MOBILE RECEIPT SCANNER FOR EASY EXPENSE SHARING

KWA KHAI MUN

UNIVERSITI TUNKU ABDUL RAHMAN

MOBILE RECEIPT SCANNER FOR EASY EXPENSE SHARING

KWA KHAI MUN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science Software Engineering with Honours

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

September 2025

DECLARATION

Name	:	Kwa Khai Mun
ID No.	:	2106348
Date	:	3 March 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Nawaf Hassan Mohammad Mohsen for his valuable advice and guidance throughout the course of this entire project. His extreme patience and persistence has helped me through the hard times of the project.

I would also like to express my appreciation to Lee Kong Chian Faculty of Engineering and Science (LKCFES) for providing me with the resources and opportunity to complete this project flawlessly.

In addition, I am deeply thankful to my friends, whose observations during a vacation trip led to the identification of the research problem. Their support, encouragement, and constructive discussions have been fundamental to the development and completion of this final year project.

Additionally, I am deeply thankful for my friends, who gave me the ideas to start the whole project during a vacation trip. Their support, feedback and encouragement gave me all the energy I needed to complete this project with ease

Lastly, I am grateful for my family for the continues mental support and encouragement given to me in order to finish my project.

ABSTRACT

The development of a bill splitting application is due to the increased complexity of shared cost dsitribution among social groups which require manual calculation to be done. This project presents a mobile application to simplify the process of bill splitting by implementing Optical Character Recognition (OCR) and Artificial Intelligence (AI). The development of the application will be guided by Rapid Application Development (RAD) while employing Tesseract as the OCR framework with DeepSeek as the AI tool to help with the success of this project. Additionally, the application is built with the React Native framework pairing it with MySQL database to manage user, events, and item tables. Crucial features regarding the application is multiple splitting methods which are the "equal" method and "pay for what you eat", real-time collaboration among participants to help out with the cost management, and also AI-based calculation to help calculate and summarize the event's cost. This application offers user a practical and user friendly solution to tackle one of the common real-world challenges. The application is easily scalable with future enhancements through different technologies such as advanced machine learning and integration with popular payment platforms. The project concludes with combining OCR and AI technologies to provide an user friendly approach to allow user collaborate with each other to manage their expenses in an efficient manner.

Keywords: mobile application; optical character recognition; artificial intelligence; cost management; bill splitting

Subject Area: QA76 Computer Science

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

AI	Artifiicial Intelligence
OCR	Optical Character Recognition
CRNN	Convolutional Recurrent Neural Network
EAST	Efficient and Accurate Scene Text
RAD	Rapid Application Development

.

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Dining out with friends, family, or colleagues has always been a common social activity for everyone. Everything is well and good, however when it comes to splitting the bill, hassle comes in especially when it comes to splitting the bill with multiple people which leads to time-consuming and tedious. Most of the community will use the traditional way to split the bill which sometimes leads to error, confusion and even disputes in the group.

With the current technology, many opportunities to solve this problem especially in the image processing, machine learning and artificial intelligence field. The process can be summarized by user taking picture of the receipt and the system will auto detect items and prices of item with the user being able to his or her group together to split the bill together in a fair manner. This can significantly improve the experience of splitting bills making it more efficient and enjoyable for everyone involved.

The function of this application allows users to take pictures of the receipt for the system to detect the item's name, price and quantity using Optical Character Recognition (OCR). User can also manually add in more item in case the OCR did not work as expected or there is more items to be added. Then user can assign participants in the item indicating the item belongs to the specific participant. One item can have one or more participants. The application will also provide two types of calculation method which includes "Split the bill evenly" and "Calculate all the meals and charges based on what the person ate" to provide the user more option on how they want to split the bills.

1.2 Importance of the Study

The motivation behind this application's development is the need for effective and user-friendly solutions of bill splitting among the public. Everyday, individuals including myself regularly be in the situation where expenses need to be divided by everybody no matter if its dining out, or going on a trip. However, existing applications often comes with challenges such as non user-friendly interface and inefficient calculations which does not solve the real problem of user instead increases frustration.

Additionally, the motive of this application is to prevent inefficient way of bill splitting and promote fairness during the process. One of the burdon faced by user is manually entering receipt details which includes the item name, price and quantity which is tedious especially if the receipt is multiple or long. A camera based OCR can easily help solve this existing frustrating problem. Other than that, having multiple and efficient option will greatly increase fairness among participants as some people may order more expensive items that others. Without a propert way to allocate the cost fairly, user will normally go with the traditional method which is prone to mistake that can lead to problems and misunderstandings amongst participants.

1.3 Problem Statement

1.3.1 Lack of Camera Function

Many existing application requires the users to manually insert information into the application, which can be time consuming and frustrating especially if there are multiple receipt and the receipt is exceptionally long . According to Taylor (2018), Optical character recognition (OCR) technology is existing apps does not work as expected, thus making the whole process confusing and adding even more work. This is because there are many types of receipt format and some can be challenging for OCR to detect.

1.3.2 Lack of Multiple Type of Bill Splitting Option

Ong (2016) stated that there are many types of ways to split the bill for example, “Split the bill evenly” which is the easiest way to split but is the least fair amost all because some may had more and some may had less during the meal. Next is “Calculate all the meals and charges based on what the person ate”, this seems to be the most fair way to calculate but it does comes with some complication for example when two or more person are sharing the same appetiser or dessert as some bill splitting application does not have the complex calculator to calculate such complicated requirements

1.3.3 Lack of Artificial Intelligence (AI) integration

Even when the bill is taken with OCR, how the information that was captured matters, To properly and accurately interpret the information from the receipt after OCR is done, the integration of AI can be done to help with the consistency of the application.

1.3.4 Lack of Invitation Function For Other Participants

Lack of invitation function for other users. All existing bill-splitting applications does not offer the feature whereby multiple users can collaborate on a single project. A collaborative feature can enable a group of users to join in a bill splitting project where they can add on their part of receipt and specify which item they consumed which can significantly improve accuracy and convenience. Without this feature, users are needed to communicate manually and one person in charge will take full responsibility if any calculation are made wrong.

1.4 Aim and Objectives

1. To employ camera function for Optical Character Recognition to avoid unessacary manually key in and increase efficiency.
2. To incorporate multiple type of calculation option for bill splitting to ensure fairness and user control.
3. To integrate Artificial Intelligence into the application for increased accurate and consistent receipt reading.
4. To facilitate the option to invite other participants to join with the calculation for efficient collaboration.

1.5 Scope and Limitation of the Study

The scope of this study includes the following

- The application will only support English as its only supporting language. All text that will appear in the application will be in English only

- The application will only use Malaysian Ringgit (MYR) as the only currency available, therefore Malaysian will be the primary user for this application.
- The application will only function in online mode. A Wi-Fi connection will be needed to use the application as AI will be implemented and Wi-Fi will be necessary.
- The OCR feature will only support English written receipts to avoid complication.

The scope of this study will exclude the following

- The application will not support any other language other than English, application will need user to be able to read and interpret English language in order to proceed.
- The application will not support any other currency other than Malaysian Ringgit, the user's receipt in other currency will not be interpreted by the system.
- The application will not support offline mode as Wi-Fi connection is required for the AI implementation to work.
- The OCR function will not support to interpret receipt written in other language other than English, limiting its function for multilanguage receipts

1.6 Contribution of the Study

The contribution this study provides is by introducing a new mobile application that is able to provide a practical and fair solutions to common challenges the community are facing currently such as splitting the bills among group at dining, trips and events. By implementing OCR in the application, the application's user can simply take a picture of the receipt instead of manually entering details which can both save time and reduce errors.

