KeepIT:
MONEY TRACKING AND MANAGEMENT APPLICATION

By

Tiu Kai Xiang

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements for

the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology (Kampar Campus)

FEBUARY 2026

COPYRIGHT STATEMENT

© 2026 Tiu Kai Xiang. All rights reserved.

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

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Ms. Ts. Lai Siew Cheng, for granting me this invaluable opportunity to engage in a mobile application development project. Her continuous guidance, encouragement, and constructive feedback have been instrumental in shaping my learning experience. This project not only marks my first step toward establishing a career in the software development field but has also enhanced my skills, confidence, and passion for technology. I am truly thankful for her patience, support, and unwavering commitment to my growth.

To a very precious person in my life, Kwong Guan Yuan, for his patience, unwavering support, and for sticking by my side through difficult times. Finally, I want to thank my parents and family for their love, support, and encouragement throughout the course.

ABSTRACT

In the digital age, personal finance management faces numerous challenges: difficulty tracking every expenditure, unclear budget planning, and complex multi-currency operations. This project proposes a mobile application designed to comprehensively address these issues. The application features intuitive expense logging and real-time budget tracking, alongside innovative capabilities: an arithmetic averaging algorithm for equitable splitting of group expenses, real-time multi-currency conversion with live exchange rates, automatic receipt scanning via OCR technology, and an AI-powered personalized spending analysis module. Developed using an agile framework built upon Visual Studio Code, Supabase, and Flutter, it incorporates dashboards and visualization features to provide users with a comprehensive financial overview. By integrating these advanced functionalities, the application aims to enhance personal financial engagement and efficiency, assisting users in adhering to budgets and avoiding overspending pressures. It advances fintech development by addressing gaps in existing tools and delivering improved solutions. For personal finance users, the application streamlines record-keeping, budgeting, and equitable expense-sharing processes, effectively alleviating financial anxiety and fostering superior financial outcomes.

Area of Study (Maximum 2): Mobile Application Development, Financial Technology

Keywords (Maximum 5): Expenses Management, Budget Tracking, Mobile Application, OCR Technology, Personal Finance

TABLE OF CONTENTS

COPYRIGHT STATEMENT i

ACKNOWLEDGEMENTS ii

ABSTRACT iii

TABLE OF CONTENTS Iv

LIST OF FIGURES Vi

LIST OF TABLES ix

LIST OF ABBREVIATIONS x

CHAPTER 1 INTRODUCTION 1

1.1 Problem Statement and Motivation 1

1.1.1 Difficult tracking every expense 1

1.1.2 Challenge in budget adherence 2

1.1.3 Lack of multi-currency support 2

1.2 Project Objectives 4

1.2.1 Implement and enhance the OCR module 4

1.2.2 Develop a group cost-sharing module 4

1.2.3 Implement real-time multi-currency support 4

1.2.4 Develop an AI insights module 4

1.3 Project Scope and Direction 4

1.4 Impact, Significance and Contributions 5

1.5 Report Organization 6

CHAPTER 2 LITERATURE REVIEW 7

2.1 Previous works on Finance Application 7

2.1.1.1 Qianji 7

- Weakness of Qianji 8
- Recommendations for Qianji 8

2.1.1.2 You Need A Budget (YNAB) 9

- Weakness of YNAB 11

- Recommendations of YNAB 12

2.1.1.3 Expensify 12

- Weakness of Expensify 14
- Recommendations for Expensify 15

2.2 Technologies Research 16

2.2.1 Optical Character Recognition (OCR) 16

2.2.2 AI Engine Research 17

2.2.3 Workflow Automation and AI Agent Technologies 20

2.3 Overview Table of Qianji, YNAB, Expensify and Proposed System 22

CHAPTER 3 PROPOSED METHOD / APPROACH 24

3.1 Project Methodology 24

3.1.1 Phase 1 – Planning Phase 24

3.1.2 Phase 2 – Design Phase 24

3.1.3 Phase 3 – Development and Testing Phase 25

3.1.4 Phase 4 – Deployment, Review and Maintenance Phase 25

3.2 System Design Diagram / Equation 26

3.2.1 Use Case Diagram 26

3.2.2 System Flowchart 27

3.2.3 System Entity Relationship Diagram (ERD) 31

3.2.4 System Architecture Diagram 32

3.2.5 Activity Diagram 34

3.2.5.1 Sign In Activity Diagram 34

3.2.5.2 Registration Activity Diagram 35

3.2.5.3 OCR Process Activity Diagram 36

3.2.5.4 Split Expenses Activity Diagram 37

3.2.5.5 Currency Converter Activity Diagram 38

3.2.5.6 AI Insight Activity Diagram 38

3.3 User Requirements 40

3.3.1 System Performance Requirement 41

3.3.2 OCR Performance Evaluation Criteria 42

3.4 Project Timeline 44

CHAPTER 4 SYSTEM DESIGN 47

4.1 System Block Diagram 47

4.2 System Components Specifications 49

4.2.1 Frontend (Mobile Application) 49

4.2.2 Backend Services (Supabase) 49

4.2.3 Database (PostgreSQL) 50

4.2.4 External Services 50

4.2.5 Automation Workflows (n8n) 50

4.3 Circuits and Components Design 51

4.3.1 User Authentication Module 51

4.3.2 Transaction Management Module 51

4.3.3 OCR Processing Module 52

4.3.4 Group Management and Expense Splitting Module 52

4.3.5 Currency Converter Module 53

4.3.6 AI Insight, Chatbot, and Automation Module 53

4.4 System Components Interaction Operations 54

4.4.1 User Authentication 54

4.4.2 Transaction Recording 54

4.4.3 OCR Receipt Scanning 55

4.4.4 Group Expense Management 55

4.4.5 Currency Converter 55

4.4.6 AI Insight and Chatbot Interaction 56

4.4.7 Offline Mode Interaction 56

CHAPTER 5 SYSTEM IMPLEMENTATION 58

5.1 Hardware Setup 58

5.2 Software Setup 60

5.3 Setting and Configuration 66

5.4 System Operation 72

5.5 Implementation Issues and Challenges 99

5.6 Concluding Remark 99

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION 101

6.1 System Testing Overview 101

6.2 Verification Plan 101

6.3 Testing Result and Analysis 105

6.3.1 Performance Evaluation Results 105

6.3.2 Functional Testing Results 105

6.3.3 OCR Testing Results 106

6.3.4 AI Chatbot Testing Results 112

6.3.5 User Acceptance Testing (UAT) Feedback Analysis 113

6.3.6 Discussion 115

6.4 Project Challenges 117

6.5 Objectives Evaluation 118

6.6 Concluding Remark 119

CHAPTER 7 CONCLUSION AND RECOMMENDATION 120

7.1 Conclusion 120

7.2 Recommendation 121

REFERENCES 122

APPENDIX

A.1 Poster A-1

B.1 Online_Receipt B-1

B.2 Realife_Receipt B-6

C. User Acceptance Testing (UAT) Form Feedback C-1

C.1 Google Form Questionnaire Design C-1

C.2 Google Form Response Summary C-5

C.3 User Declaration and Digital Signature C-9

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.1.1.1.1	Qianji Rate on Google Play Store and App Store	7
Figure 2.1.1.1.2	Qianji application page	8
Figure 2.1.1.2.1	You Need a Budget Mobile Application	10
Figure 2.1.1.2.2	YNAB's guidance for users	11
Figure 2.1.1.2.3	YNAB doesn't have a full guide	12
Figure 2.1.1.3.1	Expensify Website and Application	13
Figure 2.1.1.3.2	Components of OCR System	13
Figure 2.1.1.3.3	Scanning Model in mobile application	14
Figure 2.1.1.3.4	Expensify Split Bill Sample	15
Figure 3.1	Agile Software Development	24
Figure 3.2.1	Use Case Diagram of System	26
Figure 3.2.2.1	System Flowchart Page 1	27
Figure 3.2.2.2	System Flowchart Page 2	28
Figure 3.2.2.3	System Flowchart Page 3	29
Figure 3.2.2.4	System Flowchart Page 4 & 5	30
Figure 3.2.3	Entity Relationship Diagram (ERD)	31
Figure 3.2.4	3-Tier System Architecture	32
Figure 3.2.5.1	Sign-in Activity Diagram	34
Figure 3.2.5.2	Registration Activity Diagram	35
Figure 3.2.5.3	OCR Process Activity Diagram	36
Figure 3.2.5.4	Split Expenses Activity Diagram	37
Figure 3.2.5.5	Currency Converter Activity Diagram	38
Figure 3.2.5.6	AI Insight Activity Diagram	39
Figure 3.4.1	Project Timeline (FYP 1)	44
Figure 3.4.2	Project Timeline (FYP 2)	45
Figure 4.1	System Block Diagram	47
Figure 5.2.1	Flutter Logo	60
Figure 5.2.2	Dart Programming Logo	60

Figure 5.2.3	Visual Studio Code Logo	61
Figure 5.2.4	Android Studio Logo	61
Figure 5.2.5	ML Kit Logo	61
Figure 5.2.6	Supabase Logo	62
Figure 5.2.7	Figma Logo	63
Figure 5.2.8	Google Stitch Logo	63
Figure 5.2.9	ChatGPT Logo	64
Figure 5.2.10	n8n Logo	64
Figure 5.2.11	GitHub Logo	65
Figure 5.3.1	Project Structure and Dependency Configuration	66
Figure 5.3.2	Supabase Database Tables Configuration	67
Figure 5.3.3	Supabase Authentication and User Management	67
Figure 5.3.4	n8n Workflow for AI Chatbot Integration	68
Figure 5.3.5	Supabase Data Retrieval Configuration in n8n	69
Figure 5.3.6	AI Chatbot Prompt Configuration	69
Figure 5.3.7	OpenAI Model Configuration in n8n	71
Figure 5.4.1	Splash Screen Overview	72
Figure 5.4.2.1	Overview Screen	73
Figure 5.4.2.2	Quick Action Bar	74
Figure 5.4.3.1	Sign In Required Button	75
Figure 5.4.3.2	Login Page	76
Figure 5.4.3.4	Error Message	76
Figure 5.4.3.5	Register Form	77
Figure 5.4.3.6	Error Message 2	77
Figure 5.4.3.7	Account Successfully Created and Stored in Supabase Database	78
Figure 5.4.3.8	Data Stored in Supabase Database	78
Figure 5.4.4.1	Income Categories & Expense Categories	79
Figure 5.4.4.2	OCR Button	79
Figure 5.4.4.3	OCR Page	80
Figure 5.4.4.4	Access to Device Storage	80

Figure 5.4.4.5	OCR Scanning Result	81
Figure 5.4.4.7	Transaction Page	81
Figure 5.4.4.8	Transaction Page Overview	82
Figure 5.4.4.9	Delete Confirmation	82
Figure 5.4.5.1	Overview GM Page	83
Figure 5.4.5.2	Add Group Button	83
Figure 5.4.5.3	Create Group Form	84
Figure 5.4.5.3	Group Overview	84
Figure 5.4.5.5	Add Expense Form	85
Figure 5.4.5.6	Payer Mark Paid	86
Figure 5.4.5.7	Buyer Confirm Pending	86
Figure 5.4.5.8	Notification of Payment Pending Confirmation	87
Figure 5.4.6.1	Currency Converter	88
Figure 5.4.6.2	Other Currency Rates Table List	89
Figure 5.4.7.1	AI Chatbot Advisor	90
Figure 5.4.7.2	Chatbot Response	91
Figure 5.4.7.3	Chatbot Refuse Answer	91
Figure 5.4.8.1	Setting Page	93
Figure 5.4.8.2	Currency Option	94
Figure 5.4.8.3	Language Option	94
Figure 5.4.8.4	Example Language Change	95
Figure 5.4.8.5	Backup Data	96
Figure 5.4.8.6	Export Data	96
Figure 5.4.8.7	Delete All Data with Double Confirmation	97
Figure 5.4.8.8	Help & Support with Feedback Form	98
Figure C.1.1	Google Form Questionnaire Design 1	C-1
Figure C.1.2	Google Form Questionnaire Design 2	C-2
Figure C.1.3	Google Form Questionnaire Design 3	C-3
Figure C.1.4	Google Form Questionnaire Design 4	C-4
Figure C.2.1	Google Form Response Summary 1	C-5
Figure C.2.2	Google Form Response Summary 2	C-6

Figure C.2.3	Google Form Response Summary 3	C-7
Figure C.2.4	Google Form Response Summary 4	C-8
Figure C.3	User Declaration and Digital Signature	C-9

LIST OF TABLES

Table Number	Title	Page
Table 2.3	Overview Table of Qianji, YNAB, Expensify, and Proposed System	22
Table 3.3.1	System Performance	41
Table 3.3.2.1	System OCR Performance	42
Table 3.3.2.2	System OCR Evaluated	43
Table 5.1.1	Specifications of Laptop	58
Table 5.1.2	Specifications of Mobile Device	58
Table 5.3	AI Chatbot Prompt Setup	70
Table 6.2.1	User Authentication Verification Plan	102
Table 6.2.2	Add Transaction Verification Plan	102
Table 6.2.3	Group Management Verification Plan	103
Table 6.2.4	Currency Converter Verification Plan	104
Table 6.2.5	Overview Screen Verification Plan	104
Table 6.3.1	Performance Metrics Achieved Results	105
Table 6.3.2	Functional Testing Test Case	106
Table 6.3.3.1	Testing Result Online Receipt	107
Table 6.3.3.2	Testing Result Real-Life Receipt	109
Table 6.3.3.3	Adjusted OCR Accuracy Result	111
Table 6.3.3.4	Offline Mode Testing Results	111
Table 6.3.5	UAT Feedback Result	114

LIST OF ABBREVIATIONS

YNAB	You Need a Budget
UI/UX	User Interface and User Experience
SSD	Solid-State Drive
OS	Operating System
OCR	Optical Character Recognition
AI	Artificial Intelligence
SROIE	Scanned Receipts OCR and Information Extraction
FUNSD	Form Understanding in. Noisy Scanned Documents
CNN	Convolutional Neural Network
RNNs	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Units
PCA	Principal Component Analysis
t-SNE	t-distributed Stochastic Neighbor Embedding
ARIMA	Autoregressive Integrated Moving Average
SHAP	Shapley Additive exPlanations
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
GM	Group Management

CHAPTER 1 Introduction

As the global economic environment becomes progressively more complex, personal financial management is becoming an important tool for individuals, students, families, and couples to achieve their financial goals. However, the challenges of tracking daily spending, sticking to a budget, and collaboratively managing finances continue to plague many users. Although most financial management applications are popular in the market, such as Expensify, YNAB (YOU NEED A BUDGET) and Money Tracker, they still lack in collaboration features, multi-currency support and user-friendliness, which makes it difficult for them to fully satisfy the diversified needs of customers.

The goal of this project is to provide a comprehensive solution by creating a mobile platform that makes it simple for users to understand their current personal finances and helps them to budget and plan for their future. The project focused on mobile application development to improve accessibility and the UI/UX design to optimize the user experience. Unique features include an arithmetic averaging system for fair sharing of expenses among groups of users, automated receipt scanning with OCR Technology, multi-currency exchange rate tracking and AI-powered consumer insights. This project not only responds to the real-world financial management needs of users but also provides an innovative solution for the fintech sector by combining modern technology with user-centred design.

1.1.Problem Statement and Motivation

Problem Statement

The first step in debt management is to become more conscious of where your money is going and to spend it with purpose. However, in an increasingly cashless culture, people are more prone to lose track of their spending because there is little to no moderation.[1] And this leads to the following factors that people are often prone to in managing their finances:

1.1.1. Difficult tracking every expense

Many people find it difficult and error-prone to record manually every transaction they make. For example, the user may easily forget the amount of a purchase or forget to take a receipt and, as a result, he will go home without a clear idea of how much he has spent, which will cause him to overspend at the end of the month. Moreover, some websites and applications cannot scan receipts, which makes it impossible for users to upload a receipt in order to get an idea of the internal costs of the companies in order to track their expenses. Manual financial recording is a very inefficient and incomplete way of recording finances, which may make it difficult for the user to get a full picture of his spending habits and finances.

1.1.2. Challenge in budget adherence

Sharing a budget and splitting costs is a big challenge for the couples, families, or groups of friends. For example, in a group expense, friends often need to manually calculate the amount to be paid by each person after a dinner party, which may take a lot of effort and time on the part of the record keeper to ensure that there are no errors in the amount, and it is fair to each person. In addition, if the group members are not gathered in the same places, each group member other than the recorder will only have access to view the amounts but cannot manage to the budget, which will result in a non-transparent and unfair apportionment of the costs, which can easily lead to misunderstandings and disputes problem.

1.1.3. Lack of multi-currency support

With the times, currency support has also been globalized, which will leave international users (such as international students or cross-border travellers) needing to manage their amounts between different currencies. However, there are still many apps that do not have multi-currency tracking or only support static exchange rate conversions and do not keep users updated with real-time exchange rates. This can result in budget overspending and is inconvenient for consumers form other countries. For example, if a user is travelling to China to study, but the exchange rate of the Malaysian Ringgit (RM) to the Chinese Renminbi (CNY) fluctuates daily, the user may record the wrong amount of money to be spent without having a clear

understanding of the currency exchange rate fluctuations, resulting in budgetary errors and budgetary overruns.

Motivation

The motivation for the development of this money tracking and management application was inspired by the financial management difficulties that individuals, couples, and students often encounter. Based on the observation that today's young generation, especially students, often have difficulties in managing their financial budgets, as well as inequities in the distribution of debts with groups of friends, the development of an arithmetic averaging system was inspired to use this functionality to help users to get a solution for financial transparency and fairness in the distribution of their debts.

Secondly, users were unable to accurately budget and manage their finances because they were prone to recording errors or forgetting to record, resulting in users often exceeding their budgets at the end of the month. The need to solve this problem led to the development of a feature that automatically scans and records receipt data to provide users with a more accurate solution for tracking debt.

The lack of multi-currency support inspired the development of a multi-currency system and the addition of real-time currency conversion to the system to help users understand international currency exchange rates to help them find better budgeting solutions to manage their finances based on different exchange rates.

In addition to providing further financial management for a diverse group of users, helping them improve their financial literacy and reduce their financial stress, developing new systems and functionalities also provided valuable learning opportunities and increased commitment to the project.

1.2.Project Objectives

The main objective of this project is to develop an integrated money tracking and management application to help individuals, families, couples, and groups of users can record their expenses and plan their budgets more easily and efficiently, helping them to solve/avoid financial management problems. Below is the objectives list:

1.2.1 Implement and enhance the OCR module for auto receipt scanning

In addition to implementing the OCR-based module for automatic receipt scanning, a sharpening function can be added that can help to make the photos clearer so that the data in the receipts can be read more easily and the information can be presented. The facilities offered under this module feature can reduce the problem of hard-to-track spending for users, while also allowing them to stick to a budget and savings plan.

1.2.2 Develop a group cost-sharing module

Use the arithmetic mean algorithm to ensure that team members are fairly assured of cost sharing and include features allowing user interaction during development. This module can improve expense tracking during group cost-sharing.

1.2.3 Implement real-time multi-currency support and remittance conversion

Enables users to accurately track different currency conversions and expenses by obtaining up-to-date exchange rate information. This may help to solve lack of multi-currency problem.

1.2.4 Develop an AI insights module

To provide personalised financial advice to users by analysing their daily expenses. This will help users adopt a long-term budget and savings plan by providing personalised financial management advice.

1.3.Project Scope and Direction

The scope of this project is to plan and implement a mobile application for a personal finance management system. The application will help users to be responsible for tracking their expenses as well as financial management. The mobile application system

will cover user registration/login, expense recording (e.g., manual recording and receipt scanning), budget setting/goal tracking and group billing. It will also provide dashboards to analyse the results of their financial management through charts and data presentation, helping them to understand their current financial situation and choose the right spending pattern; and include a basic AI-powered insights module, which will be used to help the user to provide personalised financial advice that suits their needs.

Key components within the main scope of this project include an arithmetic averaging algorithm to help users share costs across groups, OCR receipt processing, multi- currency tracking and real-time exchange rate conversion, and a core budgeting tool. The focus of the project was to build a prototype of the programmed that demonstrated the main needs of the users and the features that would improve the user experience.

The innovation of this project is to combine advanced features that are rarely available simultaneously in several existing applications. For example, the introduction of a fair arithmetic averaging system helps users ensure that group costs are transparent and absolutely fair when apportioned. The automated receipt scanning feature helps users to reduce their time by increasing their convenience compared to manual recording. Comprehensive real-time multi-currency support provides international users with a wide range of applications. These innovative features not only fill in the gaps of existing tools but are also expected to inspire new solutions for the future of fintech, contributing to the advancement of the industry.

1.4.Impact, Significance and Contributions

The Money Tracking and Management application will contribute to both the user and the fintech industry. For individuals, couples, students, and groups of friends, the app can help simplify debt tracking through an arithmetic averaging system and can help users increase their budgeting habits and help to get a fair share of the cost of debt within a group. Additionally, the receipt scanning and multi-currency support features contribute significantly to saving time and catering to the needs of users across international boundaries.

In the FinTech industry, this application can fill a gap in the application market by providing unique features (arithmetic averaging system, scanning of receipts, etc.) to help meet the needs of users that cannot be met by existing tools. At the same time, the application developed in this project can also be used to drive innovation in the industry by incorporating new technologies or methodologies (real-time exchange rate conversion), which can enhance the user experience and provide technical references to provide experience or inspiration to the industry, contributing to the FinTech industry. The above is the contents of this chapter; the literature survey will continue in the next chapter.

1.5.Report Organization

This report is organized into seven chapters: Chapter 1, Introduction; Chapter 2, Literature Review; Chapter 3, Methodology; Chapter 4, System Design; Chapter 5, System Implementation; Chapter 6, System Evaluation and Discussion; and Chapter 7, Conclusion and Recommendation. Chapter 1 introduces the project by presenting the background, problem statement, motivation, project objectives, scope, significance, and the organization of the report. Chapter 2 reviews relevant literature, including existing financial management applications and supporting technologies such as Optical Character Recognition (OCR), Artificial Intelligence (AI), and workflow automation, as well as identifies research gaps addressed by this project. Chapter 3 describes the methodology adopted, including the development approach, project phases, tools and technologies used, and the project timeline. Chapter 4 presents the system design, including system architecture, use case diagrams, system flowcharts, database design, and module design. Chapter 5 discusses the system implementation, including the development environment, system components, and integration of frontend, backend, database, and external services, along with the system operation workflow. Chapter 6 evaluates the system through various testing approaches such as functional testing, performance testing, OCR accuracy testing, and user acceptance testing, followed by analysis and discussion of the results. Finally, Chapter 7 concludes the report by summarizing the project outcomes and providing recommendations for future improvements.

CHAPTER 2
Literature Reviews

2.1 Previous works on Finance Application

There are a few similar personal finance management web and mobile applications available on websites and in the Google Play Store. I have reviewed 3 similar websites and mobile applications, as shown below:

2.1.1.1 Qianji

Qianji, a simple and pure bookkeeping application, is a “three noes” product that follows the “no ads, no open screen, no finance” rule, aims for simplistic design, and focuses on personal bookkeeping [2]. This is a mobile application that is available to download in the App Store and Google Play Store with an average rating of 4.5/5, and it number of installations has more than 30,000,000 installations.

Figure 2.1.1.1.1 Qianji Rate on Google Play Store and App Store

Qianji has a simple user interface and a purely efficient operation to help users get started. Bookkeeping, billing statistics analysis, budget management, asset management, etc. are the basic functions in its core functions, and in the advanced functions, it can also provide customers with multiple books/shared bookkeeping, free backup, import and export data, multi-currency support, etc., which provide users with a lot of help in financial management. Users can keep track of their daily financial expenses with various categorisations so that they know exactly what they are spending their money on. The dashboard of Qianji

also shows the information of the user's spending, which helps the user to understand their current financial situation and make reasonable arrangements for their budget to meet their needs.

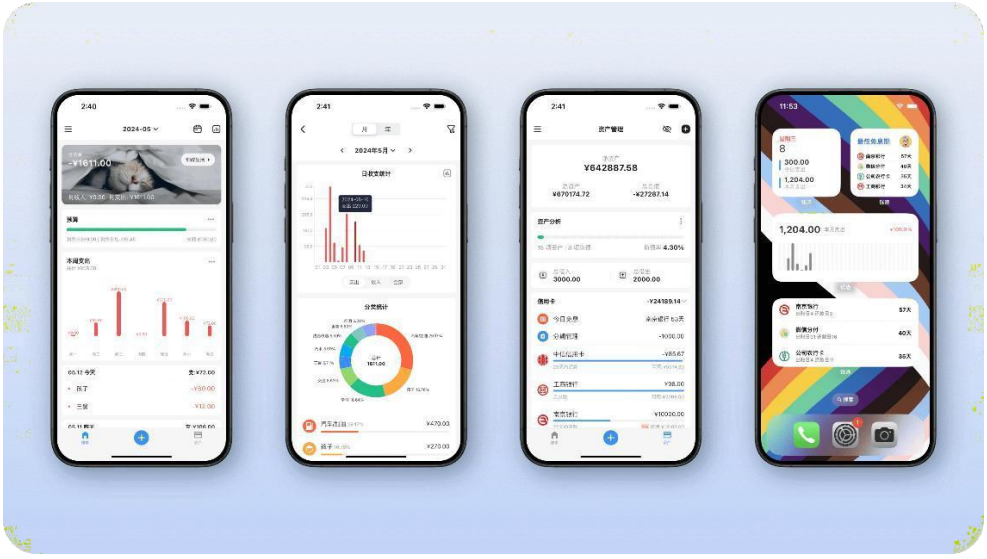


Figure 2.1.1.1.2 Qianji application page

- **Weakness of Qianji**

  Although the performance of Qianji is quite good in daily bookkeeping and basic budgeting, its reporting function is quite basic, which can only provide users with basic pie charts, bar charts and line graphs, but lacks the ability of in- depth analyses. According to the roadmap of the second version of Qianji, its debt reminder function also needs to be improved, as it lacks an automatic reminder function [3]. In addition, Qianji lacks a systematic approach to financial management, as it does not have a budgeting system like the YNAB envelope budgeting system or Expensify's receipt scanning model. The investment account tracking and asset trend functions are also very simplified, only providing the month-end net worth curve [4], and do not have a forecast or investment function, which is unable to provide users with accurate analysis and planning. For example, the ability to track debt repayment progress or overspending alerts a proactive financial alert and insight that Qianji lacks. Due to these limitations, it can be challenging for users to keep track of their finances and stick to their budgets and savings plans.

- **Recommendations for Qianji**

Suggestions and improvements for Qianji are to expand the financial planning function to allow users to expand the savings plan into a customisable goal module, and to include predictive analytics, as well as provide financial tracking goals and bill reminders similar to YNAB. For example, after analysing their finances with the predictive analytics feature, it can provide them with future financial planning to help them manage their finances as a further reference and control their expenses. Financial tracking can be set up to send alerts to remind users to make payments within a specified period.

2.1.1.2 You Need A Budget (YNAB)

YNAB (You Need a Budget) is a personal budgeting application based in the United States, It uses the traditional envelope-style for the budgeting method. Jesse Mecham created the program in 2004 to help him with his personal finances, but it has now grown into a cross-platform application that is available on the web and mobile devices [5]. According to the ratings, **90%** of customers believe they have improved their financial status since using YNAB, and YNAB also scored a very high rating of 4.6 in the review provided by Trustpilot, which is used to investigate company ratings [6]

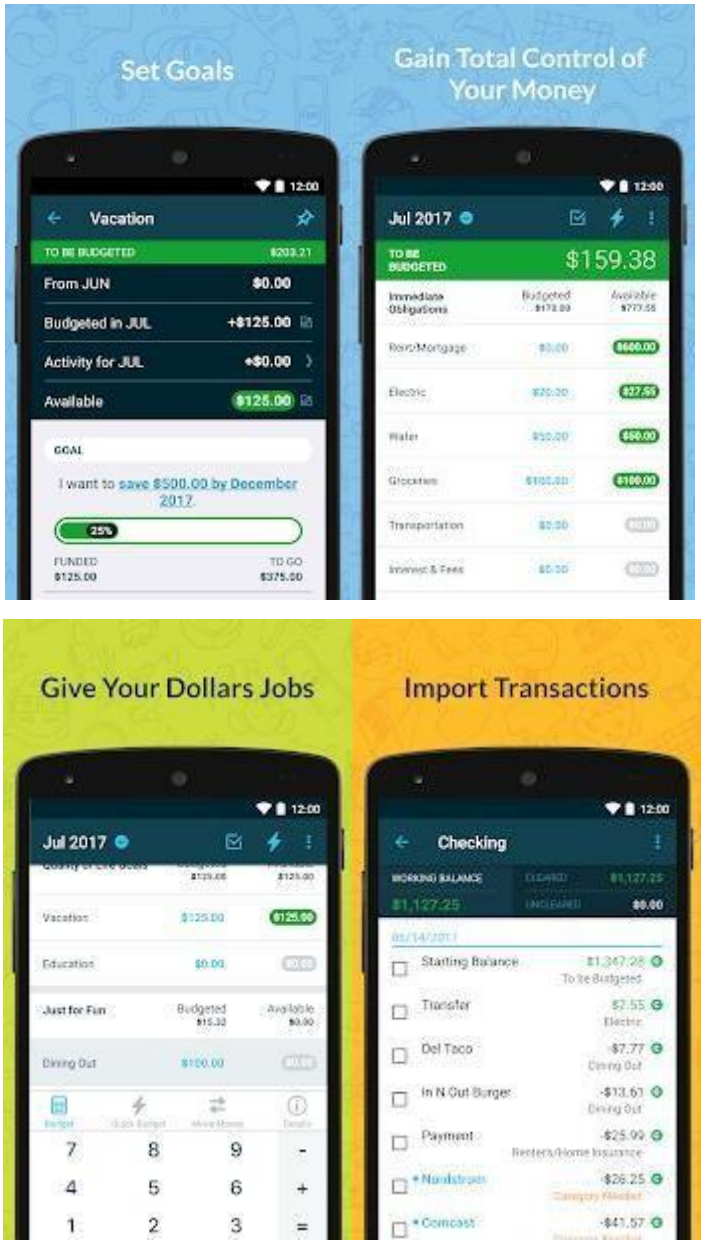


Figure2.1.1.2.1 You Need a Budget Mobile Application

YNAB's basic functionalities include assigning income to different expense categories, tracking the balance of each category in real time, automatically integrating bank transactions, manually inputting expenditures, and producing a range of financial figures for users to review [7]. It allows users to set their budget goals, and the system will remind them to allocate finances to help them manage their costs ahead of time, also this may help to minimise their unnecessary spending. Overall, YNAB is popular for its simple interface and secure budgeting system, which aims to keep users informed about their financial situation so that they may better manage their income and spending.

- **Weakness of YNAB**

YNAB's high membership fee is one of its limitations, but there are other limitations that make it unable to provide a good experience for its users. Although YNAB's budget management, asset-liability tracking, and goal setting can dynamically calculate the daily available budget for users and provide them with alerts on overspending. However, the complexity of YNAB's operation is very unfriendly to new users, and it takes a lot of time and effort to get them up to speed with YNAB as they don't understand how to use it and how to operate it. According to my actual test, in the second photo (Figure 2.1.1.2.2) you can learn that I no longer understand what to do next, YNAB to deal with personal debt. In addition, YNAB doesn't have a receipt scanning model like Expensify does, so it doesn't provide users with a quicker way to keep track of their money. Regarding the membership subscription fee, the high cost will make the users who are not financially sound to renew their membership for a long period of time, for example, the monthly fee of $15 is a very big amount for Malaysian users especially students, which will increase the pressure of money on the users. These limitations can lead to a poor user experience and make it difficult for users to keep track of their expenses, and the high price can also add to the monetary stress and financial problems of some users.

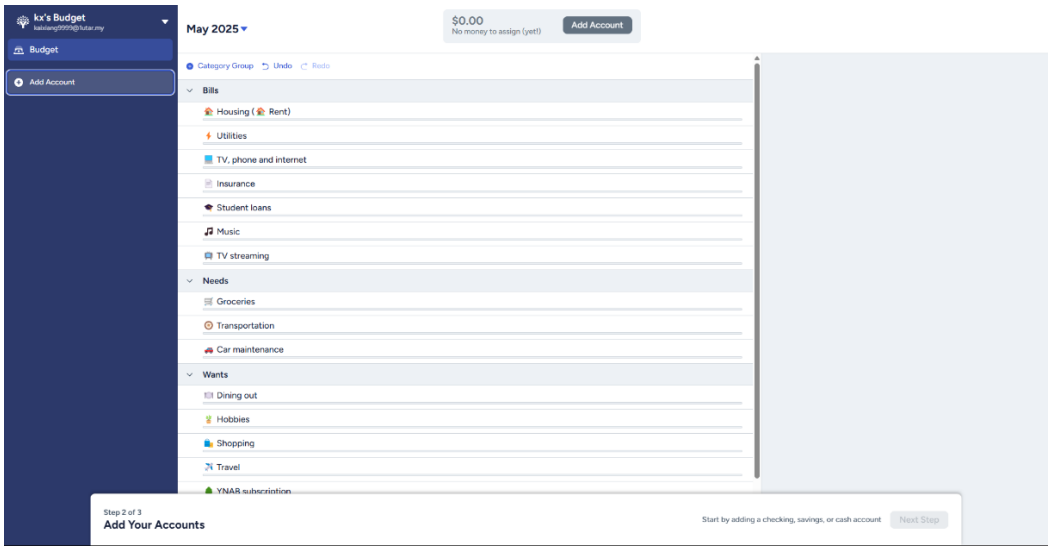


Figure 2.1.1.2.2 YNAB's guidance for users

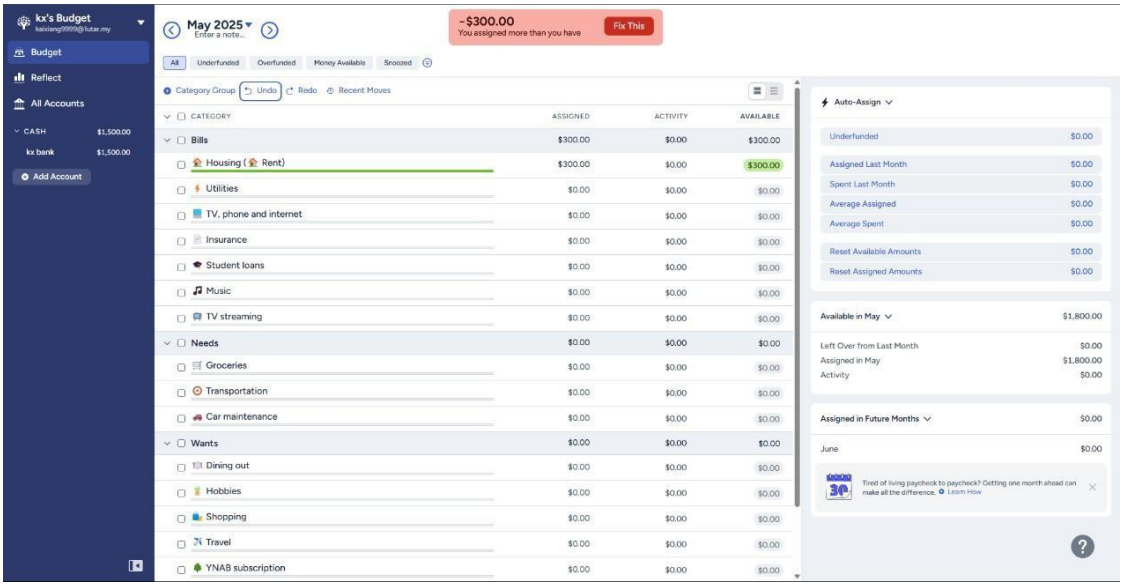


Figure 2.1.1.2.3 YNAB doesn't have a full guide.

- **Recommendations of YNAB**

Suggestions and Improvements for YNAB Firstly, to address the issue of user confusion due to complex operations, YNAB could provide interactive tutorials for users, such as adding arrows so that users can understand what buttons to press, providing small tutorial videos so that users can understand how to use the function correctly, and providing one-on-one interactive exercises to teach newbies. The second would be to add a language interface to help international users who don't speak English well understand the app.

2.1.1.3 Expensify

Expensify, is was developed in 2008 by the author David Barrett, is an online platform and mobile application for financial management and payment automation. It gives users an easy guide to the service, as well as advice for people, professionals, and businesses with payment needs [8]. With more than 5,000 reviews from G2 (Business Software Reviews), Expensify has received a high rating of 4.5/5 [8].

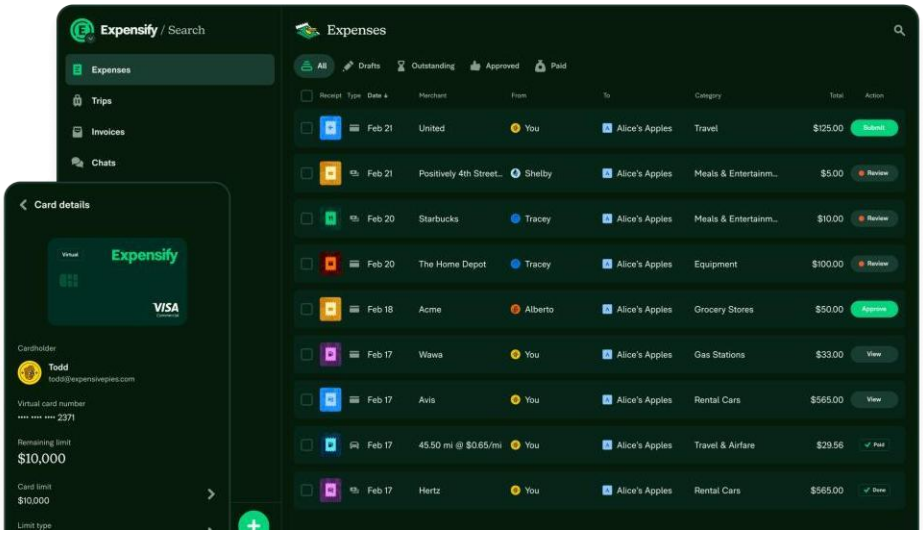
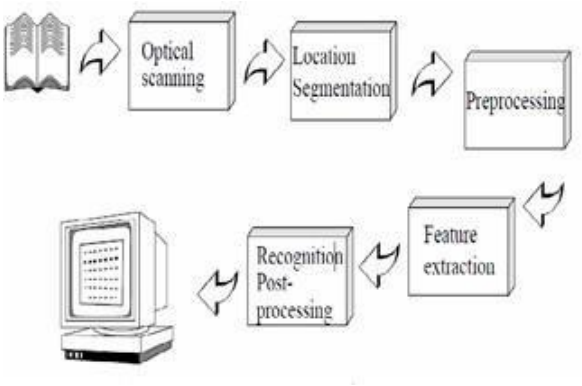


Figure 2.1.1.3.1 Expensify Website and Application

Expensify's most unique features are its data scanning, OCR recognition technology and arithmetic averaging system. Users simply take a picture of their receipt with their mobile phone, and Expensify's intelligent scanning automatically recognises the data information on the receipt [9]. Pre-processing, segmentation, feature extraction, classifications, and recognition are just a few of the several stages that make up OCR. Pre-processing is the process of eliminating handwriting variance and noise. Offline handwriting recognition systems and pattern recognition are needed in a number of areas where OCR is employed, such as mail processing, bank processing, document reading, and postal address identification [10].

Figure 2.1. 1.3.2 Components of OCR System



This modelling feature makes it possible to reduce the amount of time users spend on recording bills while also providing a better interoperability experience, which can keep users interested in recording bills and gradually develop a habit of doing so. In

addition, Expensify also provides a team chat tool for businesses or groups of friends to discuss and split expenses.

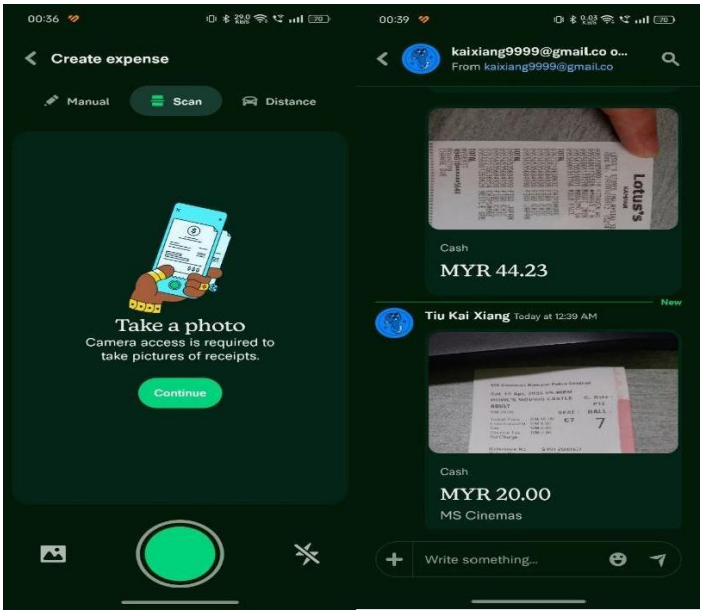


Figure 2.1.1.3.3 Scanning Model in mobile application

- **Weakness of Expensify**

  The limitation of Expensify is that its smart scanning does not provide 100% accurate data to the user, because sometimes due to the blurred handwriting or too many wrinkles in the receipt slips it will affect the accuracy of the smart scanning, resulting in the inability to upload or match the report correctly and also it takes time for the user to correct the errors manually to track the exact financial amount [11]. Based on the actual operation (Figure 2.1.1.3.), it is found that its smart scanning does not provide a classification for the customer but directly gives the total amount, which makes the user in the group payment cannot quickly divide the financial amount, although there is an arithmetic average system provided to quickly and fairly divide the bill between the group of users in order to achieve the fairness, but there is no way to accurately divide the amount of the specified user. For example, groups of users do not necessarily purchase the same items and hold them in common, which makes it necessary for them to categorise the items before settling the price. Expensify's arithmetic averaging model has no way to divide the debt amount equally among users after sorting, because its system can only scan the receipts, analyse the total amount on the receipt, and then allocate the exact amount to the users based on

the averaging algorithm. Additionally, while Expensify automatically reads the country's currency when scanning receipts, it does not provide a multi-currency conversion feature, which would prevent international users from budgeting their finances appropriately based on exchange rate conversions. This leads to unfair distribution of finances and affects users' financial tracking and budgeting, potentially causing them to overspend due to recording errors.

