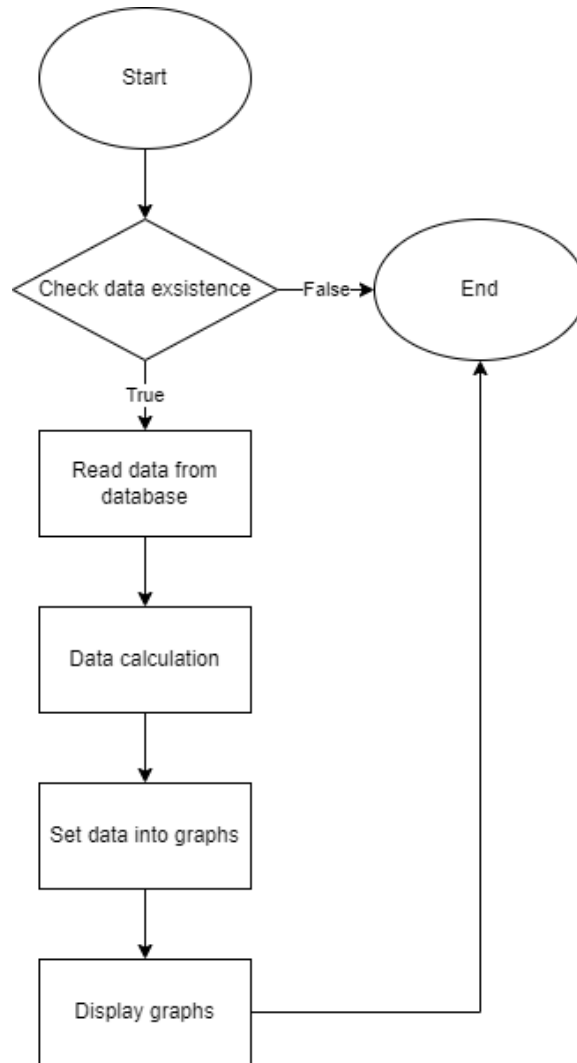


Figure 3.4.7 Show Graph Flow Chart

Figure 3.4.7 shows the process of the display of graph visualization. First, system check if there is any spending data. If there is spending data, system will read spending data from the database and sum up the data by categories. Several graphs will be generated and the data will be filled into the graphs. The graphs will be displayed after calculation complete. If there are no spending data, the flow will end.

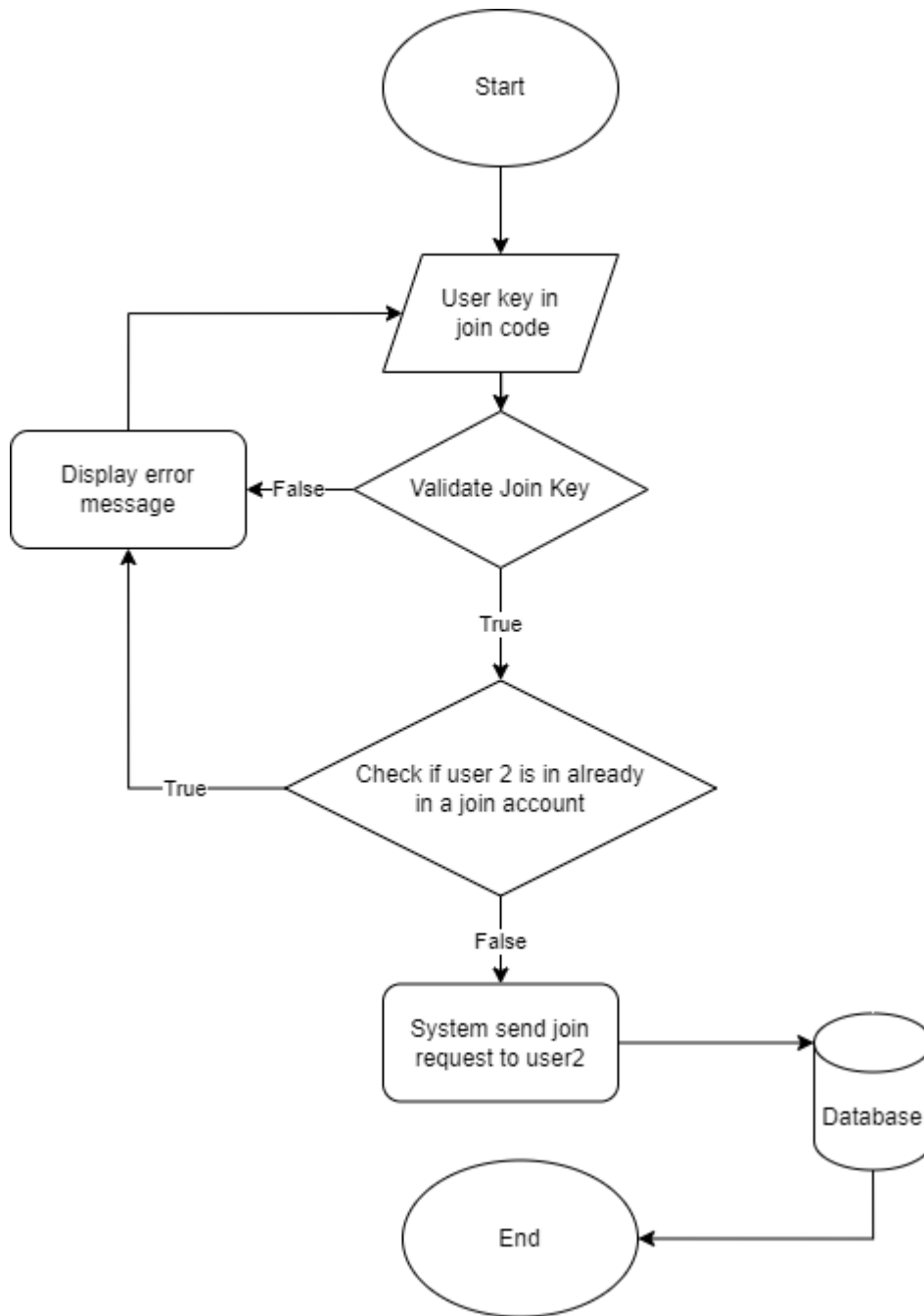


Figure 3.4.8 Profile Join Account Flow Chart

Figure 3.4.8 shows the process of the user join account at Profile page. The task start when user chooses to join an account. If user key in a join key and press join, system will check for the join key if it is bonded to another user. If the join key is valid and the 2nd user slot of the join key is empty, system will send join request to the user. If join key is not valid or the 2nd user slot of the join key is not empty, system will prompt error message to let user key in a new join key and try again.

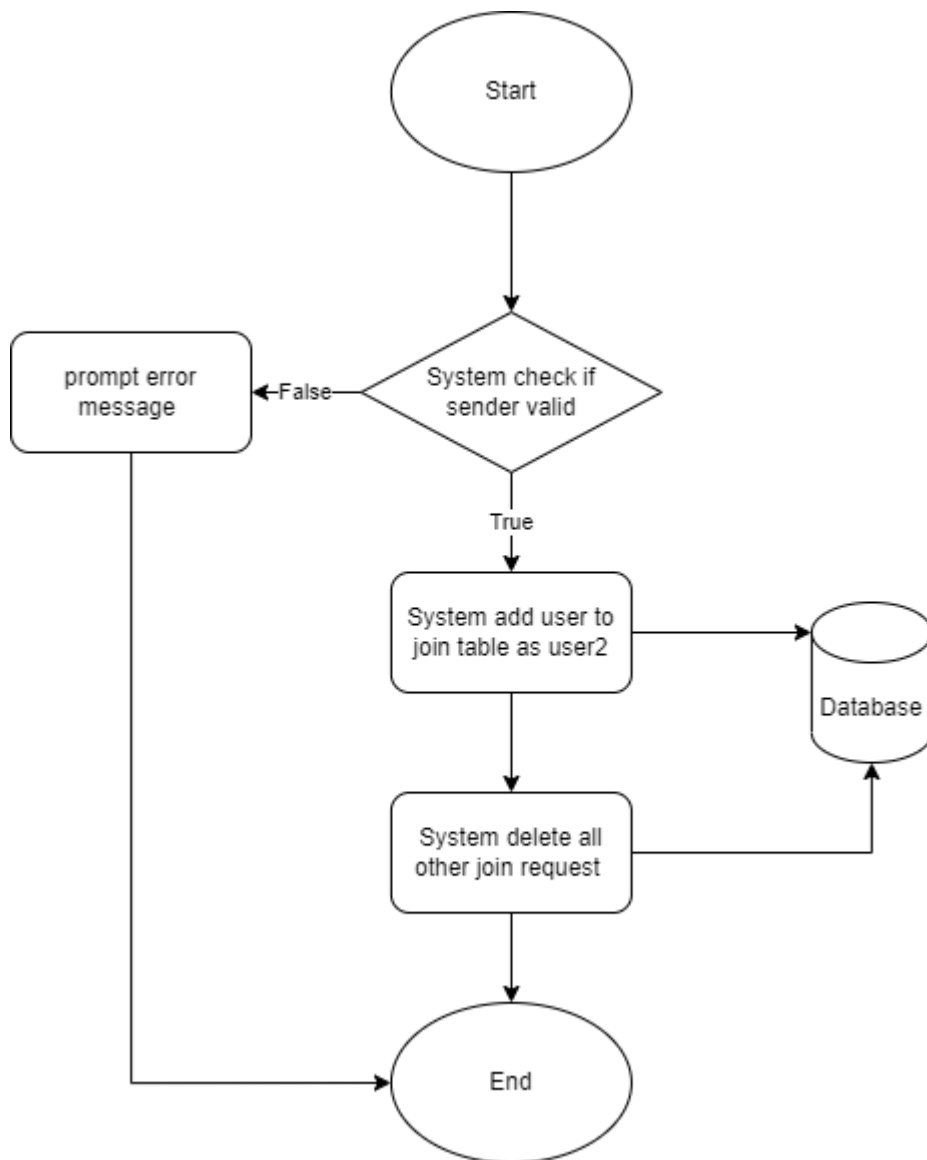
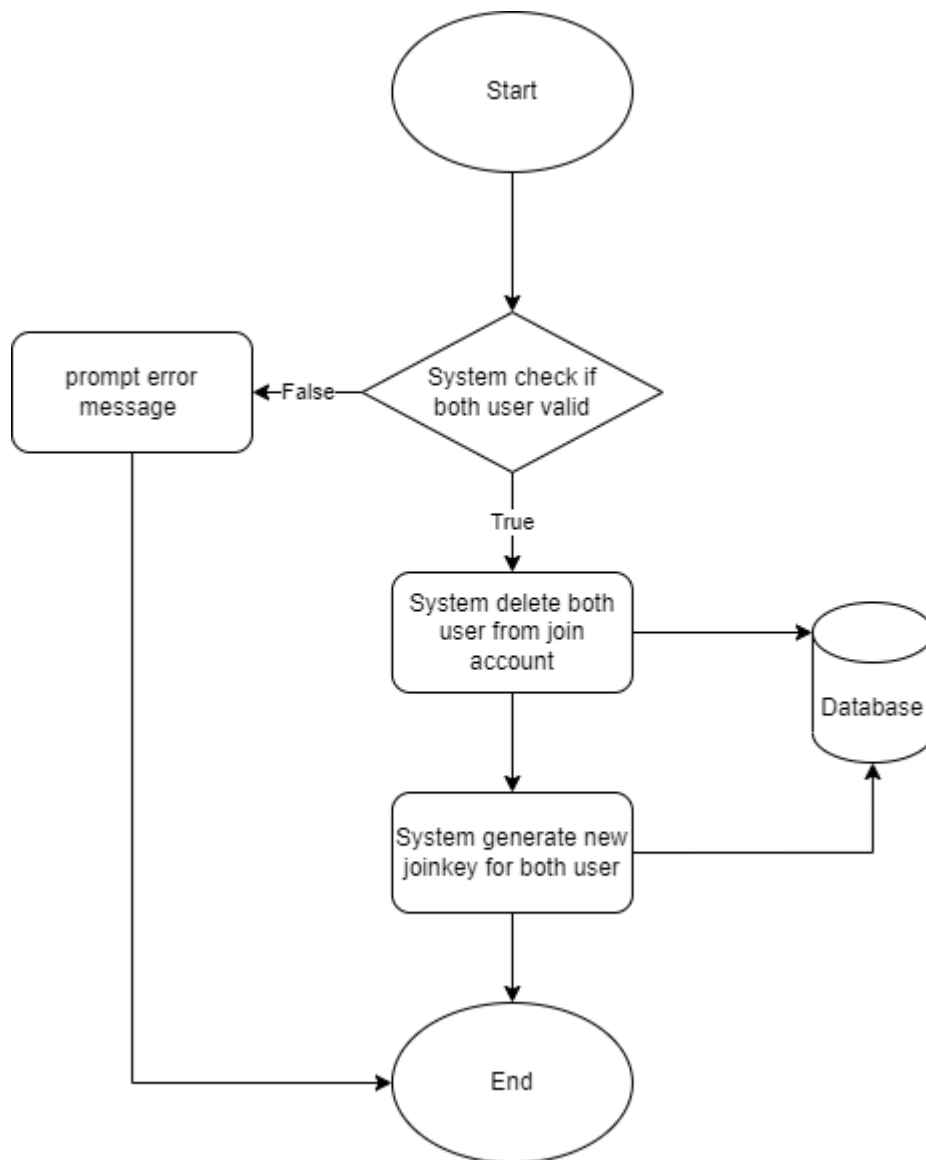


Figure 3.4.9 Accept Join Request Flow Chart

Figure 3.4.9 shows the process of the user join account at Profile page. The task start when user chooses to press the accept button for join request. System will check for the join key if it is bonded to another user. If the request is still valid and the sender is not in a joined account, system will add the sender userID as user2 and delete all other join request. If join key is not valid or the 2nd user sender is in a joined account, system will prompt error message and delete the join request.