1.7 Outline of the Report

The report will discuss about many sections of the projects including the literature review, in this section. Deep research will be made about everything, similar projects, the tools to use and methodologies. On top of that, the methodology section will discuss about the methodology used in this project and the overall initial planning of the project which gives us a good picture of the precise planning for the project. Next, the preliminary results section will talk about the first half of the project, the planning made into eh use case diagram, use case description and prototype of the project are done before developing the actual application to let the developer have a better picture of the end project will look like. The system design section explains how the backend of the application will look like such as the database structure and activity diagram of the application. Other than that, the next section will be system implementation where the application is built, the tools used and well explanation of the each of the applications' features interface will be discussed. Finally is the final chapter, the testing section where it is straightforward, discussing how the testing are made during development to ensure that the application is free of bugs and downtime.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

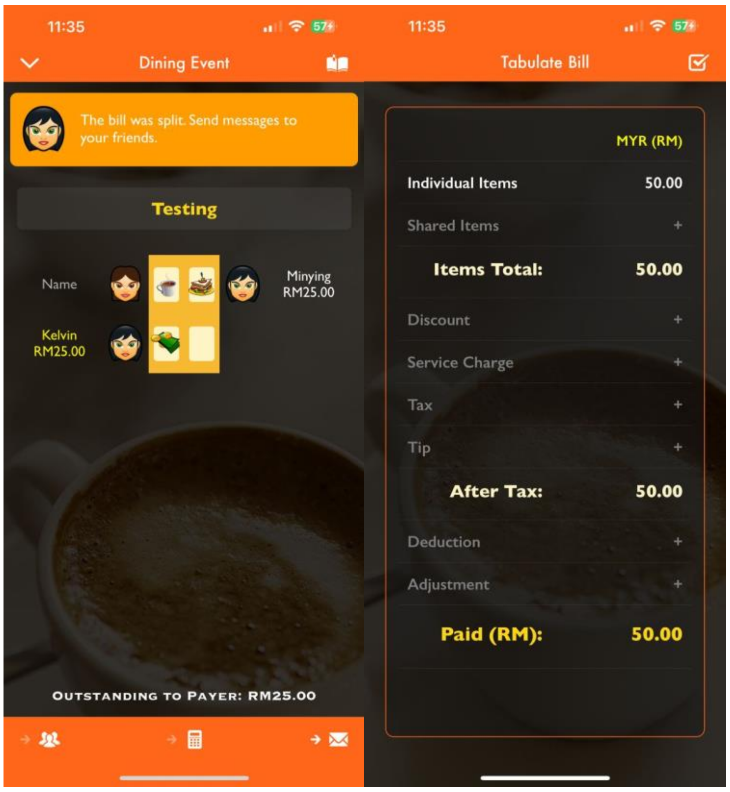
To develop an efficient and high usability system, a comprehensive literature review is crucial to understand the existing system knowing their limitations therefore being able to improve it and to choose the most suitable technology. The literature review will consists of in-depth analysis of various factor contributing to the system development which includes existing systems, development methodologies, and essential tools for development.

Firstly, an research on existing systems is carried out to understand their functionalities and identify potential issues or limitations by the application. By understanding these application, insights about their strengths and weaknesses can be analyzed, which will help in designing an improved solution. A comparison of these existing application is also provided to highlight key differences and areas for improvement. Secondly, various software development methodologies are discussed to determine the most suitable methodology approach for the project with a clear insight of their advantages and disadvantages. Understanding these methodologies ensures that the chosen approach aligns with project requirements and the flexibility needed to work. Lastly, an assessment of development tools is conducted, analyzing frameworks, databases, Artificial Intelligence (AI), and Optical Character Recognition (OCR). Each tool is analyzed in terms of its advantages and limitations to make informed decisions regarding its functions and usefulness for the project. By understanding these tools, the objective is to select the most effective solutions that will lead to the overall success of the project.

This literature review provides the necessary insights to make informed decisions in system development, ensuring that the final product is both efficient and suitable for user needs.

2.2 Existing Systems

2.2.1 iGoDutch



The bill was split. Send messages to your friends.

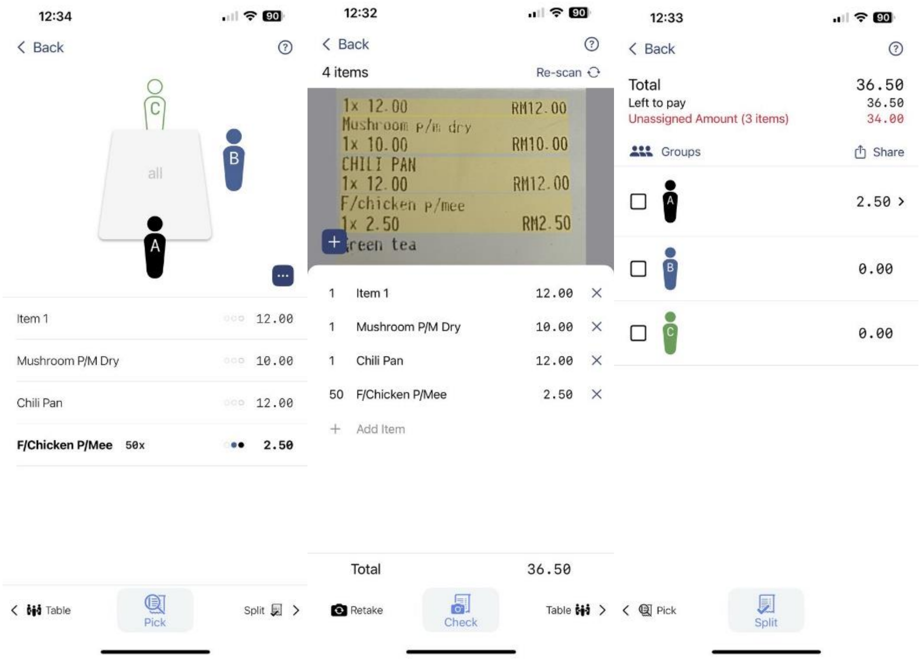
OUTSTANDING TO PAYER: RM25.00

	MYR (RM)
Individual Items	50.00
Shared Items	+
Items Total:	50.00
Discount	+
Service Charge	+
Tax	+
Tip	+
After Tax:	50.00
Deduction	+
Adjustment	+
Paid (RM):	50.00

Figure 2.2.1.1 Screenshot of iGoDutch app

IGoDutch is a free-to-use application used to help user calculate the user's expenses sharing with the option to upgrade into a more premium version with RM9.90. One of its unique functions is that user can give a default setting for themselves such as "Usual Tipping Rate" and a unique avatar. Other participant's information can be pre-added to save time for next event. The good functions that are provided by the is many components are available for user to enter such as additional discount, service charges, and even adjustment after the total deducting tax. However, IGoDutch lacks functions to make the app more efficient such as camera function to take pictures of receipt to extract information using machine learning. Also, the application does not integrate Artificial Intelligence in its calculation and lack the functions to invite other user to join in the calculation which makes it job heavy for the user. The user interface can also be hard to navigate and will need a little time to get used to it as there is many buttons and inputs. It also lacks AI integration and different typs of calculation method for the user to choose from.