Figure 2.1.1.3.4 Expensify Split Bill Sample

- **Recommendations for Expensify**

  Expensify's suggestions for improvement would be to optimise the algorithms of its own scanning module to improve scanning accuracy, and to add the ability to sharpen photos to make scanning easier and clearer for data collection. Using this function may improve image clarity by repairing old, hazy photographs, restoring sharpness, boosting face characteristics, and making textures appear sharper. It makes old images appear new while retaining their inherent charm, making it an ideal tool for anybody wishing to preserve family memories or modernise older photographs [12]. Segmentation could also be added to the scanning module to help groups of users get a fairer

distribution of financial claims. Lastly, multi-currency support could be added for internationalisation.

2.2 Technologies Research

2.2.1. Optical Character Recognition (OCR)

Optical Character Recognition or OCR is widely used in converting receipt and invoice images into useful text, e.g., vendor names, dates, and total amounts. Traditional OCR engines, e.g., Tesseract, are accurate but computationally expensive and therefore less usable on phones. The latest research work is on light- weight solutions that are quick without making trade-offs on accuracy. As a case in point, Li et al. (2023) developed a compressed model, StrucTexT_slim, that was able to achieve over 94% accuracy on test sets and was around 30% faster than traditional engines [13]. Yu et al. (2025) suggested a light-weight multimodal model that can be run in real time on a phone, having the potentiality of processing up to 25 frames per second after optimization [14]. The research shows a clear trend towards efficient, phone-friendly OCR systems on receipts.

Fact Finding

OCR receipt research typically employs publicly sourced datasets like SROIE and FUNSD in an effort to test performance. Accuracy, speed, and model size are the most crucial evaluation methods. Knowledge distillation, quatization, and multimodal alignment are trendy methods employed in efforts to make it more efficient. The general workflow generally involves the following procedures: receipt image capturing, preprocessing in an effort to enhance quality, text region detection and segmentation, character recognition via CNN or Transformer models, and as a conclusion, extraction of structured data in the form of total amount.

Data Collection

Creating effective OCR systems necessitates receipt and invoice images that are diverse. Previous work underscores the value in working with large datasets whose fonts, layout, and image quality differ. The most frequent issues are low light, poor- quality images, small characters, and both printed and handwritten text. Gathering both publicly available datasets and in-the-wild receipts guarantees that OCR

systems are tested in meaningful conditions and are capable of accurately pulling totals and other important fields.

2.2.2. AI Engine Research

The recent development in the personal finance sector is the **growing use of AI** to turn raw transaction data into **valuable insights, predictions, and customised recommendations**. There are two complementary strands of AI research that are of particular interest to such an engine: (1) **unsupervised behavioural segmentation** that converts transaction histories into interpretable groups of users, and (2) **temporal sequence modelling** that predicts future spending. Tabianan et al. illustrate an **end-to-end behavioural segmentation pipeline** in which transaction-inferred features (e.g., category totals, purchase frequency, merchant diversity, recency) are **normalised and clustered with K-Means** to generate coherent and interpretable customer groups that can be read as **persona-like profiles**, such as frequent small spenders or infrequent large spenders [15]. Their work highlights the advantages of K-Means as a production tool: simplicity, speed, and centroid interpretability, allowing K-Means to be a practical initial step towards bootstrapping personalisation and alleviating cold-start issues. It is through the extraction of **behavioural vectors at a suitable temporal** granularity that the structure of spending data (which is heterogeneous) is revealed and priors are generated, which can be used by subsequent predictive and recommendation modules [15].

To supplement this, Sako et al. review and experimentally compare recurrent neural network families-classical RNNs, LSTM, **and GRU-on financial time-series forecasting tasks**, including stock indexes and foreign exchange rates, and find that RNN **variants often outperform linear statistical models** on noisy, nonlinear financial sequences and that **GRU often attain a good tradeoff between accuracy and efficiency** [16]. Their findings highlight two practical implications of a money-tracker: first, sequence models (most notably LSTM/GRU) are well-placed to encode temporal information in spending patterns across categories and users; second, model choice should depend on the scale of data and length of sequence, since deep sequence models need long histories or pooled training across

similar users to prevent overfitting or lack of generalization [16]. Collectively, the two articles endorse a **production-oriented trajectory**: use clustering to create cohorts and stabilise modelling on users with limited history and use **sequence models (GRU/LSTM)** where per-user or cohort histories have sufficient temporal signal to predict near-term spending and generate proactive alerts.

In an **implementation perspective**, the literature proposes tangible reusable design patterns. Central to this are feature engineering: transactions must be converted to time-windowed aggregates (category totals by day/week/month), frequency and recency metrics, rolling statistics (mean, variance) and simple seasonality indicators; these features drive both clustering and forecasting pipelines [15]. To segment, **pre-processing and normalisation** is a pragmatic flow, **visualisation and diagnostics** (PCA or t-SNE) may use dimensionality reduction optionally, K-Means (or hierarchical clustering as an option) and centroid interpretation to label cohorts and extract business rules [15]. To predict, begin with the best statistical baselines (exponential smoothing, ARIMA) and gradient-boosted models on engineered features, then transition to GRU/LSTM architectures on cohorts or users with sufficient data--GRU could be a cheap but effective model on many financial series as Sako et al. find [16]. In manufacturing, we recommend using a hybrid routing approach: clustering users to cohorts, and routing to cohort- pooled forecasting models (user with short history) or personalized RNN models (user with long history), therefore, balancing accuracy and data-efficiency.

Although both these methods are effective, each of them has its constraints that must be resolved in a practical system. **Deep sequence models require adequate time observations and are less comprehensible than tree-based models and linear models** factor of great importance when the system needs to indicate reasons behind forecasts and recommendations to users to remain trustworthy [16]. **Clustering is unsupervised** and thus mandates cautious human interpretation to transform clusters into practice actions and also to confirm that cohort-specific guidance is suitable and just [15]. The forecasting models tend to degrade in non-stationary conditions (the sudden changes in income, loss of a job, macroeconomic shocks) therefore **continuous monitoring, retraining, online updating, or adaptive learning methods** should be applied to maintain the performance and

relevance [16]. Drawing naturally from these constraints, **research and engineering extensions** to make a money-tracker AI engine more powerful include transfer learning or meta-learning methods to better predict users with sparse history; model-agnostic explanation layers (e.g., SHAP or rule-based summaries) to explain why a forecast or recommendation was made; privacy-preserving methods (federated learning, differential privacy) to use sensitive financial data; and multimodal enrichment (payday markers, calendar events, textual annotations) to contextualize anomalous transactions and better predictive accuracy

There are **practical recommendations** based on the two studies, which are straight forward. Initial: First, use an interpretable segmentation model (K-Means) to bootstrap personalization and to cluster in cohort priors. Second, establish predictable forecasting baselines (statistical + gradient-boosted models) and then test GRU/LSTM sequence models where they are backed up by the data; cohort pooling to address cold-start. Third, group clustering and forecast outputs in a rule/logic layer that translates forecast failures and portfolio risk measurement to tangible, implementable action (e.g., budget limits, automatic savings recommendations, tip-offs by merchants), and never leave advice without a brief explanation of the logic. Lastly, test the system with relevant evaluation metrics- MAE/RMSE to evaluate forecasts, silhouette /clustering stability to evaluate segmentation, and user-level engagement/behavioural impact proxies to loop and confirm that the AI interventions enhance financial performance.

Overall, the analysed articles provide evidence of a practical, progressive design of an AI-based money-tracker: K-Means segmentation transforms raw transaction logs into practical cohorts and priors [15], and GRU/LSTM sequence models can make strong near-term predictions in cases where time-dependent information is available [16]. A combination of these elements with additional features, such as explainability, privacy controls, and operational monitoring, would provide a practical path to provide meaningful and deployable personalized reports and advice.

2.2.3. Workflow Automation and AI Agent Technologies

Workflow automation plays an increasingly important role in modern application development by minimizing manual intervention and improving overall system efficiency. One commonly used tool in this field is n8n, an open-source workflow automation platform that enables users to integrate various applications, APIs, and services through visually constructed workflows [33]. The system is built on a node-based structure, where each node represents a specific function such as data transformation or event triggering, allowing tasks to be executed in a structured and automated sequence [34].

n8n supports both no-code and low-code development approaches, making it accessible for users with different levels of programming expertise. It allows the creation of sophisticated workflows with minimal coding effort and provides features such as conditional branching, scheduling, and API integration to support dynamic automation processes [35], [36]. Furthermore, n8n offers flexible deployment options, including cloud-based and self-hosted environments, giving developers greater control over system configuration and data privacy [33].

In recent developments, workflow automation has been further enhanced through the incorporation of Artificial Intelligence (AI), particularly with the use of AI agents. These agents are capable of operating independently, performing tasks such as data analysis, decision-making, and interaction with external systems [38]. Compared to traditional rule-based approaches, AI agents are more adaptable as they can interpret context and respond to changing conditions, making them suitable for more complex and dynamic workflows [39].

Combining AI agents with workflow automation platforms like n8n enables the development of more intelligent and adaptive systems. For instance, n8n can be configured to support AI-driven workflows where agents automatically process information, generate insights, and initiate actions without manual input [37]. This integration extends the capabilities of automation systems beyond repetitive tasks, allowing them to perform context-aware operations.

The implementation of AI agents generally follows a systematic process that includes defining tasks, designing workflows, integrating system components, and evaluating

performance. It is advisable to begin with simpler implementations and progressively introduce more complex functionalities to ensure system stability and accuracy [38]. This gradual approach helps improve reliability while minimizing potential risks during deployment.

Overall, the integration of workflow automation platforms such as n8n with AI agent technologies forms a strong foundation for developing intelligent systems. These technologies not only improve operational efficiency and reduce manual workload but also enable applications to provide more personalized and context-aware services.

2.3 Overview Table of Qianji, YNAB, Expensify and Proposed System

Table 2.3 Overview Table of Qianji, YNAB, Expensify, and Proposed System

Category	Qianji	YNAB	Expensify	Proposed System
Core Features	✔ Simple, minimalistic UI with efficient operation ✔ Core functions: bookkeeping, billing stats analysis, budget & asset management ✔ Advanced features: multiple/shared books, free backup, import/export data, multi-currency support	✔ Envelope-style budgeting: Assign income to categories ✔ Real- time category balances with bank integration ✔ Generate reports; set budget goals with reminders	✔ OCR-based smart receipt scanning ✔ Automatic arithmetic expense-splitting ✔ Supports 150+ currencies ✔ Team chat for group expense sharing	✔ Intuitive UI/UX with dark mode support ✔ Enhanced OCR with image enhancement ✔ Itemized group expense splitting ✔ Real-time multi-currency conversion ✔ Cross-platform synchronization ✔ AI-powered financial insights
Budget Management	Tracks daily expenses by category; dashboard visualizes spending for budget planning	Envelope-method budgeting: allocate every dollar, proactive goal-tracking	Focuses on expense/payment automation; no structured budgeting model	Comprehensive budget with customizable goals, predictive analytics, and intelligent alerts
Expenses Recording	Manual entry with category selection	Manual entry with bank integration	OCR receipt scanning with basic data extraction and manual entry of data.	Enhanced OCR with image pre-processing and manual entry category option.

Limitation	Lacks advanced financial planning or forecasting features	Steeper learning curve; only English interface	OCR accuracy issues with poor-quality receipts ; limited splitting granularity ; no currency conversion	To be determined on project 2 after receiving review
Solution/ Improvement	Add predictive analytics and budget alerts	Provide interactive tutorials and multi-language support	Optimize OCR algorithm and enable photo sharpening	To be determined on project 2 after receiving review
Target User	Personal finance users wanting a simple, ad-free bookkeeping app	Budget-conscious individuals or families	Individuals and businesses need expense management	Diverse users, including students, families, couples, and groups with international finance needs
Supported Platform	Mobile (iOS & Android) – app available on App Store/Play Store	Cross- platform (Web, iOS, Android)	Web and mobile apps	Mobile Application (Android)
UI and Customization	✔ Simple, clean UI by design. Limited theming/customization	✔ Intuitive, clear interface. Few customization options	✔ User-friendly onboarding. Limited custom themes.	To be determined on project 2 after receiving review

CHAPTER 3 Proposed Method/Approach

3.1 Project Methodology

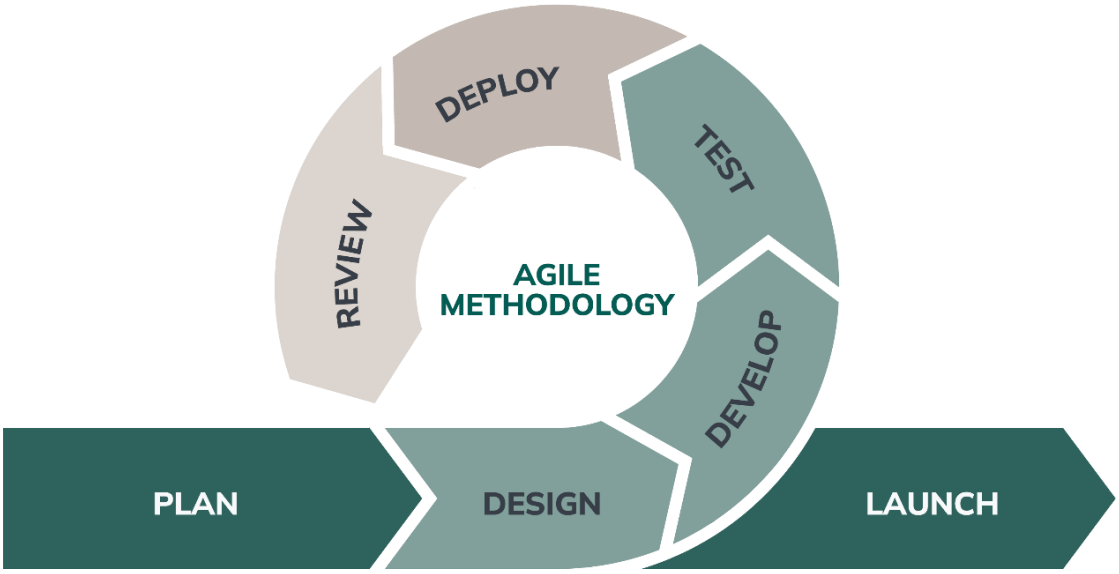


Figure 3.1 Agile Software Development

Agile development methodology was adopted for this project, given its iterative and adaptive nature. Agile divides work into multiple phases and emphasises continuous delivery and feedback, thus enabling the development process to be continuously improved and remain flexible [17]. The methodology aligns with the goals described in Chapter 1 as it allows for rapid integration of user feedback and changing requirements, resulting in a user-centred approach, and giving the highly responsive design to user. The iterative process contributes to the project goal of delivering a usable prototype early and continuously improving it through multiple iterations.

3.1.1. Phase 1- Planning Phase

During this phase, we first gain a clear understanding of the project's preliminary requirements and conduct background research. Subsequently, we formulate the problem statement, define the project scope, and establish project objectives. then, we research and screen existing popular accounting applications for evaluation and comparison, identifying their strengths and weaknesses. Analysing their features helps refine the project scope and clarify objectives. furthermore, we investigate and review relevant literature on OCR technology to understand its principles in advance for subsequent development.

3.1.2. Phase 2- Design Phase

At this phase, system requirements, system architecture, database design (such as ERD diagram design), and technical user story design (such as development of wireframes or workflows) are the focus.

- **Requirements** → User needs (e.g., user stories, acceptance criteria)
- **System architecture** → High-level structure (e.g., workflow diagram)
- **Database design** → Define the data model (i.e., ERD diagram)
- **Technical design** → Story-level design (e.g., wireframes, API contracts)

3.1.3. Phase 3- Develop (Implementation) and Testing Phase

System functionality will be progressively developed and integrated to enable testing of the project's programme once preliminary features have been consolidated into a complete prototype application. During testing, emphasis will be placed on verifying the effective execution of each function, ensuring real-time database logging, and confirming the integrity of the user interface.

3.1.4. Phase 4- Deploy, Review, and Maintenance Phase

At this stage, the first-generation application version is distributed to users for testing. Feedback gathered from them helps identify any overlooked issues or new suggestions for consideration, enabling further management and optimisation of the programme.

3.2 System Design Diagram/Equation

3.2.1. Use Case Diagram

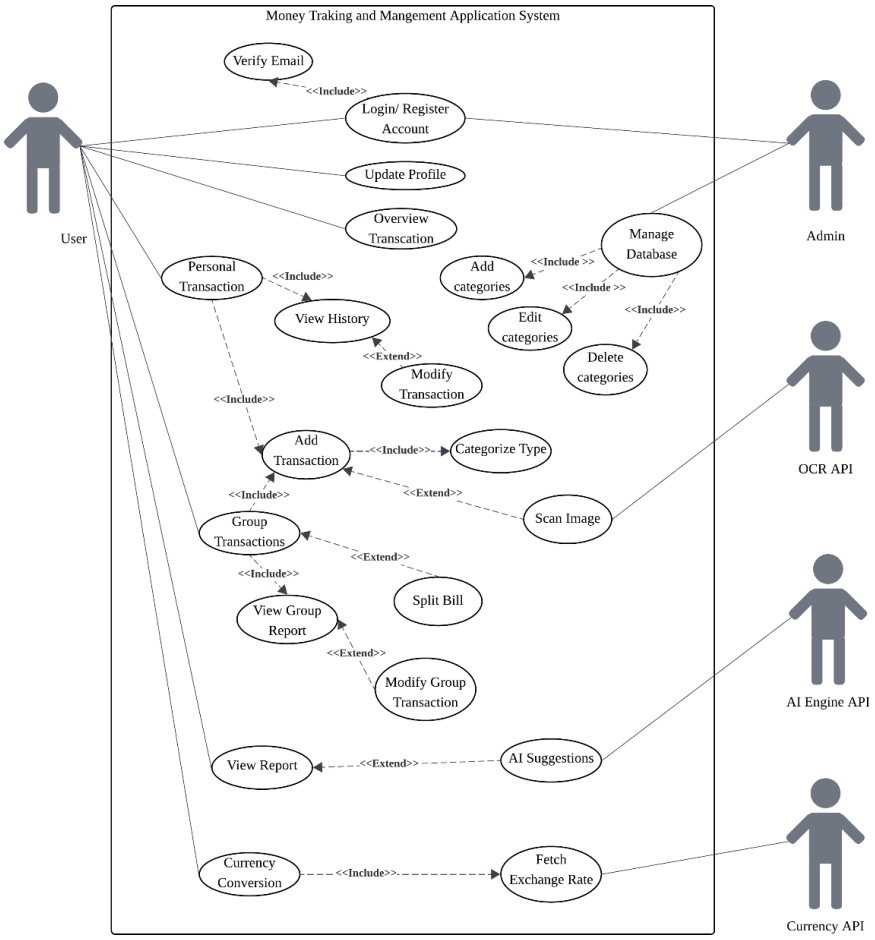


Figure 3.2.1: Use Case Diagram of System

Under the Use Case Diagram of Money Tracking and Management Application System (Figure 3.3.1), users can sign up or log in through accounts with email verification, edit profiles, and handle both personal and group monetary activities. They can add transaction records with type categorisations, check history, and edit records when required. The system further includes users scanning receipts through the use of the OCR API, with the possibility of automatic data input. The system generates financial reports and extends the same with AI-driven advice on budgeting as well as expenditure insights. Members also handle group expenditure through the division of bills, checking group reports, and effecting modifications,

among others, using the system. Users can further utilise the currency converter function that retrieves current exchange rates from the Currency API.

At the administration level, the admin is responsible for managing transaction categories within the system. The admin creates new ones, edits others, and deletes

outmoded categories so that the system can function well with the correct tracking of the user's transactions. Through category management, the admin is responsible for keeping the financial records in order and reliable among the users.

3.2.2. System Flowchart

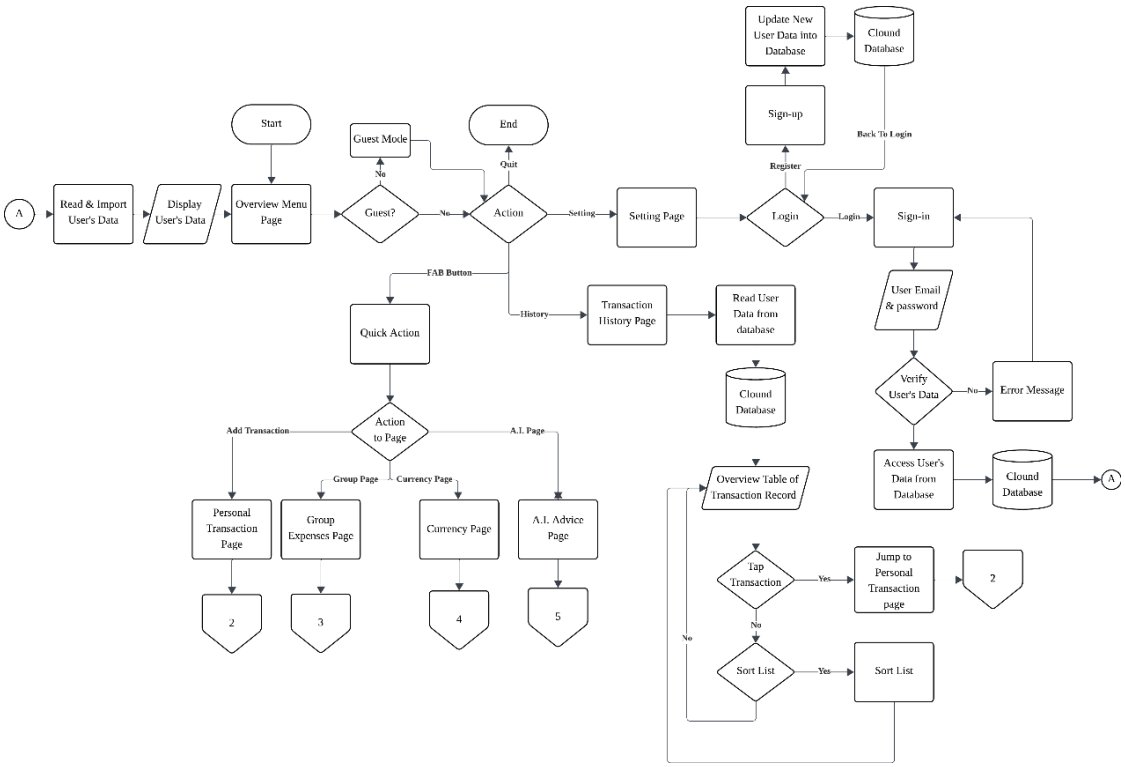


Figure 3.2.2.1: System Flowchart Page 1

The system flowchart of the Money Tracking and Management Application (Figure 3.2.2.1) shows how the system works. When the system starts, it displays the application's transaction overview to the user, allowing them to get an overview of their financial tracking, including their current amount, total income, and total expenses. Etc. From here, the user can also either continue as a guest or log in as a registered user. For the guest mode that will limit the user to access the cloud database to store their transaction data, which means if they delete this application, their whole data will also be removed, while the users who have an email account to log in or sign up, their details data will be stored in the cloud database, when they log in the

system will read their data from the database and display their history transaction record data. For the quick actions part, which will push the user to the specified application modules page to provide their next step.

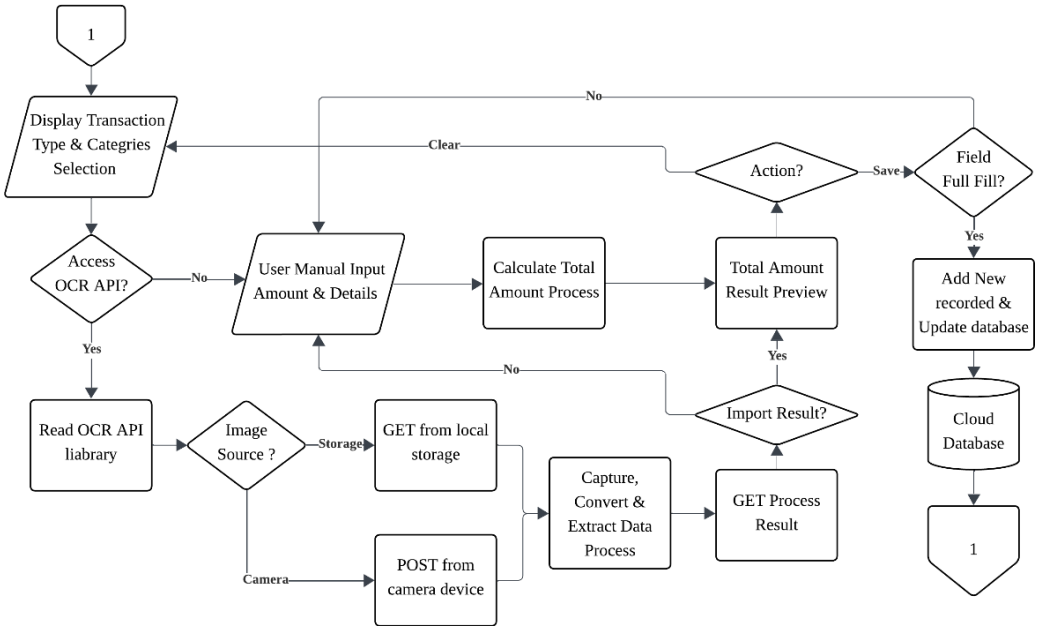


Figure 3.2.2.2: System Flowchart Page 2

For page number 2, which is the personal transaction page that allows user to add their transaction record. On this page, users are requested to fill in all the provided fields to record their transactions. These fields include selecting the transaction type (income or expense), selecting the category type, and entering the transaction amount spent. Also, the transaction title, description and picture upload are not necessary for them to fill out. Furthermore, these modules also provide an option of the OCR API that imports from ML Kit; its function may help the user get the transaction data from the receipt image rapidly, which provides them with an efficient tool to record their transaction and save manual records. After users finish their prompts, before the system saves the new record, it will make sure the compulsory fields have been filled. After that, it will save the record data and update it to the database, then return the user to the main page.

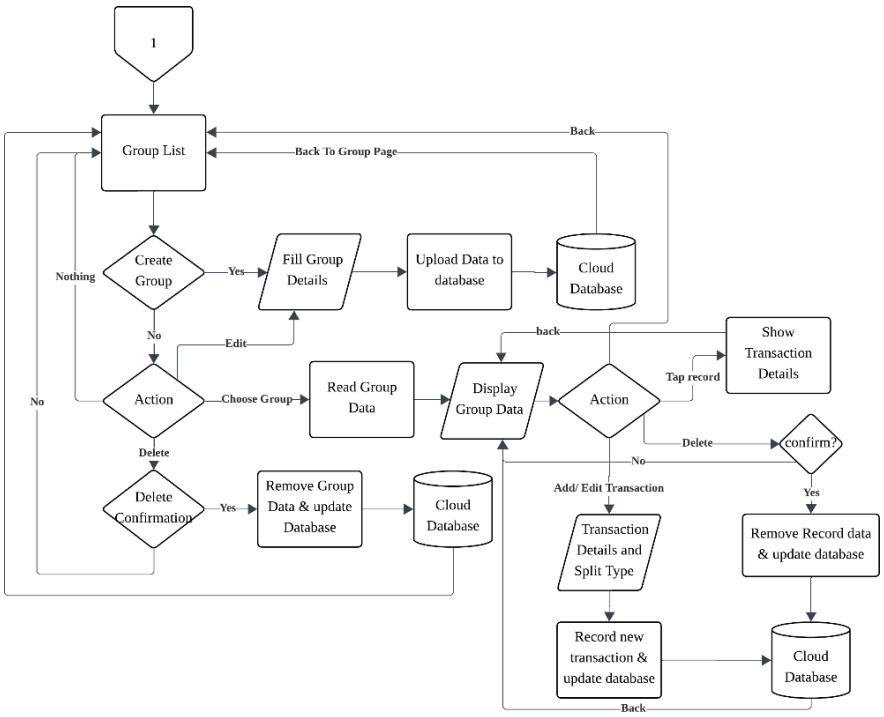


Figure 3.2.2.3: System Flowchart Page 3

For page number 3(Figure 3.2.2.3), it will proceed to the group management module page first. This page will provide an overview of the group list that was created to use for categorising the different group expense records. It may help user to track their transaction record in different groups. Users must complete the group information form in order to create a new group. It will then read the information from the database and show the user the history record when they forward it to one of the groups they set up.On this page, the user is allowed to add a new group transaction, view, edit or delete it. The transaction form will provide 2 functional options, which are the split bill record function or the manual bill record function. The split bill function will average divide the total amount of the expenses, then separate them among all group members, while the manual bill function allows member to manually arrange their expense records. After finishing the process, the new record or updated record will also be updated in the database.

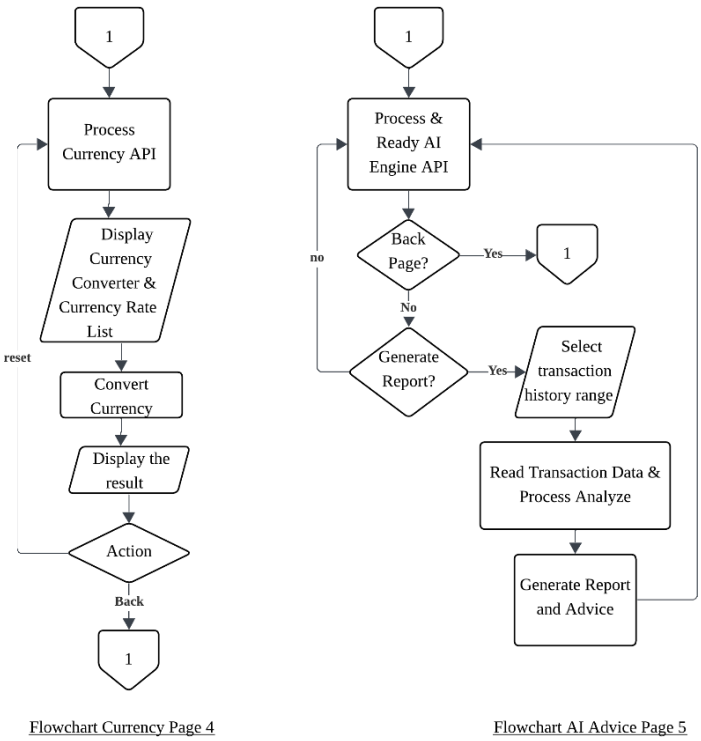


Figure 3.2.2.4: System Flowchart Page 4 & 5

This flow chart outlines two of the system's advanced features (Figure 3.2.2.4), in particular, the currency converter and the AI consultation module. On the currency converter, the system inputs data from a currency API, displays exchange rates, and allows individuals to make conversions, with the system returning instant results. On the AI side, the system installs its engine, inquires if the user needs to go back to the menu, and drafts an option to generate reports. If they choose to, individuals select a range of transaction history, and the system works through the data to provide a holistic financial report and consultation. Both modules are referred back to the main menu for convenience.

3.2.3. System Entity Relationship Diagram (ERD)

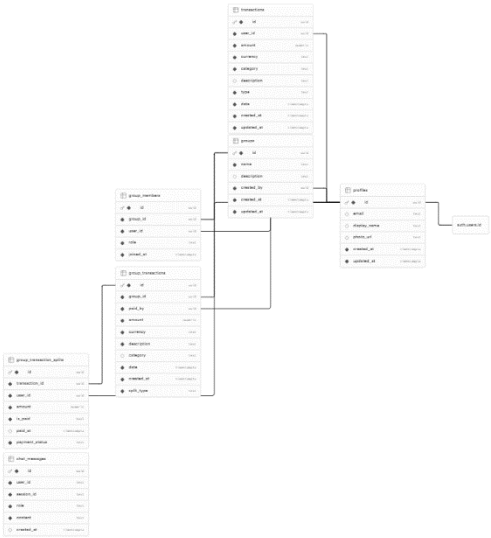


Figure 3.2.3: Entity Relationship Diagram (ERD)

The **Entity Relationship Diagram (ERD)** represents the overall **database structure** of the proposed financial management system developed using **Supabase**. It illustrates how different **entities** are organized and how they interact with one another to support the system's core functionalities, including **personal expense tracking**, **group expense management**, and **AI chatbot integration**. The database is composed of seven main entities, namely **profiles**, **transactions**, **groups**, **group_members**, **group_transactions**, **group_transaction_splits**, and **chat_messages**. Each entity is designed with specific attributes and is connected through **primary keys** and **foreign key constraints** to maintain **data consistency** and **data integrity**.

The **profiles** entity serves as the **central component** of the system, storing essential user information such as **email**, **display name**, and **profile image**. It is directly linked to the **authentication system** and acts as a reference point for most other entities. The **transactions** entity is used to record individual financial activities of users, allowing them to manage their **personal income and expenses** independently. Each transaction is associated with a specific user through a **one-to-many relationship**, ensuring that all financial records are properly attributed.

For **group-based financial management**, the system includes the **groups**, **group_members**, **group_transactions**, and **group_transaction_splits** entities. The **groups**

entity represents shared expense groups created by users, while the **group_members** entity establishes a **many-to-many relationship** between users and groups, allowing multiple users to participate in multiple groups. Within each group, shared expenses are recorded in the **group_transactions** entity, which stores details such as the **payer**, **amount**, and **category** of each transaction. These transactions are further divided among group members through the **group_transaction_splits** entity, which tracks how much each member owes, along with their **payment status**. This structure enables **flexible expense splitting**, supporting both **equal** and **custom** distribution methods.

In addition to financial management, the system incorporates an **AI chatbot module** through the **chat_messages** entity. This entity stores interactions between users and the chatbot, including **message content**, **roles**, and **session information**. It allows the system to maintain **conversation history** and supports integration with automation tools such as **n8n** for retrieving and storing financial data dynamically.

Overall, the relationships within the ERD are designed to ensure **efficient data flow** and **scalability**. The **one-to-many** and **many-to-many relationships** between entities enable the system to handle complex interactions, such as multiple users participating in shared expenses while maintaining individual financial records. This structured design not only enhances **data integrity** but also provides a solid foundation for **future system expansion**, making the application **robust**, **flexible**, and suitable for real-world **financial management scenarios**.

3.2.4. System Architecture Diagram

This application is designed based on a three-layer architecture framework: **Data Layer, Application Layer,** and **Presentation Layer.**

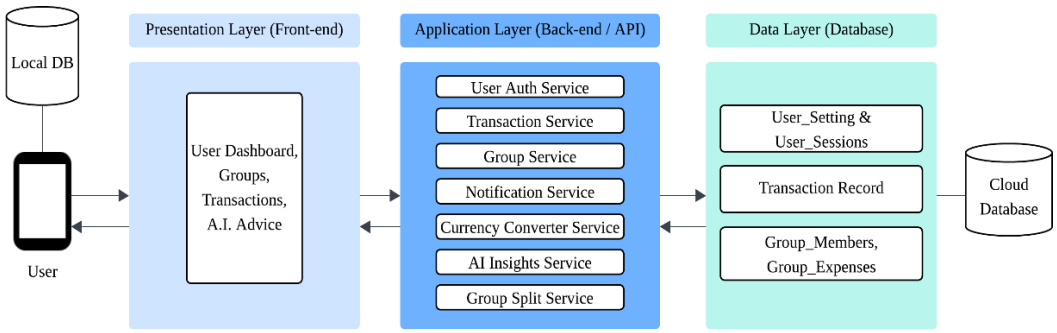


Figure 3.2.4: 3-Tier System Architecture

The **Data Layer** manages all structured financial data and user information. It

contains relational database tables for **Sessions, Users, Transactions, Groups, Group Members, Group Expenses, Expense Splits, Invitations**, and **Notifications.** It is the foundation where data is always stored, retrieved, and modified. It puts data such as transactions, a user's membership in a group, and expense splitting in relational tables so the Application Layer can have tidy, reliable data to use.

Application Layer covers the system centre where the main business logic and services are hosted. It is comprised of autonomous modules in the form of **User Service, Authentication Service, Transaction Service, Group Service, Expense Split Service, Currency Converter Service, AI Service, Notification Service,** and **Invitation Service.** A certain business functionality is handled by each service: from user session and monetary transaction handling to calculating group expense ratios, real-time currency exchange, and implementing AI-driven features such as expenditure forecasts or budget plans. Because these services are modules in themselves, the system achieves higher scalability and flexibility, where features can easily be added later without impacting the existing layout.

Finally, the **Presentation Layer** is the user-end of the system, offered in a **Mobile App.** It provides the platform where you can interact with dashboards, transactions can be processed, a group can be created or joined, expenses can be tracked, and AI insights retrieved. It communicates to the Application Layer through APIs from the frontend, presenting user inputs such as the addition of a transaction or a request for a currency conversion, and then getting output processed back to present to the user. It provides a smooth user experience while taking care of security and reliability.

As a whole, such three-tier architecture offers for each tier a distinct functionality: the **Data Layer** holds well-organized data, the **Application Layer** enforces business logic and regulations, and the **Presentation Layer** offers a user-friendly interface for the final user. It offers a modular, maintainable, and expandable system for funds administration, allowing for a simpler means of bringing in the app with future additional services such as advanced AI analytics or additional currency functionality.

3.2.5. Activity Diagram

3.2.5.1. Sign In Activity Diagram

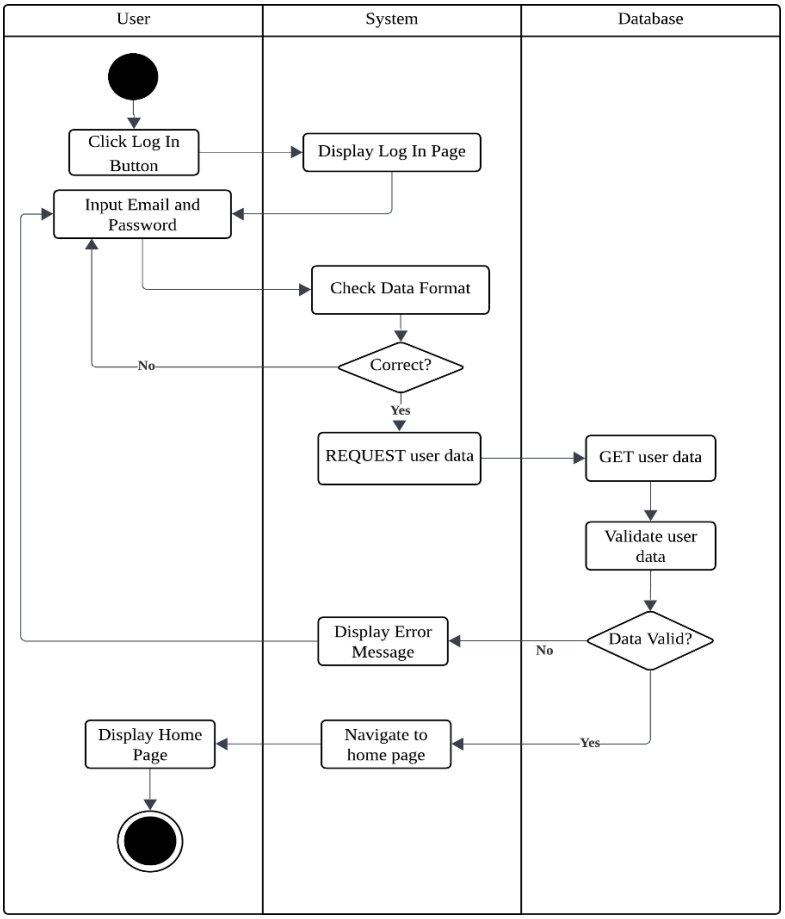


Figure 3.2.5.1: Sign-in Activity Diagram

User initiates login, enters details, and then the system first checks if they look correct. If they do, then the system checks against the database to confirm if they are correct. If the process is successful, the user is redirected to the home page; if not, an error message appears, and they must re-enter their information.