3.4.10 Cancel

Figure 3.4.10 Join Flow Chart

Figure 3.4.10 shows the process of the user join account at Profile page. The task start when user chooses to press the button for cancel request. System will check for the join key if it is bonded to another user. System will cancel join status for both users and generate new join key for both users

Chapter 4 System Design

4. 1 System Block Diagram

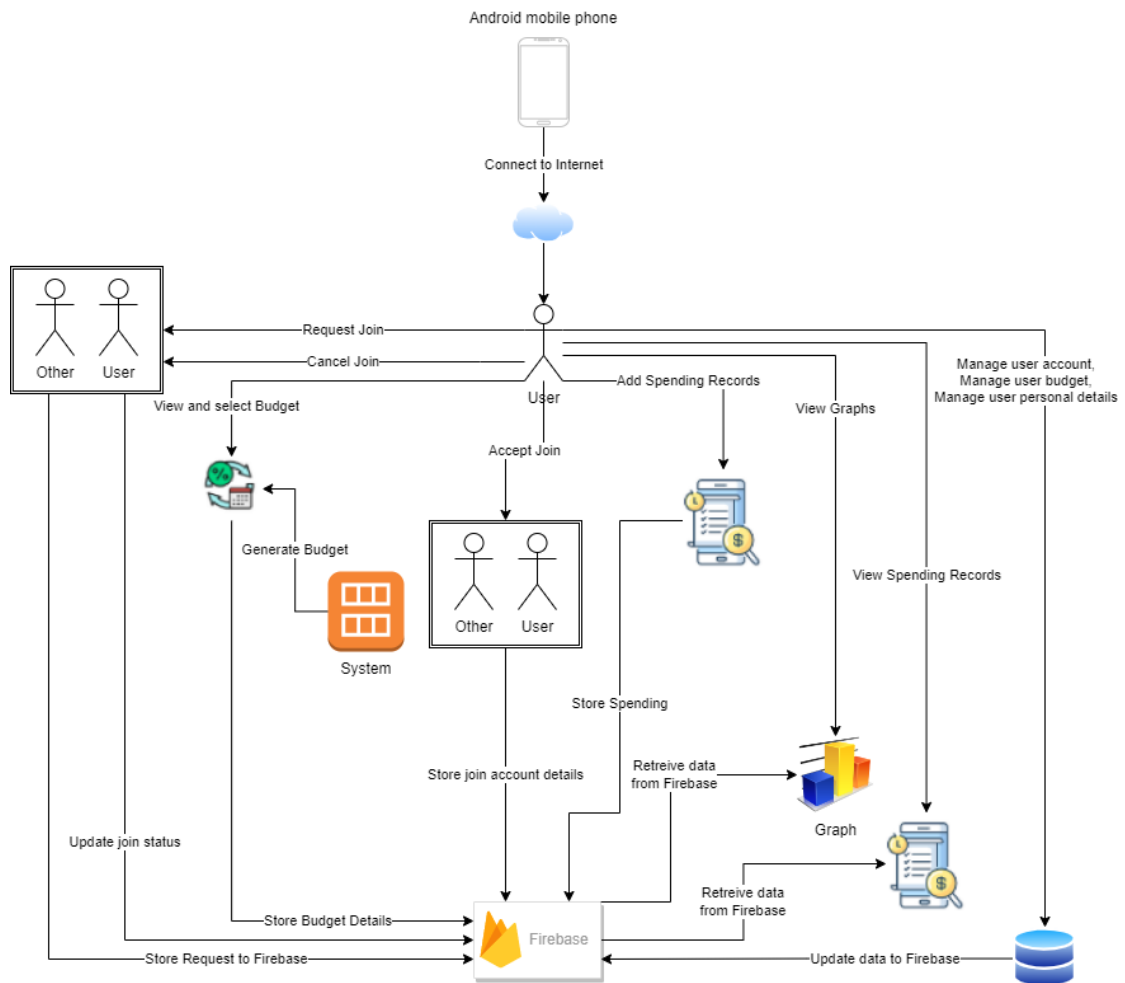


Figure 4.1.1 System Block Diagram

4.2 System Flow Description

1. signup.java

The user must first connect to the Internet in order to use the developed application. User must first register as a user. The register details will be stored in “user” table in Cloud Firestore. User are require to fill in userID, email and password. If input empty or have duplicates, system will prompt error and let user to reenter. After success registration, user account details will be stored in Firebase Cloud Firestore. User will be directed to joinAcc class.

2. joinAcc.java

The user will need to select to join other account or not. If join, user need to key in valid joinkey to send join request to other user, system will check if joinkey valid and receiver user is not in another join account. If valid, system send join request to receiver. If not valid, system prompt error message. if no join, joinkey will be generated and stored in Firebase Cloud Firestore. User will be directed to signup_personaldetails class.

3. signup_personaldetails.java

The user will need to fill in all personal details like age, income, marital status, car status, rent status, living area, and dependant. If input empty or error, system will prompt error and let user to reenter. After success registration, user account details will be stored in Firebase Cloud Firestore. User will be directed to setBudgetActivity class.

4. setBudgetActivity.java

The user will need to select which budget plan they prefer. The plan will be passed to the next page to generate respective budget. User will be directed to showBudgetActivity class.

5. showBudgetActivity.java

The class will populate all the budget amount for both budget plan. The user will need to select which budget plan they prefer. The plan type will be store in

Firebase Cloud Firestore and the plan details will be pass to next page for user to edit. User will be directed to editBudgetActivity class.

6. editBudgetActivity.java

The user can edit the budget for each category if they want. The plan details will be stored in Firebase Cloud Firestore. User will be directed to MainActivity class.

7. login.java

User will need to key in username and password to login to application. If details invalid or incorrect, system will prompt error to let user register or enter correct details. User will be directed to MainActivity class.

8. MainActivity.java

This class contains bottom navigation bar of home, spending history, add spending, graphs, and profile. User can click on respective button to go to respective page.

The page also get user spending data and user budget from Firebase Cloud Firestore and display the data.

9. recordHistoryActivity.java

The class shows all spending records of users. If user key in anything in searchbar, page populated all records related. Page spinner will prompt list of date spending records available and categories. User can select spinner to populated related records. All records are retrieve from Firebase Cloud Firestore. User can choose to download excel file for history.

10. addSpendingActivity.java

User will need to key in all required spending details. System will prompt error for empty or invalid values and let user reenter. Spending records will be stored in Firebase Cloud Firestore.

11. GraphActivity.java

The class get spending records data from Firebase Cloud Firestore and populate graphs and summarize spending.

12. ProfileActivity.java

The class get user information from Firebase Cloud Firestore. User can choose to edit personal details. User will be directed to `signup_personaldetails` class. User can choose to edit budget. User will be directed to `setBudgetActivity` class. User can choose to edit account information. System will prompt window for user to enter details. System will validate and store data to Firebase Cloud Firestore. User can choose to join account. User will be directed to `ProfileJoinAccountActivity` class. User can choose to cancel joined account. System will delete `joinstatus` from Firebase Cloud Firestore and generate new `joinkey` for user.

13. ProfileJoinAccountActivity.java

User need to key in valid `joinkey` to send join request to other user, system will check if `joinkey` valid and receiver user is not in another join account. If valid, system send join request to receiver. If not valid, system prompt error message. if no join, user will be directed back to `ProfileActivity` class.

14. JoinRequestActivity.java

The class will populate all join request from other users to current user. User can choose to accept one of the request. system will check if `joinkey` valid and receiver user is not in another join account. If valid, system write join data into Firebase Cloud Firestore and delete all request from other user. If not valid, system prompt error message.

4.3 Hardware and Software Requirements

4.3.1 Software

Before starting to develop the JFMS, there is 1 software needed to be installed and downloaded in my desktop:

1. Android Studio Iguana (Android SDK 34)

4.3.2 Hardware

An Android phone is needed to carry out testing, and a computer is needed to develop the program.

4.4 System Components Specification

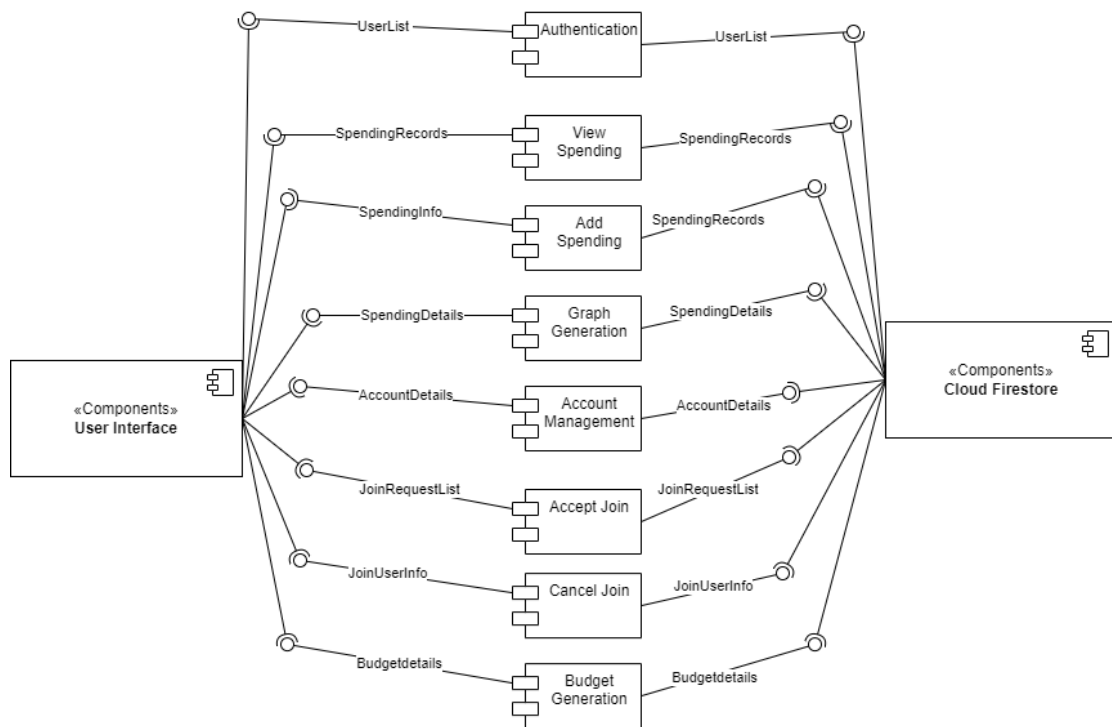


Figure 4.4.1 System Components Specification Diagram

The proposed project has 1 interface for user, and 9 module that include Authentication, View Spending, Add Spending, Graph Generation, Account Management, Accept Join, Send Join Request, Cancel Join, and Budget Genration. All data are stored on Cloud Firestore

4.5 System Components Interaction Operation

1. Authentication Module

The module will request data from user and request data from Cloud Firestore to identify user information. After verifies, user is allowed to login.

2. View Spending Module

The module will request data from Cloud Firestore to query specific records that users select.

3. Add Spending Module

The module will request spending data from user. Spending data will be pass to Cloud Firestore to be stored after verifying the data.

4. Graph Generation Module

The module will request spending details from Cloud Firestore. The module will process the data into graph and show the graph to user.

5. Account Management Module

The module will request account details from Firestore and user side. When user edit any information, account details data will be passed to Firestore and stored after verified.