2.2.2 Splitter



Item 1		12.00
Mushroom P/M Dry		10.00
Chili Pan		12.00
F/Chicken P/Mee	50x	2.50

4 items

```
1x 12.00              RM12.00
Mushroom p/m dry
1x 10.00              RM10.00
CHILI PAN
1x 12.00              RM12.00
F/chicken p/mee
1x 2.50               RM2.50
green tea
```

1	Item 1	12.00
1	Mushroom P/M Dry	10.00
1	Chili Pan	12.00
50	F/Chicken P/Mee	2.50

Add Item

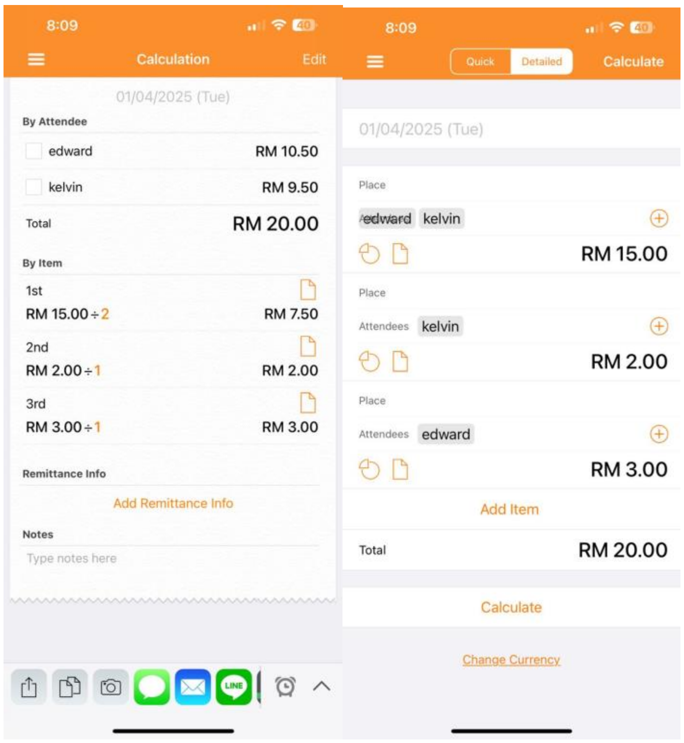
Total 36.50

Total	36.50
Left to pay	36.50
Unassigned Amount (3 items)	34.00
A	2.50
B	0.00
C	0.00

Figure 2.2.2.1 Screenshot of Splitter app

Splitter is a cost splitting application with great interface and easy to navigate which makes it easy for beginners to use it. It also has a camera function to take picture of the receipt to ease the process, additional items can be added and edited if needed. Users will differentiate alphabetically from A to Z. At the end of the calculation, a summary will be generated, every participant will need to remember their alphabet as it represents them. But Splitter is unable to track additional tax and tips that is given in the receipt. Additionally, the app does not allow user to invite another user to collaborate which makes it a one man show, and it does not provide multiple calculation options.

2.2.3 NPang



8:09

Calculation Edit

01/04/2025 (Tue)

By Attendee

edward	RM 10.50
kelvin	RM 9.50
Total	RM 20.00

By Item

1st RM 15.00÷2	RM 7.50
2nd RM 2.00÷1	RM 2.00
3rd RM 3.00÷1	RM 3.00

Remittance Info

Add Remittance Info

Notes

Type notes here

8:09

Quick Detailed Calculate

01/04/2025 (Tue)

Place

edward kelvin

RM 15.00

Place

Attendees kelvin

RM 2.00

Place

Attendees edward

RM 3.00

Add Item

Total RM 20.00

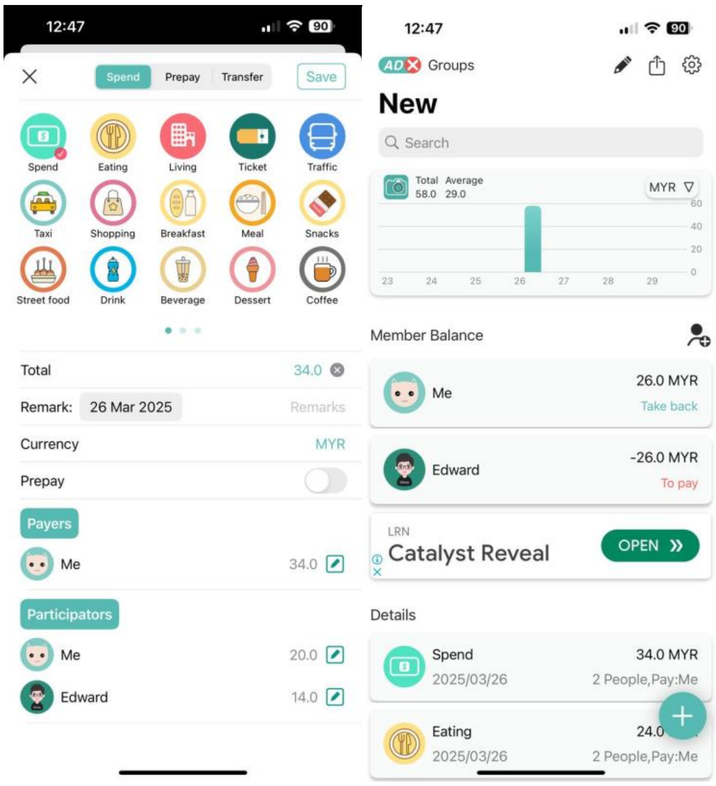
Calculate

Change Currency

Figure 2.2.3.1 Screenshot of NPang app

NPang is a simple to use cost splitting application with easy-to-understand interface, there are two types of calculation given by the application which is quick and detailed with quick being equal splitting. For every item, a picture of the item or the receipt can be taken too as a reference. However, camera function that integrate Optical Character Recognition does not work for NPang with requires user to input all the item’s price and description. Additionally, the calculation for the algorithm does not function fully as tax and tips cannot be calculated. Artificial Intelligence is also not included in the app to make it more efficient and the function of being able to invite other collaborators are not available in this application so one person must handle all the receipt.

2.2.4 GoDutch



Total 34.0
Remark: 26 Mar 2025 Remarks
Currency MYR
Prepay
Payers
Me 34.0
Participators
Me 20.0
Edward 14.0

Total 58.0 Average 29.0

Me 26.0 MYR Take back
Edward -26.0 MYR To pay

Spend 2025/03/26 34.0 MYR 2 People,Pay:Me
Eating 2025/03/26 24.0 2 People,Pay:Me

Figure 2.2.4.1 Screenshot of GoDutch app

GoDutch has a colourful interface with many useful functions to help the user with cost splitting. It works more as an interactive dashboard for the user to track their expenses and how much participants owes them. Users can enter item adn the app will help calculate the expenses for them and can have a record on how much a specific participant owes them with the beautiful display. However, it lacks camera function to apply Optical Character Recognition in the application. Other than that, Artificial Intelligence are not integrated into the application to make it more efficient. It also lacks the functions of inviting other collaborators to calculate together the cost which will be a very good idea with the nice display.

2.2.5 Comparison

Table 2.2.5.1 Comparison Table for Existing Systems

Application	Advatange	Disadvantage
iGoDutch	• Provide in-depth calculation for user.	• Lack camera function for OCR.

		• Lack of different type of calculation for cost calculation. • Lack AI for increased efficiency. • Lack collaboration function. • Hard to use interface
Splitter	• User friendly interface. • Camera function for OCR function. • Easy to understand concept therefore easy to learn	• Lack of different type of calculation for cost calculation. • Lack AI for increased efficiency. • Lack collaboration function.
NPang	• Easy to understand • Picture of receipt can be taken as reference. • Two types of calculation are given for diversity.	• Lack camera function for OCR. • Lack AI for increased efficiency. • Lack collaboration function.
GoDutch	• User friendly interface with many colours. • Interactive dashboard • East to learn	• Lack camera function for OCR. • Lack of different type of calculation for cost calculation. • Lack AI for increased efficiency. • Lack collaboration function.

Each of the application discussed have their own unique strength. IGoDutch provides in-depth calculation in its application for payment that has many components making it useful for accurate cost breakdown. Splitter on the other hand stands out on its user-friendly interface followed by its OCR functionality to increase the efficiency of the application. The easy-to-understand flow makes it very user friendly. Additionally, NPang gives the user a straightforward flow, also allowing users to take pictures of the receipt but for reference only. It also gives the user two different type of calculation which gives the user more flexibility. Finally, GoDutch focuses on delivering a colorful interactive dashboard to increase the user's experience. The design makes it nice and enjoyable.

Despite all their unique traits and strength, they share several similar weaknesses. None of the application above integrated AI to improve the application's overall efficiency. They also lack collaboration feature which limits usability of multiple user handling the cost at the same time. Also, they lack multiple calculation method which reduces the flexibility of the apps. Additionally, the apps do not have OCR function except for Splitter which will be a very great function to improve the application's overall usability.