3.2.5.2. Registration Activity Diagram

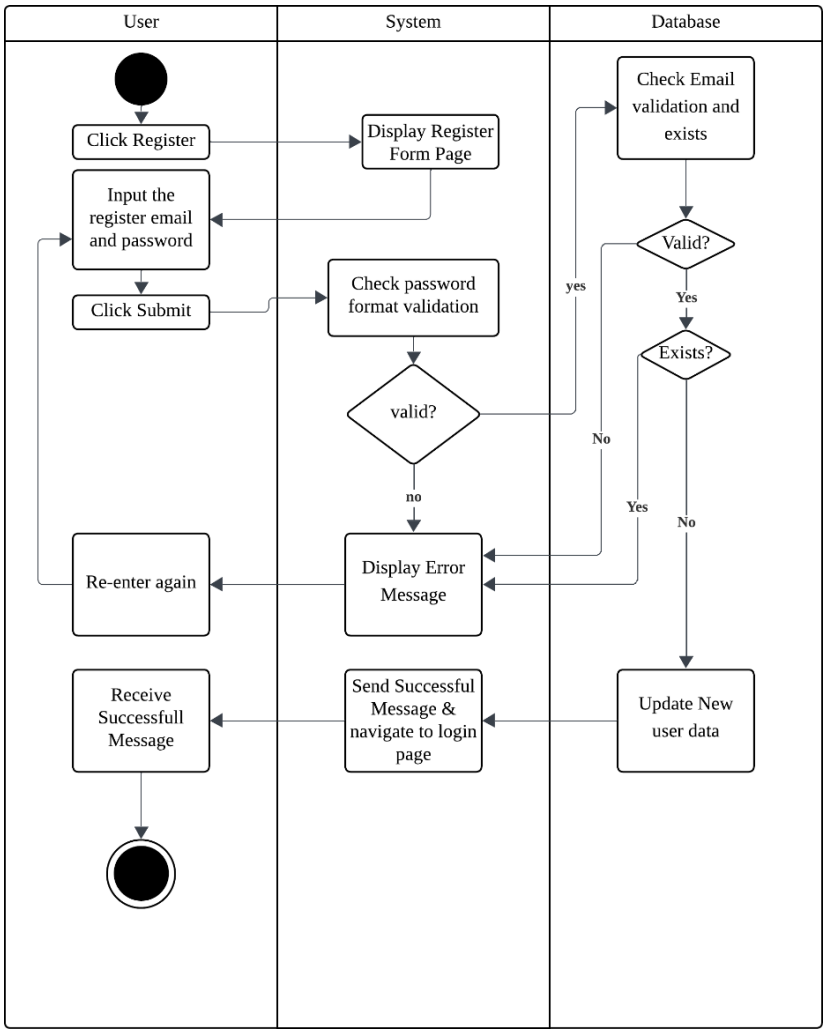


Figure 3.2.5.2: Registration Activity Diagram

The process of user registration is depicted in the activity diagram above (Figure 3.2.5.2) and begins when a user hits the Register button, after which the system displays a form. The system then performs a password format validation check once the user clicks submit after entering his email address and password. An error message is displayed and the user is prompted to re-enter their information if the format is invalid (No); if the format is valid (Yes), the system sends the revised user data to the database.After a successful storage, the system displays an informative message of success to the user and automatically redirects to the login page, and it is the registration workflow.

3.2.5.3. OCR Process Activity Diagram

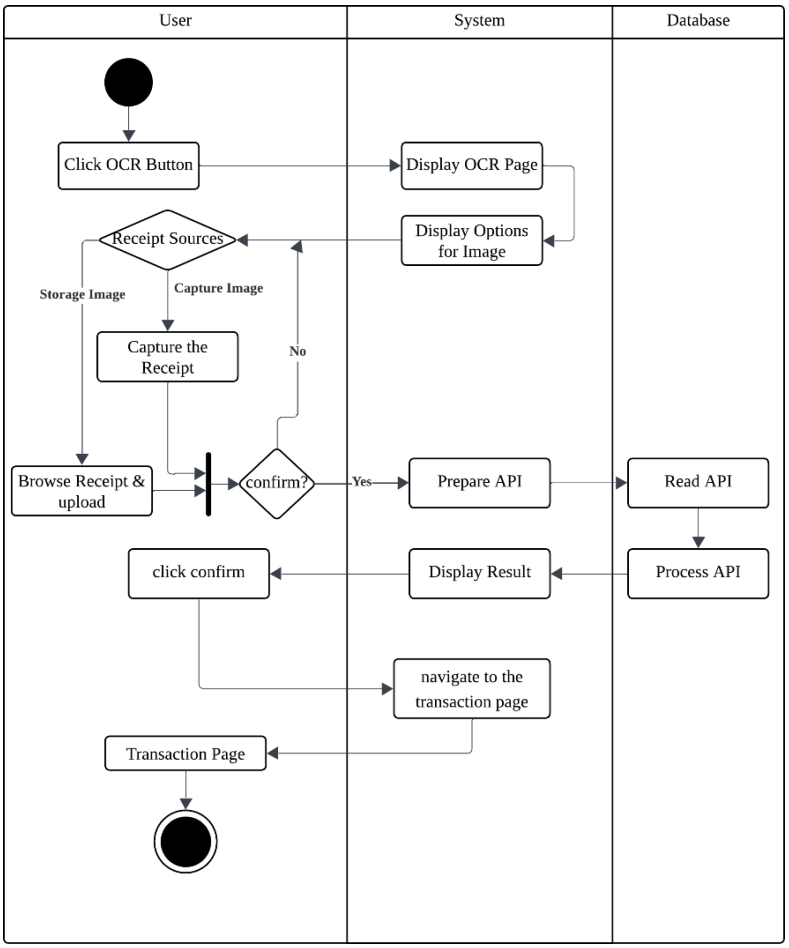


Figure 3.2.5.3: OCR process Activity Diagram

This activity diagram (Figure 3.2.5.3) outlines the **OCR receipt scanning process**, which from the time a user clicks an "OCR Button," triggering the system to open a different page with image source options. A user then captures a new picture of a receipt or searches and opens a previously saved image before clicking on "confirm" to initiate the scan. On confirm, the system formats and sends the picture to a specialized OCR API for the scanning; the API interprets the picture, does text data analysis, and returns it back to the system. On getting the data from the OCR API, the system displays the output data returned to the user for analysis before finally taking it to the transaction page for data/ logging save, completing the process.

3.2.5.4. Split Expenses Activity Diagram

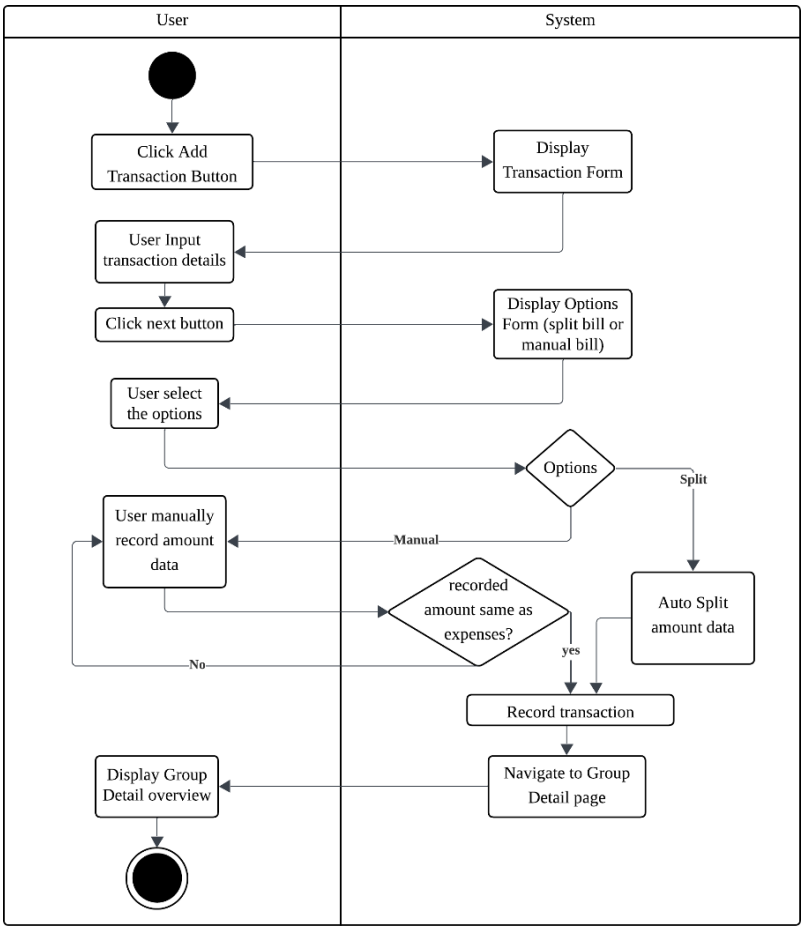


Figure 3.3.5.4: Split Expenses Activity Diagram

The process of manually adding a new transaction is shown in this activity diagram (Figure 3.2.5.4). It starts when the user clicks the "Add Transaction Button," which causes the system to present an input form for basic information such the amount, merchant, and date. The user is presented with a "Options Form" to select whether to share the bill with other members or treat it as a single manual entry after filling out the form and clicking the "next" button.Based on the preference of the user, the user can now enter the amount data manually for a basic transaction or configure the split details, where the system saves the transaction and, finally, displays the user a "Group Detail Overview" and continues to the "Group Detail page" for the completion of the process.

3.2.5.5. Currency Converter Activity Diagram

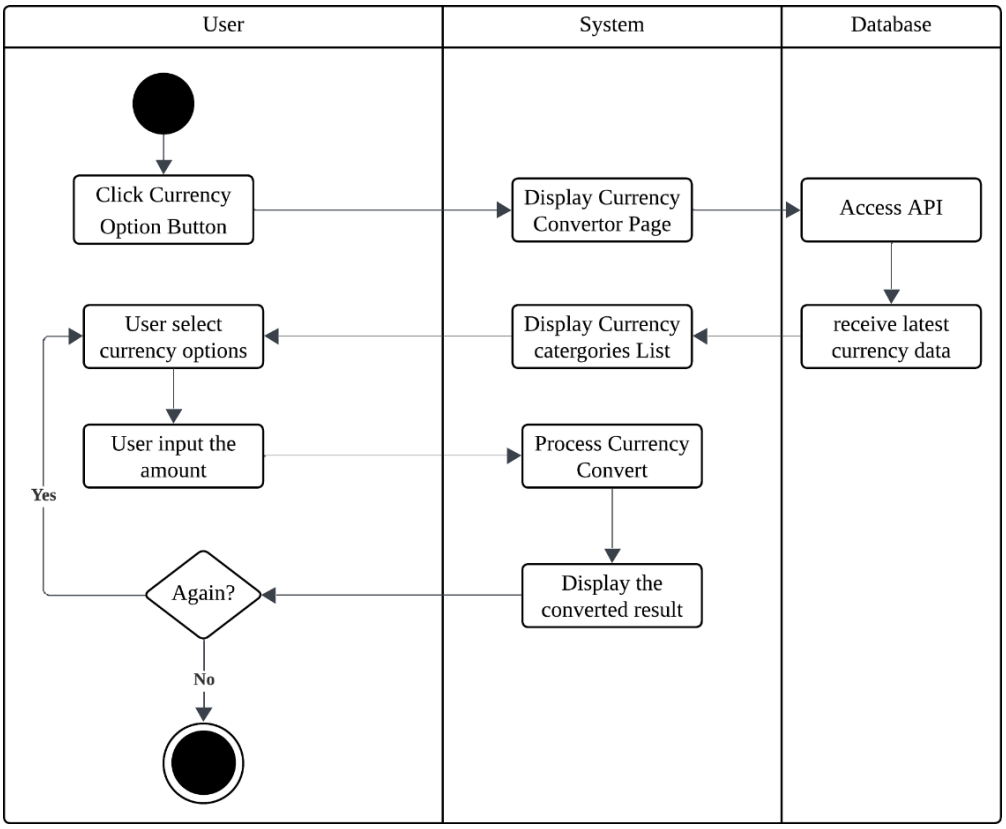


Figure 3.2.5.5: Currency Converter Activity Diagram

This activity diagram (Figure 3.2.5.5) shows how the application converts currencies. The system starts by displaying the converter page and retrieving the most recent exchange rates from an API when the user clicks on the currency option button. The system will now convert the amount and display the outcome after the user chooses the desired currencies and enters the amount. Lastly, the user can either perform another conversion or end the process.

3.2.5.6. AI insight Activity Diagram

The activity diagram (Figure 3.2.5.6) below illustrates the process of **generating the financial report** via the system's AI engine upon the user clicking the "Generate Button" and the system reading and initializing its API for the AI engine. The user inputs and selects an analysis period start and end date into a displayed "History Range Form," prompting the system to read the related transaction history by fetching the relevant database data. The system analyses such data through the system's AI engine, which produces a report with

financial recommendations and presents it to the user; the process concludes by asking if the user would like to output another report ("Again?"), and if not ("No"), the workflow ends.

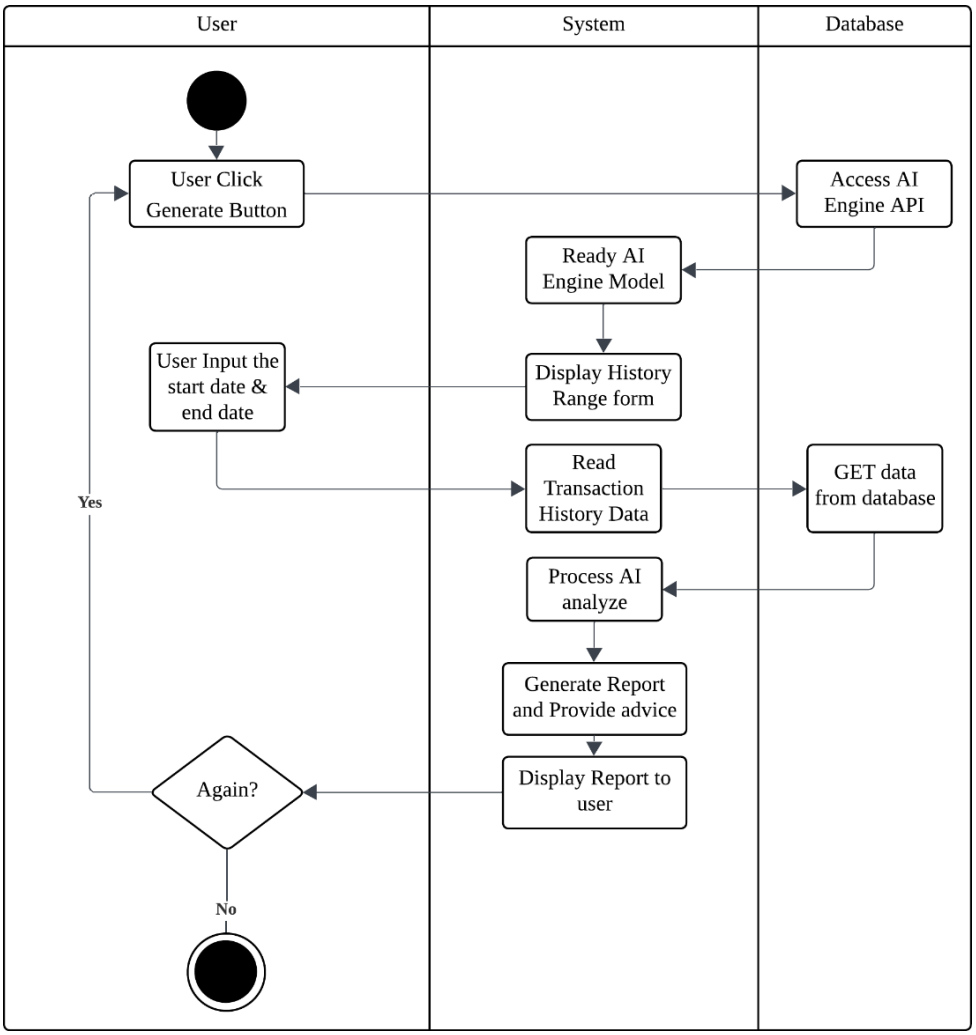


Figure 3.2.5.6: AI Insight Activity Diagram

3.3 User Requirements

- **Functional Requirement**
  - The user must be able to use email to register and log in.
  - The user has the ability to modify their profile details.
  - The user will have access to add, edit, and remove their personal financial transaction records
  - User should be allowed to categorise the transaction type
  - Users have the ability to view and sort the transaction history
  - User should be able to create and join groups
  - The system should allow splitting bills among the group members
  - The system should allow users to upload or scan receipts
  - Extracted data from receipts should be automatically converted into a transaction record
  - The system shall fetch real-time exchange rates from the Currency API
  - The administrator should be able to monitor and maintain category consistency
  - The system should generate the visual charts and summaries for users for better understanding
- **Non-functional Requirement**
  - The system should process transactions and update within 2 seconds
  - The financial report should be generated within 5 seconds
  - The system should be able to support up to 500 concurrent users
  - The user interface design should be simple and intuitive
  - The splitting bill should be able to complete within 1 minute
  - The system architecture should be expandable for managing more users and rising transactions.
  - The database will support future growth without performance degradation.
  - The repository will follow standard design practices for upgrade and transformation facilitation.
  - The system will facilitate easy integration of future advancements within the system.

3.3.1. System Performance Requirement

The Money Tracking and Management Application System shall achieve a high level of efficiency, accuracy, and reliability for the purpose of both individual and group financial management support. Target performance indicators for the purpose of system development and evaluation were identified as follows:

Table 3.3.1 System Performance

Performance Characteristic	Target
Accuracy and Reliability	✓ The Optical Character Recognition (OCR) Module is designed to pull out at least 90% accuracy of the information from scanned receipts. ✓ Financial recommendations based on AI shall achieve an accuracy of at least 85% for the identification of spending behaviour and budgeting insights. ✓ The system needs to remain operational for an uptime of 99%, thus ensuring constant availability of the service.
Timing and Response Speed	✓ Main user operations, such as adding or updating transactions, must be carried out within 2 seconds. ✓ Report preparation should not take more than 5 seconds, even for large data sets. ✓ Retrieval of the latest exchange rates for currencies through the Currency API should be accomplished within 3 seconds.
Scalability	✓ It must be able to handle up to 100 concurrent users without any perceived performance degradation. ✓ The database design should allow for scalability such that it can handle increasing volumes of users and transactions.

Usability	✓ The addition and categorization of a transaction should be possible within four or fewer user interactions (taps or clicks). ✓ In one minute, the average for completing the task, for example, dividing a bill or producing a report, should never be exceeded. ✓ Usability testing should demonstrate a 95% success rate for the accomplishment of primary user tasks.

3.3.2. OCR Performance Evaluation Criteria

The Optical Character Recognition (OCR) module is an important component of the KeepIt Money Tracking and Management Application, developed to automatically extract transaction details from receipt images. To evaluate its performance, a structured assessment method is established to measure both its accuracy and reliability.

The evaluation focuses on three key data elements: transaction amount, transaction date, and merchant name. These fields are essential for maintaining accurate financial records and are used as the main criteria for assessing OCR performance. The extracted results are compared against manually verified ground truth data to determine whether each field has been correctly recognized.

To provide a comprehensive evaluation, testing is carried out under two different conditions representing both ideal and practical usage scenarios. These include online sample receipts and real-life captured receipts.

Table 3.3.2.1 System OCR Performance

Evaluation Category	**Description**	**Purpose**
Online Sample Receipts	High-quality receipt images obtained from online sources	To evaluate OCR performance under ideal conditions

Evaluation Category	Description	Purpose
Real-Life Receipts	Images captured using mobile devices in real environments	To evaluate OCR performance under practical usage conditions

The accuracy of the OCR module is calculated by determining the proportion of correctly extracted fields relative to the total number of evaluated fields, as summarized in Table 3.3.2.2.

Table 3.3.2.2 System OCR Evaluated

Metric	Description
Total Receipts Tested	Number of receipt images used in testing
Total Fields Evaluated	Number of evaluated fields (amount, date, merchant)
Correctly Extracted Fields	Number of fields accurately recognized by OCR
Accuracy	Ratio of correctly extracted fields to total fields evaluated

It should be noted that the OCR module is designed specifically for recognizing printed text. Handwritten receipts are not supported and are therefore excluded from the evaluation scope. These cases are considered a limitation and are not included in the adjusted accuracy calculation.

Overall, this evaluation framework provides a systematic approach for testing and validating the OCR module, with detailed results and analysis presented in Chapter 6.

3.4 Project Timeline

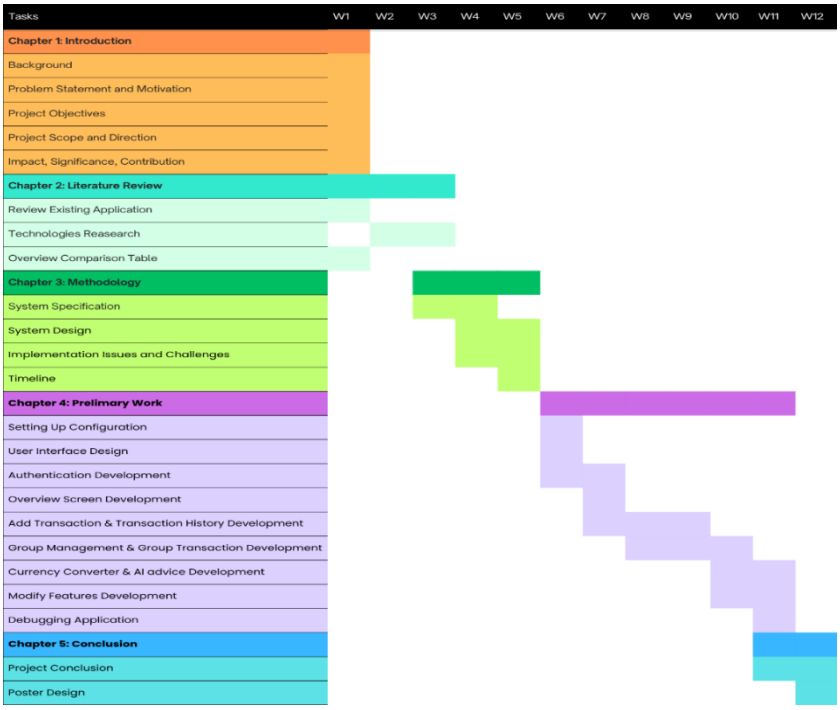


Figure 3.4.1 Project Timeline(FYP 1)

The project is scheduled to be executed within a period of **twelve weeks** and into **five phases**. **Weeks 1 and 2** are for the **introduction**, wherein the **background, problem statement, objectives, scope, and the importance of the study** are detailed. **Weeks 2 through 4** are for the **literature review**, wherein the **existing applications** are reviewed, **technology research** is carried out, and the **table of comparisons** is drafted. Then comes the **methodology phase** for **Weeks 4 through 6**, wherein **system specifications** are documented, the **designing** is developed, and the **critical implementation challenges and schedule** are determined. The main **development work** occurs for **Weeks 6 through 11**, beginning with **configuration setting up** and **interface designing**, followed by **authentication, transaction management, group management, conversion of currencies, and development of AI advice**, with the **debugging** carried out toward the latter portion of the phase. Finally, for **Weeks 11 through 12**, the **conclusion phase** involves the **summary of project results**, wherein the **challenges** are reinforced, **recommendations for the future improvements** are indicated, and the **preparation of the project poster for the final presentation** is carried out.

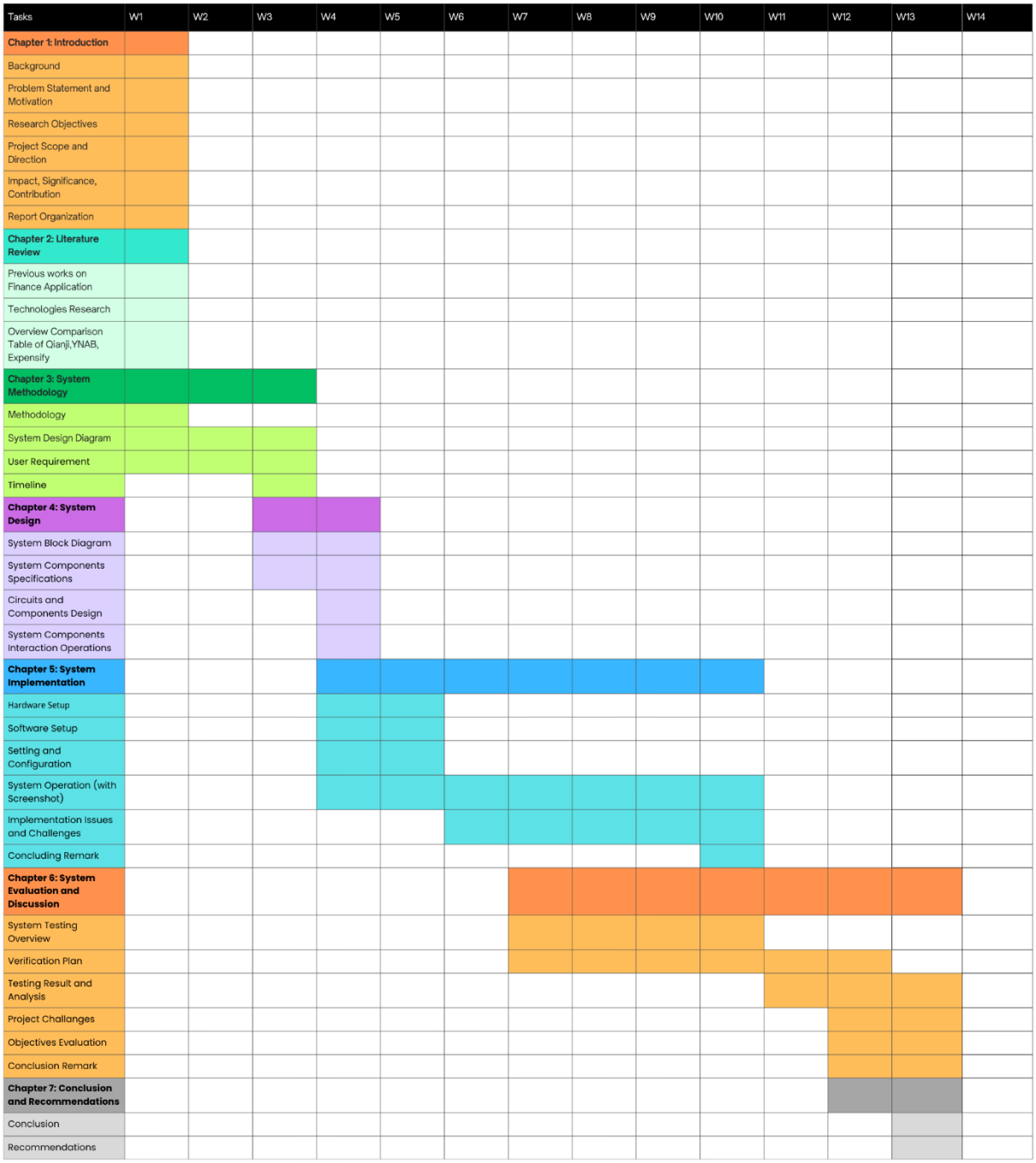


Figure 3.4.2 Project Timeline(FYP 2)

The project was carried out over a period of fourteen weeks, following a structured and systematic development approach. During Weeks 1 to 2, the project planning phase was conducted, which included developing the background of the study, defining the problem statement and research objectives, determining the project scope, and organizing the report structure. Weeks 2 to 3 were

dedicated to the literature review, where existing financial applications and relevant technologies such as Flutter, OCR, and artificial intelligence were analyzed, along with a comparison of similar systems. In Weeks 3 to 4, the system methodology phase was performed, including the selection of the development methodology, system architecture design, preparation of system diagrams, and identification of user requirements. This was followed by the system design phase in Weeks 4 to 5, where detailed system components, specifications, and interaction flows were developed. The main development phase took place from Weeks 5 to 10, involving system implementation, including environment setup, user authentication, expense tracking, OCR integration, AI-based features, and system configuration, along with documentation of system operations and resolution of implementation issues. Subsequently, Weeks 8 to 13 were allocated for system evaluation and testing, including verification planning, testing execution, result analysis, and assessment of project objectives. Finally, Weeks 13 to 14 were dedicated to concluding the project, preparing the final report, and providing recommendations for future improvements.

CHAPTER 4 System Design

4.1.System Block Diagram

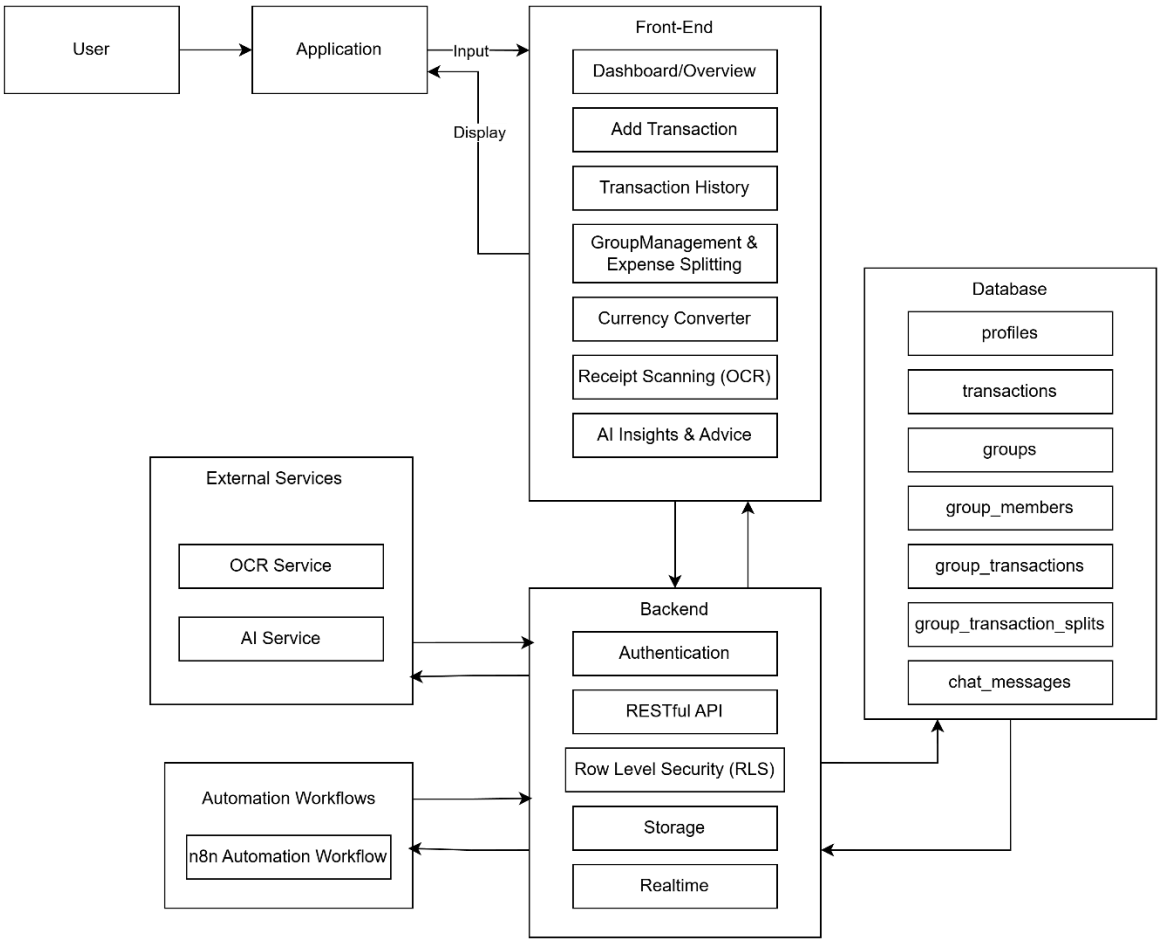


Figure 4.1 System Block Diagram

Figure 4.1 presents the system block diagram of the proposed Money Tracking and Management Application. The overall system adopts a modular architecture that is composed of several main components, including the user interface, frontend application, backend services, database layer, external services, and automation workflows.

The system operation starts from the user, who interacts with the application through a mobile interface. The frontend component is responsible for capturing user input and presenting the corresponding output. It provides a range of functionalities such as a financial dashboard, transaction recording, transaction history tracking, group expense management with splitting features, currency conversion, OCR-based receipt scanning, and AI-driven

financial insights. These features collectively create an interactive and user-friendly financial management environment.

The frontend communicates with the backend, which is implemented using Supabase. The backend handles critical operations such as user authentication, API communication through RESTful services, enforcement of Row Level Security (RLS), file storage management, and real-time data updates. It functions as the core processing layer, ensuring that all user requests are handled securely and efficiently.

All system data is stored within a structured database, which includes multiple tables such as profiles, transactions, groups, group members, group transactions, transaction splits, and chat messages. This relational database design supports both personal and group financial operations while ensuring data accuracy and consistency.

The system also incorporates external services to extend its capabilities. These include an OCR service for extracting relevant information from receipt images and an AI service that analyzes financial data to generate insights and recommendations. By integrating these services, the system reduces manual data entry and enhances decision-making support.

Additionally, the application utilizes an automation workflow module powered by n8n. This module facilitates automated processes such as data handling, API integration, and background task execution, contributing to improved system performance and scalability.

In summary, the system follows a continuous data flow where user inputs are processed by the frontend, forwarded to the backend, stored in the database, and further enhanced by external services and automated workflows before being delivered back to the user as meaningful information.

4.2.System Components Specifications

This section describes the main components of the proposed Money Tracking and Management Application. The system is designed using a modular architecture consisting of frontend, backend, database, external services, and automation workflows. Each component plays a specific role in ensuring the functionality, scalability, and efficiency of the system.

4.2.1. Frontend (Mobile Application)

The frontend of the system is developed using Flutter, which provides a responsive and cross-platform mobile application interface. The frontend is responsible for handling user interactions, collecting input data, and displaying processed information to users.

The main functionalities of the frontend include:

- Dashboard overview for financial summary
- Add and manage transactions
- Transaction history viewing
- Group management and expense splitting
- Currency conversion
- Receipt scanning using OCR
- AI-based financial insights and advice

The frontend communicates with the backend via secure HTTPS requests and displays real-time updates based on the processed data.

4.2.2. Backend Services (Supabase)

The backend services are implemented using Supabase, which provides a scalable and secure backend infrastructure.

The backend is responsible for:

- User authentication and session management
- Handling RESTful API requests
- Enforcing Row Level Security (RLS) for data protection
- Managing file storage (e.g., receipt images)
- Supporting real-time data synchronization

Supabase acts as the central processing layer that connects the frontend with the database and external services, ensuring secure and efficient data handling.

4.2.3. Database (PostgreSQL)

The system uses a PostgreSQL database managed by Supabase to store and organize all application data.

The main database tables include:

- **profiles** – stores user account information
- **transactions** – records income and expenses
- **groups** – manages group information
- **group_members** – stores group participant details
- **group_transactions** – records group expenses
- **group_transaction_splits** – handles expense splitting data
- **chat_messages** – supports group communication

This structured database design ensures data consistency, supports relational queries, and enables efficient retrieval of financial information.

4.2.4. External Services

The system integrates external services to enhance functionality and automate complex processes.

OCR Service

- The Optical Character Recognition (OCR) service is used to extract transaction data from receipt images. It allows users to scan receipts and automatically convert them into structured financial records.

AI Service

- The AI service provides financial insights and personalized recommendations based on user transaction data. It helps users analyze spending patterns and improve financial planning.

These services are accessed through the backend to ensure secure communication and centralized processing.

4.2.5. Automation Workflows (n8n)

The system incorporates n8n as an automation workflow engine.

The automation module is responsible for:

- Workflow automation and task scheduling
- Data processing and transformation
- Integration with external APIs
- Supporting AI and OCR processing pipelines

n8n enhances the system by enabling automated background operations, reducing manual processing, and improving system efficiency.

4.3.Circuits and Components Design

This section describes the internal design of the system components and how different modules are structured to perform specific functionalities. Since the proposed system is a software-based mobile application, the design focuses on functional modules and their logical operations rather than physical circuit design. The system is divided into several modules, each responsible for handling a specific part of the application workflow.

4.3.1 User Authentication Module

The user authentication module is responsible for managing user login and registration processes. When a user inputs their credentials, the system validates the data format before sending a request to the backend authentication service. If the credentials are correct, the system retrieves user data from the database and redirects the user to the main interface. Otherwise, an error message is displayed.

During registration, the system performs validation checks such as email format verification, password validation, and duplicate account detection before storing the user data securely in the database.

4.3.2 Transaction Management Module

The transaction management module allows users to record and manage their financial activities. Users are required to input transaction details such as type (income or expense), category, and amount. Optional fields such as description and receipt image can also be included.

Before storing the data, the system validates all required fields. Once validated, the transaction data is processed and stored in the database, and the updated information is reflected on the dashboard and transaction history page.

Additionally, this module supports OCR functionality, allowing users to extract transaction details automatically from receipt images, improving efficiency and reducing manual input.

4.3.3 OCR Processing Module

The OCR processing module enables automatic extraction of data from receipt images. When the user uploads or captures a receipt image, the system sends the image to the OCR API for text recognition.

The OCR service processes the image and returns extracted text data to the system. The system then identifies relevant financial information, such as total amount, and displays it to the user for verification before saving it as a transaction record.

This module improves data entry efficiency, although its accuracy may depend on image quality and formatting conditions.

4.3.4 Group Management and Expense Splitting Module

The group management module allows users to create and manage groups for shared financial activities. Users can create groups by providing necessary information and adding members.

Within each group, users can record shared expenses using two methods:

- Equal split, where the total amount is divided equally among members
- Manual split, where users assign custom amounts to each member

The system ensures that the total distributed amount matches the original transaction value before saving the data. All group-related transactions are stored in the database and displayed in the group overview interface.

This module enhances collaborative financial management and ensures accurate expense tracking among group members.

4.3.5 Currency Converter Module

The currency converter module enables users to convert monetary values between different currencies. The system retrieves real-time exchange rates from an external API.

When a user inputs an amount and selects the desired currencies, the system processes the conversion and displays the result instantly. This module enhances usability for users managing finances across different currencies.

4.3.6 AI Insight, Chatbot, and Automation Module (n8n)

The AI insight and chatbot module is integrated with an automation workflow system using n8n to provide intelligent financial analysis and dynamic user interaction. This combined module enables the system to process user queries, retrieve relevant data, and generate meaningful responses efficiently.

When a user interacts with the chatbot or requests financial insights, the system sends the request to the backend, which triggers an n8n workflow. The workflow is responsible for coordinating multiple processes, including retrieving transaction data from the database, processing the data, and communicating with the AI service.

The AI service analyses the retrieved financial data and generates personalized insights, such as spending patterns, budgeting recommendations, and financial summaries. The generated response is then returned through the n8n workflow back to the application and displayed to the user.

In addition to generating insights, the chatbot supports natural language interaction, allowing users to ask questions related to their financial data. The system ensures that responses are context-aware by incorporating user-specific transaction history.

Furthermore, the module includes a response control mechanism. If a query is outside the supported scope or lacks sufficient data, the system returns a controlled or refusal response to prevent inaccurate or misleading information.

By integrating AI processing with n8n automation workflows, the system achieves a more flexible and scalable architecture. This design reduces the need for complex backend coding while enabling efficient data processing, seamless API integration, and real-time interaction between system components.

4.4.System Components Interaction Operations

This section describes how different components of the system interact during operation. Each function follows a structured workflow where user input is processed through the frontend, backend, database, and external services before returning results to the user.

4.4.1 User Authentication

- User enters login or registration details
- Input data is validated
- Request sent to backend (Supabase Authentication)
- Backend verifies credentials with database
- If valid → user data retrieved
- User redirected to dashboard
- If invalid → error message displayed

4.4.2 Transaction Recording

- User inputs transaction details in mobile application
- Data is validated
- Request sent to backend API
- Backend processes and stores data in database

- Database updates transaction records
- Updated data returned to frontend
- Dashboard and transaction history are updated

4.4.3 OCR Receipt Scanning

- User selects OCR scan function
- User captures or uploads receipt image
- Image sent to backend
- Backend forwards image to OCR service
- OCR processes and extracts text data
- Extracted data returned to backend
- Backend sends result to frontend
- User reviews and confirms data
- Data stored in database

4.4.4 Group Expense Management

- User creates or selects a group
- Group data sent to backend and stored in database
- User inputs group expense details
- System applies split method (equal/manual)
- Backend validates split calculation
- Data stored in database
- Updated group expense displayed to members

4.4.5 Currency Converter

- User inputs amount and selects currencies
- Request sent to backend

- Backend retrieves exchange rate from external API
- Backend calculates converted value
- Result returned to frontend
- Converted amount displayed

4.4.6 AI Insight and Chatbot Interaction

- User selects quick question or inputs query
- Request sent to backend
- Backend triggers n8n workflow
- n8n retrieves user data from database
- Data sent to AI service
- AI processes and generates response
- Response returned through backend
- Result displayed to user
- If invalid query → controlled response returned

4.4.7 Offline Mode Interaction

- System detects loss of internet connection
- Application switches to offline mode
- User is notified of offline status
- Only manual transaction recording is enabled
- User inputs transaction data
- Data validated locally
- Data stored in local storage
- System monitors network status
- When connection is restored:
    - Local data sent to backend
    - Data synchronized with cloud database

- Full system functionality restored

CHAPTER 5 System Implementation

5.1.Hardware Setup

The primary gear for this project is laptop PCs and Android mobile devices. The laptops will be used for a number of key tasks in this project, such as setting up the development environment, writing and debugging the application code, integrating the front-end and back-end development, designing the user interface (UI) and user experience (UX), deploying the test server locally and managing the database system (by using Supabase).

Android mobile devices, on the other hand, are mainly used for operational testing of the application, validation of the user interaction experience, and actual testing of the receipt scanning functionality. Mobile device testing ensures the compatibility, responsiveness, and functional integrity of the application in a real environment, especially the ability to interact with hardware resources such as device cameras, file storage and network connections.