6. Accept Join Module

The module will request join user details from Firestore. Join data will stored in firestore.

7. Send Join Request Module

The module will request join user details from Firestore. Join request data will stored in Firestore.

8. Cancel Join Module

CHAPTER 4

The module will request join user details from Firestore. Join data will be deleted in Firestore.

9. Budget Generation Module

The module will request user details from user and budget data from Firestore. After calculation, budget will be generated and updated to Firestore.

CHAPTER 4

4.6 Timeline

Project Task	Project Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Study FYP1 and identify changes	■													
Create FYP2 report based on template		■												
Improve content for report			■											
Complete enhancement in report				■										
Complete all drawings of flowcharts and diagrams					■									
Changed database						■								
Develop Profile Page							■							
Develop Autogenerated Budget								■						
Develop Summary Page									■					
Fixing bugs and continue development										■				
Complete app development											■			
Test and send app for review												■		
Send evaluation form and get result													■	
Complete FYP2 Report														■

Figure 4.6.1 Gantt Chart

Chapter 5 System Implementation

5.1 Hardware Setup

The hardware involved in this project is a computer desktop and an android mobile phone as testing device. The computer is used for coding and the android mobile device is to test out the implementation.

Table 5.1.1 Specifications of desktop

Description	Specifications
Motherboard	ASUS PRIME B660M-A WIFI D4
Processor	Intel Core i5-12400F
Operating System	Windows 11
Graphic	GIGABYTE GEFORCE RTX2060 WINDFORCE OC 12G
Memory	32GB DDR4 RAM
Storage	1.5TB M.2 SDD

Table 5.1.2 Specifications of mobile phone

Description	Specifications
Manufacturer	OPPO
Model	Reno
Processor	Qualcomm SDM710 Octa-core
Operating System	Android 11
Memory	6GB RAM
Storage	256GB

5.2 Software Setup

Android Studio : Integrated Development Environment (IDE)

The proposed project will be developed using Android Studio as IDE and the operating system will only be compatible for Android OS. The drag and drop function on the design of every activity lessen the weight for developers by decrease the chance for to manually code every layout file in the activity. Android also contains Software Development Kit (SDK) tools to make development easier [10]. SDK provides many useful libraries that can be used during development. Therefore, Android Studio is a perfect IDE for developers to develop an Android based mobile application.

Java : Programming Language

Java is a programming language that is suitable for object-oriented programming and used widely in android mobile applications development [11]. Java is easier to learn and has high versatility. Therefore the project is developed using Java.

Firebase Cloud Firestore : Database

Firebase Cloud Firestore is a scalable, flexible, cloud-hosted NoSQL database provided by Firebase, a platform developed by Google [12]. It allows users to store and sync data across devices in real-time and offers powerful querying capabilities. The database have real-time data sync function, which allows changes in database propagated to all clients in real time. It also has high scalability, meaning that it can handle large amounts of data and a high number of concurrent users.

5.3 Setting and Configuration

Once development is complete, the developer may simply click the "Run" button to launch the app on an Android device or the emulator provided by Android Studio. In order to conduct this project, I modified the device developer settings on my Android phone, a OPPO Reno by enabling USB debugging. This Android smartphone features a 1080 x 2340 resolution screen and an Android 11.0 CPU, all of which are suitable for the suggested application. Firebase Cloud Storage are the databases that are utilized to store all of the data.

5.4 System Operation (with Screenshot)

5.4.1 Application

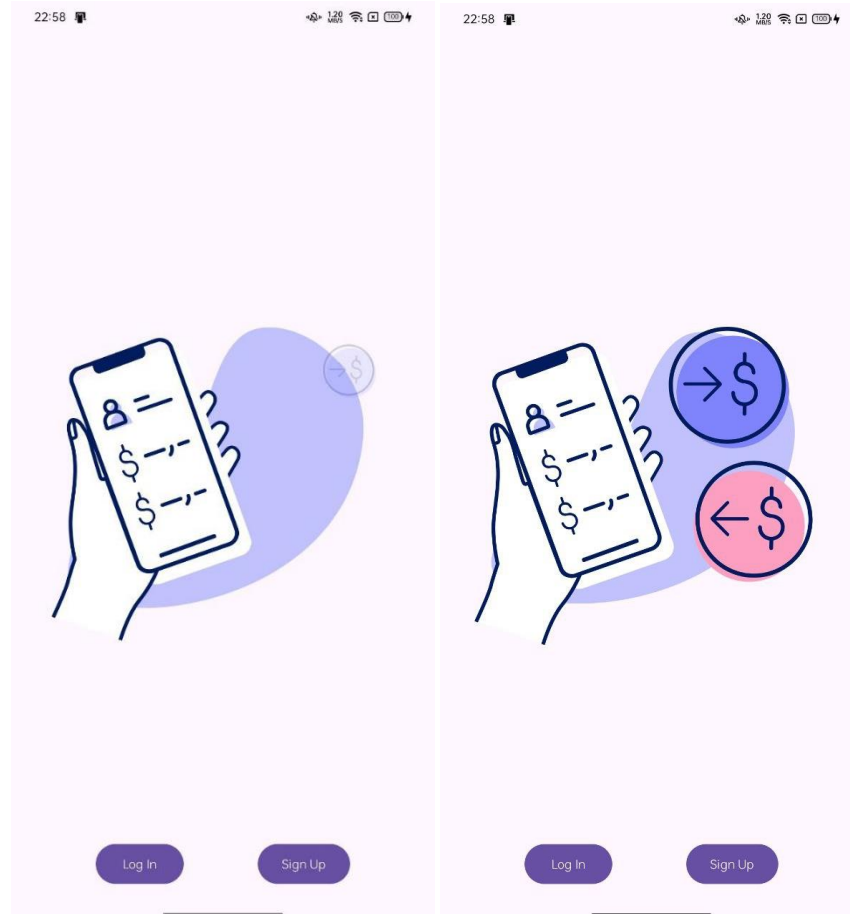


Figure 5.4.1 Splash Screen

The splash screen of the user will show up when user open the application. User can choose they want to log in to previous account or sign up a new account.

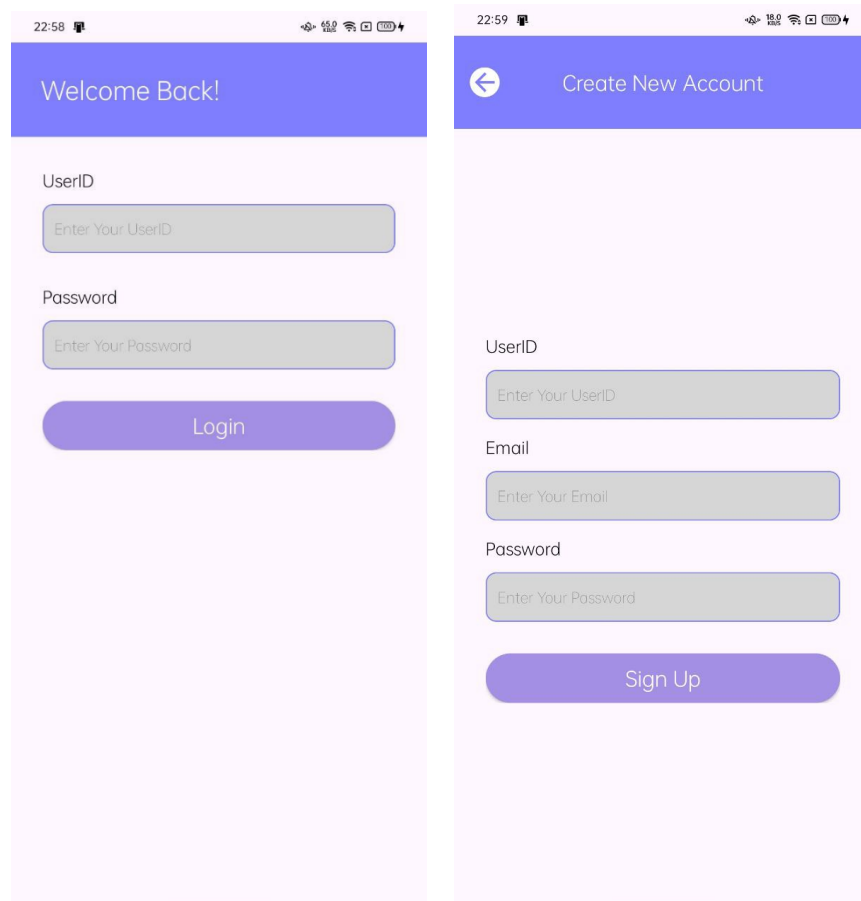


Figure 5.4.2 Login and Sign Up Page

The user is required to key in their correct user ID and password to login to their account. System will then validate if the ID is the ID and password is correct or not. Once pass system validation, the user then can only access to the main page of the proposed application.

The user is required to key in a unique userID email address and password for registering a new account. System will check if the userID or the email is duplicated. If is duplicated, system will let user key in the ID or email again. If userID and email is unique, a new account will be created. The userID, email and password will be stored in user_table in the database.

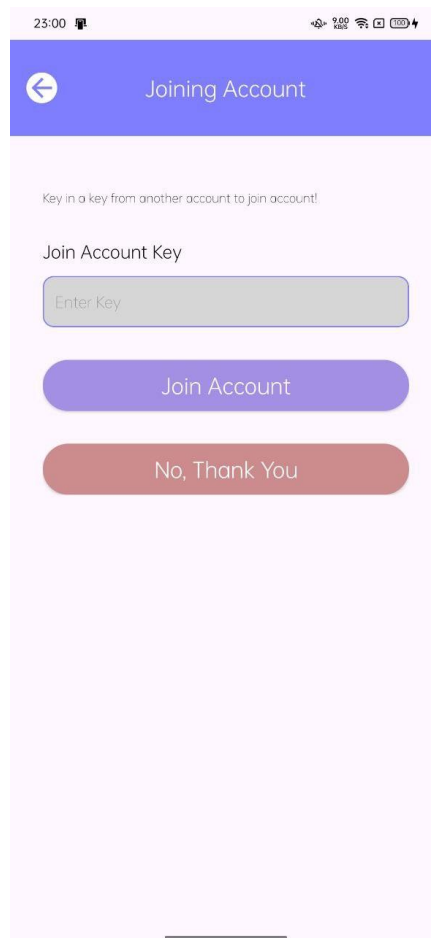
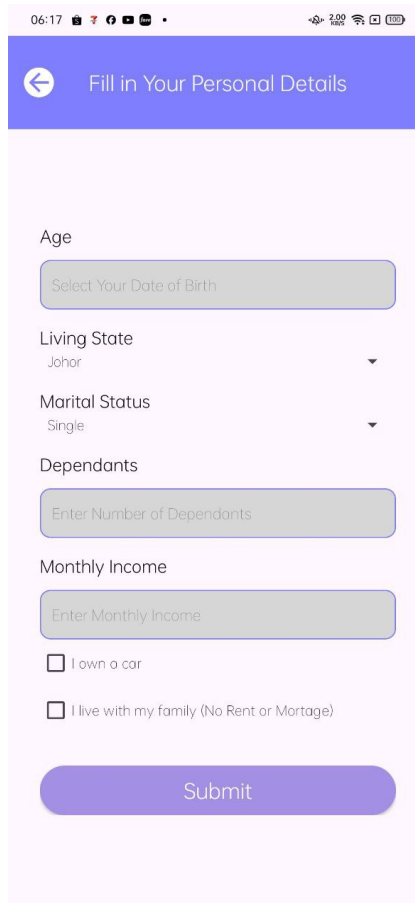


Figure 5.4.3 Join Account Module

The user will need to get the join key from another user in order to join the account. The system will check if the join key exist. If exists, it will then check if the second user in the record is empty or not. If join key exist and second user in the record is empty, user will successfully join account as second user in the table. If user key in a wrong join key, system will inform user that the join key is incorrect. If the join key is valid but the second user place is already occupied, user will fail to join too and system will tell user that the joined account is already occupied. User will need to click “No, Thank You” button to go to next page.

The “No, Thank You" button will generate a new unique join key. The join key will have 6 digits and consists of alphabets and numbers. System will check if the join key exist in the table or not. If the table does not contain this join key, the join key will be inserted as primary key and user ID as user 1 in the record. Other user can join the account by the join key.