2.3 Development Methodologies

2.3.1 Agile Development Methodology

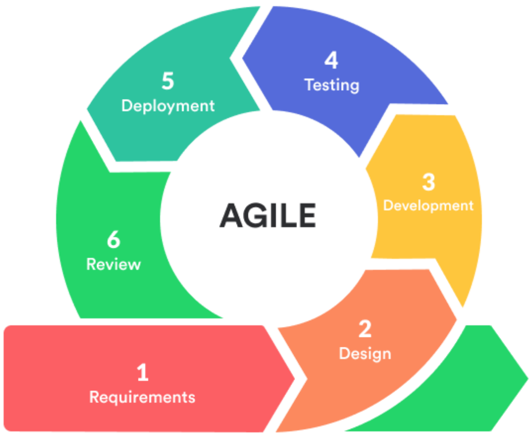


Figure 2.3.1.1 Process of Agile Development

Agile Development which is also referred as a lightweight software development method can solve traditional software development methods which includes planning the project plan in advance, fully written software's requirements and building the code and design according to the requirements. According to Alsaqqa, Sawalha and Abdel-Nabi (2020), there are four values the agile method can provide. First, Individuals and interactions over processes and tools. This value in the manifesto explains that focusing in processes and technical problem is less important than priotizing communication and interaction of the developers. The second value is working sogftware over comprehensive documentation. While documentation is important, it is time and effort used that will slow down the process especially if requirements are changed frequently. The manifesto explains that real progress is measured by testing the software as it provide feedback rather than intensive documentation. Other than that, the third value is Customer collaboration over contract negotiation. Because agile development is design to change with the changing requirements at any point. Customer feedback and negotiation are much more important that solely relying on formal agreements. Lastly, Responding to change over following a plan. Developers and customer will understand the system more as the development progresses which will lead to possible changes in requirements. Agile prioritise adapting to the changes rather than following a pre-fixed plan to achieve customer satisfaction.

Agile Development offers many advantages such as its flexibility, allowing teams to respond fast with changing requirements at any stage of the development process. Agile also promotes teamwork, collaboration and teamwork between stakeholders to ensure that leads to a more user-centric product. Also, the continuous feedback will make sure that the quality of the product will be maintained and improved to align with the user needs. However, the disadvantage that comes with Agile is having frequent user involvement which there is a possibility that not all stakeholders will be flexible with their availability. Second disadvantage is being overly complex if the project is in a large scale as there is much more components to adapt which will lead to a domino effect. Lastly, constant changes and addition in requirements can lead to scope creeps where the project expands bigger than its original requirements that can evtually lead to overbudget and over-extending time.

2.3.2 Rapid Application Development Methodology

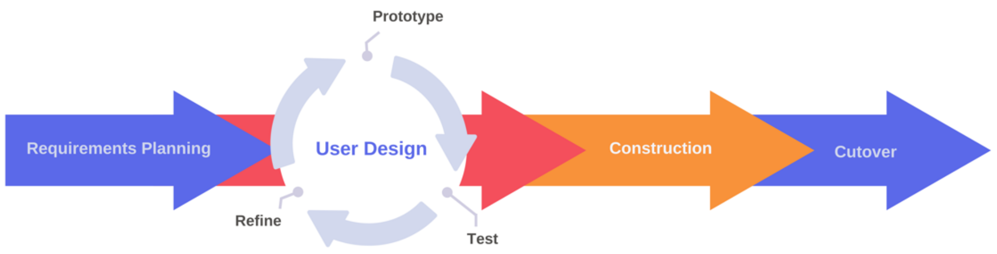


Figure 2.3.2.1 Process of Rapid Application Development

According to Pradana, Andrianto and Auliya (2022), we can understand that Rapid Application Development also known as RAD is a software development methodology designed to create functional system in short amount of time by combining techniques and prototyping. RAD consists of an initial prototype as a foundation to gather user requirements efficiently. It has four primary phases, Requirement Planning, User Design, Construction, and Cutover. At the Requirement Planning phase, the developers will collaborate with the stakeholders to understand the projects's requirements and scope. Next is the User Design Phase whereas the initial prototype is built, test, and refined based on the user's feedback ensure that the project not only align with the requirements but also with actual user needs. Then it is the Constuction phase where it will involve the developer's quick coding, testing and integration of component, capatalizing on existing reusable modules to fasten the development process. Finally, the last phase is Cutover phase which is also known as Deployment phase. It consists of final testing and deployment ensuring that the project is full functional and ready for the intended user.

The benefits of RAD is a fast development cycle which allows a quick delivery of the finished product. RAD also promotes active user involvement through the process to make sure the end product is parralel with the user needs and satisfaction. Also, RAD is is very easy to adapt with changes thus increasing its flexibility. However. It may not be suitable for large and complex system as it requires high volume of planning and a structured approach. Also the success rate of RAD depends highly on the developers skill who can work and handle stress in the fast-paced environment.

2.3.3 Waterfall Model Methodology

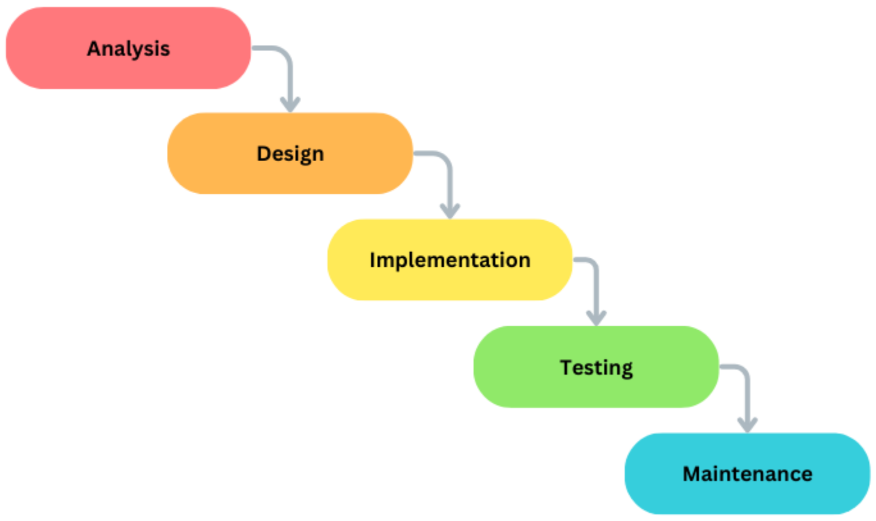


Figure 2.3.3.1 Process of Waterfall Model

Waterfall Model is the first ever model that is introduced in software development. It follows a lenear approach in which each phase must be completed before the next one, with no phases developing parraleling thus making it simple to understand and use globally in Software Engineering. Each phase contains an input that is the output from the last phase. Pradana, Andrianto and Auliya (2022) stated in their research that there are 5 total phases which is Analysis Phase, Design Phase, Implementation Phase, Testing Phase, and Maintainence Phase. Firstly is the Analysis Phase which is also known as Software Requirements Specification (SRS). Stakeholders will state out the functional and non-functional requirements, ensuring that everyone understands how the project is going to end up. Next is Design Phase, the developers and designers will then plan a system containing algorithm, architecture, database and framework for the project. Implementation Phase is up next where the developer and designers will then work on the actual coding part, working parralel with the requirements and the system itself. The next phase will then be the Testing Phase, the system will go through debugging and changing to ensure that the system meets with the initial requirements. Finally is Maintainence

Phase which deploys the system and changed to fix errors and to meet new requirements.

The advantage of Waterfall Method is all the team members have a clear understanding of what needs to be done which will reduce confusion and prevent scope creep. Other than that, Project Manager can easily monitor progress and ensure that all resources are allocated nicely. Additionally, the costs and timeline of the project can be accurately predicted making it ideal for project with strict deadlines and budget. However, the disadvantage includes lack of flexibility where if a requirement changes midway, major changes will need to be made which will lead to major delays and costs. Also, this methodology is not suitable for project that has unclear and evolving requirements. Next, the end product will only be delivered at the end of the development cycle therefore no feedback from user can be taken early. If there is changes that needs to be made, major rewords may be required.