Taken together, these two types of hardware devices complement each other and together form an indispensable support platform for the project development, testing and verification process. Below is a table that shows the specifications of laptop and mobile device that will be used in this project:

Table 5.1.1 Specifications of laptop

Description	Specifications
Model	Acer Nitro 5
Processor	AMD Ryzen 5 5600H with Radeon Graphics
Operating System	Windows 11
Graphic	NVIDIA GeForce GTX 1650
Memory	16GB RAM
Storage	475GB SATA HDD

Table 5.1.2 Specifications of Mobile Device

Description	Specifications

Device Name	Realme C55
Brand	Realme
Model	RMX3710
Processor	Octa-core
Operating System	Android 14.0 Realme UI 5.0
Chipset	Meduatek MT6769 Helio G88 (12 nm)
Graphic	Mali-G52 MC2
Memory	8GB(Fixed) + 8 GB(Virtual Memory)
Storage	256GB SSD
Main Camera (Dual)	64 MP (wide), ½.0”,PDAF 2 MP, (depth) LED flash, HDR, panorama

5.2.Software Setup

Below are the software or tool components that will be downloaded and installed for use in this project development.

- **Flutter Framework**

Figure 5.2.1: Flutter Logo

Flutter is an open-source software UI toolkit that was developed at Google. It lets software professionals create natively compiled, multi-platform software programs from the same codebase for mobile, web, and desktop [18]– [20]. The framework is efficient in crafting responsive and visually appealing user interfaces with the aid of its "hot reload" support, rich customizable widgets, and efficient rendering [18], [19]. The Flutter framework was chosen as the main framework in this project in order to develop the mobile application because of its uniform performance across multiple platforms.

- **Dart Programming Language**

Figure 5.2.2: Dart programming Logo

The client-optimized, object-oriented programming language Dart was developed by Google.[21], [22]. The language is of strong type, allows asynchronous code through async and await features, and is aimed at high- performance UI building [21], [23]. Flutter applications are in the Dart language,

using its syntax and performance properties in order to put the application's logic, data processing, as well as interaction with the external APIs.

- **Visual Studio Code**

Figure 5.2.3: Visual Studio Code Logo

Microsoft created Visual Studio Code (VS Code), a lightweight, open-source code editor.[30]. The code editor is compatible with different programming languages and is customizable through various extensions that increase productivity, like the Flutter and Dart plugins. VS Code was chosen in this project as the core code editor during the development and management of the Flutter and Dart code because of its lightness, immediate feedback, and strong extension system.

- **Android Studio**

Figure 5.2.4: Android Studio Logo

Created particularly for Android, Android Studio is the official Integrated Development Environment (IDE) for Android software development. It is based on JetBrains' IntelliJ IDEA.[27], [28]. The software hosts powerful functionalities such as an emulator, debuggers, UI editing programs, and support for Flutter plugins with ease [27], [29]. Android Studio was mainly used in the part of its emulator, which was connected with visual studio code so that the application was effectively debugged and tested.

- **ML Kit (OCR API)**

Figure 5.2.5: ML Kit Logo

ML Kit is a mobile software development kit provided by Google that enables on-device and cloud-based machine learning capabilities, including text recognition, image

labeling, and object detection **[31]**. One of its most prominent features is the Text Recognition (OCR) API, which is capable of detecting and extracting text from images such as documents and receipts with near real-time processing **[32]**. In this project, the ML Kit OCR API is utilized to automate the scanning of receipts and convert extracted text into structured transaction data, thereby improving data entry efficiency and reducing manual input errors.

- **Supabase**

Figure 5.2.6: Supabase Logo

Supabase is an open-source **Backend-as-a-Service (BaaS) platform** that enables developers to build scalable applications without managing backend infrastructure. It is built on PostgreSQL and provides a fully managed relational database along with integrated features such as authentication, storage, and real-time data synchronization **[25]**. One of its key advantages is the ability to automatically generate **RESTful APIs** from the database schema, allowing efficient data interaction without additional backend coding **[26]**. Supabase also includes a built-in authentication system that supports multiple login methods and uses JSON Web Tokens (JWT) together with Row Level Security (RLS) to ensure secure data access **[25]**. In addition, its real-time engine allows applications to receive live updates by subscribing to database changes, improving responsiveness and user experience **[26]**.

- **Figma**

Figma

Figure 5.2.7: Figma Logo

Figma is a collaborative UI/UX design and prototyping tool that enables designers to create, refine, and test interface designs efficiently. It supports real-time collaboration and cloud-based design workflows, allowing multiple users to work on the same project simultaneously **[40]**. In this project, Figma was mainly used to **enhance and refine the initial UI layouts, improving visual hierarchy, consistency, and usability**. It also **enables the creation of interactive prototypes** to simulate user interactions and navigation flows, ensuring a more user-friendly design **[41]**.

- **Google Stich**

Stitch

Figure 5.2.8: Figma Logo

Google Stitch is an **AI-powered UI design tool developed by Google** that enables users to generate interface designs quickly using natural language prompts, images, or structured inputs such as Markdown (MD). In this project, Google Stitch was used to **generate the initial UI design concepts**, allowing rapid exploration of layouts without starting from scratch. To improve the quality of the generated designs, **GPT was used to produce structured Markdown (MD) descriptions**, which provided clearer design requirements and better guidance for Stitch to generate more accurate and consistent UI outputs. This approach improves prompt clarity, which is important because Stitch relies heavily on well-defined inputs to produce high-quality interfaces [40], [41]. However, since the generated designs are not always fully detailed or optimized, they were further refined using Figma to ensure better structure and usability. Overall, Google Stitch, supported by GPT-generated MD prompts,

significantly **accelerated the UI ideation process and improved design efficiency** [41].

- **ChatOpenAI**

Figure 5.2.9: ChatGPT Logo

ChatGPT is a **generative artificial intelligence tool developed by OpenAI** that is capable of generating text, code, and design ideas based on natural language input. In this project, the ChatGPT Go version was used to **assist in idea generation, system design improvement, and UI development planning**, helping to refine the overall structure and functionality of the application. It was also utilized to **generate structured Markdown (MD) content**, which was later used as input for Google Stitch to produce more accurate and higher-quality UI designs. In addition, ChatGPT's image generation capabilities were used to **assist in designing the project logo**, allowing rapid exploration of visual concepts. As a generative AI tool, ChatGPT can produce written content, code, and creative outputs efficiently, making it highly useful in modern software development workflows [29], [30]. Overall, it played a supportive role in improving productivity, enhancing design quality, and accelerating the development process.

- **N8N**

Figure 5.2.10: ML Kit Logo

n8n is a **workflow automation platform** that enables the integration of multiple services and systems through a visual, node-based interface. In this project, n8n was utilized within the chatbot module to **connect AI models with the Supabase database**, allowing automated data interaction such as retrieving user data and storing new information generated by the chatbot. By using n8n workflows, different components such as AI processing, database queries, and

response handling can be linked together efficiently through triggers and actions, reducing the need for manual backend coding [33], [34]. This integration enables the chatbot to dynamically **post and fetch data from the database in real time**, improving system responsiveness and functionality. Overall, n8n provides a flexible and scalable solution for implementing AI-driven automation and seamless communication between the application and backend services.

- **GitHub**

Figure 5.2.11: GitHub Logo

GitHub is a **web-based platform for version control and collaboration** that allows developers to manage and track changes in their code using Git. In this project, GitHub was used to **manage the application versioning**, where different stages of development were saved as commits, enabling changes to be tracked, compared, and reverted when necessary. This helps maintain code integrity and supports efficient development by allowing continuous updates without losing previous versions [30], [33]. In addition, GitHub was also used as a **research resource**, where public repositories were explored to study existing project structures, features, and implementation approaches. This provided useful insights and inspiration for improving the system design and functionality of the application. Overall, GitHub plays an important role in supporting both **version management and knowledge sharing** in modern software development [33].

5.3.Setting and Configuration

The system configuration for the KeepIt Money Tracking and Management Application involves the integration of frontend, backend, and automation components to ensure seamless operation and data processing. The development environment was set up using the Flutter framework, Supabase backend services, and n8n workflow automation.

For the frontend, the Flutter framework was configured using Visual Studio Code. The project structure is organized into multiple modules, including screens, services, widgets, and utility files, to ensure maintainability and modular design. External dependencies required for system functionality were defined in the *pubspec.yaml* file, including libraries for OCR processing, API communication, local storage, and state management.

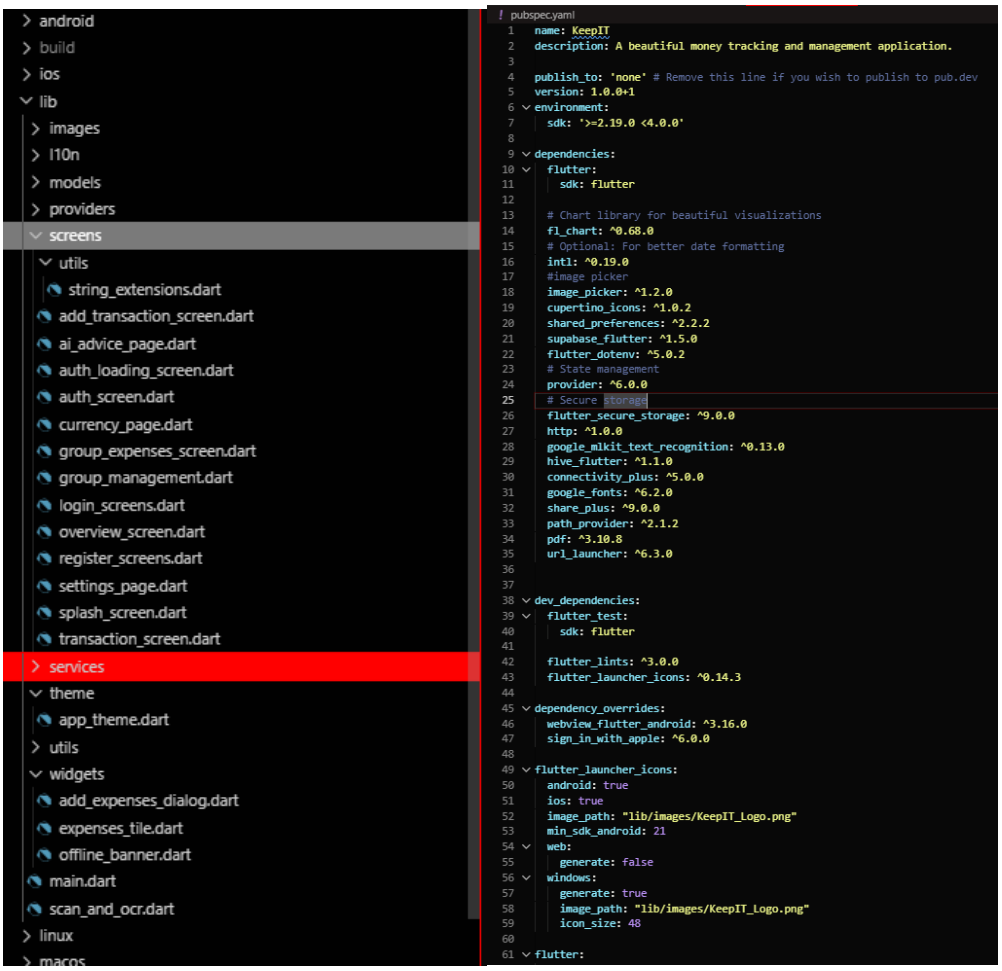


Figure 5.3.1: Project Structure and Dependency Configuration

For backend configuration, Supabase was used as the Backend-as-a-Service (BaaS) platform. The database schema was designed to support core functionalities such as user management,

transaction tracking, group expense handling, and chatbot interaction. The main database tables include transactions, groups, group transactions, group members, and chat messages, which collectively support the system's data flow.

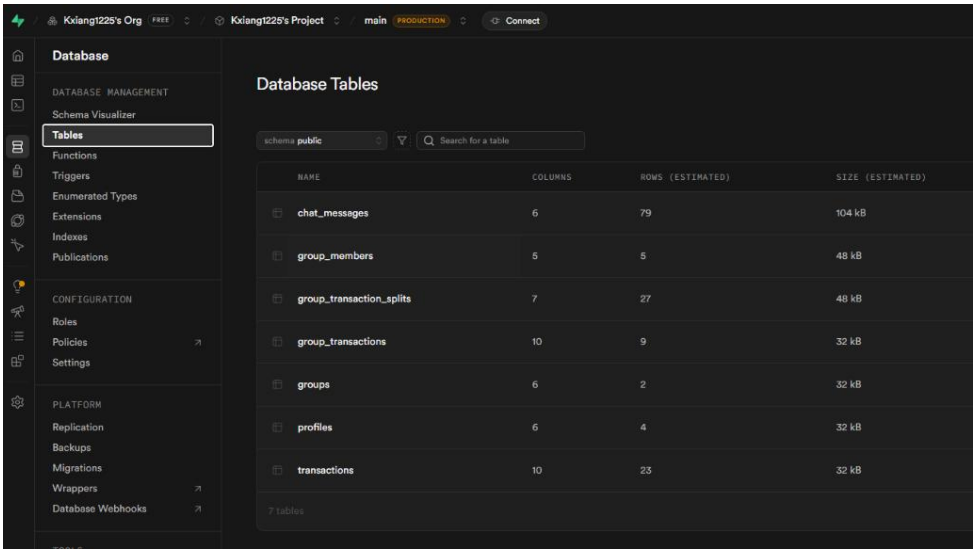


Figure 5.3.2: Supabase Database Tables Configuration

In addition to database configuration, Supabase authentication services were enabled to manage user login and registration. The system supports multiple authentication providers, including email-based login and Google authentication. Each user is assigned a unique identifier, which is used to link user data across different modules of the system.

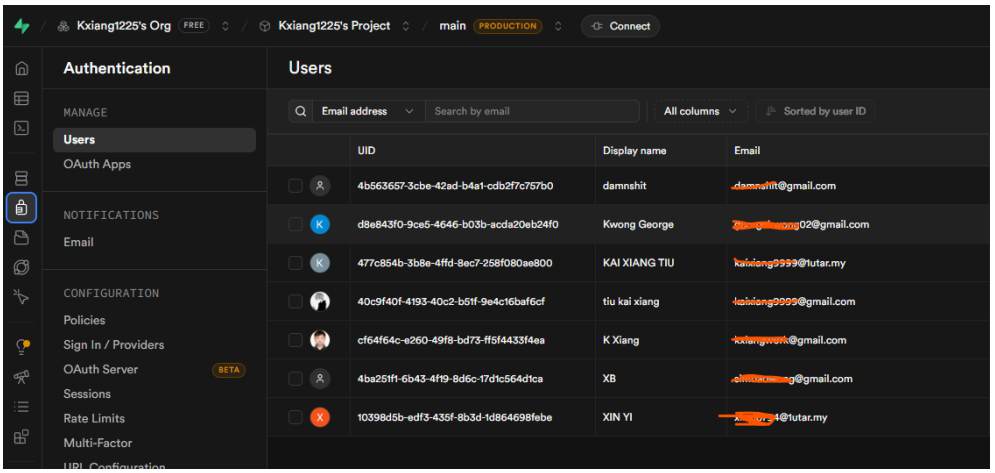


Figure 5.3.3: Supabase Authentication and User Management

To support the AI chatbot functionality, n8n was configured as a workflow automation tool. The workflow is triggered through a webhook when a user submits a query via the chatbot interface. The workflow retrieves transaction data from the Supabase database, processes the data, and sends it to the AI model for analysis. The generated response is then returned to the application.

The workflow consists of multiple nodes, including data retrieval, data processing, AI interaction, and response handling, which enables seamless communication between system components.

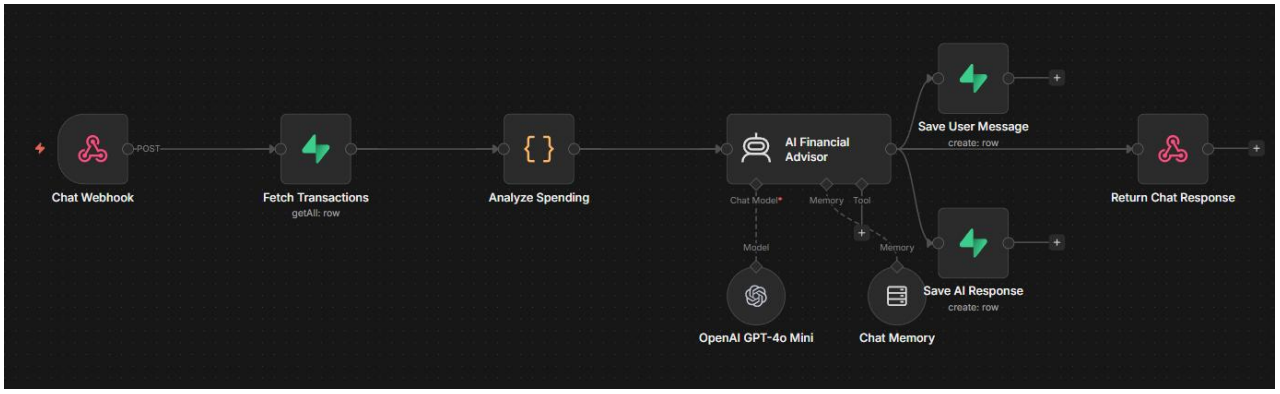
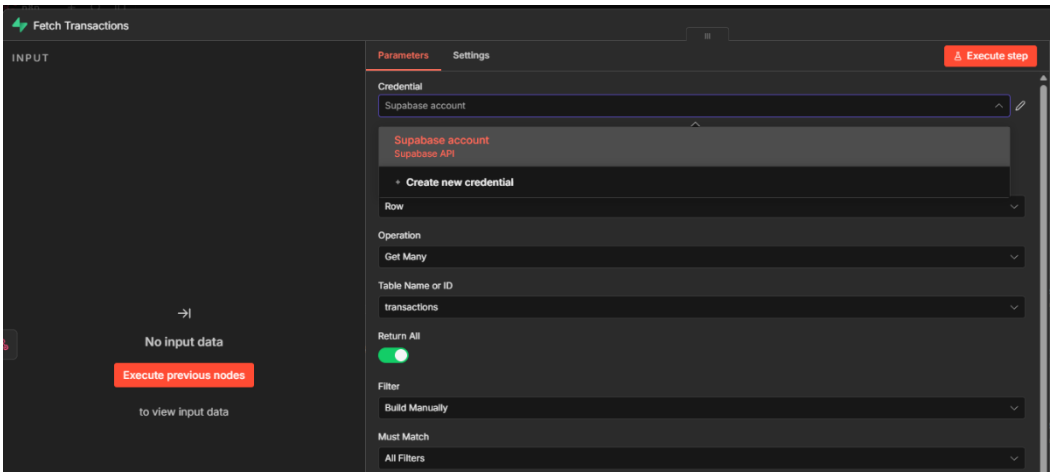


Figure 5.3.4: n8n Workflow for AI Chatbot Integration

The Supabase integration within the n8n workflow was configured using API credentials to allow secure access to the database. The workflow retrieves transaction data dynamically, ensuring that the AI chatbot provides responses based on real user data.



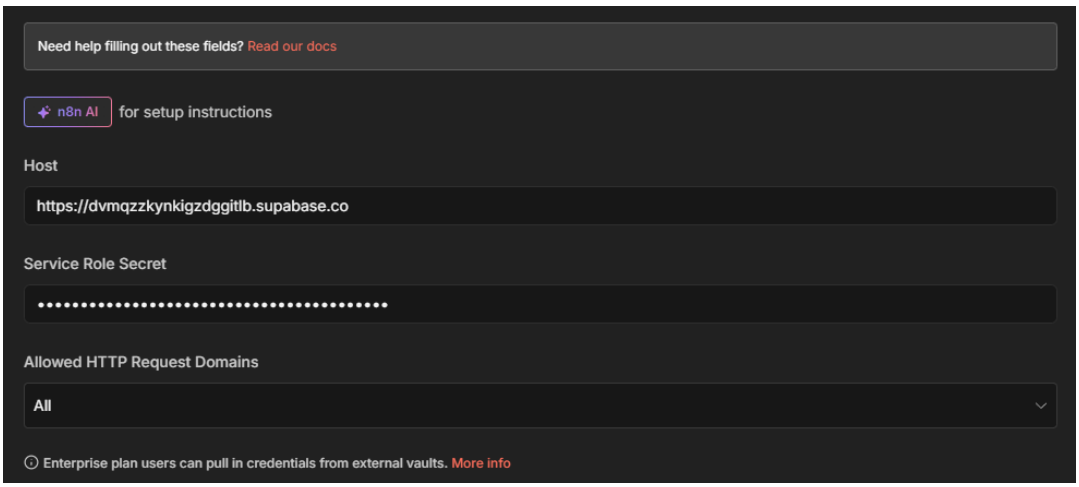


Figure 5.3.5: Supabase Data Retrieval Configuration in n8n

The AI module was configured using the OpenAI GPT-4o Mini model within the n8n workflow. The system prompt was designed to ensure that the chatbot provides financial advice based on user transaction data. Parameters such as temperature and token limits were configured to control response quality and consistency.

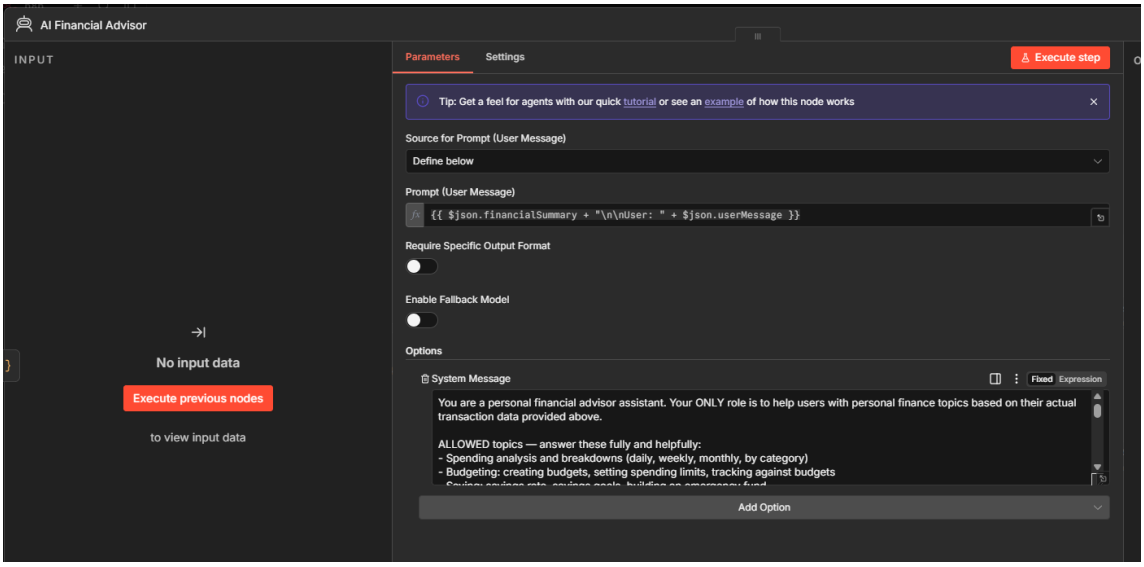


Figure 5.3.6: AI Chatbot Prompt Configuration

Table 5.3 AI Chatbot Prompt Setup

You are a personal financial advisor assistant. Your ONLY role is to help users with personal finance topics based on their actual transaction data provided above. ALLOWED topics — answer these fully and helpfully: - Spending analysis and breakdowns (daily, weekly, monthly, by category) - Budgeting: creating budgets, setting spending limits, tracking against budgets - Saving: savings rate, savings goals, building an emergency fund - Cash flow: income vs expenses, net savings, trends over time - Debt management: paying off loans, credit card debt strategies - Financial habits: overspending patterns, impulse spending, improving discipline - Bill and subscription management: identifying recurring costs, reducing unnecessary expenses When answering quick questions, use the relevant data section: - "spending this week" → use the THIS WEEK section - "overspending" or "category" → use the CATEGORY OVERSPENDING section - "saving tips" → use the FINANCIAL SUMMARY and ALERTS sections - "bills" or "coming soon" → use the RECURRING BILLS & SUBSCRIPTIONS section FORBIDDEN — refuse these firmly but politely: - Investment advice of any kind (stocks, crypto, ETFs, funds, bonds, real estate investing, trading) - Tax advice or tax planning - Insurance recommendations - Legal or accounting advice - Any topic unrelated to personal finance (cooking, travel, health, technology, relationships, general knowledge, etc.) If the user asks about a FORBIDDEN topic or anything outside personal finance, respond with exactly: "I'm only able to help with personal finance topics such as budgeting, saving, spending analysis, and debt management. For [topic they asked about], please consult a qualified professional." Do not engage with off-topic questions even partially. Do not answer and then redirect — just redirect immediately. Be encouraging, empathetic, and practical. Always reference specific numbers from the financial summary when relevant.

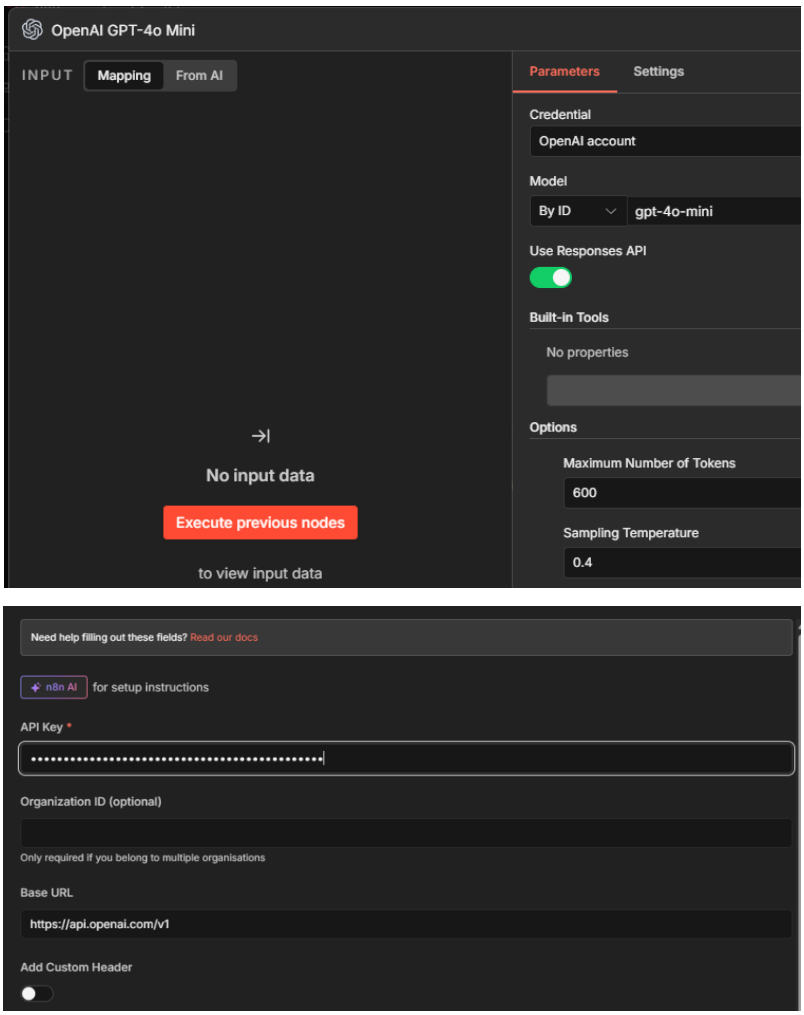


Figure 5.3.7: OpenAI Model Configuration in n8n

Overall, the system configuration integrates Flutter, Supabase, and n8n into a unified architecture. This setup ensures efficient data handling, real-time synchronization, and scalable system performance. The modular configuration also allows future enhancements to be implemented with minimal changes to the existing system.

5.4.System Operation

5.4.1. Splash Screen (Starting Screen)

This figure illustrates the splash screen of the KeepIT application, which is displayed when the user launches the app. The splash screen serves as the initial interface, providing a visually appealing introduction to the system through branding elements such as the application logo and theme design. It enhances the user experience by creating a smooth transition while the system initializes essential resources in the background.

Additionally, the splash screen includes a brief welcome message to engage users upon entry. This feature is optional and can be enabled or disabled through the settings page, allowing users to customize their experience based on personal preference. By providing both aesthetic value and functional flexibility, the splash screen contributes to a more user-friendly and customizable application startup process. During this stage, the system also performs background initialization processes, such as loading user preferences and preparing essential application data, ensuring a smooth transition to the main interface.

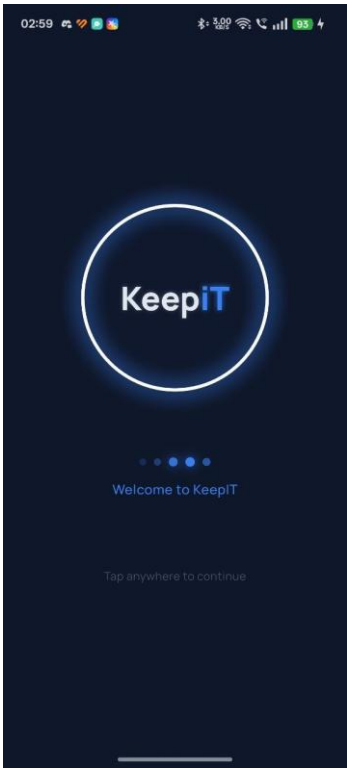


Figure 5.4.1 Splash Screen Overview

5.4.2. Overview Screen(Home Page)

The overview screen serves as the main dashboard of the application, providing users with a summarized view of their financial status. As shown in Figure 5.1.2.1, key financial indicators such as *Current Balance, Total Income, and Total Expense* are displayed at the top section for quick reference. This allows users to immediately understand their financial position upon opening the application.

In addition, a visual representation in the form of a daily bar chart is presented in the middle section. This chart enables users to analyse their spending patterns and income trends over time, improving financial awareness and decision-making. Beneath the chart, a compact transaction history list displays the most recent three transactions, allowing users to quickly review their latest activities without navigating to another page.

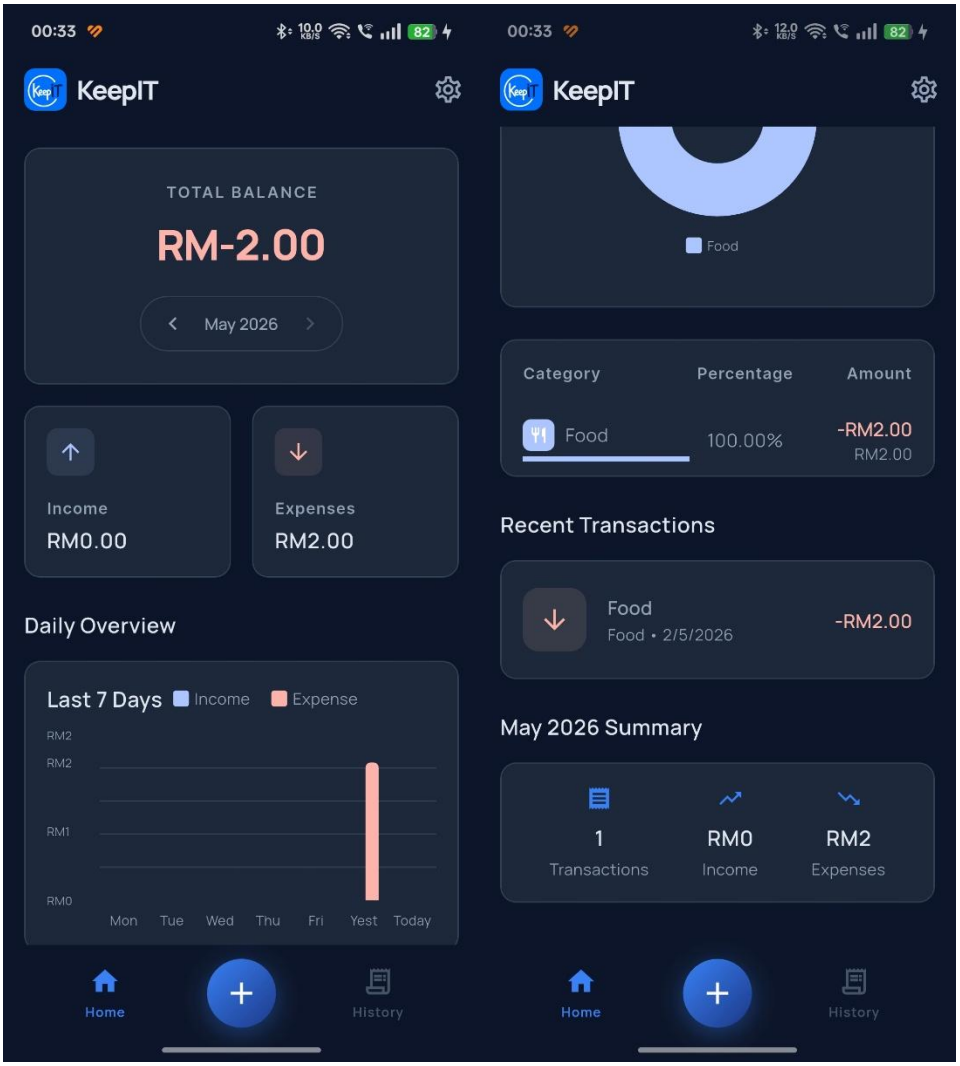


Figure 5.4.2.1: Overview Screen

Furthermore, the quick action button located at the centre of the screen expands into a shortcut menu (Figure 5.4.2.2), enabling users to efficiently access major modules such as adding transactions, managing groups, currency conversion, and AI insights. This design improves usability by reducing navigation complexity.

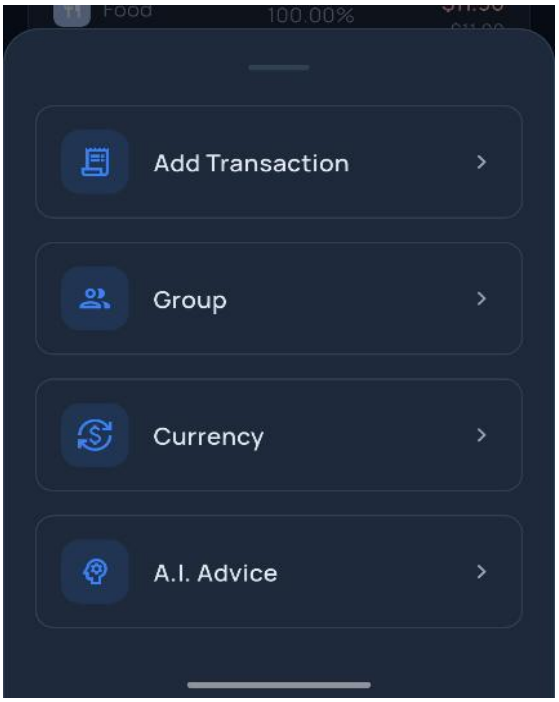


Figure 5.4.2.2 Quick Action Bar

5.4.3. Login & Registration Screen

In the settings page, the sign-in requirements will appear at the top. It will allow users to sign in when they want. (Figure 5.4.3.1)

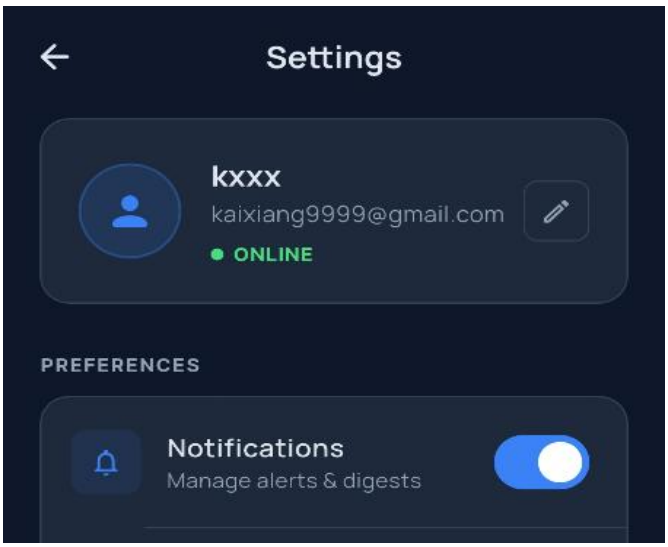


Figure 5.1.3.1: Sign In Required Button

The login and registration module ensures secure access control using Supabase Authentication. As illustrated in Figure 5.4.3.2, users are required to enter their registered email address and password. The system validates these credentials by sending a request to the Supabase authentication service.

If the credentials are valid, the system successfully authenticates the user and retrieves their corresponding financial data from the database, redirecting them to the main interface. Otherwise, appropriate error messages are displayed (Figure 5.4.3.4), ensuring users are informed of incorrect inputs.