06:17 2.00

← Fill in Your Personal Details

Age

Select Your Date of Birth

Living State
Johor

Marital Status
Single

Dependants

Enter Number of Dependents

Monthly Income

Enter Monthly Income

I own a car

I live with my family (No Rent or Mortgage)

Submit

Figure 5.4.4 Register personal details

After join account module, user will need to fill in their personal details. This includes age, living state, marital status, number of dependents, and monthly income. These data will be used to determine the amount of the autogenerated budget. System will validate each values and make sure there are no null values when user clicked submit. The data will be inserted in to the user_details table. After that, user will need to choose the type of budget they want. System will enter the selection into user_plan table in database.



Figure 5.4.5 Select Budget Type

User can choose to take the basic BudgetBuddy Guide or the Upgraded Guide. The basic guide is based on the budget data on KWSP minimum spend report data [13], and the upgraded guide is based on the mean of all users in same category.

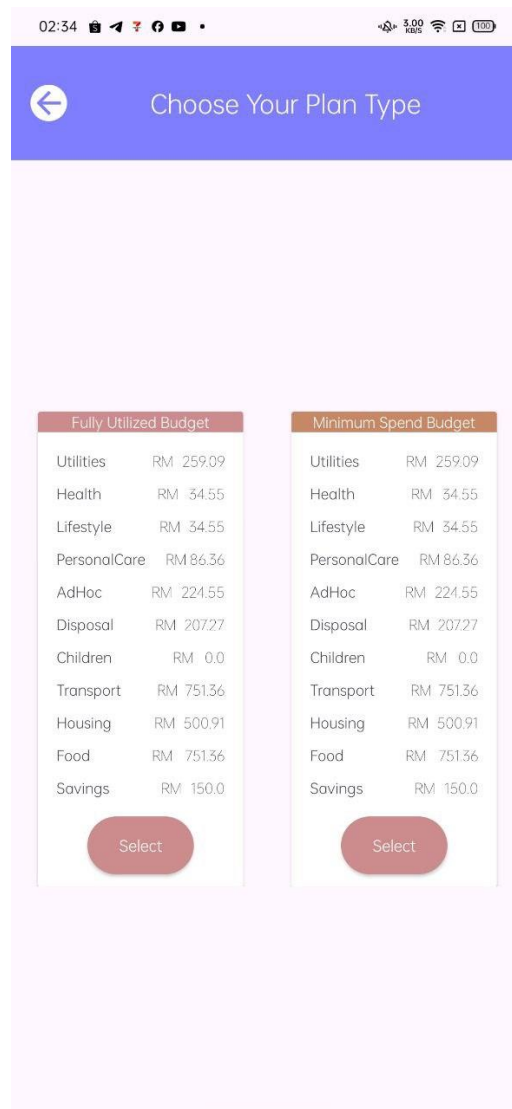


Figure 5.4.6 Show Autogenerated Budget

The autogenerated budget is shown after user select the budget guide type. User can select from the Fully Utilized Budget or Minimum Spend Budget, which that suit their needs.

02:34 2.00 100

Choose Your Plan Type

Fully Utilized Budget Total Amount: 3000.00

Utilities	RM	259.09
Health	RM	34.55
Lifestyle	RM	34.55
PersonalCare	RM	86.36
Ad-Hoc	RM	224.55
Disposal	RM	207.27
Children	RM	0.00
Transport	RM	751.36
	RM	500.00

Confirm Budget

Figure 5.4.7 Edit Budget

User can choose to edit their budget if they are not satisfied with the autogenerated budget. The budget data will be stored after confirmation of the budget.

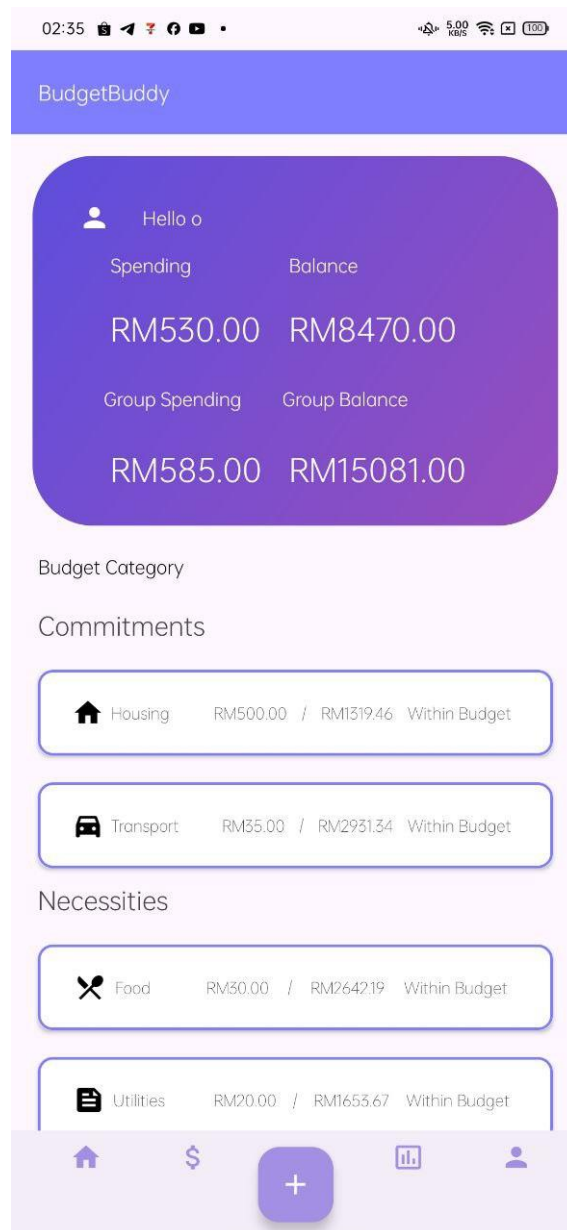


Figure 5.4.8 Home Page

The main page of the proposed application is used for displaying user budget and spending. The spending and balance of personal and group will be shown after user login. The cardview below will display the total spending of the users and whether it is within budget or not.

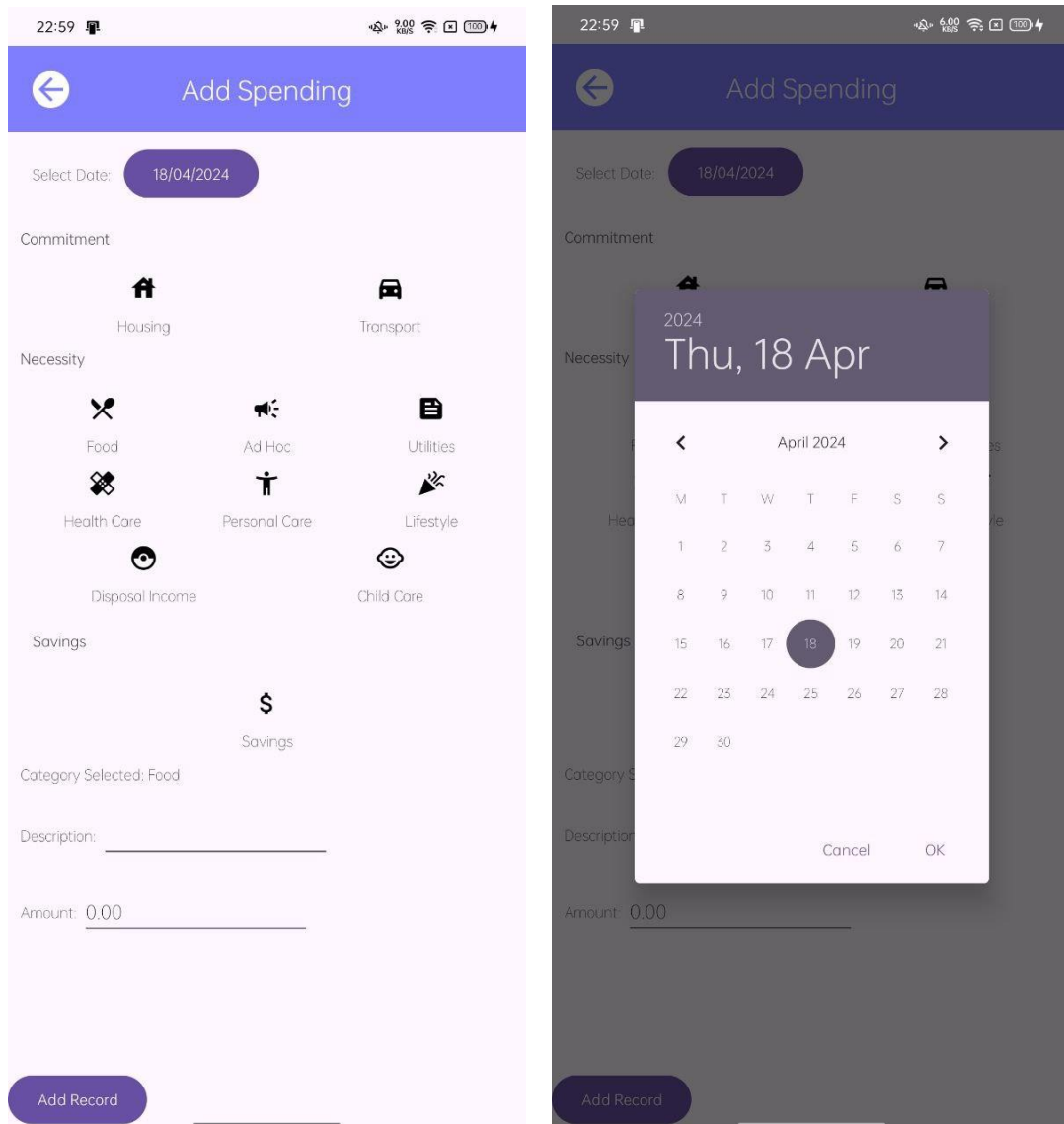


Figure 5.4.9 Add Records

The add spending records page can let users to add their spending records. User can select the date of the record or stick by today's date for date of record. Next, user need to select the category for the spending. The description and amount need to be filled in too. System will check and make sure there are no null values. If there are null values, system will prompt error message and request user to fill all the empty spaces. The validated data will then be inserted into the database in the table `user_spending`.

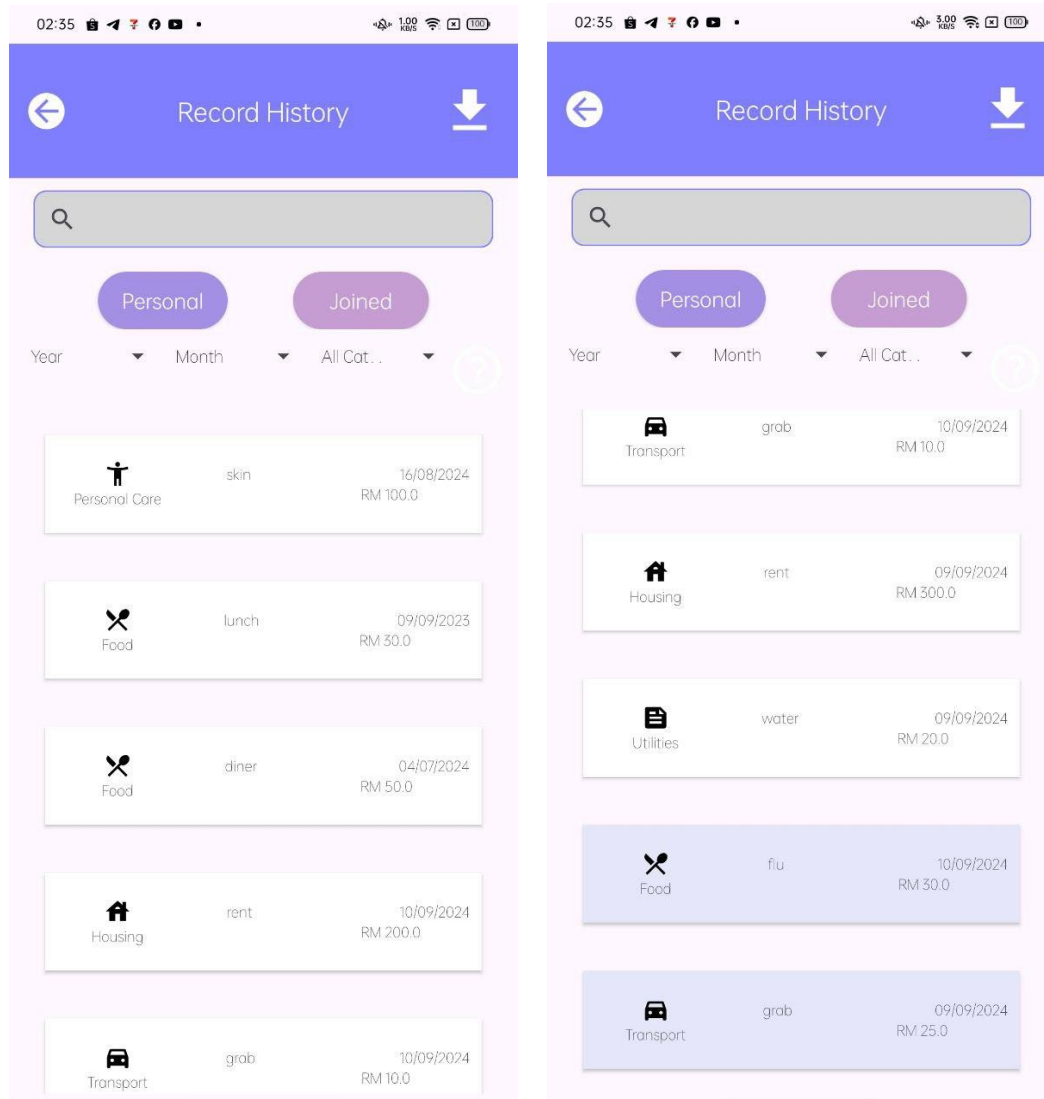


Figure 5.4.10 User Spending Records History

When user enter the Record History page, system will populate the recycle view with the user personal spending records if available. When user clicked the joined button, system will display spending records of user and joined user. The joined user records will be highlighted in purple. The spinner on the page will be populated according to the records. User can filter the records based on year, month, and categories. User can select the spinner data to filter the records they wanted to showcase. User can also type in the search bar to search for the records they wanted. The search will query all the records by spending description. All relevant records will be displayed. User can also click on the download button to download the data to an excel file.