2.3.4 Prototyping Model Methodology

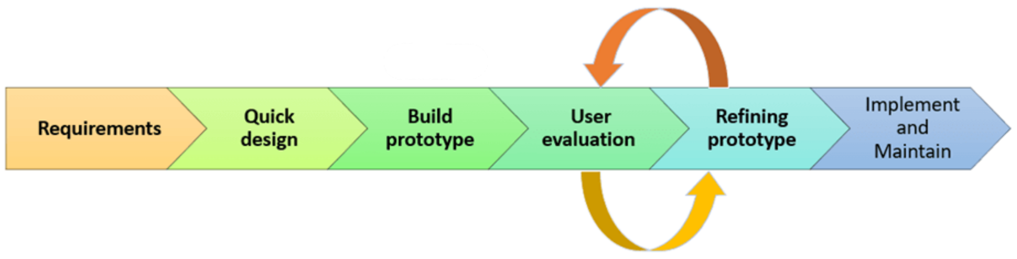


Figure 2.3.4.1 Process of Prototyping Model

The next methodolodgy to be understood is Prototyping Model, Chandran and Das Aundhe (2021) stated in a research that Protyping Model is where a prototype are built, tested and changed until the point where the end product meets project requirements. This is suitable for project that has no clear project requirements also allowing developers and stakeholders to collaborate. There are a total of six phases which is Requirement Gathering, Quick Design, Build Prototype, User Evaluation, Refining Prototype, Implement and Maintain. In Requirement Gathering, it involves doing research and collecting the project's requirement or interview between developers and stakeholders to understand their expectation towards the project. Next, is Quick Design which is a phase where a simple design of the project is developed is a rough idea of the system, it gives all the stakeholders the help understand the system. Build Prototype is where a small working prototype will be build based on the quick design which can be interacted by the users too. Next up will be User Evaluation where the feedback is given by client after evaluation to improve the system. Next is Refining prototype where the developers will do changes to improve the system based on the feedback given by the client. User Evaluation and Refining Prototype will be done repeatedly until all stakeholders are satisfied with the end product. Finally, Implement and Maintain, When the final prototype is approved, fully developed and tested. Additionally, ongoing maintenance are done to maintain stability.

The advantages of prototyping model consist of its active user involvement which allows better expectations for the user and better work done by the developers and also enable developers to identify and refine what modules and functions are confusing. Because user can interact with an actual working model, they can get a clearer understanding of the system during development. Early bug detection in the prototype can help reduce the resources needed to complete the project. Other than that, the disadvantage is the over-reliance towards the prototype which can lead to insufficient requirements analysis because developer will focus more on improving the model rather that a fixed plan. If strategy is not properly done, effort done in building prototype will be a wate of resources and inefficiency.

2.3.5 Comparison

Table 2.3.5.1 Comparison Table for Development Methodologies

Methodology	Advantages	Disadvantages
Agile Development	• Highly flexible in terms of requirement changes. • Continuous feedback from user improves	• Frequent user involvement is required which may be hard for to schedule.

	the end product's quality	• The process can be too complex for project that is big scale • Changes can lead to scope creep leading to increased budget and timeline
Rapid Application Development (RAD)	• Fast development for quick system development • High user involvement to align with user needs	• The process can be too complex for project that is big scale • Successful of the project will depend on the skill of the developers in a fast-paced environment.
Waterfall Model	• Clear structure and defined phases. • Project managers have good control over project's resources allocation. • Predictable costs and timeline, ideally for project with strict timeline and budget.	• Lacks flexibility as major changes are needed for little rework • Not suitable for changing and evolving requirements. • No early user feedback which leads to issue found late
Prototyping Model	• Active user involvement leads to better user expectations.	• Over-relying on prototypes will lead to insufficient requirement analysis.

	• Users are to interact with a working prototype giving a clearer picture. • Early bug detection by user and developer can reduce overall resources needed.	• If poorly planned, resources used in prototype will be wasted.

All the methodology mentioned above have their strength and weakness therefore making them suitable for different types of projects. Agile development is highly flexible which makes it ideal for project that has evolving and changing requirements as it allows continuous changes. However, it requires frequent user involvement and can be complex for big scale project. Additionally, Rapid Application Development are suitable for small to medium sized project that requires fast delivery with active user participation offering flexibility. But it requires highly skilled developers to make it feasible and it is not ideal for large project. Other than that, the Waterfall model is best methodology for large project as it is structured well defined which will ensure predictability in terms of cost and timeline. However, it lacks flexibility as small requirements change will require a major change. Finally, the Prototyping Model is best for project with unclear requirements as it allow improvements based on user's feedback. On the contrary, over reliance towards prototypes can lead to inefficient planning and possible wasting resources.

2.4 Development Tools

2.4.1 Framework

2.4.1.1 ReactJS

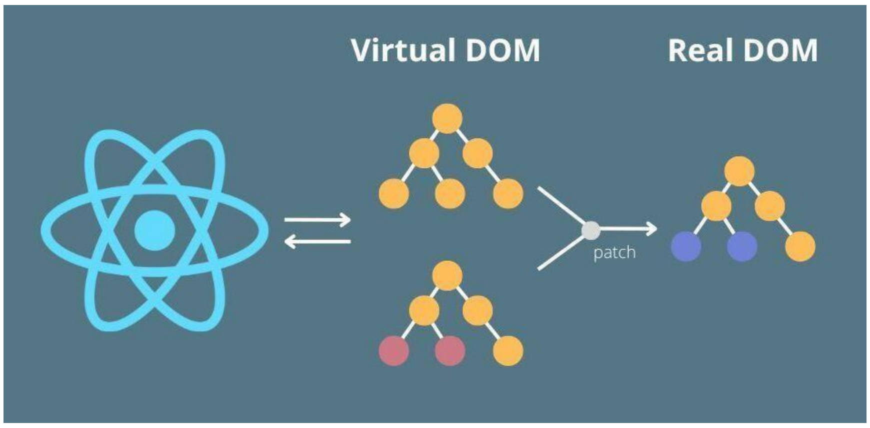


Figure 2.4.1.1.1 Figure of how ReactJS work

ReactJS is one of the popular JavaScript library that is used by developers to build user interfaces with high efficiency and is primarily used in web applications. Vara and Rakshitha (2024) stated in their research that many big companies are using it such as Instagram, Netflix, Facebook, and Airbnb. It is usually referred to as a component-based library. One of the key features of ReactJS is component-based architecture whereas the application are built with reusable component, these components are assembled to create complex interface while keeping the code organized and the same time. Other than that, React uses a Virtual Document Object Model (DOM) to improve the framework's performance. DOM only update the modified components instead of the whole webpage to reduce rendering time. Next is about its Cross-Platform Compatibility where it supports both IOS and android with only uising a single codebase, this can save cost and time in the development process.

The advantage of React includes increased performance because of Virtual DOM reducing re-renders. Other than that, Component-Based-Architecture promotes resuable AI component making the development more maintainable. The flexibility of React is also good as it works well with various other backend technology. However, it can be quite advanced for beginners as it requires understanding of multiple component, hooks and state management. Additionally, Frequent changes in React also require developers to be constantly up to date. Lastly, React only focuses on the UI later thus requiring developers to external libraries for additional functions.