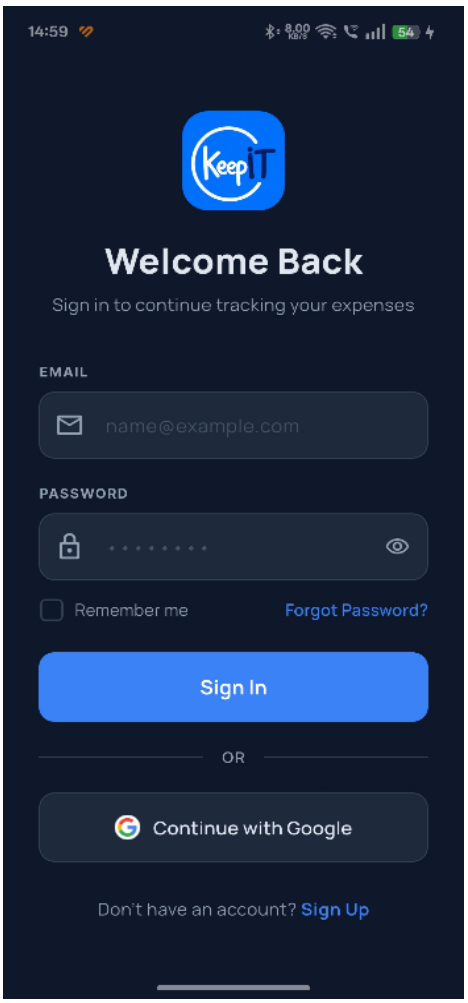


Figure 5.4.3.2: Login Page

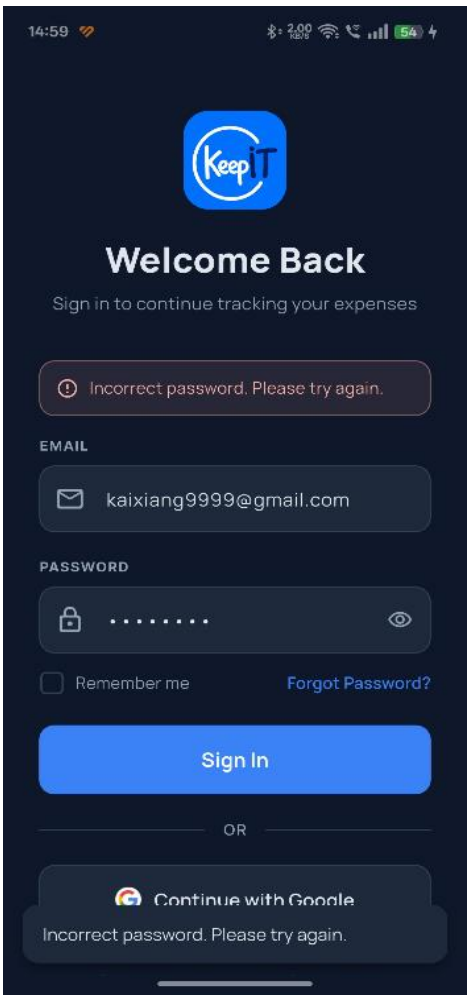
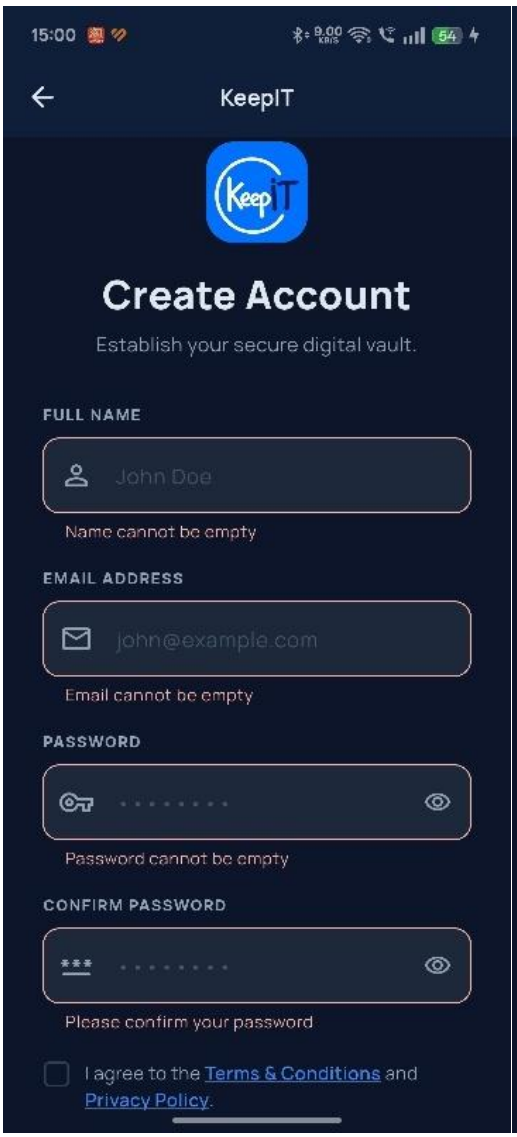


Figure 5.4.3.4 Error Message



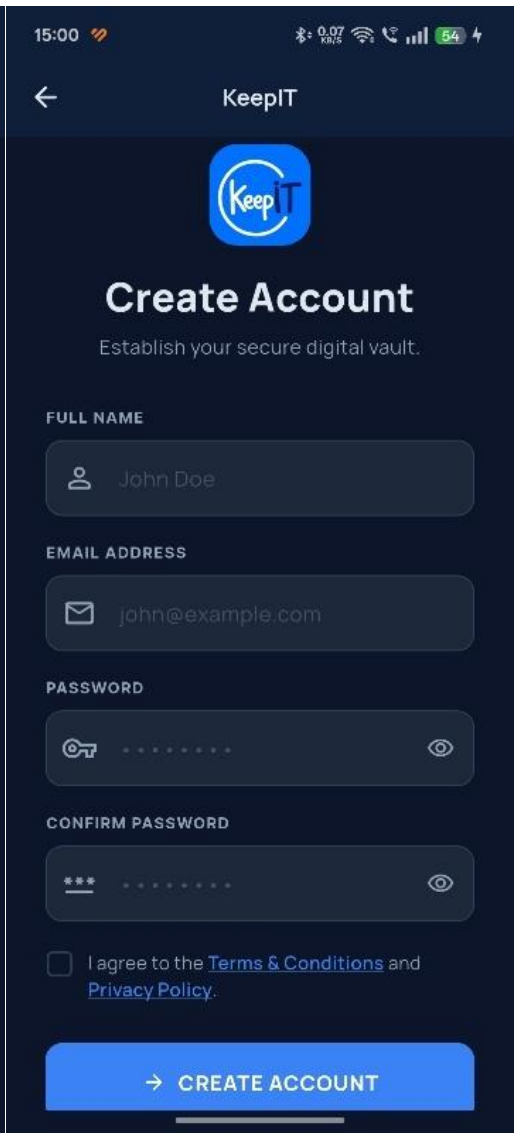
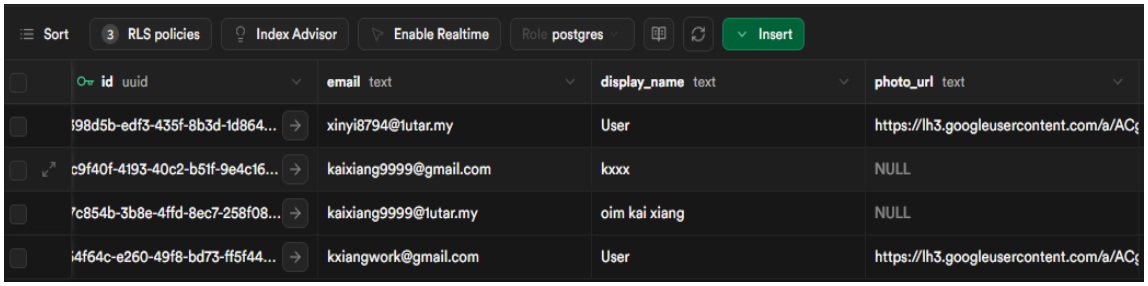


Figure 5.4.3.5: Register Form Figure 5.4.3.6: Error message2

For new users, the registration process (Figure 5.4.3.5) requires all mandatory fields to be completed. The system performs validation checks such as:

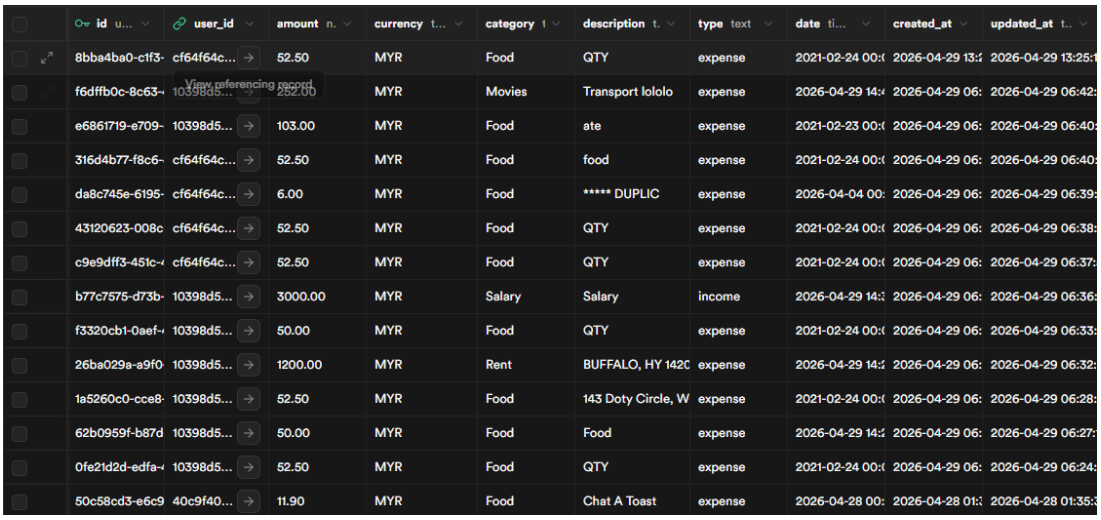
- Email format verification
- Password length and format validation
- Duplicate email detection

If validation fails, error messages are shown (Figure 5.4.3.6). Upon successful registration, user data is securely stored in both Supabase Authentication and the database (Figure 5.4.3.7 & 5.4.3.9). This ensures that user credentials and transaction-related data are properly managed and synchronized.



id uuid	email text	display_name text	photo_url text
98d5b-edf3-435f-8b3d-1d864...	xinyi8794@1utar.my	User	https://lh3.googleusercontent.com/a/AC
c9f40f-4193-40c2-b51f-9e4c16...	kaixiang9999@gmail.com	kxxx	NULL
7c854b-3b8e-4ffd-8ec7-258f08...	kaixiang9999@1utar.my	oim kai xiang	NULL
4f64c-e260-49f8-bd73-ff5f44...	kxiangwork@gmail.com	User	https://lh3.googleusercontent.com/a/AC

Figure 5.4.3.7 Account successfully created and stored in Supabase Database



id u...	user_id	amount n.	currency t...	category t	description t.	type text	date ti...	created_at	updated_at t..
8bba4ba0-c1f3-	cf64f64c...	52.50	MYR	Food	QTY	expense	2021-02-24 00:(	2026-04-29 13:2	2026-04-29 13:25:1
f6dffb0c-8c63-	10398d5...	252.00	MYR	Movies	Transport lololo	expense	2026-04-29 14:4	2026-04-29 06:	2026-04-29 06:42:
e6861719-e709-	10398d5...	103.00	MYR	Food	ate	expense	2021-02-23 00:(	2026-04-29 06:	2026-04-29 06:40:
316d4b77-f8c6-	cf64f64c...	52.50	MYR	Food	food	expense	2021-02-24 00:(	2026-04-29 06:	2026-04-29 06:40:
da8c745e-6195-	cf64f64c...	6.00	MYR	Food	***** DUPLIC	expense	2026-04-04 00:	2026-04-29 06:	2026-04-29 06:39:
43120623-008c	cf64f64c...	52.50	MYR	Food	QTY	expense	2021-02-24 00:(	2026-04-29 06:	2026-04-29 06:38:
c9e9dff3-451c-	cf64f64c...	52.50	MYR	Food	QTY	expense	2021-02-24 00:(	2026-04-29 06:	2026-04-29 06:37:
b77c7575-d73b-	10398d5...	3000.00	MYR	Salary	Salary	income	2026-04-29 14:3	2026-04-29 06:	2026-04-29 06:36:
f3320cb1-0aef-	10398d5...	50.00	MYR	Food	QTY	expense	2021-02-24 00:(	2026-04-29 06:	2026-04-29 06:33:
26ba029a-a9f0-	10398d5...	1200.00	MYR	Rent	BUFFALO, HY 1420	expense	2026-04-29 14:2	2026-04-29 06:	2026-04-29 06:32:
1a5260c0-cce8-	10398d5...	52.50	MYR	Food	143 Doty Circle, W	expense	2021-02-24 00:(	2026-04-29 06:	2026-04-29 06:28:
62b0959f-b87d	10398d5...	50.00	MYR	Food	Food	expense	2026-04-29 14:2	2026-04-29 06:	2026-04-29 06:27:
0fe21d2d-edfa-	10398d5...	52.50	MYR	Food	QTY	expense	2021-02-24 00:(	2026-04-29 06:	2026-04-29 06:24:
50c58cd3-e6c9	40c9f40...	11.90	MYR	Food	Chat A Toast	expense	2026-04-28 00:	2026-04-28 01:3	2026-04-28 01:35:3

Figure 5.4.3.8 Data stored in Supabase Database

After filling in all the required fields in the registration form, the application sends the user data to the Supabase backend for account creation. The system first checks whether the entered email already exists in the Supabase Authentication service. If the email is already registered, an error message will be returned to the user, prompting them to use a different email address. If the verification is successful, the user account is created and securely stored in the Supabase Authentication system (Figure 5.4.3.7).

In addition, the user's information and related data are stored in the Supabase database (Figure 5.4.3.8). This database is also responsible for storing user transaction records, such as income and expenses. When the user logs into the application, the system retrieves the stored data from Supabase and displays it on the user's homepage, ensuring that all financial information is up-to-date and easily accessible.

5.4.4. Add-Transaction Screen & Transaction History Screen

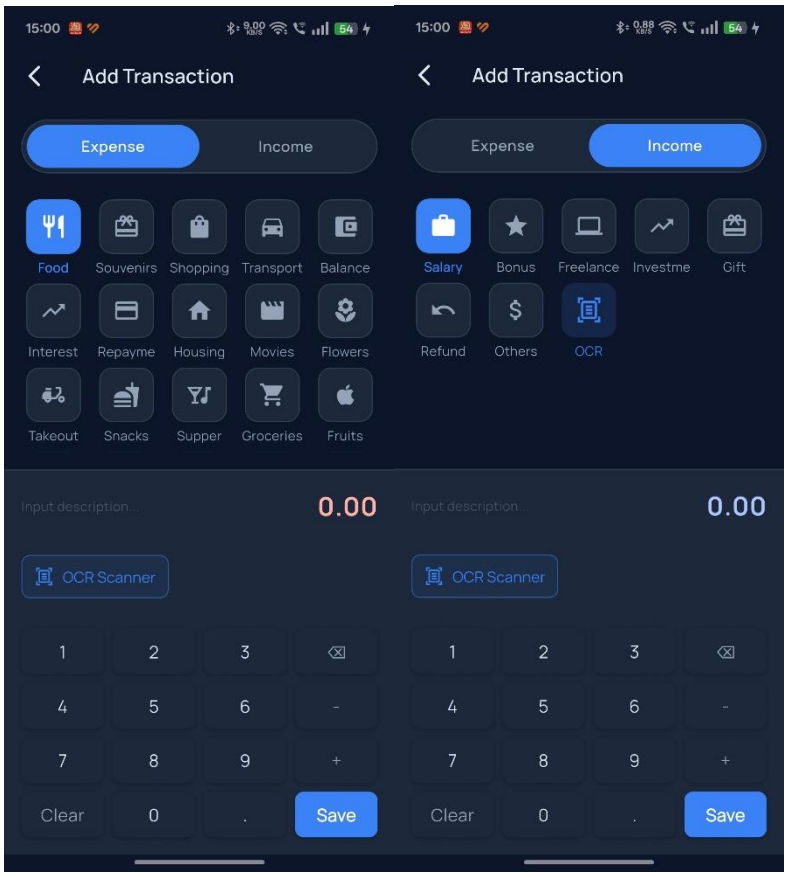


Figure 5.4.4.1 Income Categories & Expense Categories

This page allows the user to record their transaction by selecting the type of transaction (income or expense), the type of categories, details of the transaction and the amount of the transaction. (Figure 5.4.4.1) Additionally, this page also developed a mini calculator to help the user save time to total up the total amount of their transaction and provide the OCR functional button (Figure 5.4.4.2).

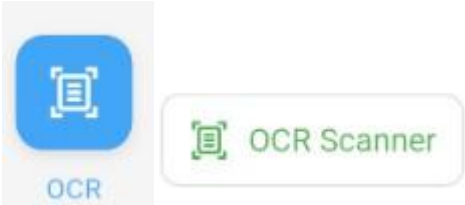
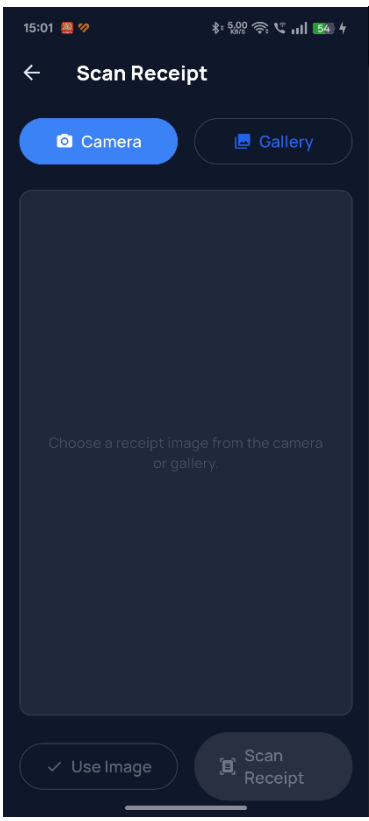


Figure 5.1.4.2 OCR Button

This will forward the user to the OCR page (Figure 5.4.4.3) that uses the API to capture or upload the image from the user's device (Figure 5.4.4.4). It will provide more efficiency for the transaction record by extracting the data inside the image. (Figure 5.4.4.3)



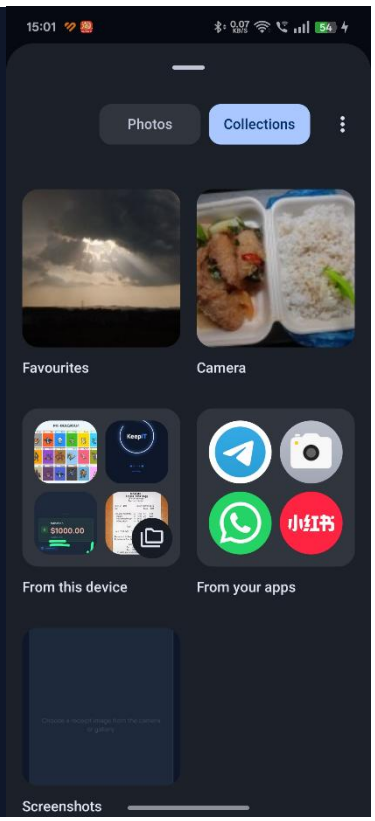


Figure 5.4.4.3 OCR Page Figure 5.4.4.4 Access to device storage

Figure 5.4.4.5 illustrates the OCR scanning result after processing a receipt image. The system utilizes an OCR API to extract textual information from the uploaded or captured image. The extracted data, such as transaction amount and relevant details, is then displayed to the user for verification.

To enhance extraction accuracy, a keyword detection mechanism is implemented during processing. The system searches for predefined keywords such as *"Total", "Total Amount", and "Amount"* to identify the most relevant financial value within the receipt. This reduces errors caused by irrelevant or noisy text.

However, the OCR accuracy may still be affected by factors such as image quality, lighting conditions, and receipt formatting. Therefore, users are allowed to manually review and edit the extracted data before saving it. This ensures that the final recorded transaction remains accurate and reliable.

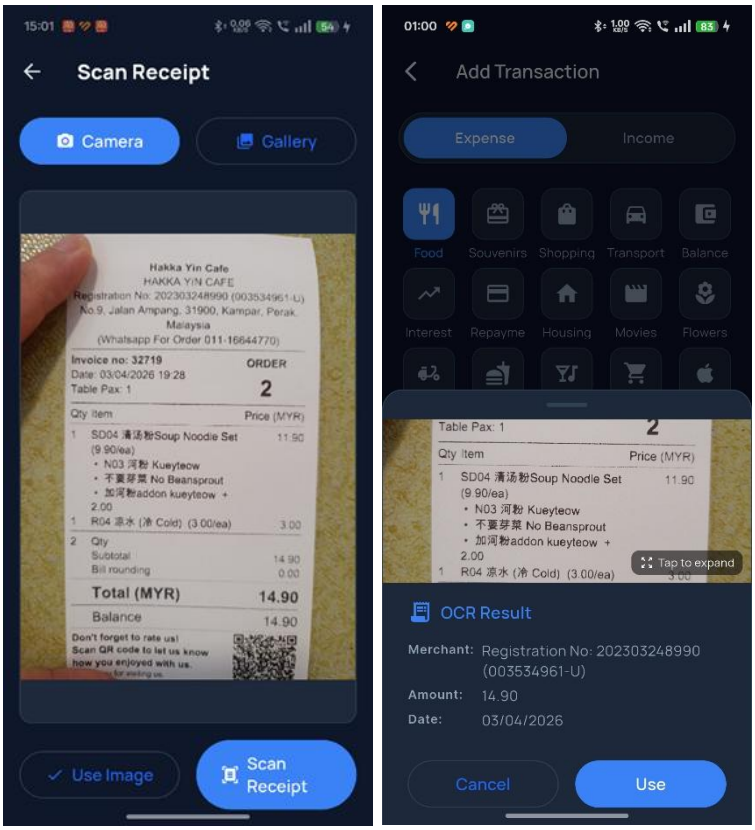


Figure 5.1.4.5 OCR Scanning Result

After a transaction is successfully recorded, it is displayed in the transaction history page (Figure 5.4.4.7 and Figure 5.4.4.8). This page provides a structured list of all user transactions, allowing users to review their financial activities in detail.

Figure 5.4.4.7 Transaction Page

Figure 5.4.4.8 Transaction Page Overview

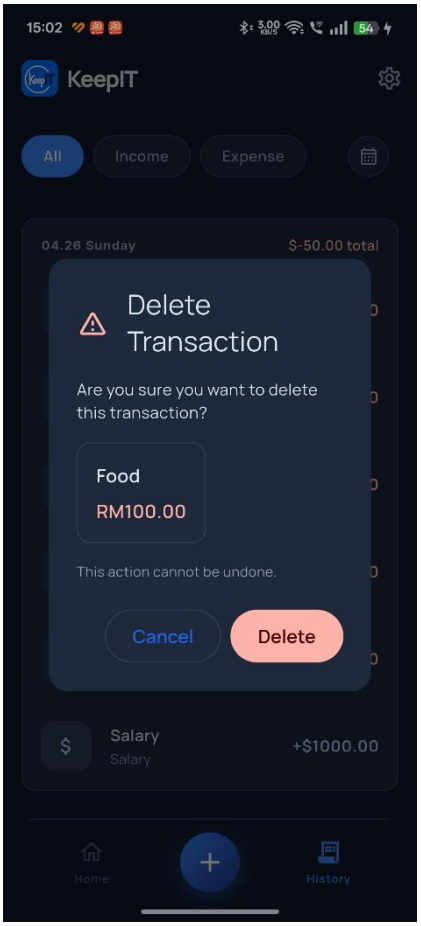


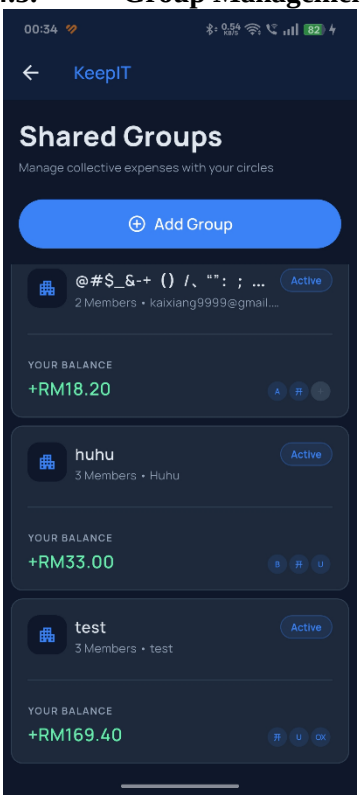
Figure 5.4.4.9 Delete Confirmation

Each transaction entry supports interactive operations:

- **Swipe left** → triggers delete function with confirmation (Figure 5.4.4.9)
- **Tap record** → opens edit mode with pre-filled data

These interaction features enhance usability by allowing users to easily manage and update their financial records. The double confirmation mechanism in deletion prevents accidental data loss, improving system reliability.

5.4.5. Group Management(GM) Screen & Group Screen



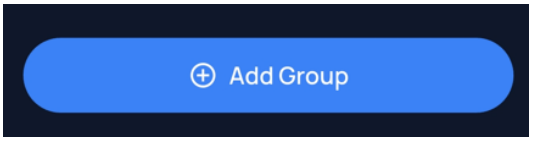


Figure 5.4.5.1 Overview GM Page

Figure 5.4.5.2 Add Group Button

Figure 5.4.5.1 and Figure 5.4.5.2 collectively illustrate the main interface of the Group Management (GM) module. The Overview GM Page displays a list of all groups that the user has created or joined, with each group representing a separate context for managing shared expenses, such as trips, events, or household budgeting. This structure allows users to organize financial activities efficiently and avoid mixing unrelated transactions.

Within this interface, the “Add Group” button serves as the primary interaction point for initiating the creation of a new group. When the user selects this button, the system redirects them to the group creation form. By integrating group overview and creation access within the same screen, the system enhances usability and provides a seamless workflow for managing multiple shared expense groups.

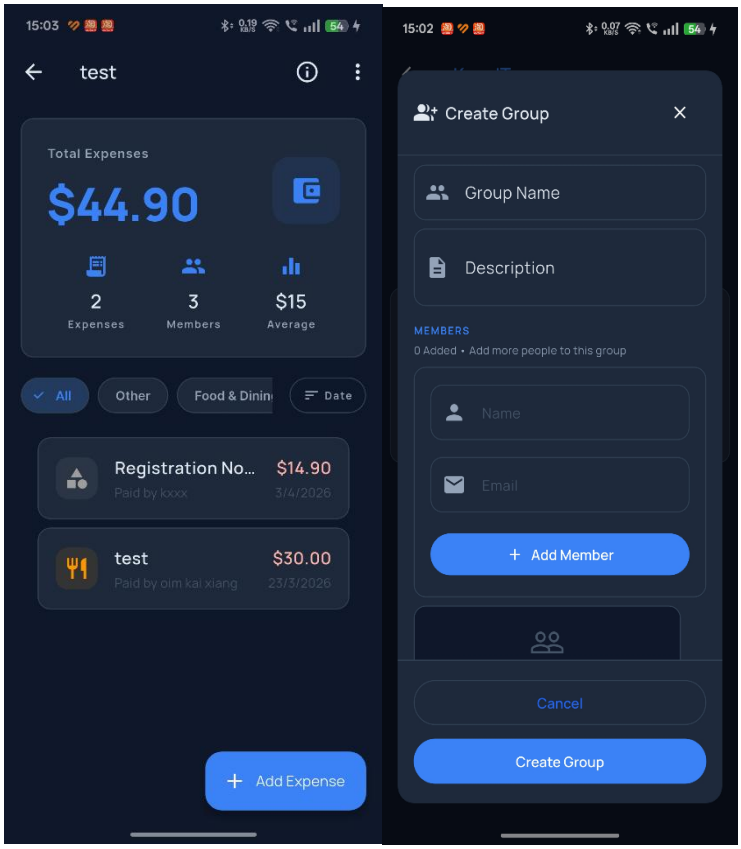


Figure 5.4.5.3 Create Group Form Figure 5.4.5.4 Group Overview

Figure 5.4.5.3 and Figure 5.4.5.4 demonstrate the process of creating and managing a group within the system. The Create Group Form requires users to input essential information such as group name, description, and member list. The system performs validation to ensure all required fields are completed before allowing the group to be created. Once submitted, the group data is stored in the database and linked to the respective users.

After successful creation, users are redirected to the Group Overview page, where detailed information about the group is displayed. This includes total expenses, number of members, average spending, and a list of recorded transactions. The Group Overview serves as the central hub for monitoring and managing group financial activities. By combining group creation and overview functionalities, the system ensures a smooth transition from setup to active financial tracking, improving overall user experience and operational efficiency.

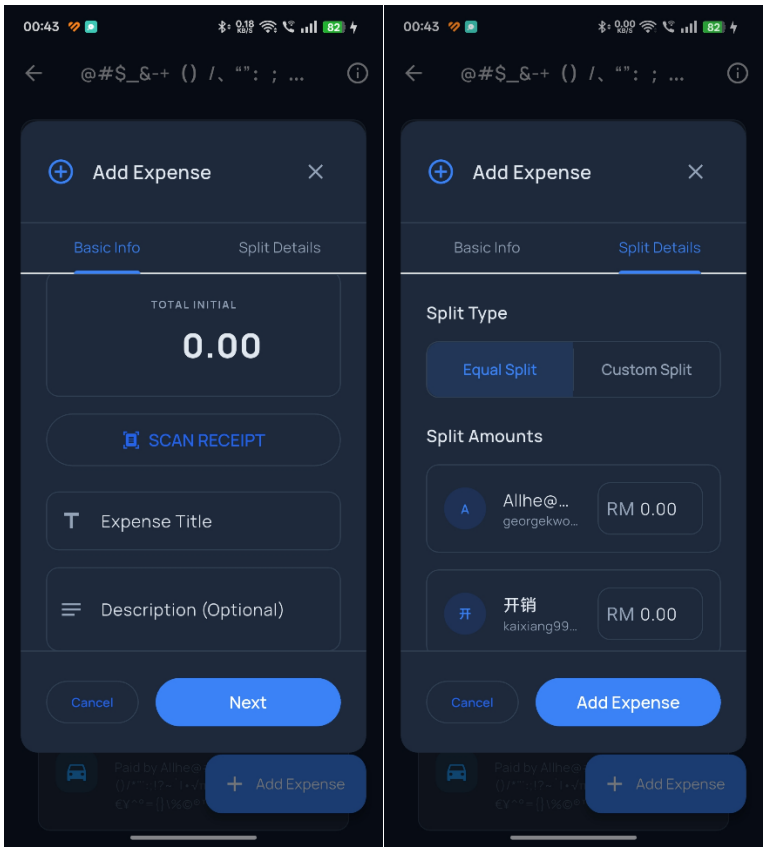
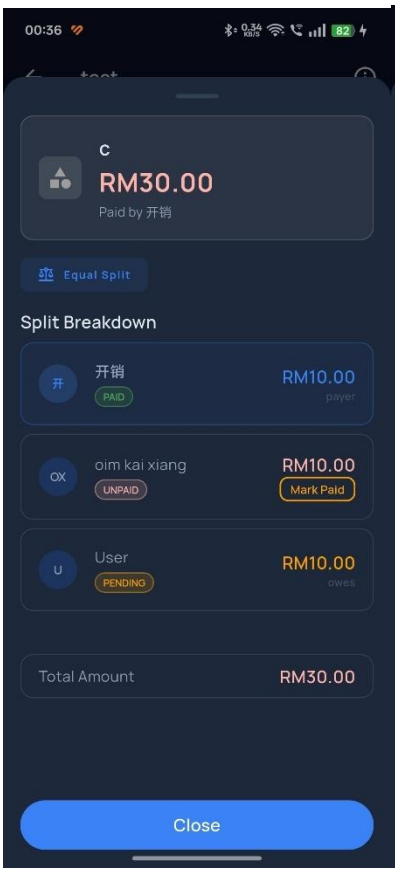


Figure 5.4.5.5 Add Expense Form

Figure 5.4.5.5 presents the Add Expense Form used for recording group transactions. This form includes two primary expense distribution methods: **equal split** and **manual split**.

- In the equal split method, the system automatically divides the total expense equally among all group members.
- In the manual split method, users can assign specific amounts to each member, allowing flexibility when expenses are not shared equally.

The system performs validation to ensure that the total distributed amount matches the original expense value before allowing submission. This validation mechanism prevents inconsistencies in financial records and ensures accurate expense distribution. By supporting both automatic and manual splitting methods, the system provides flexibility while maintaining data integrity.



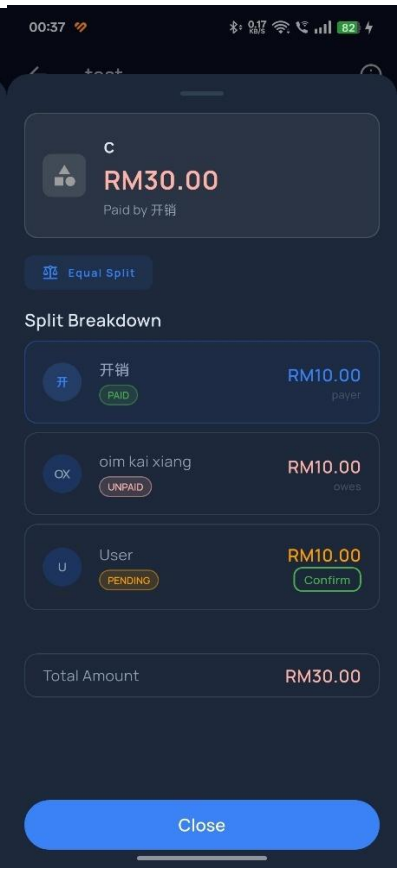


Figure 5.4.5.6: Payer Mark Paid Figure 5.4.5.7: Buyer Confirm Pending

These figures collectively illustrate the payment status workflow within the Group Management module. When a group expense is recorded, all involved members are initially assigned a payment status. Members who owe money are marked as “pending,” indicating that their payment has not yet been completed.

The “Buyer Confirm Pending” state represents users who have outstanding balances and are required to confirm or complete their payment. Once a payment is made, the payer can update their status by marking the transaction as “paid.” This action triggers a real-time update in the system, reflecting the completed payment in the group’s transaction records.

This dual-state mechanism ensures clear tracking of financial responsibilities among group members. It enhances transparency by allowing all users to monitor who has settled their payments and who still has pending obligations, thereby reducing confusion and improving accountability within shared expense management.

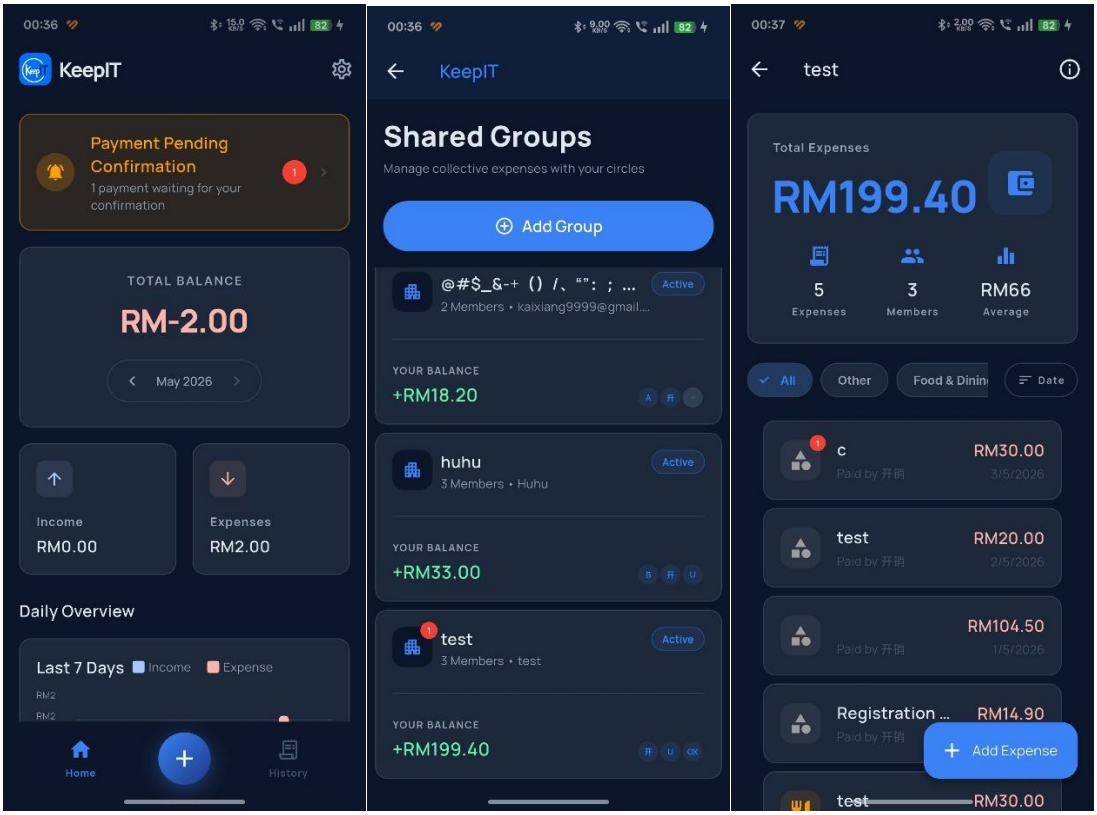


Figure 5.4.5.8 notification of payment pending confirmation

Figure 5.4.5.8 illustrates the notification system that supports the payment status workflow within the Group Management module. When a transaction remains in a “pending” state, the system automatically generates notifications to remind users of outstanding payments.

These notifications are triggered based on real-time transaction status updates and serve as prompts for users to confirm or complete their payments. By integrating notification functionality with payment tracking, the system improves communication among group members and ensures that financial obligations are settled promptly.

This feature enhances overall system efficiency by reducing delays in payment confirmation and maintaining accurate group financial records.

5.4.6. Currency Converter Screen

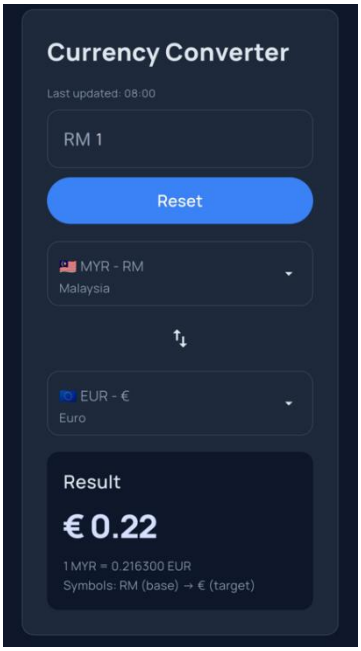


Figure 5.4.6.1 Currency Converter

The currency converter module (Figure 5.4.6.1) allows users to convert monetary values between different currencies using real-time exchange rates obtained from an external API.

When the user inputs an amount and selects the source and target currencies, the system retrieves the latest exchange rate and performs the conversion instantly. The result is displayed dynamically, ensuring accuracy and convenience.

Additionally, a currency rate table (Figure 5.4.6.2) is provided to give users an overview of multiple exchange rates. This enhances user understanding of global currency trends.

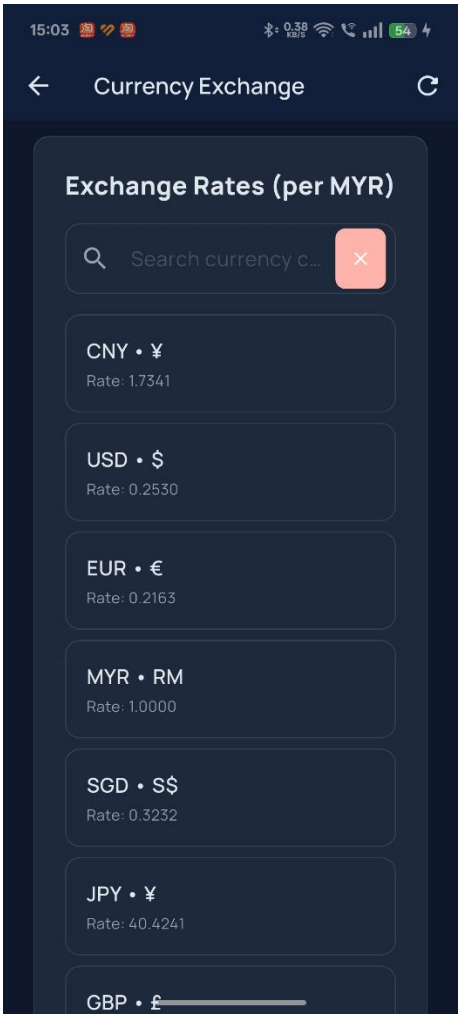


Figure 5.4.6.2 Other Currency Rates Table List

5.4.7. AI Chatbot Screen

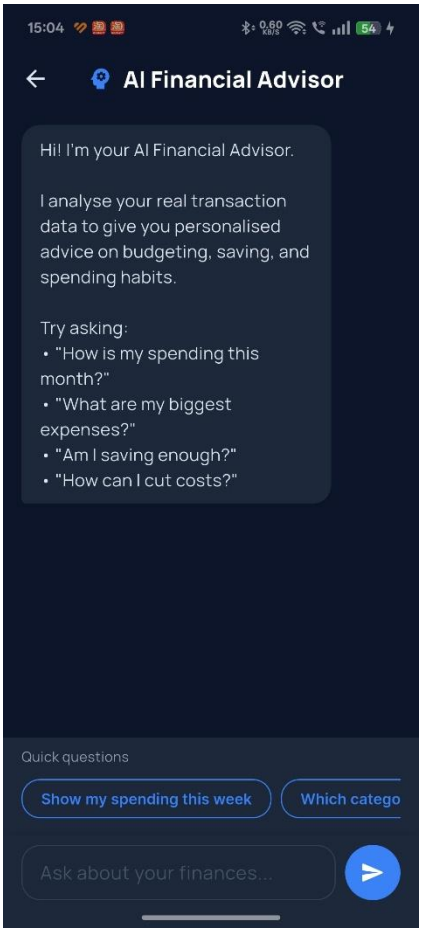


Figure 5.4.7.1 AI Chatbot Advisor

Figure 5.4.7.1 presents the AI Chatbot Advisor interface, which acts as an intelligent assistant within the application. The chatbot is integrated with the backend database and AI engine, enabling it to analyse user transaction data and provide personalized financial advice. Users can interact with the chatbot through natural language queries, making the system more user-friendly and accessible. This feature enhances decision-making by offering insights into spending habits, budgeting strategies, and financial planning.

The chatbot is connected to the backend system through workflow automation, where user queries trigger data retrieval from the database and are processed by the AI model. The integration ensures that responses are context-aware and based on the user's actual financial records, improving the relevance and accuracy of the advice provided.

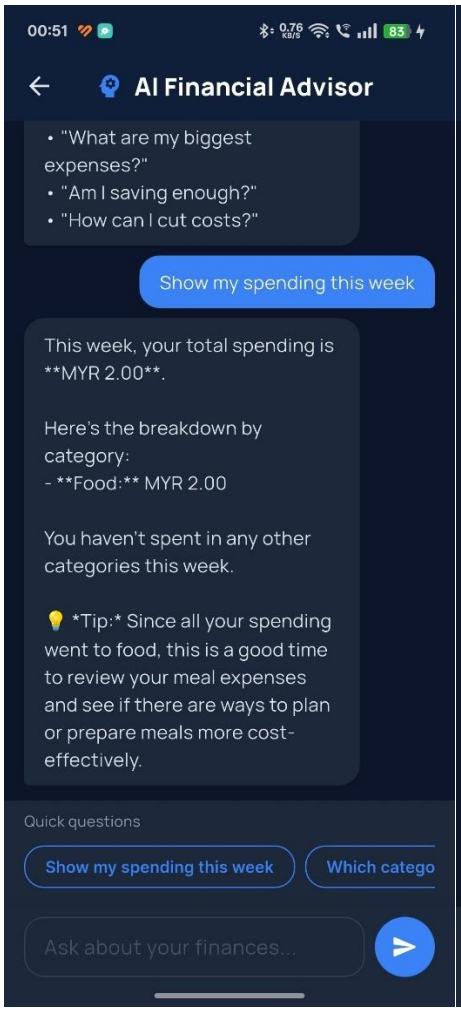


Figure 5.4.7.2: Chatbot Response Figure 5.4.7.3: Chatbot Refuse Answer

These figures illustrate the response handling mechanism of the AI chatbot within the application. When a user submits a query, the system processes the request by analysing relevant financial data retrieved from the database and generating an appropriate response using the integrated AI model.

In normal scenarios, the chatbot produces meaningful responses such as spending analysis, budgeting recommendations, or financial insights based on the user's transaction history. This demonstrates the system's ability to transform raw financial data into useful and actionable information.

However, in cases where the query is outside the supported scope, lacks sufficient data, or cannot be processed reliably, the chatbot returns a refusal response. This controlled

behaviour prevents the generation of inaccurate or misleading information and ensures the reliability of the system.

By supporting both valid responses and refusal handling, the chatbot maintains a balance between functionality and accuracy, enhancing user trust and ensuring consistent system performance. This response control mechanism ensures that the AI operates within defined boundaries, preventing unreliable outputs and maintaining system credibility.