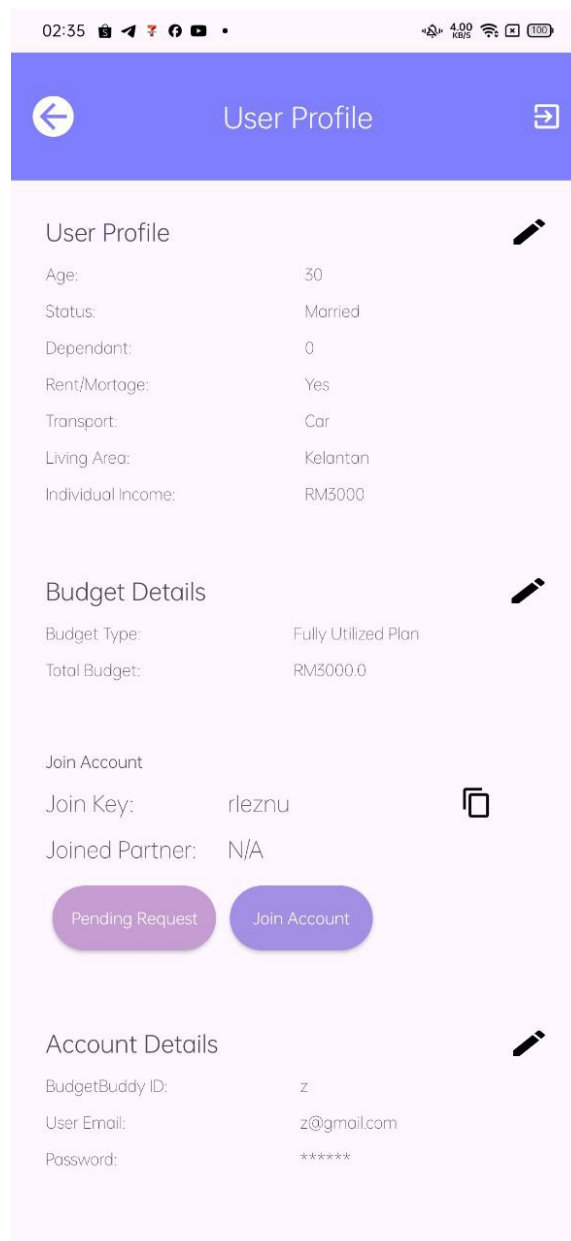


Figure 5.4.11 Profile Activity

The profile of the user will consist of 4 major parts. User personal details, user budget, join key data, and account details. The first part of the personal details include age, marital status, dependency status, rent, transport, living area and income. User can choose to edit the details by pressing the edit button, and system will prompt a warning window informing that the user will need to setup the budget again if they modify their personal information. The user will be directed to `signup_personaldetails` class to fill up all information.

The second part of the profile is about the budget details. It will show the budget type and total budget amount of the user. User can choose to edit the details by pressing the edit button, and

CHAPTER 5

system will prompt a warning window informing that the user will need to setup the budget again . The user will be directed to `setBudgetActivity` class to fill up all information.

The third part of the profile is about join account details. It will show the join key and the join partner. User can choose to join account by pressing the join account button. The user will be directed to `ProfileJoinAccountActivity`. This works similarly to `JoinRequest` activity, just it did not generate a new `joinKey`. As for `Remove Join Partner` button, the system will prompt a warning window to let user confirm deletion, then the join status will be deleted. Both users will be deleted from the join account. `Pending Request Button` will direct user to `JoinRequestActivity` class.

The last part is account details. It shows the ID and email of the user, and user can press edit button to edit their email and password.

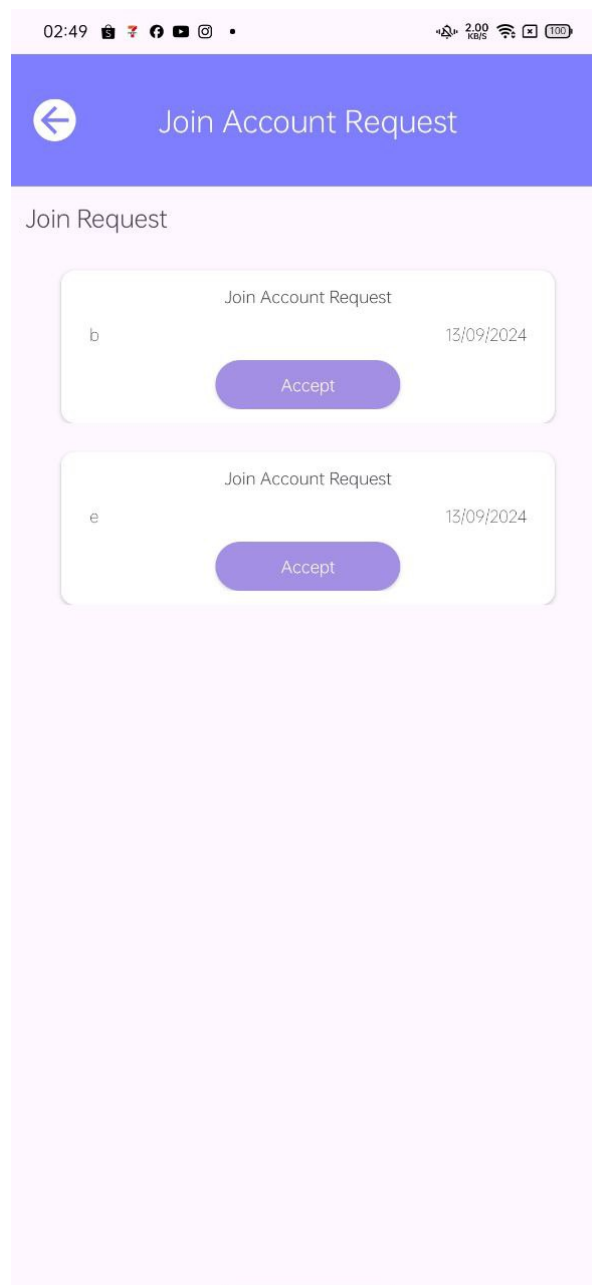


Figure 5.4.12 JoinRequestActivity

The JoinRequestActivity Page will show all the join request that is pending accept by user. All the records will be shown with the time sender sent request, the id of sender. User can choose one to accept and if the sender is still available, the join status will be updated in Firebase Cloud Firestore for both sender and user. If sender is not available, the request will be deleted and user can choose to accept other request. After success join, all pending request will be deleted.

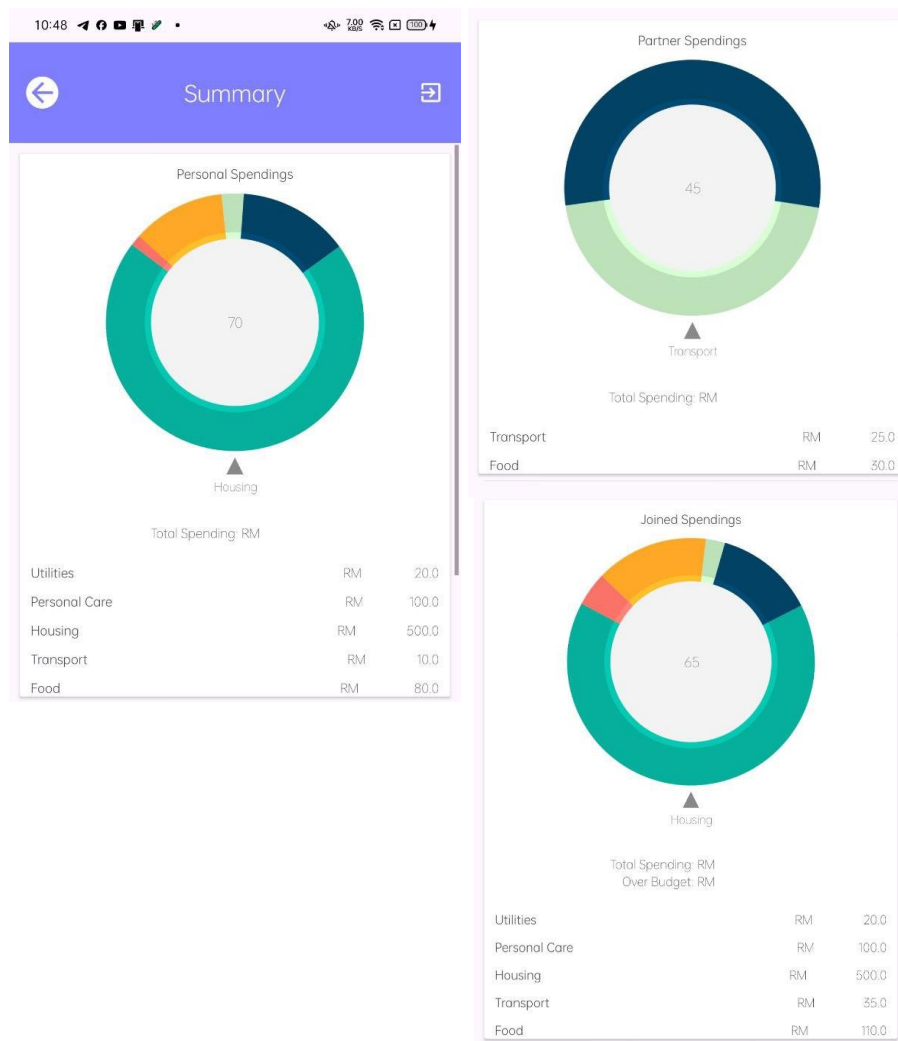


Figure 5.4.13 Graph Activity

The graph page of the user will display all the graphs that is generated using the user's spending data. The graph can be rotated to show the percentage of the spending category. All of the spending category sum will also be populated. There is 3 graphs, personal, partner and joined graphs.

Chapter 6 System Evaluation and Discussion

6.1 System Evaluation Survey Results

The developed app is shared among some UTARians, and an evaluation form is shared for their feedback for the app. There are a total of 15 respondents.