2.4.1.2 AngularJS

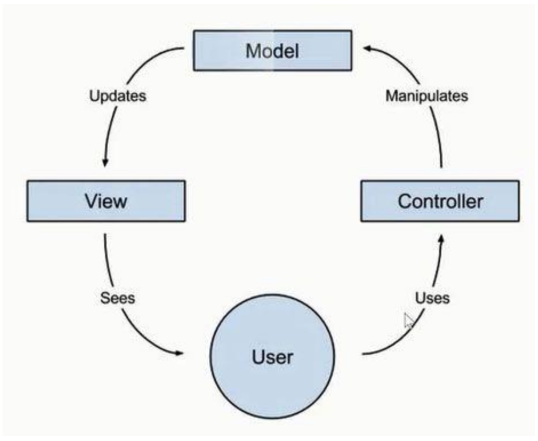


Figure 2.4.1.2.1 Figure of MVC

AngularJS is JavaScript-based front-end framework that is developed by a company named google, it enables HTMP with its additional functions, allowing developers to create interactive and data-driven web applications. AngularJS is mainly meant for dynamic, single paged applications where the content will update without the whole page reloading such as real-time applications, E-commerce sites, Content management system. Company such as Gmail, Weather.com, and Forbes are currently using this framework for increased efficiency. Vyas (2022) explains that AngularJS works by binding data between model and view for example, if a user types something in an input field, the model will update without additional JavaScript. It also have special attributes for HTML to make it easier for developers to develop the application. Angluar JS also has a MVC Arcitecture which concists of Model-View-Controller whereas the Model will represent the data, View representing the UI, and Controller which conntect the model and all the view by managing functions.

The advantages that comes with AngularJS is a full-fledged framework where it has built-in tools for necessary functions and routing making th developer's life easier. The two-way data binding also ensures that automatic synchronization is done between the model and view. AngularJS is especially well-suited for large applications which is ideal for organization that has a structured architecture. However, AngularJS has a steep learning curve as it has TypeScript and decorators which can be hard for beginner to learn. Performance bottleneck is also a risk for large application because of two-way data binding. Finally it is considered less flexible as it has a strict structure and less third-party integrations.

2.4.1.3 NativeScript

Figure 2.4.1.3.1 Logo of NativeScript

Sufyan Bin Uzayr (2022) explained in his book that NativeScript is a cross-platoform development framework that is used to build apps using JavaScript, TypeScipt, Angular or Vue.js. NativeScript allows a direct interaction between the native APIs without the need of a seperate bridge. NativeScript are suitable for Enterprise applications for their business and productivity apps that requires deep native access to work efficiently. But, it is not suitable for high graphic games because it runs with JavaScript virtual machines which will need a high-demand rendering for performance overhead. The key features for NativeScript is its direct native API access which does not require third-party bridges. It supports both Angular & Vue.js which allow flexibility for the developer to choose.

The advantages of Native Script is its excellent performance because of the usage of native UI component. Other than that, it works for both IOS and Android which will save time and cost. Native Script also supports multiple frameworks together such as JavaScript, TypeScript, Angular and Vue.js. However, Native Script has limited support from the community compared to other framework such as React Native and Flutter. Furthermore, it lacks nice UI components therefore requires extra effort to create custom UI elements and functions. There are some concerns about the performance compared to React Native and Flutter in some scenarios.

2.4.1.4 React Native

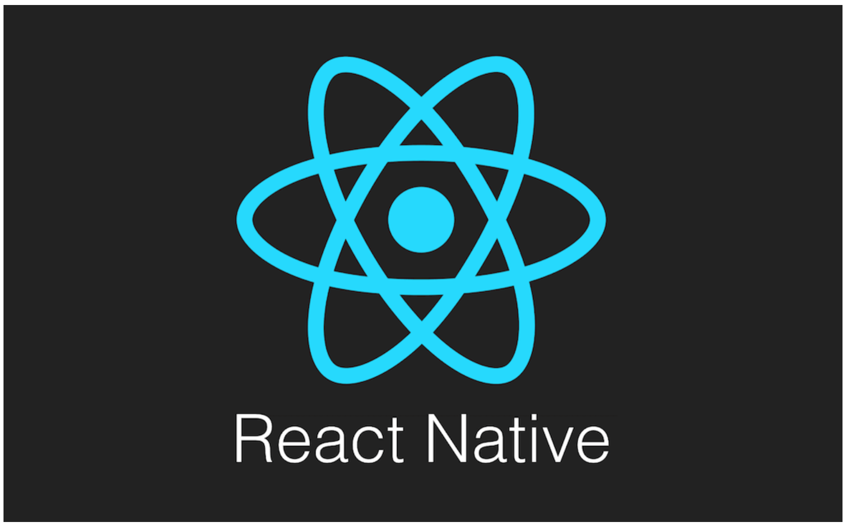


Figure 2.4.1.4.1 Logo of React Native

React Native is also a cross-platform development framework that allows developers to create app for both Android and IOS using JavaScript or TypeScript and it is developed by Meta also known as Facebook. Shevtsiv and Striuk (2021) stated that React Native ensures a better performance than traditional hybrid frameworks because of its utilization of native UI components rather than relying on WebViews. This framework is suitable for business and consumers applications for social media and e-commerce. It is also suitable for data-driven application that is focused on content display such as news feed and dashboard. One of the key features of React Native is its fast development because of its **Hot Reloading** which allow quick UI update without reloading. React Native also has a strong community and meta support which makes it having a wide and extensive third-party libraries.

What make React Native strong is its code reusability where around 90% of the code can be shared between IOS and android. The performance of the app will also be near-native because it used native components instead of

WebView. Strong community also help React Native to have well-supported libraries.Additionally, Hot Reloading help speeds up the application's development process. However, in complex scenarios React Native is not able to work as fast as fully native application. Next, there are many third-party modules which leads to many features that requires external libraries.

2.4.1.5 Flutter

Figure 2.4.1.5.1: Logo of Flutter

Windmill (2020) explained that Google created an open-source cross-platform development framework, Flutter that is used to build applications for multiple operating systems such as Android, iOS, web, and desktop by only using a single codebase. Many large companies such as Google Ads, BMW, and eBay have opted for Flutter because of its high performance and appealing interfaces

The advantages Flutter can bring to the table includes its high performance due to its own rendering engine named Skia which allows the framework to compile into native ARM code. Flutter also can support multiple devices and operating systems such as mobile, web, and desktop from one single codebase. Other than that, another important feature is its Hot Reload which can allow developers to instantly see the changes they made without the need to restart the application. Flutter also uses widget-based architecture making the UIs highly customizable for every device's file available. Another green flag is Flutter is backed by Google itself and have a strong community that ensure its growth and improvement. However, applications developed by Flutter normally have large size folder compared to other purely native apps. Additionally, because Flutter is still considered new compared to other frameworks such as React Native, some third-party package may not be as mature.

2.4.1.6 Comparison

Table 2.4.1.6.1 Comparison Table for Framework

Development Tools	Advantage	Disadvantage
ReactJS	• Virtual DOM increases efficiency by reducing re-renders. • Component-based architecture for easy reusable UI component. • Cross-platform compatibility	• Steep learning curve because of components, hooks, and state. • Developers needs to stay up to date with updates regularly. • Only focus on UI therefore requires additional library for full function.
AngularJS	• Full-stacked framework with tools built for routing and state management. • Two-way data binding ensuring automatic synchronization. • Suitable for large application that requires structured architecture.	• Steep learning curve because of TypeScript and advanced concept. • Less third-party integration makes it less flexible. • Possibility of bottleneck because of two-way data binding.

NativeScript	• Contain direct native API without the need of third-party bridge. • Support many other framework such as JavaScript, TypeScript,Angular, and Vue.js • Cross-platform compatibility	• Small community support compared with React Native • There is limited built-in component so will need extra work for customization. • Possible performance issue for certain situations.
ReactNative	• Large and strong community support • Hot Reloading to speed up development • Cross-platform compatibility	• Performance may differ for highly complex application • Rely heavily on third party modules for additional functionality. • Performance is not as fastr as fully native applications.
Flutter	• High performance due to compilation into native ARM code • Compatible with mobile, web and desktop from one codebase	• Larger app file size compared to other framework • Environment is newer than React Native making it fewer mature

	• Highly customizable UIs from rich widget library • Hot reload for rapid development • Supported by Google and strong community ensuring stability and growth	third-party package.