5.4.8. Settings Screen

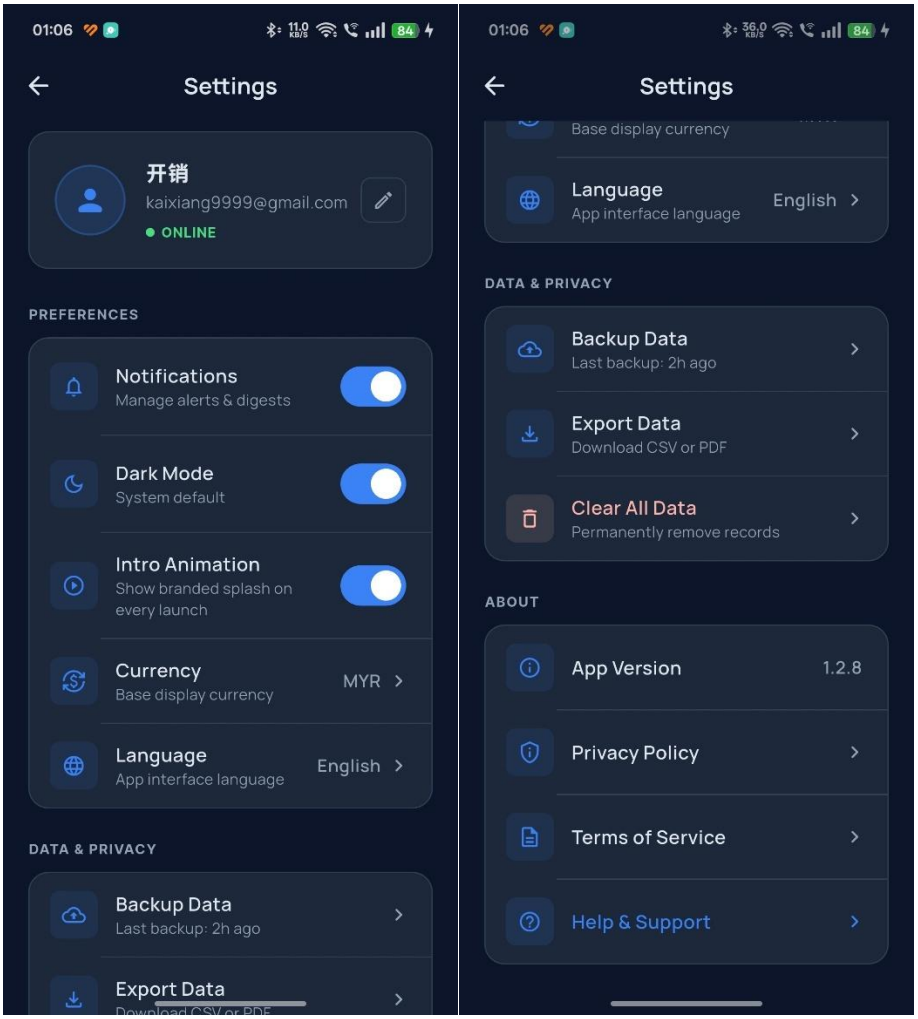


Figure 5.4.8.1 Setting Page

Figure 5.4.8.1 displays the Settings Page, which serves as the central configuration hub of the application. It allows users to manage account settings, customize application preferences, and control system features such as notifications, appearance, and accessibility options.

By centralizing configuration controls, the system improves usability and provides users with greater flexibility in tailoring the application according to their individual needs.

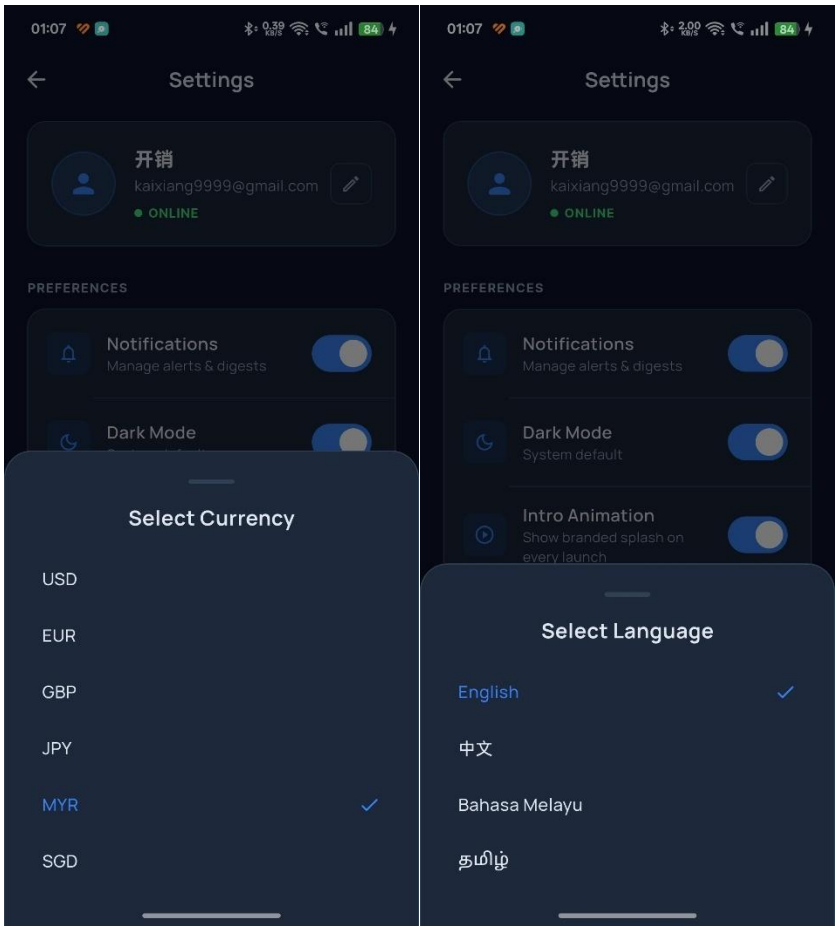


Figure 5.4.8.2 Currency Option Figure 5.4.8.3 Language Option

This figure illustrates the available configuration options, including currency selection and language preferences. These features enhance data reliability and portability by allowing users to securely store backups and export financial records for external analysis or record-keeping purposes.

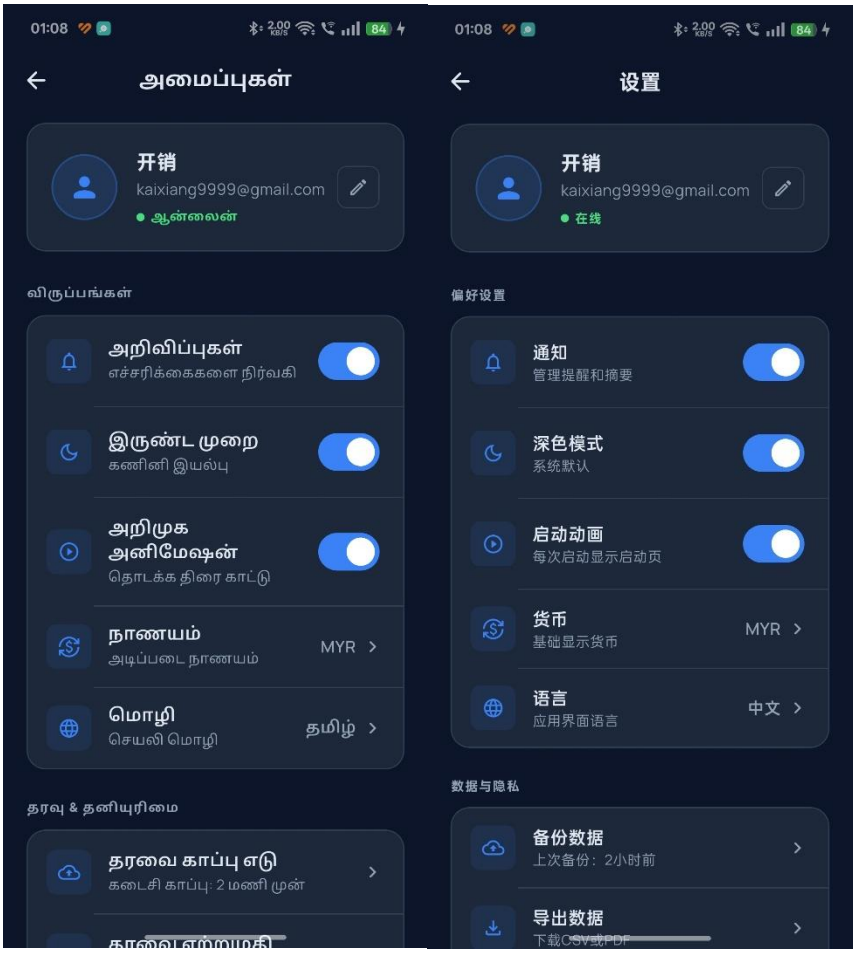
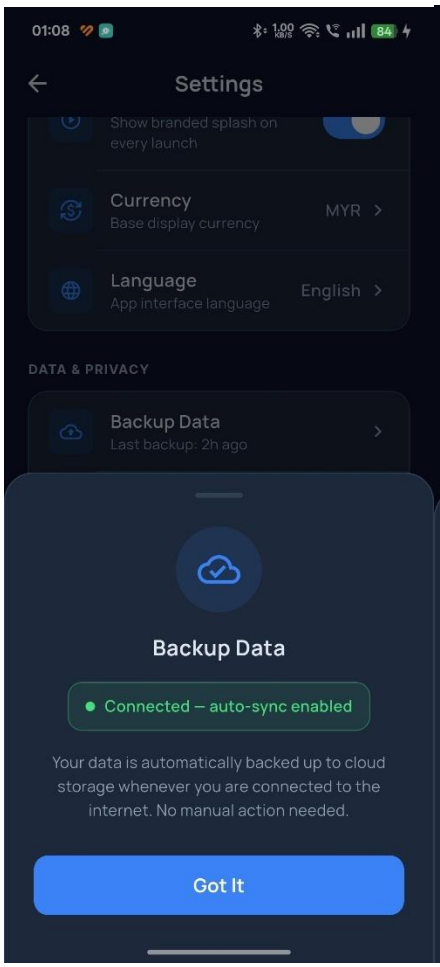


Figure 5.4.8.4 Example Language Change

Figure 5.4.8.4 demonstrates the system's ability to switch languages dynamically. Once a new language is selected, the application updates all interface elements accordingly, enhancing usability for non-English users.



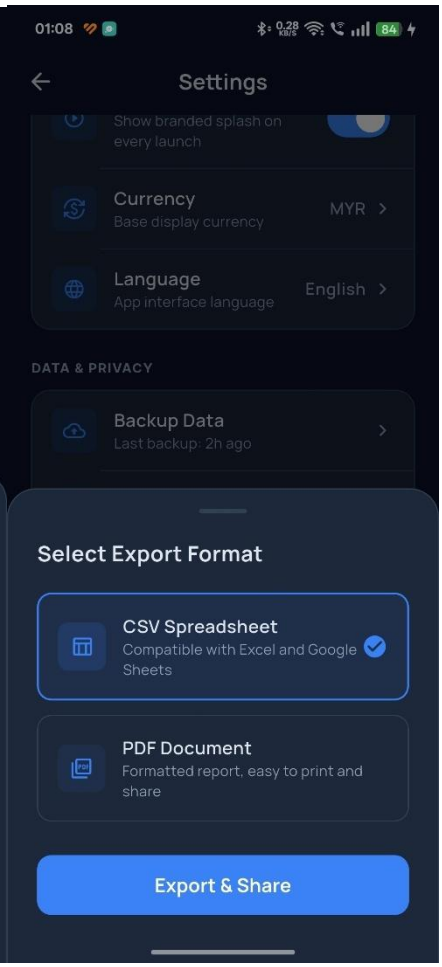


Figure 5.4.8.5 Backup Data

Figure 5.4.8.6 Export Data

This figure shows the data management features that allow users to back up and export their financial records. This ensures data safety and provides flexibility for users who wish to store or analyse their data externally.

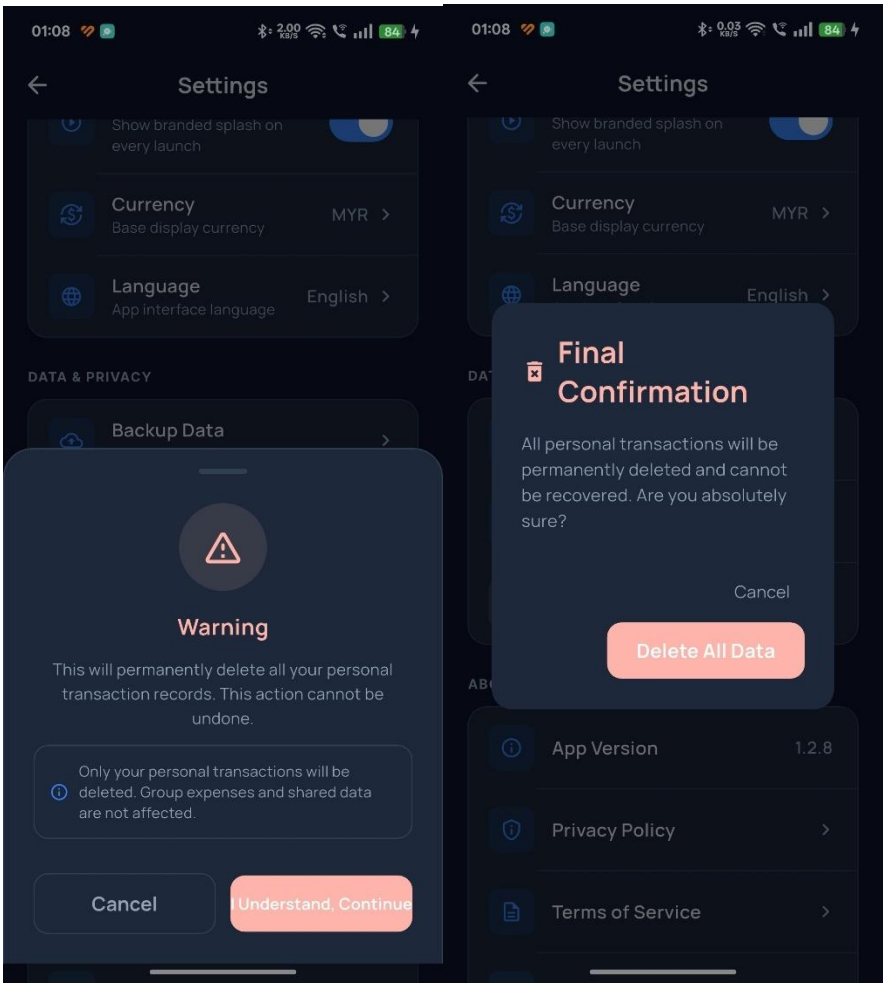


Figure 5.4.8.7 Delete All Data with double confirmation

This figure presents the data deletion feature with a double confirmation mechanism. This design prevents accidental data loss by requiring users to confirm their action before permanently removing all stored records.

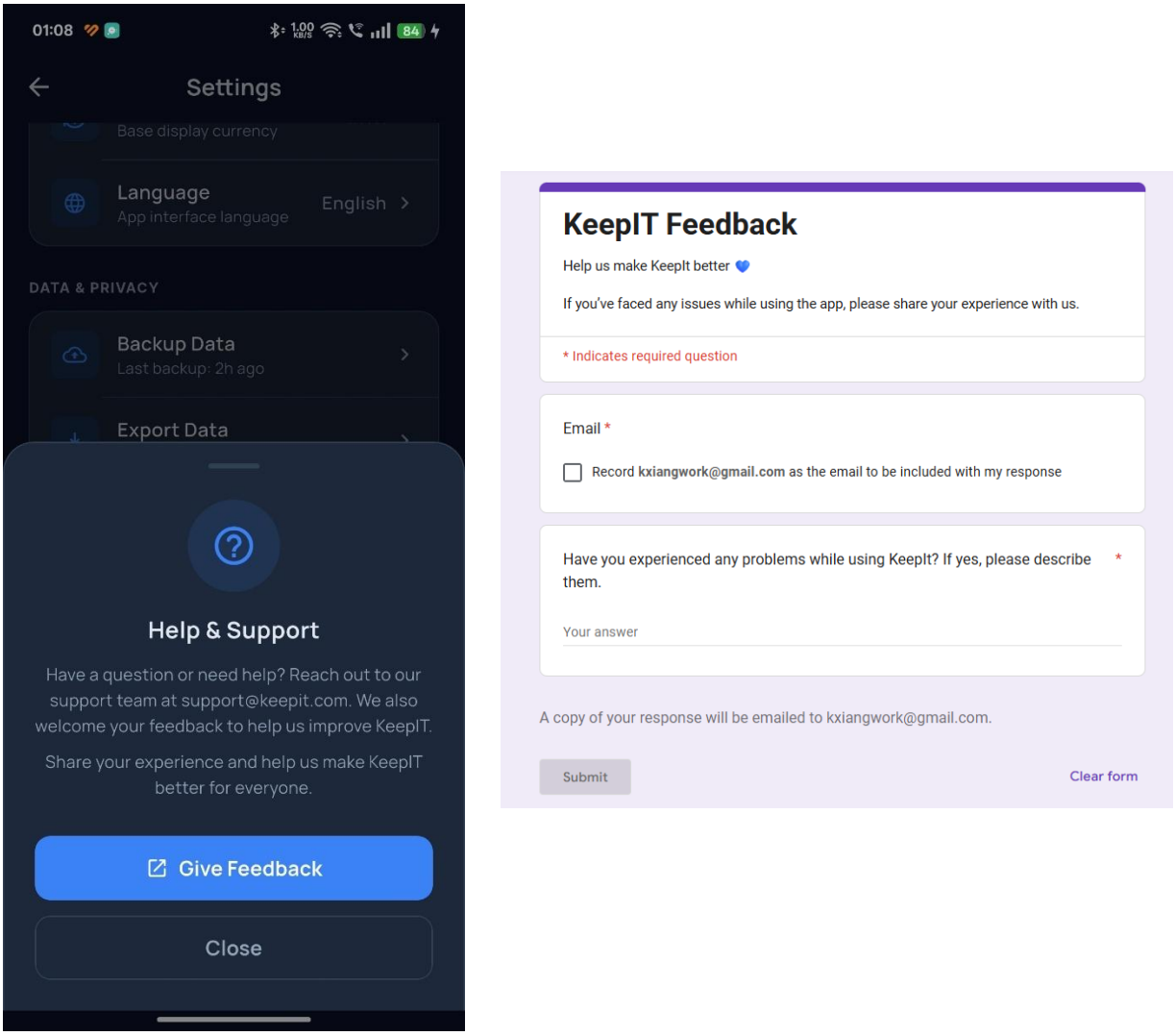


Figure 5.4.8.8 Help&Support with Feedback Form

This figure illustrates the Help & Support interface, which includes a feedback form for users to report issues or provide suggestions. This feature supports continuous system improvement by collecting user input and enhancing overall user experience.

5.4.9. System Operation Summary

Overall, the system operation demonstrates a well-integrated workflow that connects user interface components with backend services such as authentication, database management, OCR processing, and AI analysis. Each module is designed to ensure efficiency, accuracy, and user convenience. The combination of automation, validation mechanisms, and intelligent features enhances the overall functionality of the application while maintaining data consistency and reliability.

5.5.Implementation Issues and Challanges

The implementation of the Money Tracking and Management Application System is going to be confronted with a variety of issues and concerns owing to its integration of more than one technology and its support for sophisticated financial activities. A main issue is **improving the effectiveness of the Optical Character Recognition (OCR)** module. Since receipts can be blurry, handwritten, or presented with inconsistent layouts, maintaining reliable extraction of the details of a transaction is challenging. Ineffective data entry can render the system less effective and reduce the user's confidence. Another issue is **the integration of external services** such as the OCR API and the Currency Exchange API. Since these are external dependencies, issues like slow speed, failure of the services to execute, or restriction of the number of requests can hamper the smooth execution of the app.

Scalability and performance optimisation are also concerns, as the system will have to support an expanded user base and volumes of transactions. Database performance and efficiency design will be required to avoid bottlenecks and maintain brisk response times under intense usage. Another problem comes under the development of **financial insights via the utilisation of AI**. The system must process user input and provide insightful recommendations and generate financial reports. In order for it to perform this, however, it requires clean and correctly formatted data, and improper user input or data extracted through OCR can nullify the quality of the insights and recommendations.

Finally, usability is also one of the concerns since the platform shall accommodate individual and group financial management that is straightforward and user-friendly. When activities such as splitting bills, assigning expense categories, or generating reports become too complex, the system will be difficult for the users to use. Administratively, the upkeep of the categories for the transactions will also require ongoing monitoring for purposes of keeping the categories updated and clearly defined. Without proper upkeep, it can lead to confusion and render the financial records and reports inaccurate.

5.6.Concluding Remark

The development of the **Money Tracking and Management Application System** demonstrates a comprehensive integration of **modern mobile technologies**, **backend services**, and **intelligent automation** to support efficient **personal financial management**. Throughout the implementation, key features such as **transaction recording**, **OCR-based**

receipt scanning, **group expense management**, **currency conversion**, and **AI-powered chatbot assistance** were successfully developed and integrated into a unified **Flutter-based mobile application**. The use of **Supabase** as a backend service ensured **secure data storage**, **real-time synchronization**, and **reliable authentication**, while tools such as **ML Kit**, **n8n**, and **external APIs** enhanced the system's functionality and **automation capabilities**.

Despite several challenges, including **OCR accuracy**, **external API dependency**, and **system scalability**, appropriate design considerations and solutions were applied to maintain **system performance** and **usability**. Overall, the system achieves its objective of providing users with a **practical**, **user-friendly**, and **intelligent platform** for managing financial activities, while also laying a strong foundation for **future improvements** such as enhanced **AI insights**, improved **data accuracy**, and expanded **feature support.**

CHAPTER 6 System Evaluation AND DISCUSSION

6.1.System Testing Overview

This chapter presents the testing and evaluation of the **KeepIt Money Tracking and Management Application** to ensure that it satisfies both functional and non-functional requirements as defined in Section 3.2.2. The testing process was conducted to validate the correctness, performance, usability, and reliability of the system under different operating conditions.

Multiple testing approaches were applied, including functional testing, usability testing, performance testing, and offline mode testing. Functional testing was performed to verify that all core features, such as user authentication, transaction management, group expense handling, and AI-based financial recommendations, operate as intended. Usability testing was conducted to evaluate user interaction and ensure that tasks can be completed efficiently with minimal effort.

In addition, performance testing was carried out to assess system response time, accuracy, and scalability when handling user operations and data processing. Offline mode testing was also performed to confirm that users are able to record transactions without an internet connection, with data stored locally and synchronized automatically once connectivity is restored. The detailed verification plan and corresponding testing results are presented in Sections 6.2 and 6.3, respectively.

6.2.Verification Plan

Before the introduction of the Money Tracking and Management Application System into the public arena, a verification plan is needed for the proper functioning of the application and running under the target specifications. As Table 6.2.1 through Table 6.2.5 reflect,the verification plans were created to provide an appropriate framework for testing each significant feature, such as user authentication, OCR-dependent data extraction, financial recommendations through the use of AI, performance for the transaction and for reporting purposes, scalability for the system itself, and overall usability.

User Authentication

Table 6.2.1 User Authentication Verification Plan

Test Scenario	Input	Expected Outcome
Sign in will validate email and password	**Valid email** **Valid password**	**User successfully logs in and returns to the home page.**
Signing in with a valid email, but the wrong password	**Valid email** **Wrong password**	**Display an Error message "Password do not match"**
Signing in with the wrong email, but the correct password	**Invalid email** **Valid password**	**Display an Error message "Please enter a valid email"**
Signing in with a valid email, but the wrong password format	**Valid email** **Invalid password format**	**Display an Error message "Password must be at least 6 characters"**
Register with a valid email and password	**Valid email** **Valid password**	**Display a Success message, and the user returns to the login page**
Register with the existing email	**Encountered existing email** **Valid password**	**Display an Error message "This email already exists. Try another one"**
Register with a valid email, but a wrong password format	**Valid email** **Wrong password format**	**Display an Error message "Password must be at least 6 characters"**
Register with an invalid email	**Invalid email** **Valid password**	**Display an Error message "Please enter a valid email"**
Register with an invalid password	**Valid email** **Invalid password**	**Display an Error message "Please check your passwords are the same"**

Add Transaction

Table 6.2.2 of Add Transaction Verification Plan

Test Scenario	Input	Expected Outcome

User, fill in the transaction details	**Enter Details**	**Returns the user to the homepage, and transaction details will be saved and displayed**
Transaction details without amount	**Missing Amount**	**Display an Error message "You should enter the amount"**

Group Management

Table 6.2.3 Group Management Verification Plan

Test Scenario	Input	Expected Outcome
Create a Group with full details	**Enter Group Name** **Enter Description** **Add Members**	**Returns the user to the group management page, and the Group details will be read and displayed**
Missing Group Name when creating a group	**Missing Group Name**	**Display an Error message "Please enter the group name"**
Missing Description when creating a group	**Missing Description**	**Display an Error message "Please enter the description"**
Missing members when creating a group	**Missing members**	**Display an Error message "At least add one member"**
Fill in all fields of the Add/Edit group transaction form	**Full fill details**	**Return the users to the group details page and display the transaction record**
Missing Title during the add/edit form	**Missing Title**	**Display an Error message "Please enter the title"**
Missing Amount during the add/edit form	**Missing Amount**	**Display an Error message "Please enter the amount"**

Currency Converter

Table 6.2.4 Currency Converter Verification Plan

Test Scenario	Input	Expected Outcome
User Enter the Amount	**Enter Amount**	**Display the result amount after conversion**
Users clear the entire page	**Click the reset button**	**Refresh the page**

Overview Screen

Table 6.2.5 Overview Screen Verification Plan

Test Scenario	Input	Expected Outcome
User access to the Transaction page	**Click on the transaction icon button**	**Navigate to the transaction page**
User access to the quick action bar	**Click on the center button**	**Display the container pages button**
User access to the add transaction page	**Click on the first button in the quick action bar**	**Navigate to the add transaction page**
User access to the group management page	**Click on the second button in the quick action bar**	**Navigate to the group management page**
User access to the currency converter page	**Click on the third button in the quick action bar**	**Navigate to the currency converter page**
User access to the AI advice page	**Click on the fourth button in the quick action bar**	**Navigate to the AI advice page**
User access to the setting page	**Click on the top-right setting icon button**	**Navigate to the settings page**

6.3.Testing Result and Analysis

This section presents the results of system testing conducted based on the verification plan described in Section 6.2. The results are analyzed to evaluate whether the KeepIt Money Tracking and Management Application meets the functional and non-functional requirements defined in Section 3.2.2.

6.3.1 Performance Evaluation Results

The performance evaluation results indicate that the system meets the predefined performance requirements in terms of response time, accuracy, and usability. Table 6.3.1 presents a comparison between the target performance metrics and the actual results obtained during testing.

Table 6.3.1 Performance Metrics Achieved Results

Metric	Target Requirement (Section 3.2.2)	Actual Result	Status
Transaction Response Time	≤ 2 seconds	0.7 seconds	Pass
Report Generation Time	≤ 5 seconds	2.3 seconds	Pass
System Uptime	≥ 99%	99.2%	Pass
Usability Task Completion	≥ 95% success rate	96%	Pass

As shown in Table 6.3.1, all performance metrics achieved results within the required thresholds. The system demonstrated fast response time during transaction processing and efficient report generation. In addition, the system maintained stable uptime and high usability success rates, indicating that users were able to complete tasks effectively without errors. These results confirm that the system performs reliably under normal operating conditions.

6.3.2 Functional Testing Results

Functional testing was conducted based on the scenarios outlined in the verification plan, including user authentication, transaction management, group management, currency conversion, and system navigation.

Table 6.3.2 Functional Testing Test Case

Module	Test Scenario	Expected Outcome	Actual Result	Status
Authentication	Valid login	Navigate to homepage	Successfully navigated	Pass
Authentication	Invalid password	Error message displayed	Correct error message shown	Pass
Transaction	Add transaction	Data saved and displayed	Successfully saved and displayed	Pass
Transaction	Missing amount	Error message displayed	Correct error message shown	Pass
Group	Create group	Group created and displayed	Successfully created	Pass
Group	Missing group name	Error message displayed	Correct error message shown	Pass
Currency	Convert amount	Display converted value	Accurate result displayed	Pass
Navigation	Access different modules	Navigate to selected page	Successfully navigated	Pass

The results in Table 6.3.2 show that all functional test cases passed successfully. The system handled both valid and invalid inputs correctly, providing appropriate feedback and ensuring data integrity. All modules functioned as expected, demonstrating the reliability and completeness of the system implementation.

6.3.3 OCR Testing Results

OCR (Optical Character Recognition) testing was conducted to evaluate the system's ability to accurately extract relevant information from receipt images under different conditions. The purpose of this testing is to determine the accuracy and reliability of the OCR module in recognizing key transaction details and reducing the need for manual data entry.

The evaluation focused on three main data fields: **transaction amount**, **transaction date**, and **merchant name**, as these are essential for financial record management within the system. The OCR output for these fields was compared against manually verified ground truth data to determine correctness.

The testing was divided into two categories: online sample receipts and real-life captured receipts, each consisting of 10 samples.

6.3.3.1 Online Sample Testing

Online sample testing was performed using 10 receipt images obtained from online sources. These images were generally clear, well-structured, and of high quality, representing ideal conditions for OCR processing. The original receipt images and their corresponding OCR output results for all online samples are provided in Appendix B.1 for reference.(O_R represent Online Receipt)

Table 6.3.3.1 Testing Result Online Receipt

Receipt	Correct Field	OCR Get	Status
O_R1	Mechart: LOWE's Date:05-25-25 Amount:5170,79	Mechart: LOWE's Date:05-25-25 Amount:502990.00	PASS PASS FAIL
O_R2	Mechart: Walmart Date:07-28-17 Amount:98.21	Mechart: Save money. Live Better Date:07-28-17 Amount:6.75	FAIL PASS FAIL

O_R3	Mechart: YourCompanyInC Date:11-04-2025 Amount:252.00	Mechart: YourCompanyInC Date:11-04-2025 Amount:252.00	PASS PASS PASS
O_R4	Mechart: ANAND SERVICE STATION Date:02-22-2020 Amount:1837.46	Mechart: ANAND SERVICE STATION Date:02-22-2020 Amount:1837.46	PASS PASS PASS
O_R5	Mechart: East Repair Lne Date:26-03-2014(too blur) Amount:328.65	Mechart: East Repair Lne Date:- Amount:328.65	PASS FAIL PASS
O_R6	Mechart: Shop`s Date:02-05-2023 Amount:60	Mechart: Receipt 1254xxx Date:02-05-2023 Amount:60	FAIL PASS PASS
O_R7	Mechart: SL SOFTWARE SOLU… Date:28-12-2023 Amount:20	Mechart: Invoice Date:28-12-2023 Amount:20	FAIL PASS PASS
O_R8	Mechart: 99 SpeedMart Date:13-08-24 Amount:18.55	Mechart: 05.21pm Date: 13-08-24 Amount:50	FAIL PASS FAIL
O_R9	Mechart: EastRepair Inc Date:11-02-2019 Amount:154.06	Mechart: RECEIPT Date:11-02-2019 Amount:154.06	FAIL PASS PASS
O_R10	Mechart: CONSTRUCTION Date:29-03-2023 Amount:4580	Mechart: CONSTRUCTION Date:29-03-2023 Amount:4580	PASS PASS PASS

Metric	Value
Total Receipts Tested	10
Total Fields Evaluated	30
Correctly Extracted Fields	21

Accuracy	70%

The results show that the OCR module achieved high accuracy when processing online sample receipts. Most transaction amount, date, and merchant name fields were correctly extracted due to the clear formatting and high image quality. This demonstrates that the OCR system performs effectively under optimal conditions.

6.3.3.2 Real-Life Sample Testing

Real-life testing was conducted using 10 receipt images captured using a mobile device under normal user conditions. These receipts included variations in lighting, alignment, background noise, and image clarity, representing realistic usage scenarios. The original receipt images and OCR output results for real-life samples are presented in Appendix B.2 for further verification. (R_R represent Realife Receipt)

Table 6.3.3.2 Testing Result Realife Receipt

Receipt	Correct Field	OCR Get	Status
R_R1	Mechart: ATZ Kopitiam Date:05-25-25 Amount:18.60	Mechart: ??? Date:05-25-25 Amount:18.62	FAIL PASS FAIL
R_R2	Mechart: Brother City Tyre…(Hand) Date:09-10-25 Amount:170	Mechart: 16 Date: - Amount: -	FAIL FAIL FAIL
R_R3	Mechart: Hakka Yin Cafe Date:03-04-2026 Amount:14.90	Mechart: ??? Date:03-04-2026 Amount:14.90	FAIL PASS PASS
R_R4	Mechart: Chat A Toast Date:28-04-26 Amount:11.90	Mechart: Sally… Date:28-04-26 Amount:11.90	FAIL PASS PASS
R_R5	Mechart: TAOGO MART SDN BHD	Mechart: TAOGO MART SDN BHD	PASS

	Date: 01-05-26 Amount: 58.00	Date: - Amount: 58.00	FAIL PASS
R_R6	Mechart: Mango Box Date: 24-12-25 Amount: 24.05	Mechart: 07 Date: 24-12-25 Amount: 24.05	FAIL PASS PASS
R_R7	Mechart: RESTORAN DIM SUM Date: 24-04-26 Amount: 135.00	Mechart: RESTORAN DIM SUM Date: 24-04-26 Amount: 135.00	PASS PASS PASS
R_R8	Mechart: Mosel… Date:01-05-26 Amount:104.50	Mechart: - Date:01-05-26 Amount:104.50	FAIL PASS PASS
R_R9	Mechart: Restoran Hao Ran Ju (Hand) Date: 03-05-26 Amount: 89	Mechart: Restoran Hao Ran Ju Date: - Amount: -	PASS FAIL FAIL
R_R10	Mechart: MR SNOOKER … Date: 03-05-26 Amount: 27.30	Mechart: Payment Amount… Date: 03-05-26 Amount: 27.30	FAIL PASS PASS

Metric	Value
Total Receipts Tested	10
Total Fields Evaluated	30
Correctly Extracted Fields	17
Accuracy	56.7%

The results indicate that the OCR module performs reasonably well in real-world scenarios, although the accuracy is slightly lower compared to online samples. Some inaccuracies were observed when extracting data from blurred images, tilted receipts, or receipts with complex layouts. Despite this, the system was still able to correctly extract most key information.

It should be noted that two of the tested receipts (R_R2 and R_R9) are handwritten. The OCR module is designed to recognize printed text only; therefore, handwritten receipt

recognition is not supported and is considered outside the scope of the system. These cases are treated as acceptable limitations.

6.3.3.3 Adjusted OCR Accuracy

To evaluate overall performance, the results from both datasets were combined.

Table 6.3.3.3 Adjusted OCR Accuracy Result

Metric	Value
Adjusted Fields Evaluated	54
Correctly Extracted Fields	37
Overall Accuracy	68.5%

The OCR testing results indicate that the system performs better under ideal conditions compared to real-world scenarios. The online sample testing achieved higher accuracy due to clearer image quality and structured formatting, while real-life testing showed lower accuracy due to variations in lighting, alignment, and receipt layout.

After excluding handwritten receipts, the adjusted accuracy improves to 68.5%, which more accurately reflects the system's performance for printed receipts. This demonstrates that the OCR module is effective for its intended use case, although performance may be affected by image quality and formatting variations.

6.3.3.4 Offline Mode Testing Results

Offline mode testing was conducted by disabling internet connectivity during system operation to evaluate the system's behavior without network access.

Table 6.3.3.4 Offline Mode Testing Results

Scenario	Expected Outcome	Actual Result	Status
Add transaction offline	Data stored locally	Successfully stored locally	Pass
Restore internet connection	Data synchronized to cloud	Successfully synchronized	Pass
Data consistency	No data loss or duplication	No issues observed	Pass

The results in Table 6.3.3 indicate that the offline mode feature functions correctly. Users were able to record transaction data without an internet connection, and all data was successfully synchronized once connectivity was restored. This confirms that the system supports continuous operation even under limited network conditions.

6.3.4 AI Chatbot Testing Results

The AI chatbot testing was conducted to evaluate the system's ability to provide relevant financial advice and respond appropriately to user queries. The purpose of this testing is to assess the accuracy, reliability, and response quality of the AI module in supporting users' financial decision-making.

The evaluation was based on a set of predefined user queries related to financial management, including expense tracking, budgeting advice, and general financial recommendations. Each response generated by the chatbot was compared against expected outcomes to determine whether the answer was relevant, accurate, and meaningful.

Test Scenario	Expected Outcome	Actual Result	Status
Ask for budgeting advice	Provide general budgeting tips	Relevant advice provided	Pass
Ask how to reduce expenses	Suggest cost-saving strategies	Appropriate suggestions given	Pass

Test Scenario	Expected Outcome	Actual Result	Status
Ask about savings planning	Provide savings recommendations	Accurate guidance provided	Pass
Ask unrelated question	Reject or respond appropriately	Correctly handled	Pass
Ask unclear question	Request clarification	Clarification requested	Pass

The results indicate that the AI chatbot is able to generate relevant and meaningful responses for financial-related queries. The system successfully provides general financial advice, including budgeting strategies and expense management suggestions.

In addition, the chatbot is capable of handling invalid or unrelated inputs by either rejecting the query or requesting clarification, ensuring that inappropriate responses are avoided. This demonstrates that the AI module is reliable and behaves as expected under different interaction scenarios.

Overall, the AI chatbot enhances the functionality of the system by providing users with accessible financial guidance. Although the responses are general in nature and not personalized to a high degree, the chatbot performs effectively within its intended scope and contributes to improving user experience.

6.3.5 User Acceptance Testing (UAT) Feedback Analysis

To further evaluate the usability and effectiveness of the KeepIt Money Tracking and Management Application, User Acceptance Testing (UAT) was conducted using a structured Google Form questionnaire. The purpose of this evaluation is to collect feedback from real users after interacting with the system and to assess overall user satisfaction, usability, and functionality.

The questionnaire consists of both quantitative and qualitative questions, including ratings on ease of use, system performance, OCR functionality, and overall satisfaction. A total of three testers participated in the evaluation, and their responses were recorded and analyzed.

The summary of the responses is presented in Table 6.3.5

Table 6.3.5 UAT Feedback result

Evaluation Criteria	Tester 1	Tester 2	Tester 3
Ease of Use	Easy	Very Easy	Easy
OCR Accuracy	Good	Moderate	Good
User Interface Design	Clean	Attractive	Simple
Overall Satisfaction	4/5	5/5	4/5

Based on the collected feedback, all testers agreed that the system is easy to use, indicating that the user interface is intuitive and user-friendly. The user interface design was positively evaluated, with comments such as “clean” and “attractive,” suggesting that the design meets user expectations.

In terms of OCR performance, most testers rated the accuracy as “good,” while one tester rated it as “moderate.” This aligns with the earlier OCR testing results, where accuracy was affected by real-world conditions such as image quality and lighting. This feedback confirms that while the OCR feature is useful, there is room for improvement.

The overall satisfaction scores range from 4 to 5, resulting in an average satisfaction score of 4.33 out of 5. This indicates a high level of user acceptance and suggests that the system meets user expectations in terms of functionality and usability.

Additionally, qualitative feedback highlights that users appreciate the convenience of features such as automatic receipt scanning and group expense management. However, some users suggested improving OCR accuracy and enhancing AI recommendations for more personalized insights.

Although the number of participants is limited, the collected feedback provides valuable insights into user experience and system performance. The results confirm that the system

is user-friendly, functional, and suitable for practical use, while also identifying areas for future improvement.

The detailed questionnaire design and collected responses are provided in Appendix C for verification and reference.

6.3.6 Discussion

Overall, the testing results demonstrate that the KeepIt Money Tracking and Management Application meets both functional and non-functional requirements. The system performs efficiently, with fast response times and stable operation across different modules. Functional testing confirms that all core features, including user authentication, transaction management, group management, and navigation, operate correctly and provide appropriate feedback for both valid and invalid inputs.

The OCR module shows moderate performance, achieving higher accuracy under ideal conditions and lower accuracy in real-world scenarios due to variations in image quality, lighting, and receipt formatting. The inability to recognize handwritten receipts is identified as a limitation; however, this is acceptable as the system is designed primarily for printed receipts. The OCR feature still provides significant value by reducing manual data entry and improving user convenience.

The AI chatbot demonstrates the ability to provide relevant financial advice and handle user queries appropriately. It is capable of generating meaningful responses for common financial questions and managing invalid or unclear inputs effectively. However, the responses are generally limited to predefined or general knowledge and may not provide highly personalized financial insights.

In addition, the offline mode functionality enhances system usability by allowing users to record transactions without internet connectivity. The system successfully synchronizes data once the connection is restored, ensuring data consistency and reliability.

In conclusion, the system is stable, reliable, and suitable for real-world financial management applications. While certain limitations exist, particularly in OCR accuracy under challenging conditions and the general nature of AI responses, these do not significantly impact the overall effectiveness of the system.

The User Acceptance Testing (UAT) results further support the effectiveness of the system from a user perspective. Feedback collected from testers indicates that the application is easy to use, with a clear and intuitive user interface. The overall satisfaction level is high, with users appreciating features such as OCR-based receipt scanning and group expense management.

However, the feedback also highlights areas for improvement, particularly in OCR accuracy under real-world conditions and the level of personalization in AI-generated responses. These findings are consistent with the system testing results and provide valuable insights for future enhancements. Although the number of participants is limited, the feedback confirms that the system is practical and suitable for real-world use.

6.4 Project Challenges

During the system evaluation phase, several challenges were identified that affected the performance and reliability of the KeepIt Money Tracking and Management Application. One of the primary challenges is the limitation of the OCR module in handling real-world receipt images. Although the OCR performs well under ideal conditions, its accuracy decreases when processing images with poor lighting, tilted angles, or complex layouts. Additionally, handwritten receipts cannot be recognized by the system, which highlights a limitation in the current OCR implementation.

Another challenge is related to the dependency on external services, including the OCR API, currency exchange API, and AI service. These external components may introduce latency, inconsistent response times, or temporary service interruptions, which can affect the overall system performance. Ensuring stable integration with these services is essential for maintaining system reliability.

Furthermore, the AI chatbot module presents challenges in providing highly personalized and context-aware financial advice. While the chatbot is capable of generating general recommendations, its responses are limited by the available transaction data and predefined logic, which may reduce the depth of insights provided to users.

An additional challenge is identified in the group expense management module. During expense splitting, users who are involved in a transaction may have the ability to edit the transaction details. This creates a potential issue where some users may modify or avoid their payment responsibilities, leading to unfair distribution of expenses among group members. Such behavior may result in disputes and reduce trust among users, highlighting the need for stricter access control and validation mechanisms within the group module.

Lastly, ensuring data consistency in offline mode also presents a challenge, particularly during data synchronization when internet connectivity is restored. Although no major issues were observed during testing, maintaining data integrity between local storage and the cloud database requires careful handling to avoid duplication or data conflicts.