Email
15 responses

zhijing914@1utar.my
teh@1utar.my
ziyann0200@1utar.my
cjiazhi9731718@1utar.my
anderslim368@1utar.my
yitan02@1utar.my
yb0128@1utar.my
loklokchung@1utar.my
weixinkaw@1utar.my

Figure 6.1.1 Evaluation Form Participants



Figure 6.1.2 Evaluation Form Results Part 1

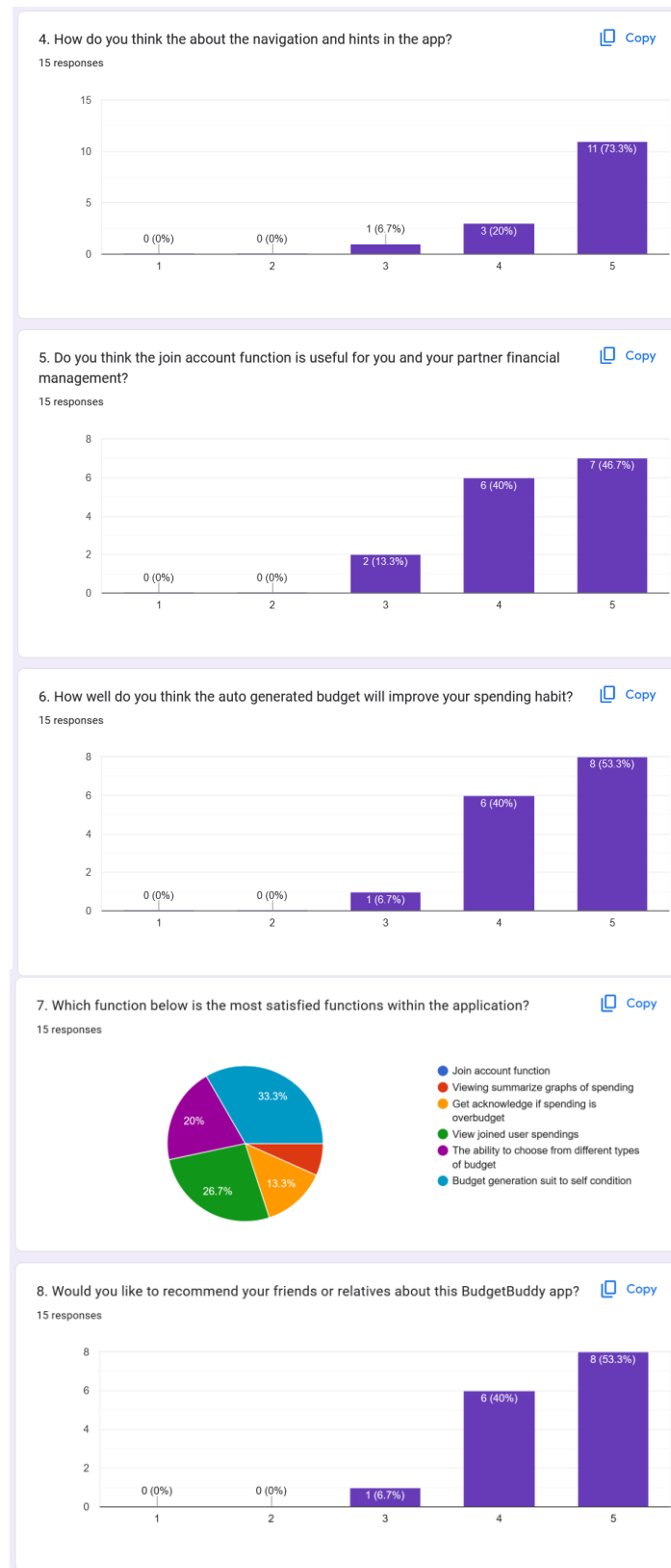


Figure 6.1.3 Evaluation Form Results Part 2

CHAPTER 6

The email of the respondents are recorded and shown in Figure 6.1.1. Through the evaluation in Figure 6.1.2, for question 1, 46.7% of the respondents gives a 5 mark for overall design, another 46.7% of the respondents gives a 4 mark for overall design, and 6.7% of the respondent gives a moderate 3 mark.

For question 2, 60% of the respondents gives a 5 mark thinking that the auto generated budget is reasonable, and the other 26.7% gives a 4 mark, while 13.3% gives a 3 mark. This may because some of the users thinks that the spending category does not suit their normal spending.

For question 3, 60% of the respondents give a 4 mark on how well the automated budget help to manage their financial needs. 33.3% of the users give a 5 mark, and 6.7% of the respondent gives a 3 mark. It can be concluded that majority of the testers are positive towards the autogenerated budget and find it useful.

Figure 6.1.3 shows question 4. 73.3% of the respondents thinks that the buttons and hints in the program are very clear and easy to understand. 20% of the respondents gives a 4 mark, and 1 person give a 3 mark. Overall, it can be concluded that the navigation of the app is user friendly and easy to use.

For question 5, 46.7% of the respondents gives a 5 mark for thinking join account function is useful for couple financial management. There are also 40% of the users gives a 4 mark, and 13.3% person who gives a 3 mark. This may because not every couple manage their spending together.

For question 6, 53.3% of the respondents gives a 5 mark for thinking that the autogenerated budget will improve their spending habits. There are also 40% of the respondents gives a 4 mark, and 6.7% of the respondents give 3 mark.

For Figure 6.1.4 question 7, the highest percentage of 33.3% respondents are most satisfied with the budget generation that suit to self-condition. The second place goes to view joined user spendings, which has a 26.7% vote. The third places goes to the ability to choose from different types of budget, with a 20% vote. The fourth place goes to get acknowledge if spending is overbudget, with a 13.3% vote, and a 6.7% of the respondents vote for viewing summarize graphs of spending, which is the fifth place. The last place is join account function, with no votes.

For question 8, 53.3% respondents gives a 5 mark and 40% of the respondents gives a 4 mark on recommending the BudgetBuddy app. 6.7% of the respondents give a 3 mark.

To summarize the evaluation, majority of the respondents are positive about the app, and have the most satisfaction on the auto generated budget.

6.2 System Testing and Performance Metrics

Login Test Case				
Test Case: Login				
Test Case ID: F01				
Date Created: 6/9/2024				
Role: User				
No.	Input Values	Expected Results	Actual Results	Pass/Fail
1	Enter valid user details and click the 'Login' button in the Login page.	The system authenticates the user details.	User was authenticated and redirect into the application.	Passed.
2	Enter invalid user details and click the 'Login' button in the Login page.	The system displays an error message.	Error message displayed.	Passed.

Sign Up Test Case				
Use Case: Sign Up				
Function ID: F02				
Date Created: 7/9/2024				
Role: User				
No.	Input Values	Expected Results	Actual Results	Pass/Fail
1	Enter valid user details and click the "Sign Up" button in the Login page.	System record data into database and redirected to next page	System record data into database and redirected to next page	Passed.
2	Enter duplicate userID or user email and click the "Sign Up" button in the Login page.	The system displays a message account already exist.	The system displays a message account already exist.	Passed.
3	Empty values and click the "Sign Up" button in the Login page.	The system displays an error message.	Displayed error message.	Passed.

Fill User Details Test Case				
Use Case: Fill User Details				
Function ID: F03				
Date Created: 7/9/2024				
Role: User				
No.	Input Values	Expected Results	Actual Results	Pass/Fail
1	Enter valid user details and click the “Submit” button.	System record data into database and redirected to next page	System record data into database and redirected to next page	Passed.
2	Empty values and click the “Submit” button.	The system displays an error message.	Displayed error message.	Passed.

Join Account Test Case				
Use Case: Fill User Details				
Function ID: F04				
Date Created: 7/9/2024				
Role: User				
No.	Input Values	Expected Results	Actual Results	Pass/Fail
1	User enter valid join key and click “Join Account” button	System record data into database and redirected to next page	System record data into database and redirected to next page	Passed.
2	User click “No, Thank You” button	System record data into database and redirected to next page	System record data into database and redirected to next page	Passed.
3	User enter invalid join key and click “Join Account” button	The system displays an error message.	Displayed error message.	Passed.

Add New Spending Test Case				
Use Case: Add New Spending				
Function ID: F05				
Date Created: 7/9/2024				
Role: User				
No.	Input Values	Expected Results	Actual Results	Pass/Fail
1	Click the 'Add Record' button after entering pet details partially in Add Post page.	The system prompt user to fill in all the fields.	The system prompted user to fill in all the fields.	Passed.
2	Click the 'Add Record' button after entering all the pet details in Add Post page.	Record is stored	Record is stored	Passed.
3	Click the 'Add Record' when spending Amount is 0	The system prompt user "Spending amount could not be 0"	The system prompt user "Spending amount could not be 0"	Passed.

View Post Test Case				
Use Case: View Spending Records				
Function ID: F06				
Date Created: 7/9/2024				
Role: User				
No.	Input Values	Expected Results	Actual Results	Pass/Fail
1	Click the 'Personal' button.	System display all personal records.	System display all personal records.	Passed.
2	Click the 'Joined' button.	System display all personal records and joined records.	System display all personal records and joined records.	Passed.
3	Select respective year, month, category on spinner.	System display all records related.	System display all records related.	Passed.
4	Type in search bar to search for records.	System display all records related.	System display all records related.	Passed.

Profile Page Test Case				
Use Case: Profile				
Function ID: F07				
Date Created: 7/9/2024				
Role: User				
No.	Input Values	Expected Results	Actual Results	Pass/Fail
1	Click the edit button on the right of User Profile	User directed to fill user details page	User directed to fill user details page	Passed.
2	Click the edit button on the right of Budget Details	User directed to fill budget details page	User directed to fill budget details page	Passed.
3	Click the edit button on the right of Account Details	System prompt window for user to edit account details	System prompt window for user to edit account details	Passed.
4	Click the copy button on the right of Join Account	Join Key copied	Join Key copied	Passed.
5	Click the “Pending Request” button	User directed to pending request page	User directed to pending request page	Passed.
6	Click the “Join Account” button	User directed to join account page	User directed to join account page	Passed.
7	Click the “Remove Join Partner” button	Window prompt warning window.	Window prompt warning window.	Passed.
8	Click “Ok” in 7 warning window.	System remove join status for user and joined user	System remove join status for user and joined user	Passed.

Profile Page Edit Account Test Case				
Use Case: Edit Profile				
Function ID: F08				
Date Created: 7/9/2024				
Role: User				
No.	Input Values	Expected Results	Actual Results	Pass/Fail
1	User fill in valid email or valid password or both and submit	Record is stored	Record is stored	Passed.
2	User fill in invalid email and submit	The system prompt user invalid email.	The system prompt user invalid email.	Passed.
3	User fill in existing email and submit	The system prompt user duplicate email.	The system prompt user duplicate email.	Passed.

6.3 Objectives Evaluation

To propose a system that can provide budget recommendations based on location, dependents, and marital status.

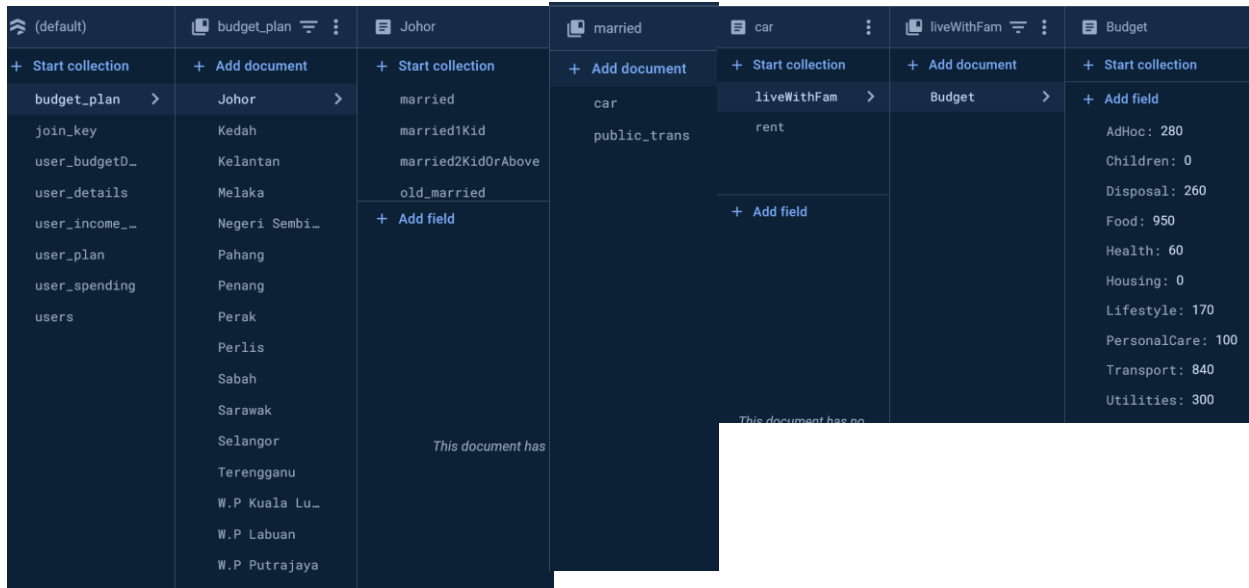


Figure 6.3.1 Budget Plan Table in Cloud Firestore

The users are required to fill in their details therefore they can be categorized into different groups. Different categories will get different budget types. The budget data is obtained from the report of KWSP of minimum spend budget guide, in collaboration with University Malaya and Social Wellbeing Research Centre (SWRC) [13]. The following is the summary of the sum of budget for the report.