Each framework has its own strengths and weaknesses, making them suitable for different project eith different requirements. ReactJS is highly rusable because of its Virtual DOM and component-based architecture, allowing reusable UI components and easy integration with multiple backend framework and technology. However, its complexity and regular updates makes it challenging for beginners to learn, and it requires additional libraries for full functionality if to be applied. AngularJS, on the other hand, is a full-stacked framework with tools built-in to help developer and two-way data binding, thus suitable for large, structured applications. However, its steep learning curve and potential performance bottlenecks in large applications can be a disadvantage. NativeScript provides direct native API access and supports for multiple frameworks such as JavaScript, TypeScript, Angular, and Vue.js, making it flexible and efficient for big applications. However, its small community support, limited UI components, and potential performance concerns may have some drawback. Next, React Native has high near-native performance, and very strong community support especially from META, thus making it a popular choice for mobile app development. However, it relies on third-party libraries and perform fully as compared to native applications in complex scenarios. Finally, Flutter has high performance because it is compiled into the native ARM code. It is also compatible to many sort of device and has highly customizable UI from its rich widget library, Flutter also have hot reloading and it supported by Google which has a strong community behind its stability and growth. But, it has larger app fill size compared to other framework and third-party package are less mature because of newer environment .Choosing the right framework depends on the project's requirements, in terms of performance, flexibility, and scalability.

2.4.2 Database

2.4.2.1 SQLite

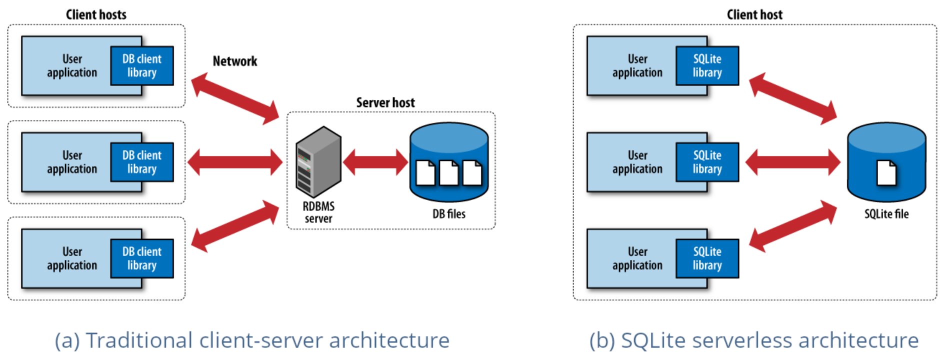


Figure 2.4.2.1.1 Figure of difference between traditional database and SQLite

SQLite is a flexible, serverless relational database management system that stores the entire database in a single file making it hassle free, excluding the need for a separate server process. According to Hummert and Pawlaszczyk (2022, pp.129–155) SQLite requires little setup unlike traditional databases such as PostgreSQL and MySQL making it ideal for small-scale and mobile application. SQLite supports the standards SQL features, which includes transactions, triggers, and views, while maintaining its high reliability and performance for operations that are read-heavy. Its simplicity and flexibility allow smooth integration across different operating system such as Windows, macOS, and Linux.

SQLite advantages includes its zero-setup design while being self-cointained which reduce the complexity of the database and administrative side. It can also be easily shared and migrate between system easilty as it s database are stored as a single file. Other than that, SQLite is open source and free to use making it cost-effective for developers. However, SQLite has some disadvantages such lack of built-in user management thus less secure for environment that has multi-user. Additionally it will lock the database during operations which may cause performance bottleneck in the application. Finally, it is not suitable for large-scale application as SQLite has limited scalability.

2.4.2.2 MySQL

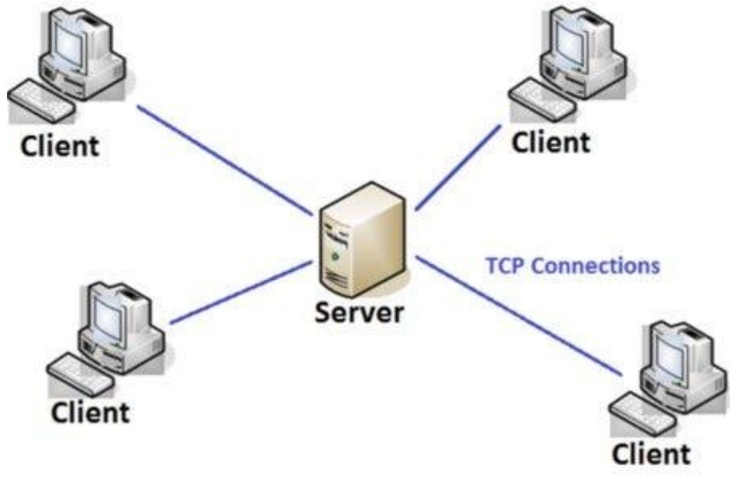


Figure 2.4.2.2.1 Figure of how MySQL works

MySQL is a efficient, open-source relational database management system that uses a client-server model to handle large-scale data storage and retrieval. Nixon (2021) stated in her book that MySQL runs and has a dedicated database server which allows multiple users and applications to interact with the database at the same time in a network unlike SQLite, which operates in a serverless, file-based database. It is used widely in various application such as web applications, and online services due to its high scalability, and performance. MySQL supports highly complex queries, stored procedures, triggers, and replication, making it compatible for website with high traffic and data-intensive applications.

The advantage of MySQL includes its ability to handle large amount of data while at the same time being able to maintain its high performance. Also, MySQL has strong security feature such as user access control and encryption to ensure data integrity and protection for the database. MySQL works well with various programming language which makes it an ideal choice for developers. However, MySQL have some drawbacks such as bring inconvenience for lightweight applications because it will require more setup. Performance can also be low if under extremely high transaction if done without proper optimization even if it supports high-write concurrency. Also, because MySQL strict adherence towards SQL standards causes it to be inconsistent sometimes and certain advanced feature are limited.

2.4.2.3 MongoDB

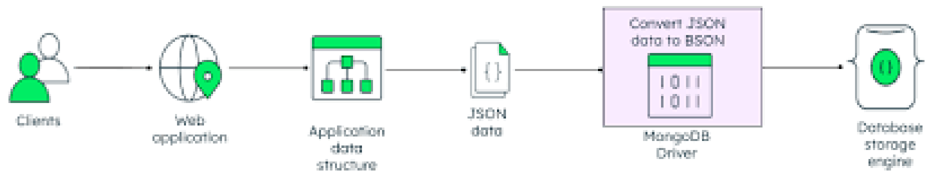


Figure 2.4.2.3.1 Figure of how MongoDB works

Makris et al. (2020) shared that MongoDB is a leading NoSQL database designed for flexibility, scalability, and high-performance data storage. MongoDB stores data in JSON-like documents (BSON format) rather than tables such as traditional relational databases like MySQL, thus suitable for unstructured or semi-structured data. MongoDB supports dynamic schemas, which allows developers to modify data structures without downtime, this function is particularly useful in agile development environments.

One of MongoDB's advantages is its horizontal scalability through sharding, which ables data to distribute across multiple servers to handle large-scale applications efficiently. Other than that, It also offers high availability with replica sets, to ensure automatic failover and data redundancy. MongoDB's query language is powerful, having to support indexing, aggregation pipelines, and also real-time analytics. Additionally, its native drivers for various programming languages such as Python, JavaScript, and Java simplify integration into applications in the modern world. MongoDB is widely used in many industry including big data, and Internet of Things due to its ability to handle different type of data types and high-velocity workloads. However, MongoDB also has some disadvantages. While it provides flexibility for developers. Its performance can degrade when handling complex queries involving multiple joins because it lacks traditional relational constraints.

Moreover, storage efficiency is another concern, this is because BSON format often results in larger data sizes compared to traditional relational databases. Additionally, although horizontal scaling is a strong feature, properly configuring and maintaining sharded clusters requires expertise and experience to set it up properly.

2.4.2.4 PostgreSQL

PostgreSQL

Figure 2.4.2.4.1 Figure of PostgreSQL logo

PostgreSQL which is also known as Postgres, is an open-source object-relational database management system known for its unique, scalable and standards compliance. Makris et al. (2020) also stated that unlike simpler databases like SQLite or traditional relational systems like MySQL, PostgreSQL supports queries that are more complex, custom data types if needed, and advanced SQL features like window functions, common table expressions, and full-text search. Furthermore, it comply closely to SQL standards at the same time giving extensions for other frame such as JSON and machine learning integrations.