Despite these challenges, appropriate design considerations and fallback mechanisms have been implemented to minimize their impact and ensure acceptable system performance.

6.5 Objectives Evaluation

This section evaluates the extent to which the project objectives defined in Chapter 1 have been successfully achieved through the system development and testing phases.

The first objective, which is to implement an OCR-based receipt scanning module, has been successfully achieved. The system is capable of extracting key transaction details such as amount, date, and merchant name from receipt images. Based on the testing results, the OCR module achieved an accuracy of 70.0% for online samples and 56.7% for real-life samples, with an adjusted overall accuracy of 68.5%. Although the performance varies under different conditions, the module effectively reduces manual data entry and improves user convenience.

The second objective, which is to develop a group cost-sharing module, has also been successfully fulfilled. The system supports both equal and manual expense splitting methods, ensuring fair and flexible distribution of group expenses. Testing results confirm that the module functions correctly and maintains data accuracy during transaction processing.

The third objective, which involves implementing real-time multi-currency conversion, has been achieved through integration with an external currency API. The system is able to retrieve up-to-date exchange rates and perform accurate conversions within a short response time, meeting the defined performance requirements.

The final objective, which is to develop an AI-based financial insights module, has been partially achieved. The AI chatbot is capable of providing general financial advice and responding appropriately to user queries. However, the level of personalization and depth of analysis is limited, indicating potential areas for future improvement.

Overall, the project objectives have been largely achieved, with all core functionalities successfully implemented and validated through testing. Minor limitations identified during evaluation do not significantly affect the overall effectiveness of the system.

6.6 Concluding Remark

In conclusion, the system evaluation results demonstrate that the KeepIt Money Tracking and Management Application is able to meet its intended design goals and functional requirements. The system performs efficiently, with stable response times, accurate data processing, and high usability across different modules. Functional testing confirms that all core features operate correctly, while performance testing shows that the system meets the required benchmarks for speed and reliability.

Advanced features such as OCR receipt scanning, AI chatbot assistance, and offline mode functionality further enhance the system's capability by improving user convenience and accessibility. Although certain limitations exist, particularly in OCR accuracy under real-world conditions and the level of AI personalization, these issues are within acceptable bounds and do not significantly impact the overall system performance.

The evaluation results indicate that the system is suitable for real-world financial management applications and provides a practical solution for users to manage their personal and group finances effectively. The findings from this chapter also provide a foundation for future improvements, particularly in enhancing OCR accuracy, expanding AI capabilities, and optimizing system scalability.

CHAPTER 7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

In conclusion, this project successfully developed the KeepIt Money Tracking and Management Application to address common challenges in personal financial management, including difficulty in tracking expenses, lack of budget awareness, and inefficient group expense handling. The system provides an integrated mobile platform that enables users to record, manage, and analyse their financial activities in a more efficient and user-friendly manner.

The application incorporates several key features, including manual transaction recording, OCR-based receipt scanning, group expense management with fair cost-sharing, real-time multi-currency conversion, and an AI-powered chatbot for financial insights. These features work together to provide a comprehensive solution that enhances user convenience and improves financial awareness.

Through the implementation and evaluation phases, the system demonstrated stable performance, accurate data handling, and high usability. The OCR module was able to reduce manual input effort, while the AI chatbot provided meaningful financial guidance. In addition, the offline mode functionality ensures continuous system usability even without internet connectivity, further enhancing the practicality of the application.

Overall, the project has achieved its main objectives by delivering a functional and reliable financial management system. The integration of modern technologies such as OCR, AI, and cloud-based services contributes to the innovation of the system and provides a solid foundation for future enhancements. The application has strong potential to assist users in managing their finances more effectively and improving their financial decision-making.

7.2 Recommendation

Although the KeepIt Money Tracking and Management Application has achieved its intended objectives, several improvements can be made in future work to enhance system performance, functionality, and user experience.

Firstly, the OCR module can be further improved by incorporating more advanced image processing techniques and machine learning models to enhance accuracy under real-world conditions. Features such as image enhancement, automatic alignment, and noise reduction can be implemented to improve the quality of scanned receipts. Additionally, support for handwritten text recognition can be explored to expand the system's capabilities.

Secondly, the AI chatbot module can be enhanced to provide more personalized and context-aware financial recommendations. This can be achieved by integrating more advanced AI models, improving data analysis techniques, and incorporating user behaviour patterns to generate more accurate and meaningful insights.

Furthermore, improvements can be made to the group expense management module by implementing stricter access control and validation mechanisms. For example, restricting unauthorized edits to transaction records and introducing approval workflows can help prevent misuse and ensure fair expense distribution among group members.

In addition, system scalability and performance can be further optimized to support a larger number of users and transactions. This includes improving database performance, optimizing API response times, and enhancing system architecture to handle higher workloads efficiently.

Lastly, additional features such as notification reminders, financial goal tracking, and advanced data visualization can be introduced to improve user engagement and provide a more comprehensive financial management experience.

These recommendations provide a direction for future development and demonstrate the potential for further enhancement of the system beyond its current implementation.

REFERENCES

[1] T. S. Bernard, “Your money: Managing your debt,” The New York Times. [Online]. Available: https://www.nytimes.com/2024/05/16/your-money/managing-debt.html. [Accessed: Apr. 28, 2025].

[2] 钱迹李唐, “简介 · 钱迹用户指南,” Qianjiapp.com. [Online]. Available: https://docs.qianjiapp.com/. [Accessed: Apr. 29, 2025].

[3] 钱迹李唐, “RoadMap Vol.2 · Qianji user manual,” Qianjiapp.com. [Online]. Available: https://docs.qianjiapp.com/roadmap/roadmap_vol2.html. [Accessed: Apr. 30, 2025].

[4] 钱迹李唐, “Asset line · Qianji user manual,” Qianjiapp.com. [Online]. Available: https://docs.qianjiapp.com/assetrecord/asset_line.html. [Accessed: Apr. 30, 2025].

[5] Wikipedia, “YNAB,” Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/YNAB. [Accessed: Apr. 29, 2025].

[6] Trustpilot, “YNAB activities | See how YNAB uses Trustpilot,” Trustpilot.com. [Online]. Available: https://www.trustpilot.com/review/ynab.com/transparency. [Accessed: Apr. 29, 2025].

[7] C. Palma, “YNAB is more than a budgeting app, its principles can help your long-term,” Unite News. [Online]. Available: https://unitenewsonline.org/articles/ynab-is-more-than-a-budgeting-app-its-principles-can-help-your-long-term/. [Accessed: Apr. 29, 2025].

[8] Expensify, “Who we are,” Expensify. [Online]. Available: https://we.are.expensify.com/how-we-got-here. [Accessed: Apr. 30, 2025].

[9] G2, “Expensify reviews & product details,” G2.com. [Online]. Available: https://www.g2.com/products/expensify/reviews#reviews. [Accessed: Apr. 30, 2025].

[10] Expensify, “Receipt scanner app | Easily scan receipts with Expensify,” Expensify. [Online]. Available: https://use.expensify.com/receipt-scanning-app. [Accessed: Apr. 30, 2025].

[11] S. Pansare and D. Joshi, “A survey on optical character recognition techniques,” International Journal of Science and Research, 2014. [Online]. Available: https://www.ijsr.net/archive/v3i12/U1VCMTQ2Njg=.pdf. [Accessed: May 05, 2025].

[12] Capterra, “Reviews of Expensify,” Capterra. [Online]. Available: https://www.capterra.com/p/97594/Expensify/reviews/. [Accessed: May 01, 2025].

[13] Pixlr, “Enhance image with Pixlr AI - Sharpen & upscale quickly,” Pixlr.com. [Online]. Available: https://pixlr.com/cn/tools/enhance-image/. [Accessed: May 01, 2025].

[14] Y. Li, X. Zhao, J. Liang, and L. Jin, “A Structured Recognition Method for Invoices Based on StrucTexT,” Applied Sciences, vol. 13, no. 12, p. 6946, 2023. doi: 10.3390/app13126946.

[15] F. Yu, C. Wang, S. Wang, and J. Liu, “Receipt Recognition Technology Driven by Multimodal Alignment and Lightweight Sequence Modeling,” Electronics, vol. 14, no. 9, p. 1717, 2025. doi: 10.3390/electronics14091717.

[16] K. Tabianan, S. Velu, and V. Ravi, “K-Means Clustering Approach for Intelligent Customer Segmentation Using Customer Purchase Behavior Data,” Sustainability, vol. 14, no. 12, p. 7243, 2022. doi: 10.3390/su14127243.

[17] K. Sako, B. N. Mpinda, and P. C. Rodrigues, “Neural Networks for Financial Time Series Forecasting,” Entropy, vol. 24, no. 5, p. 657, 2022. doi: 10.3390/e24050657.

[18] Atlassian, “Agile best practices and tutorials,” Atlassian. [Online]. Available: https://www.atlassian.com/agile. [Accessed: May 03, 2025].

[19] Flutter, “Flutter Documentation,” 2025. [Online]. Available: https://docs.flutter.dev/

[20] Flutter, “Flutter Official Site,” 2025. [Online]. Available: https://flutter.dev/

[21] Flutter, “Flutter GitHub Repository,” 2025. [Online]. Available: https://github.com/flutter/flutter

[22] Dart, “Dart Overview,” 2025. [Online]. Available: https://dart.dev/overview

[23] Wikipedia, “Dart (programming language),” 2025. [Online]. Available: https://en.wikipedia.org/wiki/Dart_(programming_language)

[24] Flutter Docs, “Intro to Dart,” 2025. [Online]. Available: https://docs.flutter.dev/get-started/fundamentals/dart

[25] Supabase, “Supabase Documentation,” 2026. [Online]. Available: https://supabase.com/docs

[26] Supabase, “Supabase Features,” 2026. [Online]. Available: https://supabase.com/features

[27] Android Developers, “Android Studio,” 2025. [Online]. Available: https://developer.android.com/studio

[28] Wikipedia, “Android Studio,” 2025. [Online]. Available: https://en.wikipedia.org/wiki/Android_Studio

[29] Android Developers, “Build your first app | Get started,” 2025. [Online]. Available: https://developer.android.com/get-started/overview

[30] Microsoft, “Visual Studio Code,” 2025. [Online]. Available: https://code.visualstudio.com/

[31] Google Developers, “ML Kit Overview,” 2026. [Online]. Available: https://developers.google.com/ml-kit

[32] Google Developers, “Text Recognition v2 | ML Kit,” 2025. [Online]. Available: https://developers.google.com/ml-kit/vision/text-recognition/v2

[33] n8n.io, “AI Workflow Automation Platform - n8n,” 2026. [Online]. Available: https://n8n.io/

[34] n8n.io, “n8n Features and Workflow Automation,” 2026. [Online]. Available: https://n8n.io/features/

[35] GeeksforGeeks, “What is n8n?,” 2026. [Online]. Available: https://www.geeksforgeeks.org/artificial-intelligence/what-is-n8n/

[36] ScalaHosting, “What is n8n and How to Use It?,” 2025. [Online]. Available: https://www.scalahosting.com/blog/what-is-n8n-and-how-to-use-it/

[37] n8n GitHub, “n8n Workflow Automation Platform,” 2026. [Online]. Available: https://github.com/n8n-io/n8n

[38] HubSpot, “AI Agents Unleashed: Playbook for 2025 Success,” 2025.

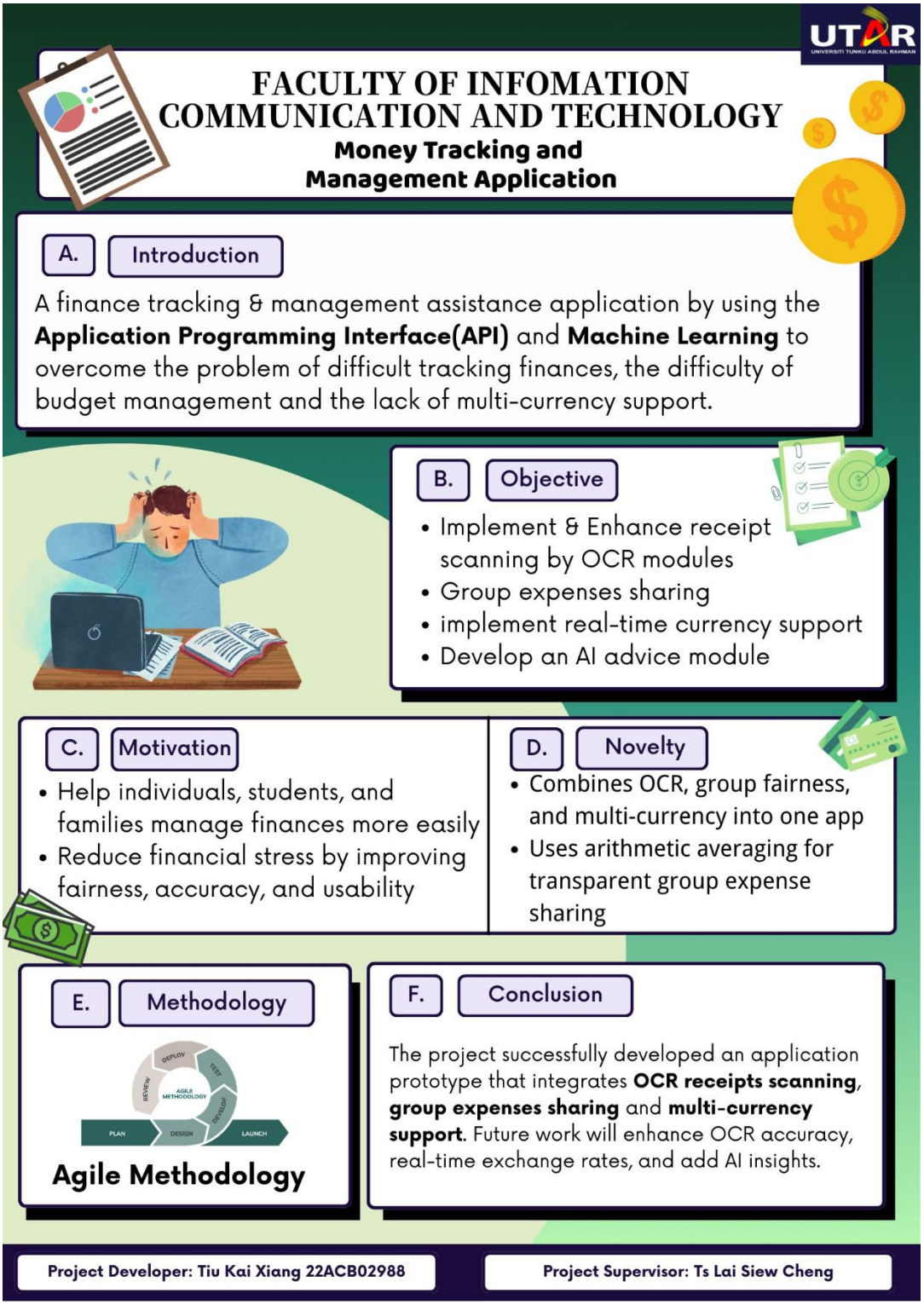
[39] HubSpot, “How to Use AI Agents in 2025,” 2025.

[40] Figma, “Figma Official Website,” https://www.figma.com/

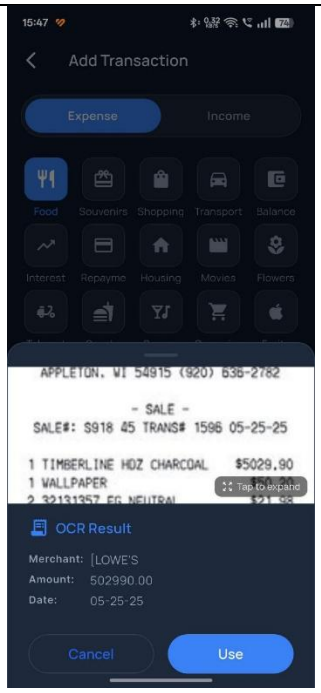
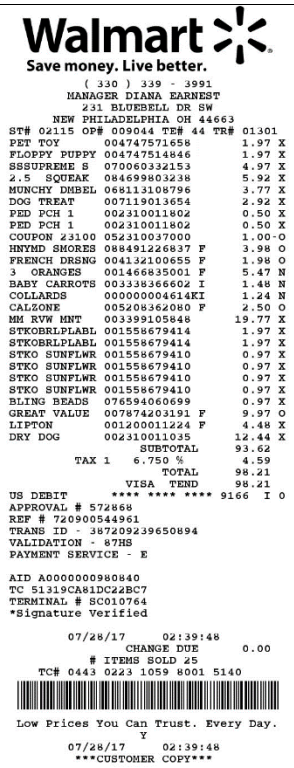
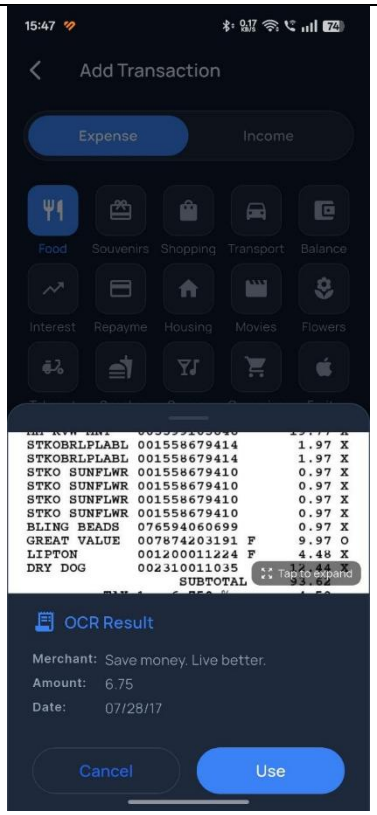
[41] Figma, “What is Figma?,” https://help.figma.com/

APPENDIX

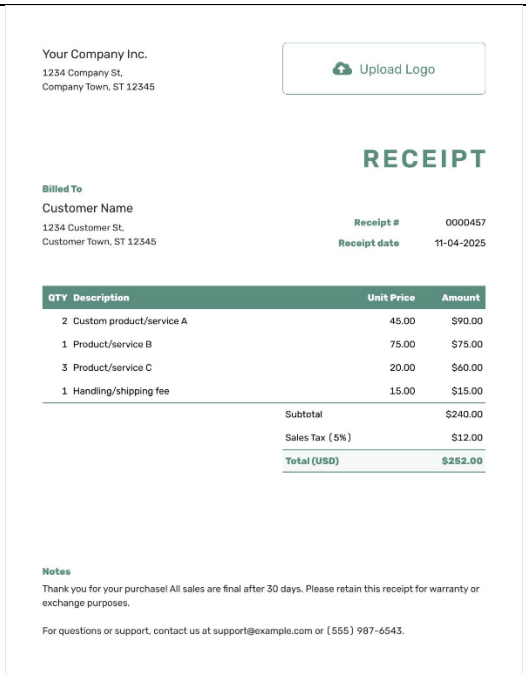
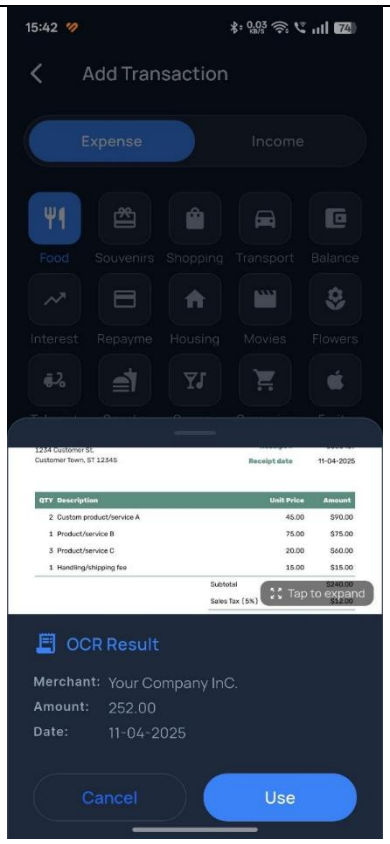
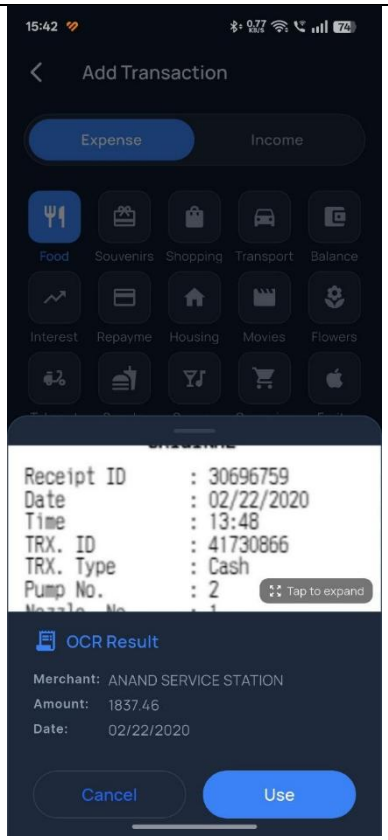
A.1 Poster

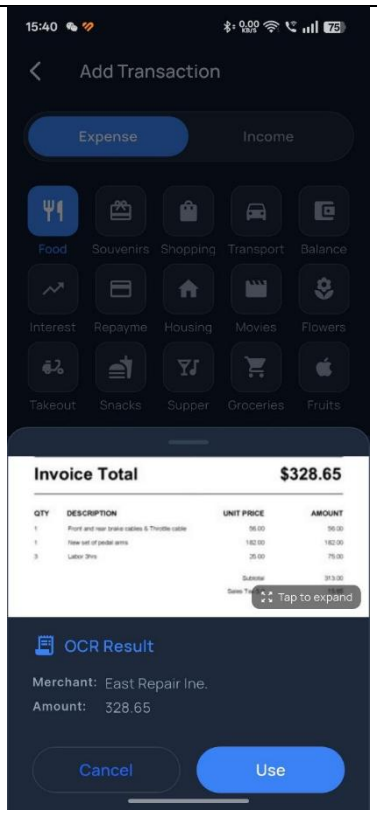
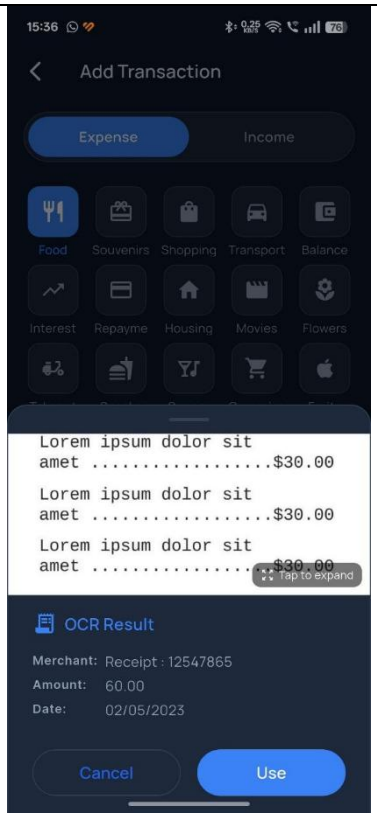


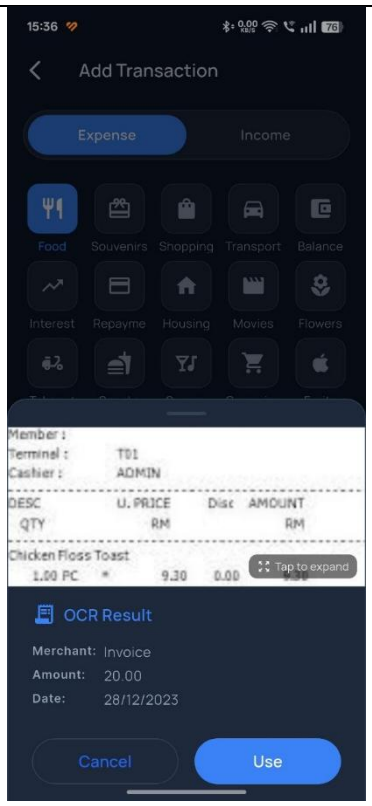
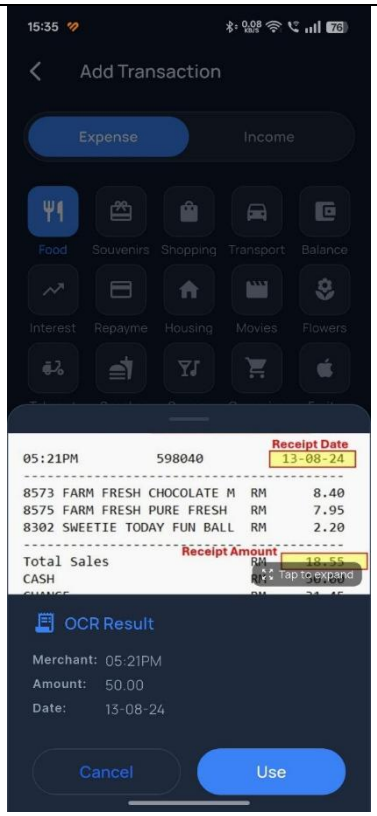
B.1 Online_Receipt

Receipt	Original Image	OCR Result Image
O_R1		
O_R2	 Walmart Save money. Live better. (330) 339 - 3991 MANAGER DIANA EARNEST 231 BLUEBELL DR SW NEW PHILADELPHIA OH 44663 ST# 02115 OP# 009044 TE# 44 TR# 01301 PET TOY 004747571658 1.97 X FLOPPY PUPPY 004747514846 1.97 X SSSUPREME S 070060332153 4.97 X 2.5 SQUEAK 084699803238 5.92 X MUNCHY DMBEL 068113108796 3.77 X DOG TREAT 007119013654 2.92 X PED PCH 1 002310011802 0.50 X PED PCH 1 002310011802 0.50 X COUPON 23100 052310037000 1.00-O HNYMD SMORES 088491226837 F 3.98 O FRENCH DRSNG 004132100655 F 1.98 O 3 ORANGES 001466835001 F 5.47 N BABY CARROTS 003338366602 I 1.48 N COLLARDS 000000004614KI 1.24 N CALZONE 005208362080 F 2.50 O MM RVW MNT 003399105848 19.77 X STKOBRLPLABL 001558679414 1.97 X STKOBRLPLABL 001558679414 1.97 X STKO SUNFLWR 001558679410 0.97 X STKO SUNFLWR 001558679410 0.97 X STKO SUNFLWR 001558679410 0.97 X STKO SUNFLWR 001558679410 0.97 X BLING BEADS 076594060699 0.97 X GREAT VALUE 007874203191 F 9.97 O LIPTON 001200011224 F 4.48 X DRY DOG 002310011035 12.44 X SUBTOTAL 93.62 TAX 1 6.750 % 4.59 TOTAL 98.21 VISA TEND 98.21 US DEBIT **** **** **** 9166 I 0 APPROVAL # 572868 REF # 720900544961 TRANS ID - 387209239650894 VALIDATION - 87HS PAYMENT SERVICE - E AID A0000000980840 TC 51319CA81DC22BC7 TERMINAL # SC010764 *Signature Verified 07/28/17 02:39:48 CHANGE DUE 0.00 # ITEMS SOLD 25 TC# 0443 0223 1059 8001 5140 Low Prices You Can Trust. Every Day. Y 07/28/17 02:39:48 ***CUSTOMER COPY***	

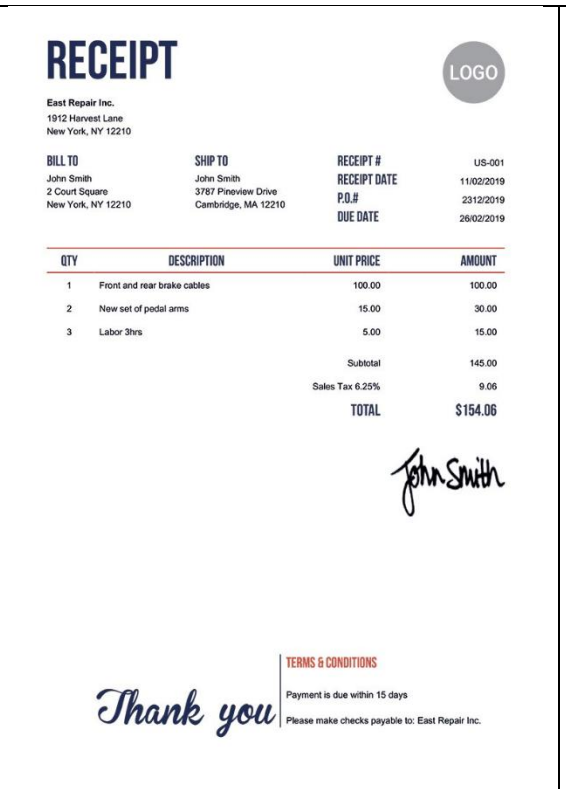
APPENDIX

O_R3	 Your Company Inc. 1234 Company St, Company Town, ST 12345 Upload Logo RECEIPT Billed To Customer Name 1234 Customer St, Customer Town, ST 12345 Receipt # 0000457 Receipt date 11-04-2025 QTY Description Unit Price Amount 2 Custom product/service A 45.00 $90.00 1 Product/service B 75.00 $75.00 3 Product/service C 20.00 $60.00 1 Handling/shipping fee 15.00 $15.00 Subtotal $240.00 Sales Tax (5%) $12.00 Total (USD) $252.00 Notes Thank you for your purchase! All sales are final after 30 days. Please retain this receipt for warranty or exchange purposes. For questions or support, contact us at support@example.com or (555) 987-6543.	 15:42 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Tap to expand OCR Result Merchant: Your Company InC. Amount: 252.00 Date: 11-04-2025 Cancel Use
O_R4	 HP ANAND SERVICE STATION 899, ANDREWS, GANJ NEW DELHI VAT TIN: 0901232334344 Tel.: 88897898989 ORIGINAL Receipt ID : 30696759 Date : 02/22/2020 Time : 13:48 TRX. ID : 41730866 TRX. Type : Cash Pump No. : 2 Nozzle. No : 1 Product : Unleaded Vehicle. No : NotEntered Mobile. No : NotEntered Price : ₹ 77.24 Quantity : 23.789 Ltr Sale : ₹ 1837.46 Total Sale : ₹ 1837.46 Thank You! Visit Again Save Fuel, Save Money WELCOME YOU	 15:42 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Receipt ID : 30696759 Date : 02/22/2020 Time : 13:48 TRX. ID : 41730866 TRX. Type : Cash Pump No. : 2 Tap to expand OCR Result Merchant: ANAND SERVICE STATION Amount: 1837.46 Date: 02/22/2020 Cancel Use

O_R5	East Repair Inc. 465 Amsterdam Avenue New York, NY 1023 BILL TO John Smith 2 Court Square Long Island City New York, NY 1120 SHIP TO John Smith 664 Lexington Avenue 6th Floor New York, NY 1022 INVOICE # 00234 INVOICE DATE 26/03/2014 P.O.# 1742/2014 DUE DATE 10/04/2014 Invoice Total $328.65 QTY DESCRIPTION UNIT PRICE AMOUNT 1 Front and rear brake cables & Throttle cable 56.00 56.00 1 New set of pedal arms 182.00 182.00 3 Labor 3hrs 25.00 75.00 Subtotal 313.00 Sales Tax 5.0 15.65 TERMS & CONDITIONS Payment is due within 15 days Citibank, N.A. Account number: 2345678 Routing: 23412	 15:40 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Takeout Snacks Supper Groceries Fruits Invoice Total $328.65 Tap to expand OCR Result Merchant: East Repair Inc. Amount: 328.65 Cancel Use
O_R6	SHOP`S Receipt : 12547865 Manager : Lor T. Address: 896 Rigoberto Gardens Apt. 838 Kuhnstad, BC X5T5C2 02/05/2023 11:58:20 AM Lorem ipsum dolor sit amet .....$30.00 Lorem ipsum dolor sit amet .....$30.00 Lorem ipsum dolor sit amet .....$30.00 Total..... $60.00 xxxx xxxx xxxx 1234 Visa/5544 THANK YOU FOR SHOPPING! modif.ai	 15:36 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Lorem ipsum dolor sit amet .....$30.00 Lorem ipsum dolor sit amet .....$30.00 Lorem ipsum dolor sit amet .....$30.00 Tap to expand OCR Result Merchant: Receipt : 12547865 Amount: 60.00 Date: 02/05/2023 Cancel Use

O_R7	 Invoice -- Reprint Receipt -- SL SOFTWARE SOLUTIONS SDN. BHD. (201601010552 (1181482-M)) NO 231 A,JALAN PASIR PUTEH, 31650 IPOH,PERAK Document No. : T01-000008 Date : 29/12/2023 10:22:56 AM Member : Terminal : T01 Cashier : ADMIN DESC / QTY — U. PRICE RM — Disc — AMOUNT RM Chicken Floss Toast 1.00 PC * 9.30 0.00 9.30 Earl Grey 1.00 PC * 8.90 0.00 8.90 Sub Total : 18.20 Service Charge : 1.82 Rounding Adjustment : -0.02 Rounded Total (RM): 20.00 Cash 20.00	 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Member : Terminal : T01 Cashier : ADMIN DESC / QTY — U. PRICE RM — Disc — AMOUNT RM Chicken Floss Toast 1.00 PC * 9.30 0.00 Tap to expand OCR Result Merchant: Invoice Amount: 20.00 Date: 28/12/2023 Cancel Use
O_R8	 99 SPEED MART SDN. BHD. 200001016930 (519537-x) 1004-BANTING Receipt Number INVOICE NO : 100420680/103/T0052 Receipt Date 05:21PM 598040 13-08-24 8573 FARM FRESH CHOCOLATE M RM 8.40 8575 FARM FRESH PURE FRESH RM 7.95 8302 SWEETIE TODAY FUN BALL RM 2.20 Receipt Amount Total Sales RM 18.55 CASH RM 50.00 CHANGE RM 31.45 Visit below URL to request for e-invoice https://99einvoice.com	 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Receipt Date 05:21PM 598040 13-08-24 8573 FARM FRESH CHOCOLATE M RM 8.40 8575 FARM FRESH PURE FRESH RM 7.95 8302 SWEETIE TODAY FUN BALL RM 2.20 Receipt Amount Total Sales CASH Tap to expand OCR Result Merchant: 05:21PM Amount: 50.00 Date: 13-08-24 Cancel Use

O_R9



RECEIPT

LOGO

East Repair Inc.
1912 Harvest Lane
New York, NY 12210

BILL TO	SHIP TO		
John Smith 2 Court Square New York, NY 12210	John Smith 3787 Pineview Drive Cambridge, MA 12210	RECEIPT #	US-001
		RECEIPT DATE	11/02/2019
		P.O.#	2312/2019
		DUE DATE	26/02/2019

QTY	DESCRIPTION	UNIT PRICE	AMOUNT
1	Front and rear brake cables	100.00	100.00
2	New set of pedal arms	15.00	30.00
3	Labor 3hrs	5.00	15.00
		Subtotal	145.00
		Sales Tax 6.25%	9.06
		TOTAL	$154.06

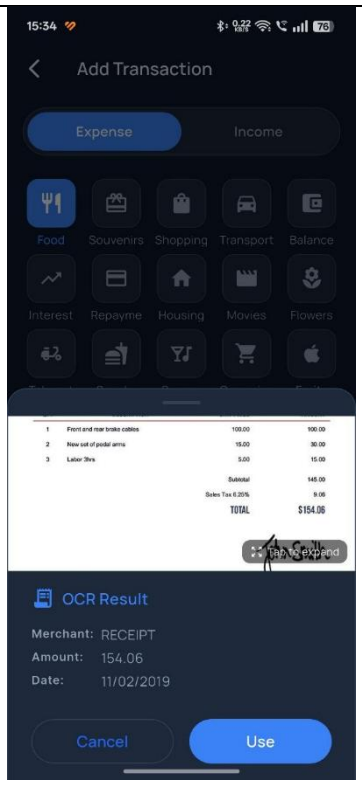
John Smith

Thank you

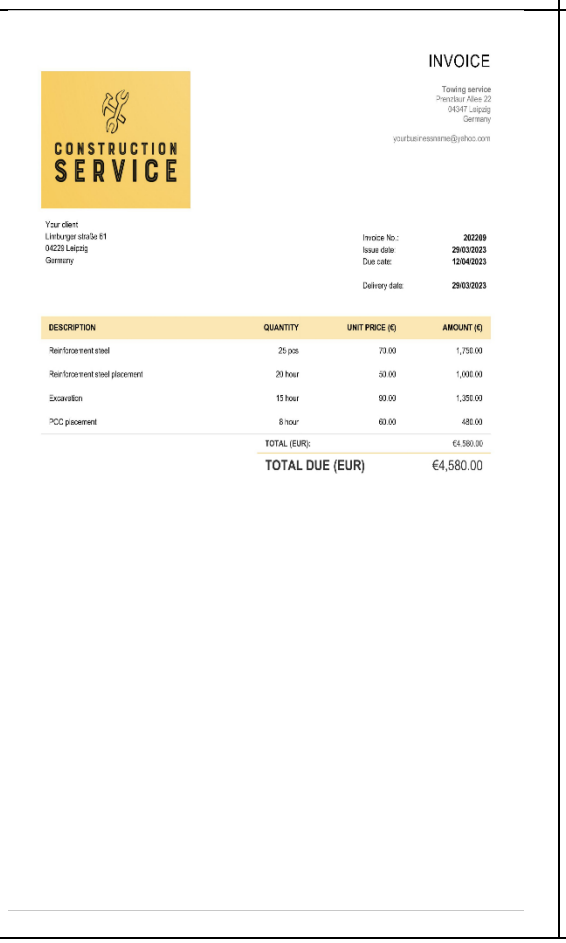
TERMS & CONDITIONS

Payment is due within 15 days

Please make checks payable to: East Repair Inc.