	BUJANG PENGGUNA PENGANGKUTAN AWAM	BUJANG PEMILIK KERETA	PASANGAN BERKAHWIN TANPA ANAK	PASANGAN BERKAHWIN SEORANG ANAK	PASANGAN BERKAHWIN DUA ORANG ANAK	IBU/BAPA TUNGGAL SEORANG ANAK	IBU/BAPA TUNGGAL DUA ORANG ANAK	WARGA EMAS PASANGAN	WARGA EMAS INDIVIDU
Lembah Klang	RM1,930	RM2,600	RM4,630	RM5,980	RM6,890	RM4,740	RM5,650	RM3,210	RM2,520
Johor Bahru	RM1,760	RM2,290	RM4,110	RM5,360	RM6,100	RM4,200	RM4,940	RM3,020	RM2,330
Kota Kinabalu	RM1,710	RM2,230	RM4,000	RM5,130	RM5,840	RM4,000	RM4,710	RM2,930	RM2,290
Kuala Terengganu	RM1,630	RM2,160	RM3,820	RM4,920	RM5,610	RM3,820	RM4,510	RM2,730	RM2,090
Alor Setar	RM1,530	RM2,060	RM3,680	RM4,760	RM5,430	RM3,660	RM4,340	RM2,630	RM2,020
Kuching	RM1,680	RM2,210	RM3,920	RM5,020	RM5,720	RM3,890	RM4,590	RM2,790	RM2,160
Kuantan	RM1,710	RM2,230	RM3,910	RM5,030	RM5,740	RM3,900	RM4,610	RM2,780	RM2,130
Kota Bharu	RM1,540	RM2,110	RM3,750	RM4,830	RM5,520	RM3,720	RM4,400	RM2,690	RM2,050
Georgetown	RM1,830	RM2,430	RM4,360	RM5,640	RM6,370	RM4,460	RM5,190	RM3,140	RM2,450
Ipoh	RM1,680	RM2,270	RM3,970	RM5,140	RM5,850	RM4,020	RM4,730	RM2,840	RM2,190
Soremban	RM1,720	RM2,300	RM4,170	RM5,400	RM6,130	RM4,250	RM4,980	RM2,910	RM2,250
Bandar Melaka	RM1,730	RM2,270	RM4,010	RM5,140	RM5,850	RM4,020	RM4,720	RM2,830	RM2,180

Figure 6.3.2 Summary of Budget on Report Guide

To integrate salary-specific budgeting recommendations to the diverse income levels and financial needs of users in Malaysia.

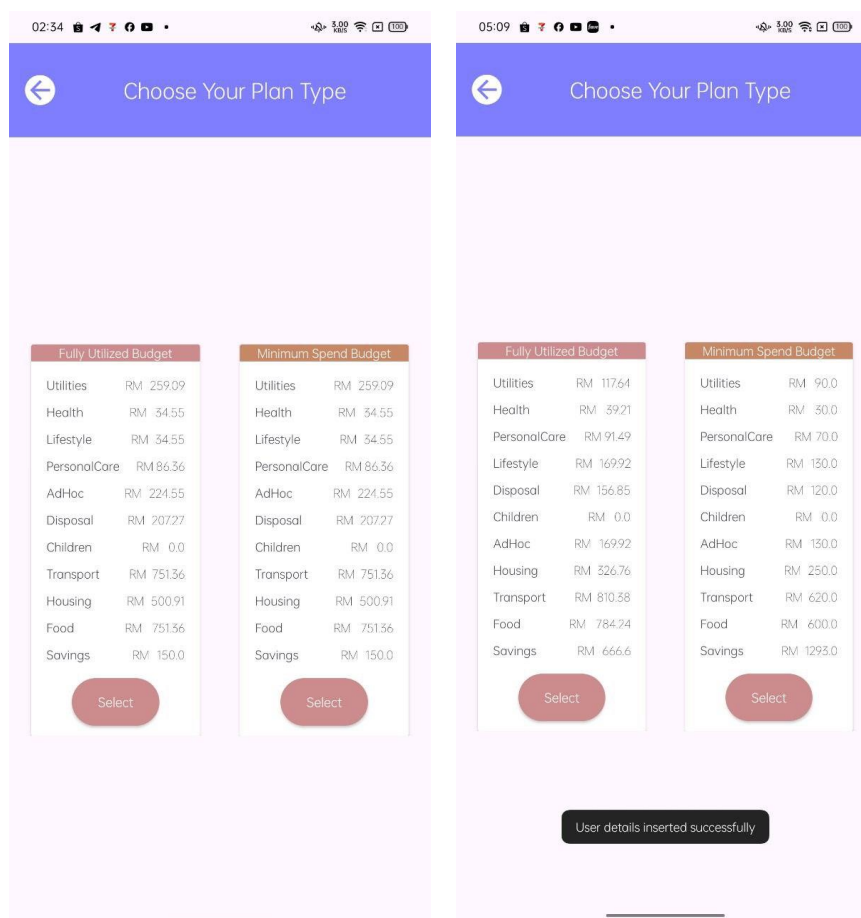
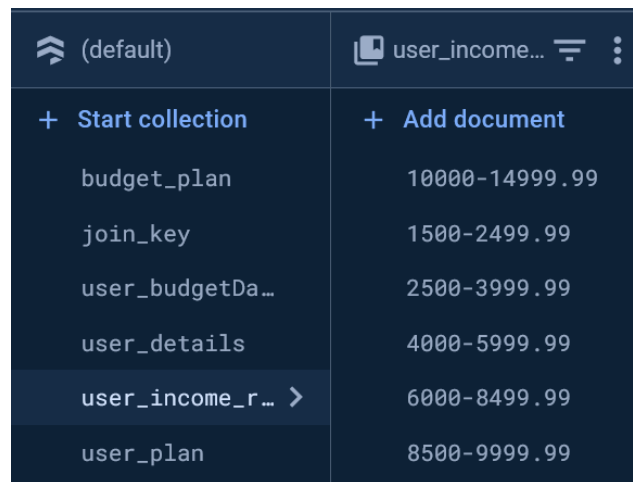


Figure 6.3.3 Autogenerated budget example for different income

Other than categories, the budget is also tailored for every user. The autogenerated budget is multiplied by the budget ratio and every user can get their budget by their income. For basic BudgetBuddy Budget, there are 3 situations for Fully Utilized Budget. The figure 6.3.3 shows 3 income of the users that has the same age, marital status, dependency status and living area. As an example, if the user at least 80% of the income is higher than the basic budget in the database, system will calculate 20% of the income as savings, and 80% of the income will be multiply by budget ratio to get the fully utilized budget.

If user income at least 95% of the income is higher than the basic budget, the fully utilized budget will follow the budget in the table and the difference will become savings.

If user income is less than the budget, 5% of the income will be classified as savings and 95% of the income will be multiplied with the budget ratio to get the budget. This can ensure that all users can get a suitable budget.



(default)	user_income...
+ Start collection	+ Add document
budget_plan	10000-14999.99
join_key	1500-2499.99
user_budgetDa...	2500-3999.99
user_details	4000-5999.99
user_income_r... >	6000-8499.99
user_plan	8500-9999.99

Figure 6.3.4 Income Range Table in Cloud Firestore

As for the upgraded version of BudgetBuddy autogenerated budget, the logic of the budget is that it will get all users from the same income range and get the mean for every category in the budget. If more and more user uses this app, it will be more suitable for users to know how much other users in their income range is managing their budget. This will be more useful for users that have higher income.

To implement collaborative financial planning and management for households or shared financial arrangements within budgeting apps.

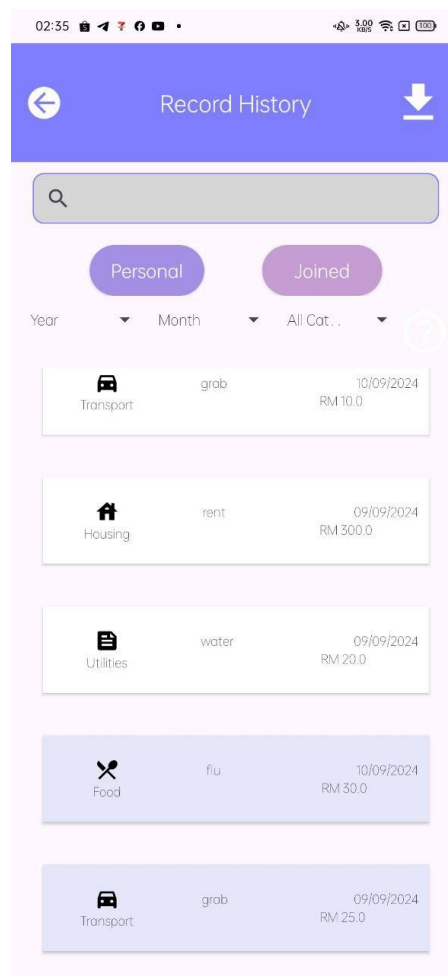


Figure 6.3.5 Spending Records for both users

The joint account function is implemented to let users to join their incomes and let both users can monitor their spendings and have a same budget together. The budget can let them track joined spendings such like rent. It will clearly show who is contributing and who is overspending. This can let couples to improve their spending habits and make the financial records clear.

6.4 Concluding Remark

In this chapter, the developed application is tested using various test case to ensure the app have the least vulnerabilities. The system is also evaluated using the objectives to made sure that all objectives are achived.

Chapter 7 Conclusion and Recommendation

7.1 Conclusion

The development of the proposed project using Java in Android studio is important to fulfil the needs for users that need finance management. The auto generated budget and joined account feature is the key feature of the mobile app and is crucial for users who wants to manage their spendings with their partner or just budgeting themselves. Various test case have been carried out to ensure the app is working fine, and a evaluation form is generated to evaluate the mobile app. The objectives defined in the project is fulfilled, thus it is believed that the BudgetBuddy Joined Finance Management App can help users to achieve success in finance planning and management.

7.2 Recommendation and Future Work

Although the current development of the proposed project had already achieved its objectives, but there are still some features that can still be improved in future development. The first recommendation is moving all calculations to the cloud. The app still has a small user base now all of the calculations are still on the client side. The calculation may use Cloud Function in Firebase to let the calculations to be run on server side, speeding up the process and save resource.

The next improvement is that the summary page can include more charts. The current Summary page only summarize the spendings of the month and does not include any other charts. Charts like line graph of balance for each month, highest transaction each month can be implemented to let users to be clearer of their spendings.

Image processing can also be implemented to detect records from receipt, making it easier for user to insert records. User will not need to key in records manually, and this can save time and energy for the user.

A feature for user to record constant records like rent and bills can be implemented too to save the hassle for user to key in same records every month. Notifications could also be implemented to remind users to remind users to record their spendings if they didn't open the app for a day.

REFERENCES

REFERENCES

- [1] “Fact.MR – Increasing Penetration of Smartphones to Propel the Demand for Personal Finance Mobile Apps during 2021-2031,” www.factmr.com. <https://www.factmr.com/report/personal-finance-mobile-app-market> [3]“Beware the debt trap, Young ones ,” *New Straits Times*, 15-Mar-2021. [Online]. Available: <https://www.nst.com.my/opinion/letters/2021/03/673935/beware-debtrap-young-ones>. [Accessed: 05-Dec-2022].
- [2] “Beware the debt trap, young ones | New Straits Times,” NST Online, Mar. 15, 2021. <https://www.nst.com.my/opinion/letters/2021/03/673935/beware-debtrap-young-ones>.
- [3] B. Holzhauer, “Goodbudget Budgeting App Review,” Forbes Advisor, May 06, 2021. <https://www.forbes.com/advisor/banking/goodbudget-budgeting-app-review/>
- [4] Feeman, “Bluecoins App Review – Finances on your fingertips: Money sonata,” *Money Sonata*. <https://www.moneysonata.com/bluecoins-app-review-personal-finance-android/>.
- [5] J. Chen, “Android Operating System: What You Need to Know,” Investopedia, Jun. 05, 2022. <https://www.investopedia.com/terms/a/android-operating-system.asp>
- [6] “Money Manager Expense & Budget,” App Store. <https://apps.apple.com/us/app/money-manager-expense-budget/id560481810>
- [7] “Belanjawanku,” Softonic. <https://belanjawanku.en.softonic.com/android> (accessed Apr. 19, 2024).
- [8] “Money Manager Expense & Budget,” App Store. <https://apps.apple.com/us/app/money-manager-expense-budget/id560481810>
- [9] Atlassian, “Agile Best Practices and Tutorials,” Atlassian, 2019. <https://www.atlassian.com/agile>
- [10] “SDK Platform Tools release notes,” Android Developers. <https://developer.android.com/tools/releases/platform-tools>
- [11] “The Java® Language Specification,” docs.oracle.com. <https://docs.oracle.com/javase/specs/jls/se11/html/index.html>
- [12] “Cloud Firestore | Store and sync app data at global scale,” *Firebase*. <https://firebase.google.com/products/firestore>

REFERENCES

[13] D. Norma, M. Nik, N. Ainoon, and N. Osman, "BELANJAWANKU EXPENDITURE GUIDE -A REFERENCE BUDGET FOR MALAYSIAN," 2022. Available: https://rmke12.ekonomi.gov.my/ksp/storage/event/124_24_puan_nik_noor_ainoon_belanjawanku_reference_budget_for_malaysian.pdf

APPENDIX SYSTEM EVALUATION SURVEY FORM

BudgetBuddy Finance Management App Review

This is a review form for BudgetBuddy Finance Management App. All results are only for FYP purposes.

netaliewan0809@1utar.my [Switch account](#)

Not shared

* Indicates required question

Email *

Your answer _____

1. How do you think about the overall design of the BudgetBuddy App? *

	1	2	3	4	5	
Very Bad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excellent

2. How reasonable is the auto generated budget to your opinion? *

	1	2	3	4	5	
Very Unreasonable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Reasonable

3. Please indicate how well the automated budget function can help to manage your financial needs? *

	1	2	3	4	5	
Not useful at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Useful

4. How do you think the about the navigation and hints in the app? *

	1	2	3	4	5	
Very Confusing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Easy to Understand

APPENDIX

5. Do you think the join account function is useful for you and your partner financial management? *

1 2 3 4 5

Not useful at all Very Useful

6. How well do you think the auto generated budget will improve your spending habit? *

1 2 3 4 5

No improvement at all Major improvement


7. Which function below is the most satisfied functions within the application? *

- Join account function
- Viewing summarize graphs of spending
- Get acknowledge if spending is overbudget
- View joined user spendings
- The ability to choose from different types of budget
- Budget generation suit to self condition

8. Would you like to recommend your friends or relatives about this BudgetBuddy app? *

1 2 3 4 5

Will Not Recommend At All Very Recommended

Submit  Page 1 of 1 [Clear form](#)

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	Study week no.:2
Student Name & ID:Wan Xin 20ACB02288	
Supervisor: Mr. Ku Chin Soon	
Project Title: Faculty of Information and Communication Technology	

1. WORK DONE

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

Studied FYP1 report and identify changes that need to do and created FYP2 report based on template

2. WORK TO BE DONE


Improve content for report

3. PROBLEMS ENCOUNTERED


N/A

4. SELF EVALUATION OF THE PROGRESS

Good. Project on schedule.



 Supervisor's signature



 Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	Study week no.:4
Student Name & ID:Wan Xin 20ACB02288	
Supervisor: Mr. Ku Chin Soon	
Project Title: Faculty of Information and Communication Technology	

1. WORK DONE

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

Improve content for report and enhance report

2. WORK TO BE DONE


Complete all diagrams and app development

3. PROBLEMS ENCOUNTERED


N/A

4. SELF EVALUATION OF THE PROGRESS

Good. Project on schedule.



 Supervisor's signature



 Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	Study week no.:6
Student Name & ID:Wan Xin 20ACB02288	
Supervisor: Mr. Ku Chin Soon	
Project Title: Faculty of Information and Communication Technology	

1. WORK DONE

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

Completed all diagrams on report and change database from SQLite to Cloud Firestore

2. WORK TO BE DONE

Continue development

3. PROBLEMS ENCOUNTERED

N/A

4. SELF EVALUATION OF THE PROGRESS

Good. Project on schedule.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	Study week no.:8
Student Name & ID:Wan Xin 20ACB02288	
Supervisor: Mr. Ku Chin Soon	
Project Title: Faculty of Information and Communication Technology	

1. WORK DONE

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

Develop Profile page and autogenerated budget function

2. WORK TO BE DONE


Continue app development

3. PROBLEMS ENCOUNTERED


N/A

4. SELF EVALUATION OF THE PROGRESS

Good. Project on schedule.



 Supervisor's signature



 Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	Study week no.:10
Student Name & ID:Wan Xin 20ACB02288	
Supervisor: Mr. Ku Chin Soon	
Project Title: Faculty of Information and Communication Technology	

1. WORK DONE

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

Develop Summary Page and fix edited previous code

2. WORK TO BE DONE

Continue app development


3. PROBLEMS ENCOUNTERED

N/A

4. SELF EVALUATION OF THE PROGRESS

Good. Project on schedule.

Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	Study week no.:12
Student Name & ID:Wan Xin 20ACB02288	
Supervisor: Mr. Ku Chin Soon	
Project Title: Faculty of Information and Communication Technology	

1. WORK DONE

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

Complete app development and start testing for bugs, Sent app to multiple users for review

2. WORK TO BE DONE

Complete report


3. PROBLEMS ENCOUNTERED

N/A

4. SELF EVALUATION OF THE PROGRESS

Good. Project on schedule.

Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4S1	Study week no.:12
Student Name & ID:Wan Xin 20ACB02288	
Supervisor: Mr. Ku Chin Soon	
Project Title: Faculty of Information and Communication Technology	

1. WORK DONE

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

Send evaluation form and get result, complete FYP2 report

2. WORK TO BE DONE


N/A

3. PROBLEMS ENCOUNTERED


N/A

4. SELF EVALUATION OF THE PROGRESS

Good. Project on schedule.



 Supervisor's signature



 Student's signature

POSTER

JOINED ACCOUNT FINANCE MANAGEMENT SYSTEM



INTRODUCTION

A good finance management system application can decrease the rate of youths having high credit card debt, let users success in financial planning, and let couples can manage family expenses accurately.



OBJECTIVE

1. To propose a system that can provide budget recommendations based on location, dependents, and marital status.
2. To integrate salary-specific budgeting recommendations to the diverse income levels and financial needs of users in Malaysia.
3. To implement collaborative financial planning and management for households or shared financial arrangements within budgeting apps.



PROPOSED METHOD

Develop a joined account finance management system app using Android Studio by java programming language, and using cloud firestore as database



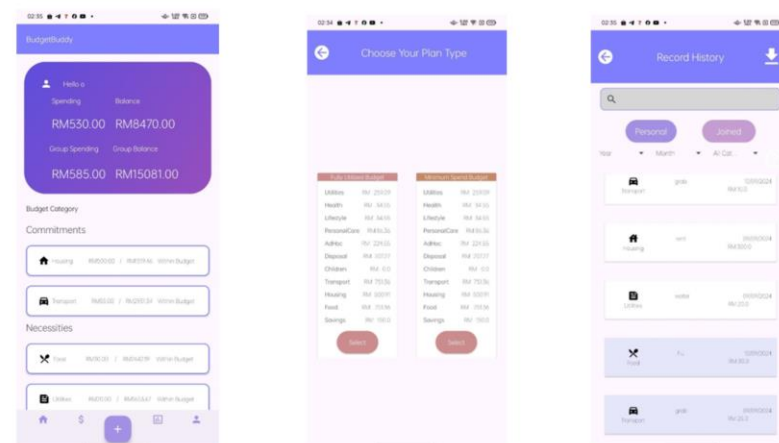
CONCLUSION

Successfully fulfilled the objective by:

- Auto generating budget by user category
- Joined account function that can let 2 users to have a same budget and record spending together



RESULT



PLAGIARISM CHECK RESULT

PLAGIARISM CHECK RESULT

Turnitin Originality Report	
Processed on: 13-Sep-2024 03:28 +08 ID: 2452270940 Word Count: 11148 Submitted: 1	
WanXin2002288 By WAN Xin	
Similarity Index 6%	Similarity by Source Internet Sources: 4% Publications: 0% Student Papers: 3%

1% match (Internet from 21-Jul-2024) https://audio.com/kenji/audio/mp3-chapter-4-conversation-starter-deep-blue-card
< 1% match (Internet from 25-Oct-2022) http://irigs.iuu.edu.pk:64447/gsd/collect/00electron/import/T08953E2012CSMCS.pdf
< 1% match (student papers from 23-Feb-2017) Submitted to UT, Dallas on 2017-02-23
< 1% match (student papers from 18-Apr-2023) Submitted to College of Banking and Financial Studies on 2023-04-18
< 1% match (Internet from 20-Nov-2022) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3720/P13906%20%2826%25%29.pdf?isAllowed=y&sequence=1
< 1% match (Internet from 20-Nov-2022) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/4814/161-15-849%2021_.pdf?isAllowed=y&sequence=1
< 1% match (Internet from 20-Nov-2022) http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5175/161-15-7441%20%2826_%29.pdf?isAllowed=y&sequence=1
< 1% match (Internet from 20-Apr-2024) https://www.coursehero.com/file/230638360/adsaaspdf/
< 1% match (Internet from 07-Nov-2022) https://www.coursehero.com/file/145775872/Test-Cases-Tested-Davadocx/
< 1% match (Internet from 07-Aug-2024) https://docslib.org/doc/11366817/voice-based-data-entry-through-android-phone-by
< 1% match (student papers from 03-Sep-2024) Submitted to Universidad TecMilenio on 2024-09-03
< 1% match (Internet from 11-Dec-2021) http://staff.utar.edu.my/gohhg/stu_work/ug.htm
< 1% match () Marashdeh, Qussai. "Development and Implementation of 3-D, High Speed Capacitance Tomography for Imaging Large-Scale, Cold-Flow Circulating Fluidized Bed", "Office of Scientific and Technical Information (OSTI)", 2012
< 1% match (student papers from 22-Apr-2024) Submitted to Joseph Leckie Academy on 2024-04-22
< 1% match () Alrashoud, Eiman. "CHILDREN'S SOCIAL NETWORK: KIDS CLUB", CSUSB ScholarWorks, 2017
< 1% match (Internet from 29-Jan-2023) https://www.slideshare.net/ssuserf37169/design-and-implementation-of-an-obstacle-avoiding-robot
< 1% match (student papers from 26-Mar-2010) Submitted to University of Greenwich on 2010-03-26
< 1% match (student papers from 22-Mar-2013) Submitted to Manchester Metropolitan University on 2013-03-22
< 1% match (student papers from 25-May-2020) Submitted to University of Queensland on 2020-05-25
< 1% match (Internet from 11-Oct-2020) https://www.mdpi.com/1424-8220/16/6/931/htm
< 1% match (Internet from 23-Dec-2022) https://5dok.org/document/dzox84z-development-motivationaltool-employees-increasetheir-physical-activity-improving-occupational.html

PLAGIARISM CHECK RESULT

<p>< 1% match () Hall, EJ. "Hot dogs – advancing the epidemiology and clinical definition of heat-related illness in UK dogs", 2024</p>
<p>< 1% match (Internet from 30-Jan-2021) https://ringgitplus.com/en/blog/personal-finance-news/epf-releases-belanjawanku-an-expenditure-guide-for-malaysians.html</p>
<p>< 1% match (Internet from 30-Aug-2024) https://www.androidpure.com/samsung-galaxy-a52-5g-and-galaxy-a72/</p>
<p>< 1% match (Internet from 17-Aug-2024) https://api.mountainscholar.org/server/api/core/bitstreams/5bb11448-4d9d-436c-b72f-ca8c6993ba19/content</p>
<p>< 1% match (Internet from 10-Nov-2021) https://fict.utar.edu.my/documents/FYP/FYP2_template/FYP2_Report_Template_IA.docx</p>
<p>< 1% match (Zeinab Shahbazi, Debapriya Hazra, Sejoon Park, Yung Cheol Byun. "Toward Improving the Prediction Accuracy of Product Recommendation System Using Extreme Gradient Boosting and Encoding Approaches", Symmetry, 2020) Zeinab Shahbazi, Debapriya Hazra, Sejoon Park, Yung Cheol Byun. "Toward Improving the Prediction Accuracy of Product Recommendation System Using Extreme Gradient Boosting and Encoding Approaches", Symmetry, 2020</p>
<p>< 1% match (Internet from 20-Mar-2023) https://vdocuments.mx/autoplug.html</p>

PLAGIARISM CHECK RESULT

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



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Wan Xin
ID Number(s)	20ACB02288
Programme / Course	IA
Title of Final Year Project	Joined Account Finance Management System

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u> 6 </u> % Similarity by source Internet Sources: <u> 4 </u> % Publications: <u> 0 </u> % Student Papers: <u> 3 </u> %	
Number of individual sources listed of more than 3% similarity: <u> 0 </u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

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

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

Signature of Supervisor

Name: Ku Chin Soon

Date: 13/09/2024

Signature of Co-Supervisor

Name: _____

Date: _____

CHECKLIST



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB02288
Student Name	Wan Xin
Supervisor Name	Mr Ku Chin Soon

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 12 September 2024

CHECKLIST