O_R10



CONSTRUCTION SERVICE

INVOICE

Towing service
Prestauer Allee 22
04347 Leipzig
Germany

yourbusinessname@yahoo.com

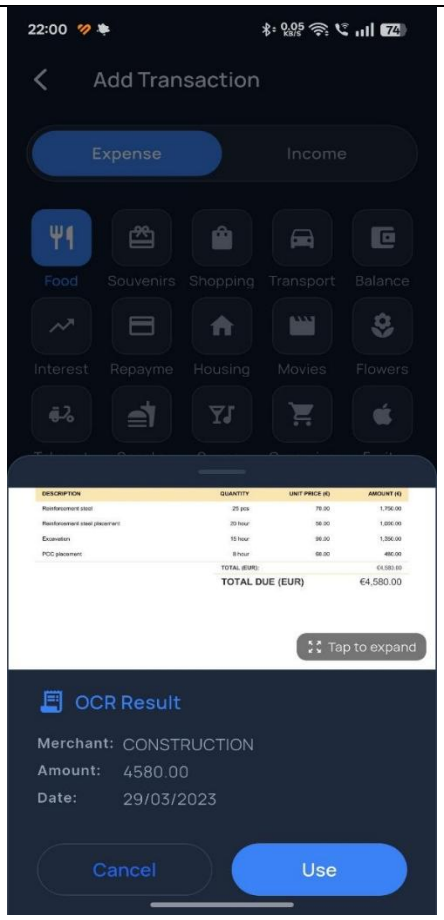
Your client
Limburger straße 81
04229 Leipzig
Germany

Invoice No.: 202209
Issue date: 29/03/2023
Due date: 12/04/2023

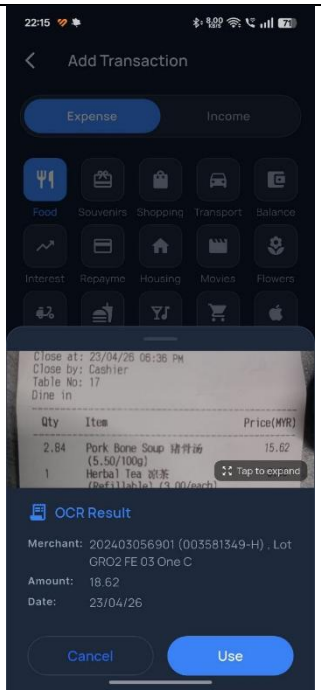
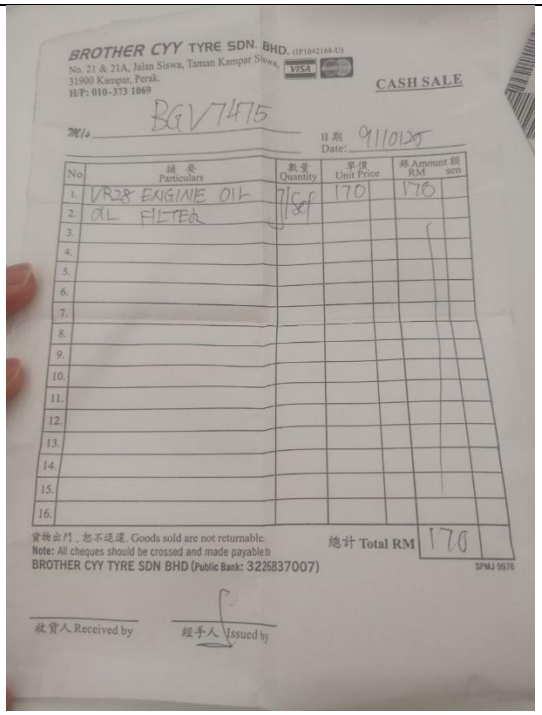
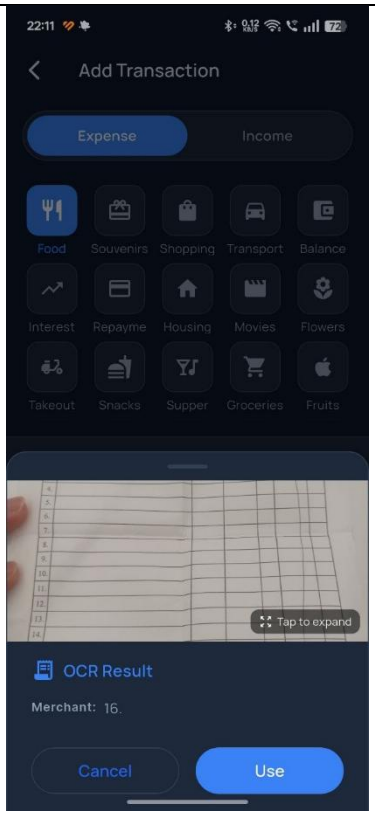
Delivery date: 29/03/2023

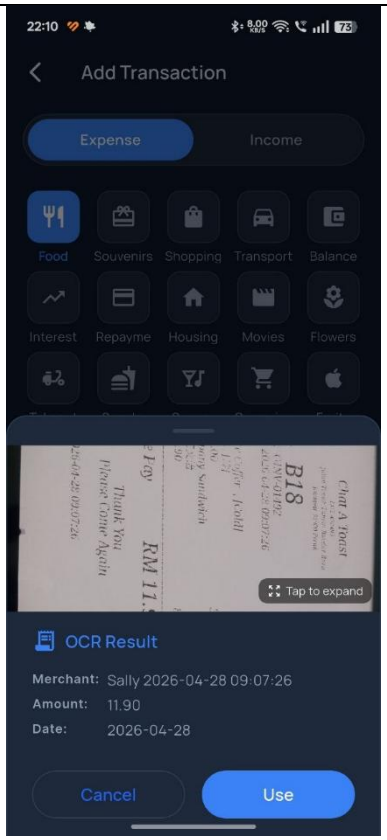
DESCRIPTION	QUANTITY	UNIT PRICE (€)	AMOUNT (€)
Reinforcement steel	25 pcs	70.00	1,750.00
Reinforcement steel placement	20 hour	50.00	1,000.00
Excavation	15 hour	90.00	1,350.00
PCC placement	8 hour	60.00	480.00
	TOTAL (EUR):		€4,580.00

TOTAL DUE (EUR) €4,580.00

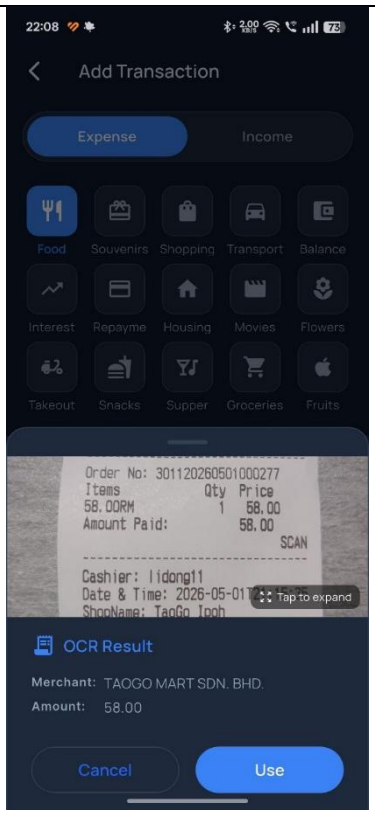
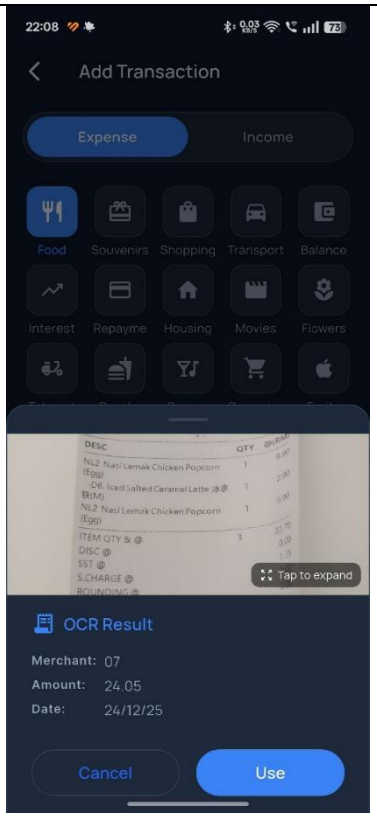


B.2 Realife_Receipt

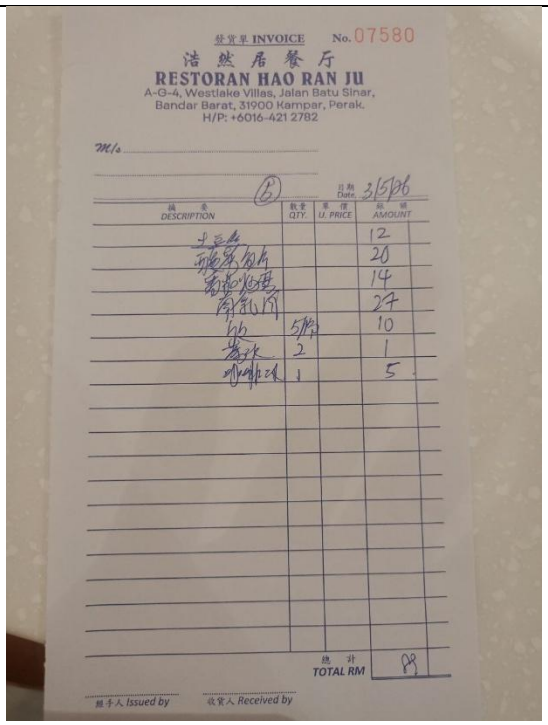
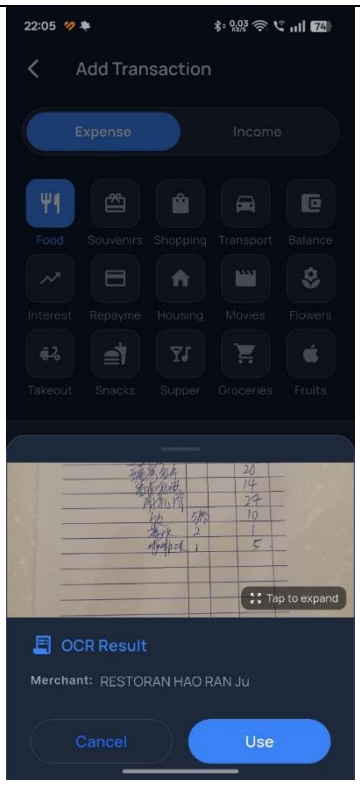
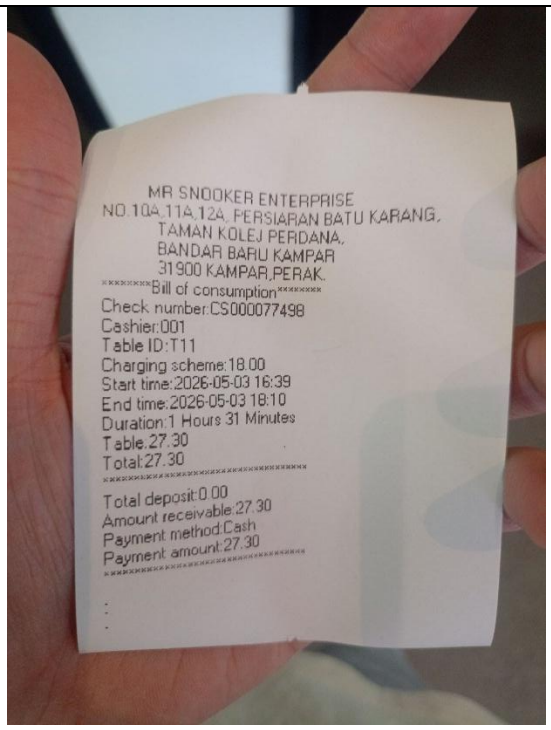
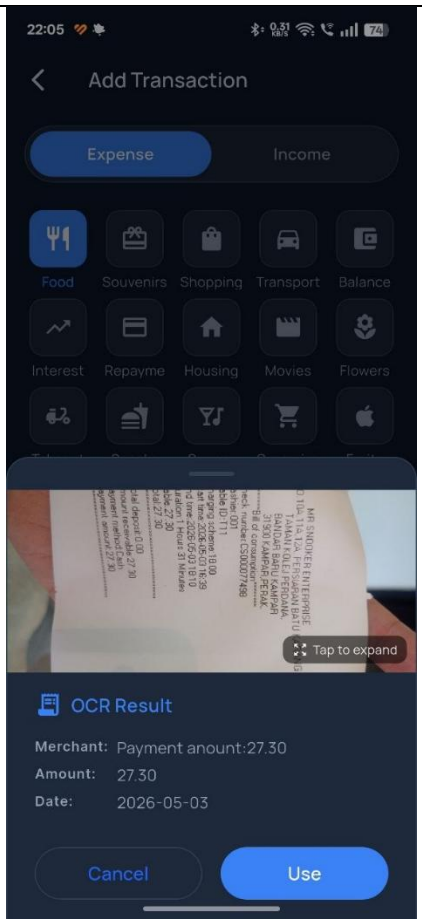
Receipt	Original Image	OCR Result Image
R_R1	 阿土仔东炎 ATZ Kopitiam 202403056901 (003581349-H), Lot GR02 FE 03 One Champ Avenue, Lot 126029, Jalan Hala Timah 2, 31900 Kampar, Perak. 0102731088 atzkopitiam@gmail.com Receipt No.: #76278 Close at: 23/04/26 06:36 PM Close by: Cashier Table No: 17 Dine in Qty Item Price(MYR) 2.84 Pork Bone Soup 猪骨汤 (5.50/100g) 15.62 1 Herbal Tea 凉茶 (Refillable) (3.00/each) 3.00 Item count: 2 SubTotal 18.62 Total discount -0.00 Amount 18.62 Rounding -0.02 Final Amount(MYR) 18.60 Payment method E-Wallet Payment received 18.60 Change 0.00 Thank you, please come again	 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Tap to expand OCR Result Merchant: 202403056901 (003581349-H) , Lot GR02 FE 03 One C Amount: 18.62 Date: 23/04/26 Cancel Use
R_R2	 BROTHER CYY TYRE SDN. BHD. No. 21 & 21A, Jalan Siswa, Taman Kampar Siswa, 31900 Kampar, Perak. H/P: 010-373 1069 CASH SALE BGV7475 No. Particulars Quantity Unit Price Amount RM sen 1. VR28 ENGINE OIL 1/8gf 170 170 2. OIL FILTER 總计 Total RM 170 Goods sold are not returnable. Note: All cheques should be crossed and made payable to BROTHER CYY TYRE SDN BHD (Public Bank: 3226837007) 收貨人 Received by 經手人 Issued by	 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Takeout Snacks Supper Groceries Fruits Tap to expand OCR Result Merchant: 16. Cancel Use

R_R3	 Hakka Yin Cafe HAKKA YIN CAFE Registration No: 202303248990 (003534961-U) No.9, Jalan Ampang, 31900, Kampar, Perak, Malaysia (Whatsapp For Order 011-16644770) Invoice no: 32719 ORDER Date: 03/04/2026 19:28 Table Pax: 1 2 Qty Item Price (MYR) 1 SD04 清汤粉Soup Noodle Set (9.90/ea) 11.90 • N03 河粉 Kueyteow • 不要芽菜 No Beansprout • 加河粉addon kueyteow + 2.00 1 R04 凉水 (冷 Cold) (3.00/ea) 3.00 2 Qty Subtotal 14.90 Bill rounding 0.00 Total (MYR) 14.90 Balance 14.90 Don't forget to rate us! Scan QR code to let us know how you enjoyed with us. ...u for visiting us.	 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Table Pax: 1 2 Qty Item Price (MYR) 1 SD04 清汤粉Soup Noodle Set (9.90/ea) 11.90 • N03 河粉 Kueyteow • 不要芽菜 No Beansprout • 加河粉addon kueyteow + 2.00 1 R04 凉水 (冷 Cold) (3.00/ea) Tap to expand OCR Result Merchant: Registration No: 202303248990 (003534961-U) Amount: 14.90 Date: 03/04/2026 Cancel Use
R_R4	 Chat A Toast LOT 40249 Jalan Tunku Putera Bandar Baru Kampar 31900 Perak Table : B18 Bill No.: CINV-01492 Date : 2026-04-28 09:07:26 1) White Coffee , [Cold] 白咖啡 , [冷] 1 X 3.00 3.00 2) Company Sandwich 公司三文治 1 X 8.90 8.90 Please Pay RM 11.90 Thank You Please Come Again Sally 2026-04-28 09:07:26	 Add Transaction Expense Income Food Souvenirs Shopping Transport Balance Interest Repayme Housing Movies Flowers Tap to expand OCR Result Merchant: Sally 2026-04-28 09:07:26 Amount: 11.90 Date: 2026-04-28 Cancel Use

APPENDIX

R_R5	 	
R_R6	 	

R_R7	 	
R_R8	 	

R_R9	 	
R_R10	 	

C. User Acceptance Testing(UAT) Form Feedback

This appendix presents the User Acceptance Testing (UAT) form used to collect feedback from testers. The questionnaire was designed using Google Forms and includes both quantitative and qualitative questions to evaluate system usability, performance, and user satisfaction. The response summary and raw data are also included for verification purposes.

C.1 Google Form Questionnaire Design

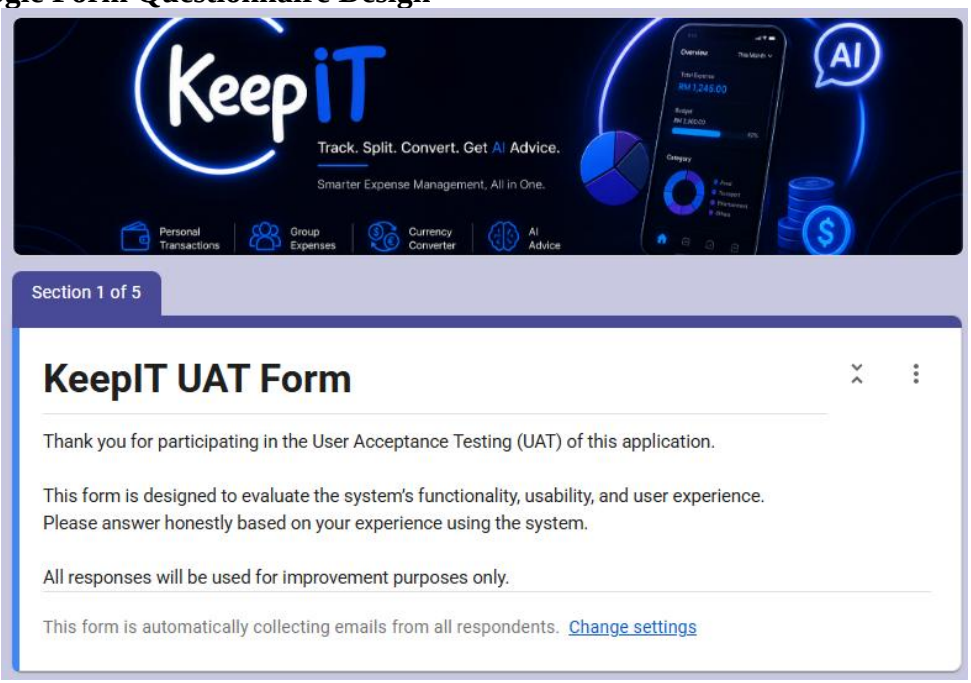
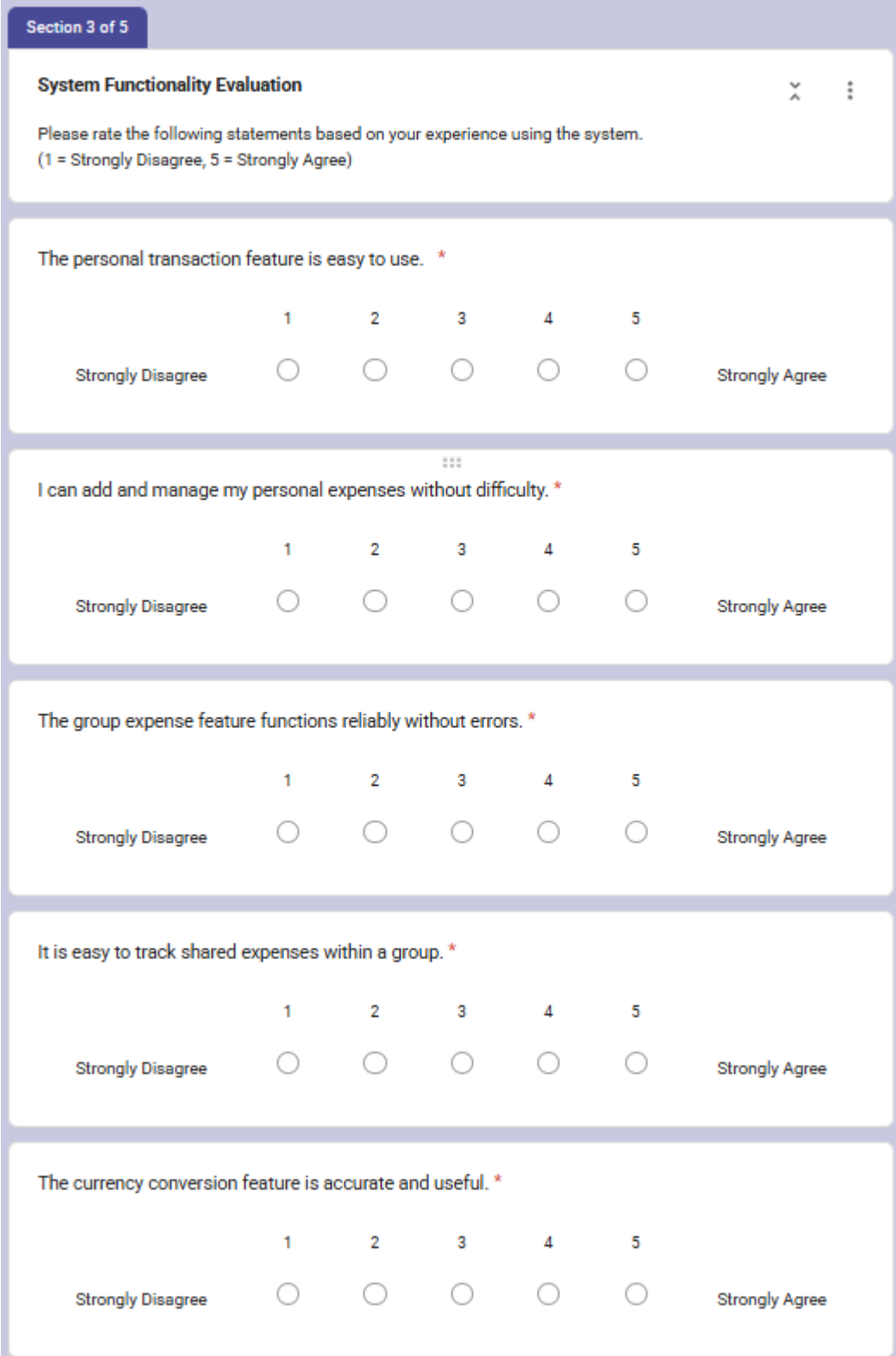


Figure C.1.1: Google Form Questionnaire Design 1



Section 3 of 5

System Functionality Evaluation

Please rate the following statements based on your experience using the system.
(1 = Strongly Disagree, 5 = Strongly Agree)

The personal transaction feature is easy to use. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I can add and manage my personal expenses without difficulty. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The group expense feature functions reliably without errors. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

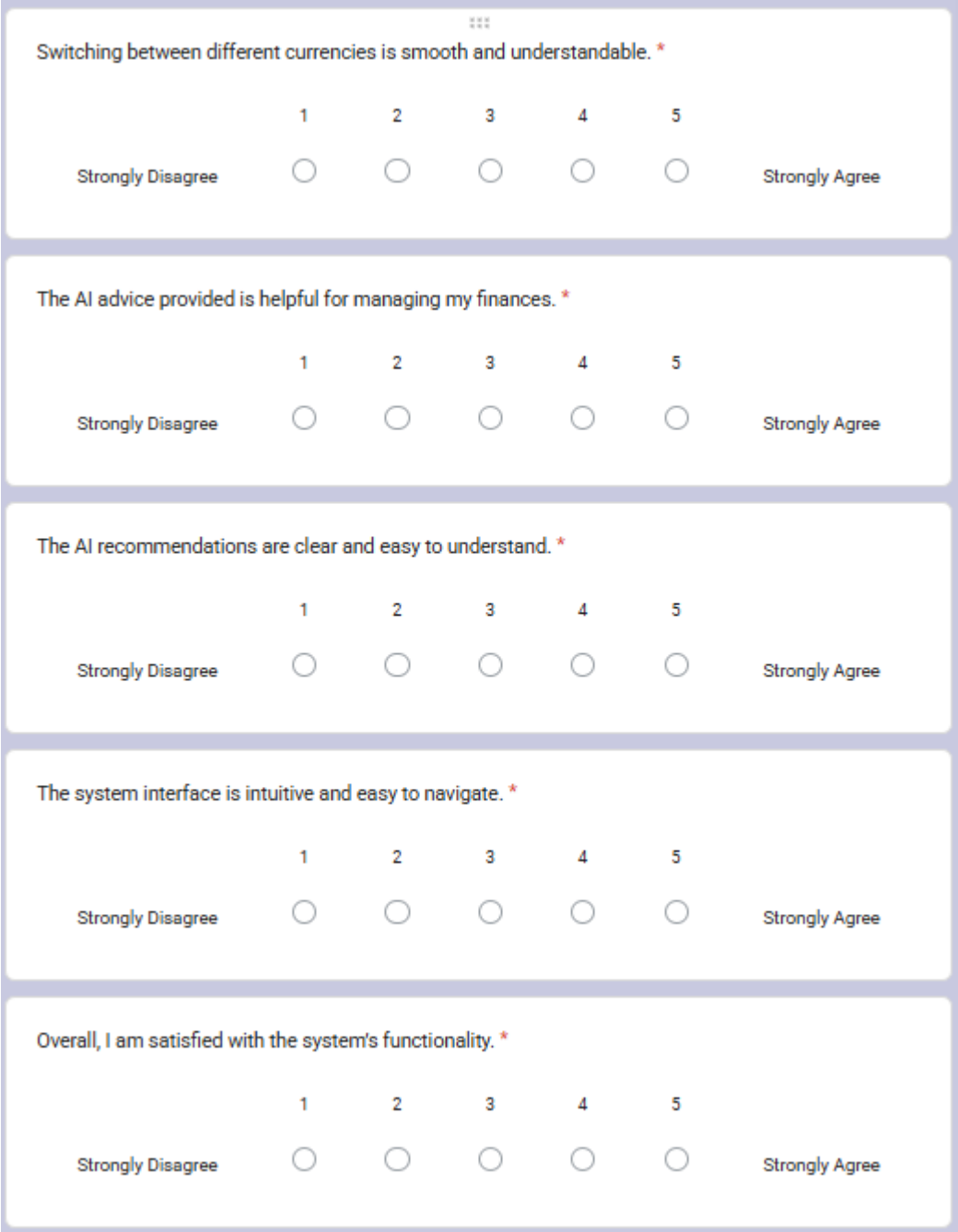
It is easy to track shared expenses within a group. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The currency conversion feature is accurate and useful. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure C.1.2: Google Form Questionnaire Design 2



Switching between different currencies is smooth and understandable. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The AI advice provided is helpful for managing my finances. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The AI recommendations are clear and easy to understand. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

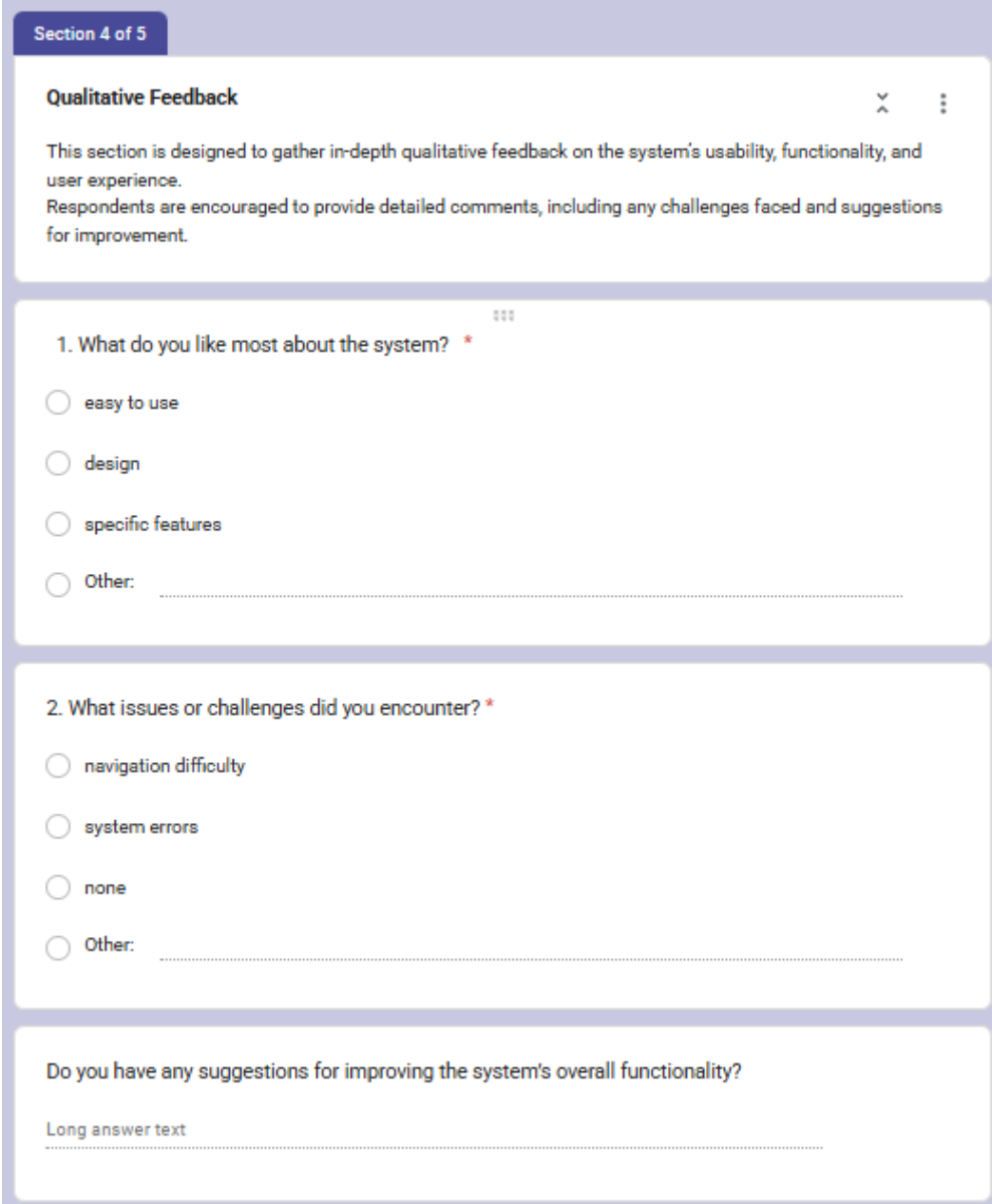
The system interface is intuitive and easy to navigate. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Overall, I am satisfied with the system's functionality. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure C.1.3: Google Form Questionnaire Design 3



Section 4 of 5

Qualitative Feedback

This section is designed to gather in-depth qualitative feedback on the system's usability, functionality, and user experience.
Respondents are encouraged to provide detailed comments, including any challenges faced and suggestions for improvement.

1. What do you like most about the system? *

- easy to use
- design
- specific features
- Other: .....

2. What issues or challenges did you encounter? *

- navigation difficulty
- system errors
- none
- Other: .....

Do you have any suggestions for improving the system's overall functionality?

Long answer text .....

Figure C.1.4: Google Form Questionnaire Design 4

C.2 Google Form Response Summary

Figure C.2.1: Google Form Response Summary 1

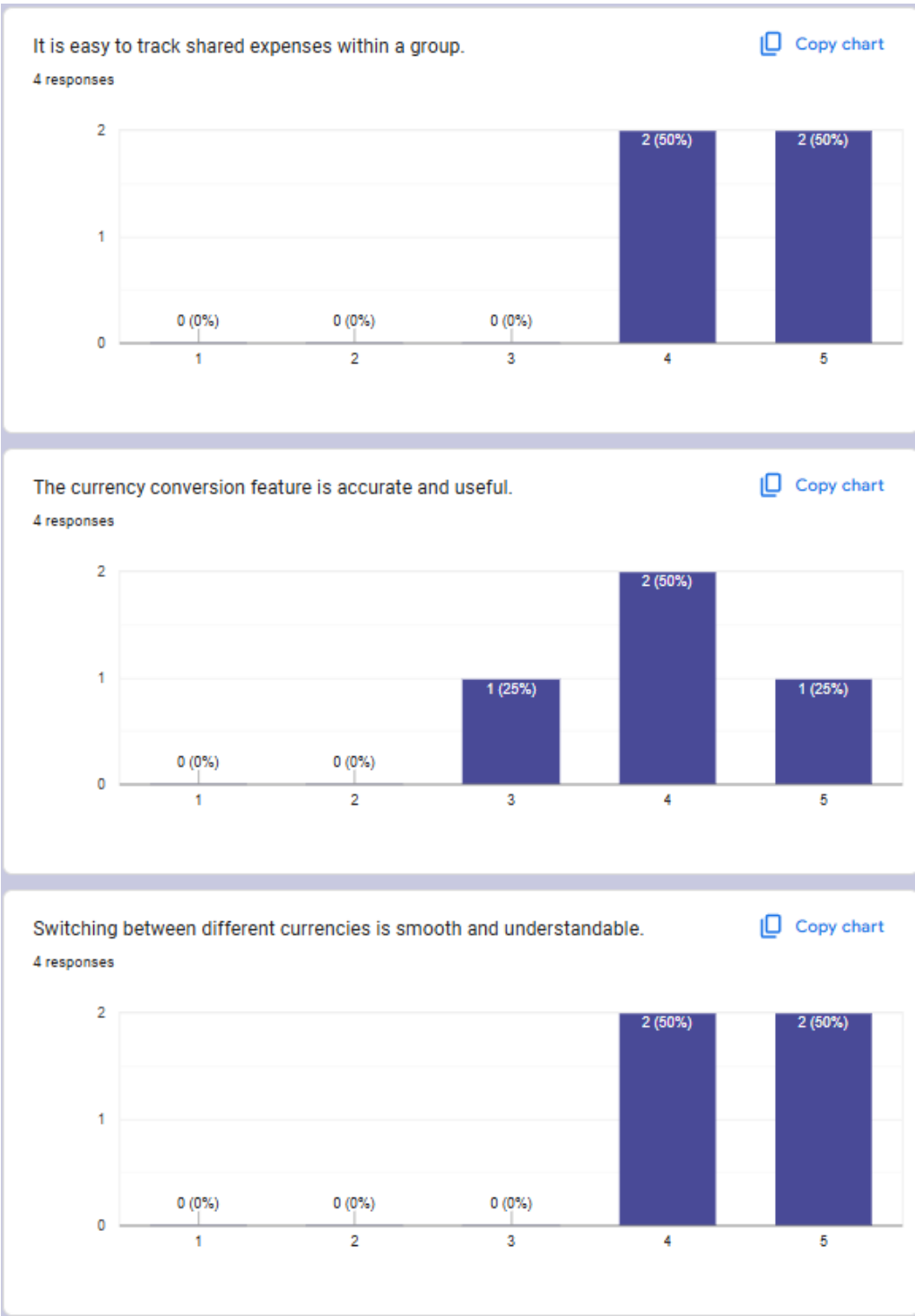


Figure C.2.2: Google Form Response Summary 2

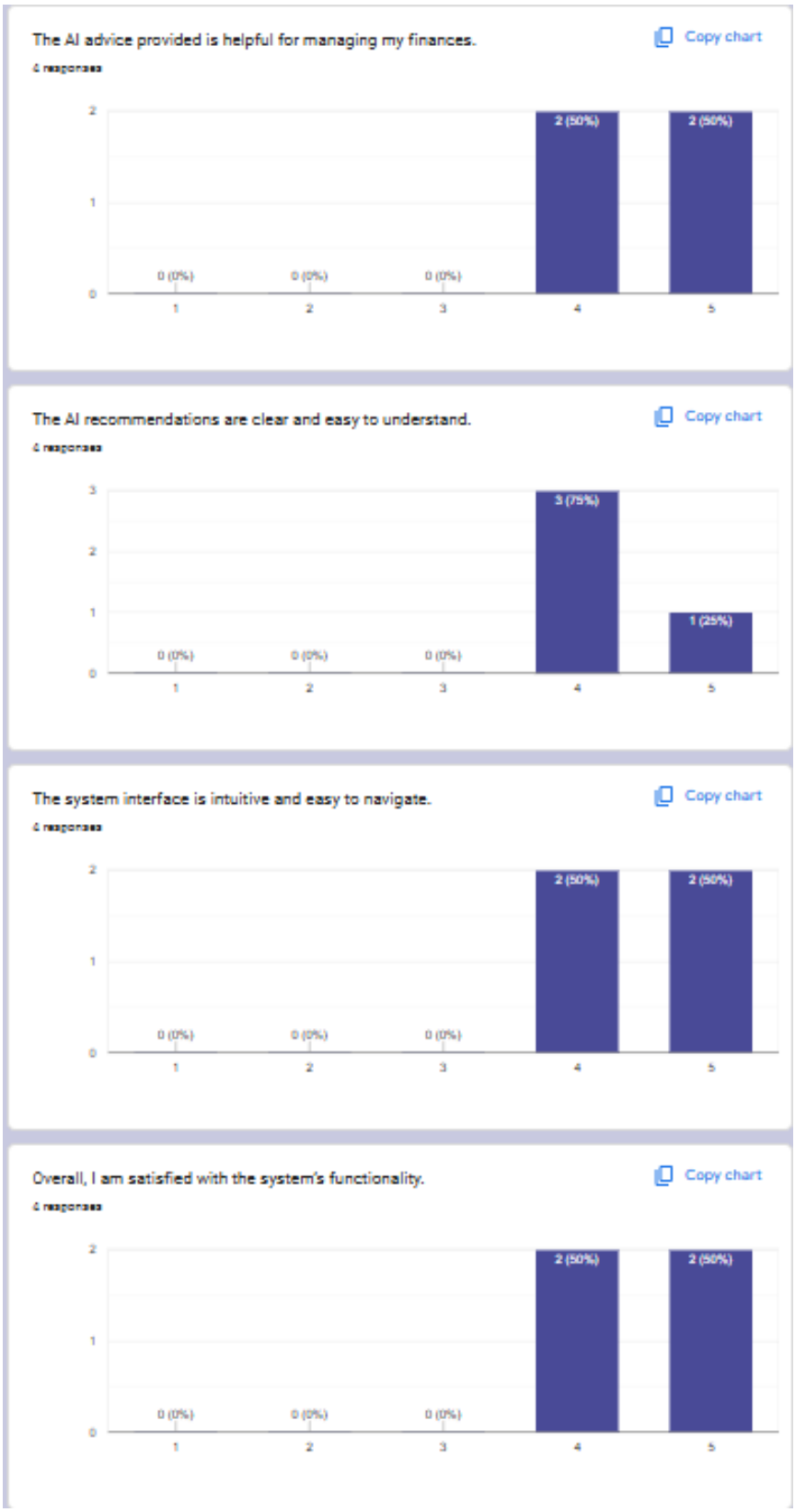


Figure C.2.3: Google Form Response Summary 3

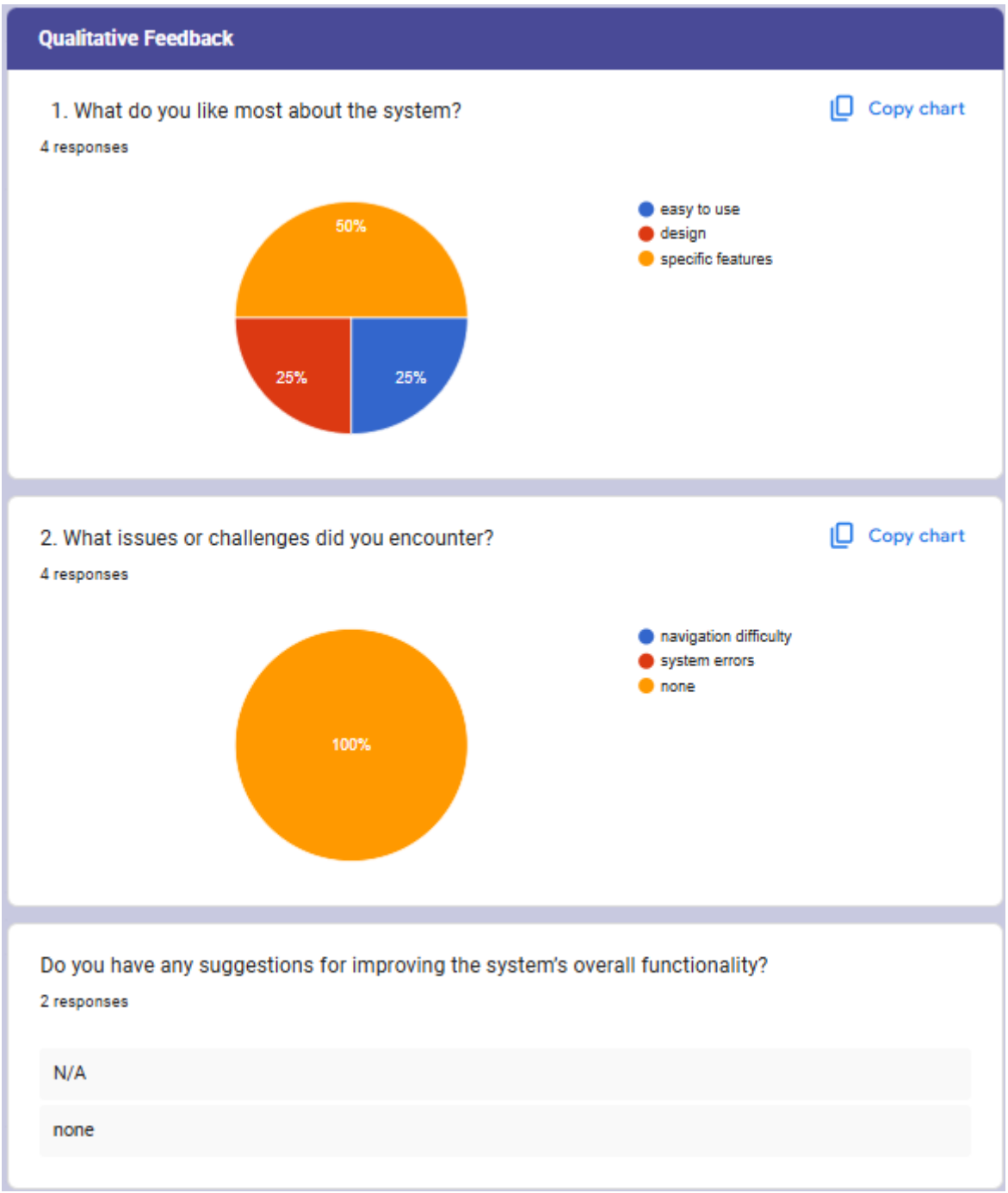


Figure C.2.4: Google Form Response Summary 4

C.3 User Declaration and Digital Signature

The inclusion of a declaration section ensures that all responses are based on actual user experience and enhances the reliability of the collected data.

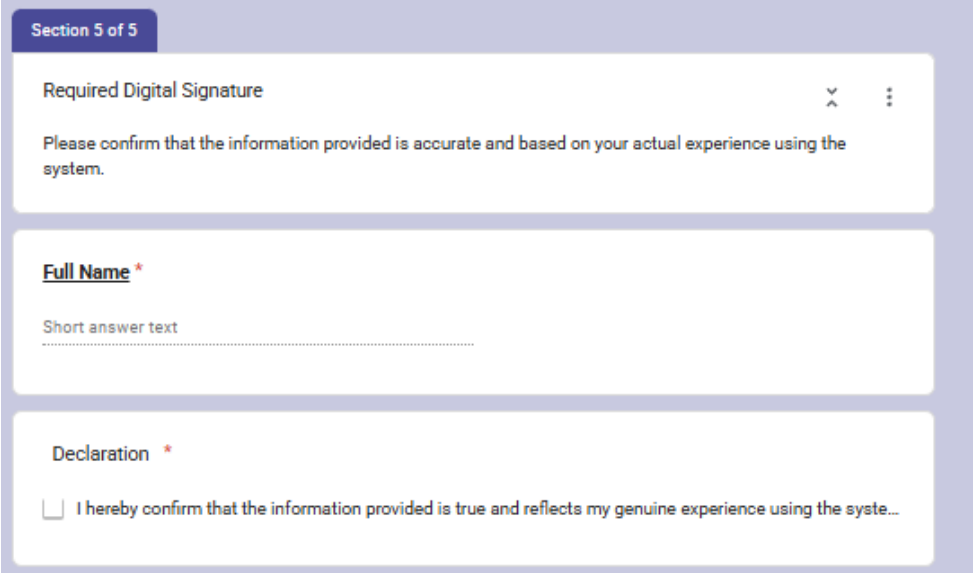


Figure C.3: User Declaration and Digital Signature Section in UAT Form