A key advantage of PostgreSQL is its characteristic to scale and reliability in handling high-volume transaction and analytical workloads. It provides Atomicity, Consistency, Isolation, Durability (ACID) compliance and also multi-version concurrency control (MVCC) for improved query performance. Like MongoDB, PostgreSQL maintains strong data integrity with strict compliance while still able to support support semi-structured data through JSONB which is a binary JSON storage. However, PostgreSQL has some disadvatange. It is regularly considered to be more resource-intensive than MySQL or SQLite, requiring higher memory and CPU usage from the user, which can impact performance if used in low-powered systems. Additionally, its complexity makes it harder to set up and maintain, requiring more database administration experience to handle. Additionally, write-heavy workloads may not perform as well as other specialized NoSQL databases like MongoDB, while similar setup will be more challenging compared to MySQL. Finally, although it has strong community support, PostgreSQL's adoption in some application's environments is lower than MySQL, which means fewer third-party tools and hosting solutions can cater specifically to it.

2.4.2.5 Oracle

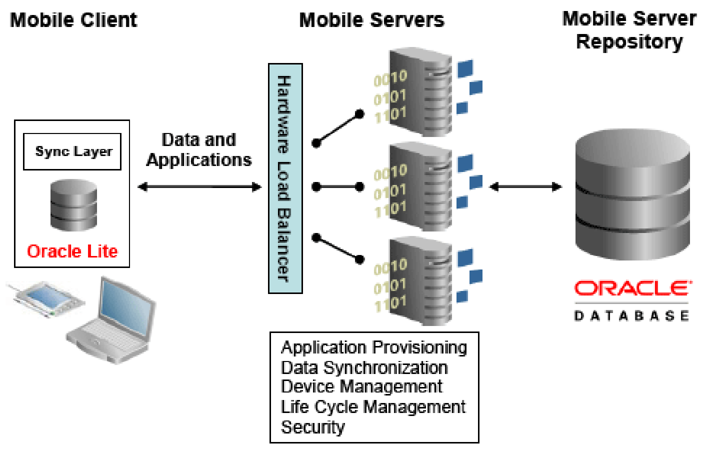


Figure 2.4.2.5.1 Client-server architecture by Oracle Database

Oracle Database works using a client-server architecture as shown in Figure 2.4.2.5.1, the Mobile Client will operate with Oracle Lite which is a lighweight version of Oracle Database, and a synchronization later to manage data exchange with the server. The clients will communicate with multiple mobile servers via a hardware olad balancer which will ensure efficient request distribution. In the Mobile Servers, it will handle critical task such as data synchronization, security and device management and finally store allt he data in a central Oracle Database repository which will serves as the main backend system for storing and managing enterprise data.

In the book written by Kuhn and Kyte (2022), the advantage that Oracle Database can bring is it enables mobile users to work offline with their own local data, which will be synchronized with the central database after connectivity is found. The use of load balancer and multiple server also supports system scalability. Oracle Database also make sure of strong data integrity and also enterprise-level security. However, Oracle Database requires higher setup complexity, especially because of multiple components and synchronization layers which will recquire more resources and expertise to manage the whole database.

2.4.2.6 Comparison

Table 2.4.2.6.1 Comparison Table for Databases

Database	Advantage	Disadvantage
SQLite	• Self contained and serverless thus easy to set up • High read-performance for small-scale application • Cross platform support • Open source free to use	• Lacks user management function and security features • Poor scalability for big-scale application • Locking database during write operation which can cause bottleneck • Not suitable for multi-user applications
MySQL	• Supports large number of dataset with complex queries. • Have high performance for both write and read operations. • Strong security features with user	• Requires more setup compared to SQLite • Obeying strict SQL compliance may lead to inconsistensies • Performance may be reduced during high transaction if not optimized properly.

	access control and encryption. • Easily scalable • Widely supported by many third-party tools.	
MongoDB	• Incredibly flexible schema design ideally for unstructured or semu-structured data. • Bale to scale horizontal for large-scale application. • High availability with replica sets and automatic failover. • Useful for real-time analytic because of powerful querying with indexing • Supports integration with multiple programming language	• Performance issue may occurs due to multiple joins • BSON format result in larger storage space • Requires developer with experience to confirgure and maintain the database. • Lacks built in user authentication and security features for multi-user application.
PostgreSQL	• ACID compliant with strong data integrity and reliable • Supporting complex queries with custom data types and full-text search	• Resource intensive than other SQL database needing to use more memory and CPU • More complex setup and maintainence

	• MVCC enhances the overall performance. • Easily scalable with replication and parallel processing. • Supports relational and semiu-structured data with JSONB	needing experienced administrator. • Write-heavy workload may not perform as well as NoSQL database.
Oracle Database	• Mobile users are able to work offline with local data • Supports system scalability • Strong data integrity and enterprise-level security	• Setup complexity • Requires significant resource and knowledge to manage

Each database system has their own unique strengths and weaknesses, making them suitable for different types of application. SQLite is a simple, serverless database that is easy to set up and great for small-scale applications but lacks scalability and built-in security features. MySQL is a widely used relational database because it is able to supports large datasets, high-performance transactions, and strong security compliance, making it a popular choice for web applications among developers, it only requires proper setup and optimization to make it ready to use. MongoDB, a NoSQL database, is designed and reccomended for its flexibility and scalability, making it ideal to handle unstructured and semi-structured data in big data and real-time analytics applications, but it struggles with complex relational queries and has higher storage because of BSON format. PostgreSQL is a highly reliable database with strong ACID compliance, making it ideal for big company applications that requires complex queries and high data integrity in their database. However, it is resource-intensive and requires experienced developer to handle the administration. Oracle Database enables user to work offline while also supports the application to scale with strong data integrity and security. But the setup is way more complex and the resources and knowledge needed to maintain the database is much more.

2.4.3 Artificial Intelligence

2.4.3.1 ChatGPT

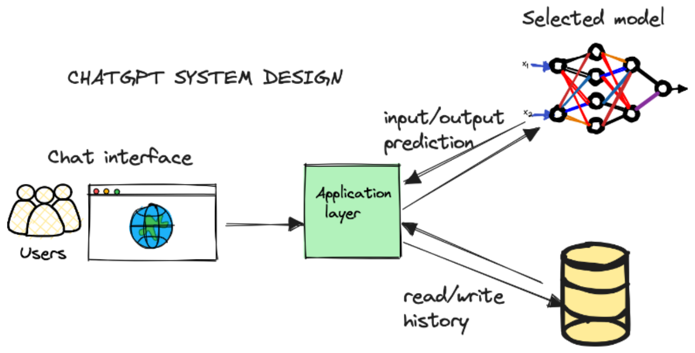


Figure 2.4.3.1.1 Figure of how ChatGPT works

ChatGPT is an advanced AI language model created by the company OpenAI, it is designed for natural language understanding and generation. Singh, Kumar and Pawan Singh Mehra (2023) explains that ChatGPT integrates deep learning techniques, specifically transformer-based neural networks, which results in human-like text responses. ChatGPT works by analyzing input prompts by the user, predicting the most relevant response based on its extensive training data, and adapting its output based on current user interaction. It is widely used in every industry for example customer support, content creation, coding assistance.

The advantage of ChatGPT includes high versatile AI model that can generate human-like text to interact with the user, making it useful for customer support, content creation, coding assistance, and general conversation. Additionally, its ability to quickly process and respond to user’s queries promotes productivity and assists users in problem-solving. However, ChatGPT has its drawback, as it has a chance of generating inaccurate or biased information. It also struggles with deeply technical queries or specialized content that requires precise factual accuracy and have a possibility of not always provide factually perfect answers.

2.4.3.2 DeepSeek

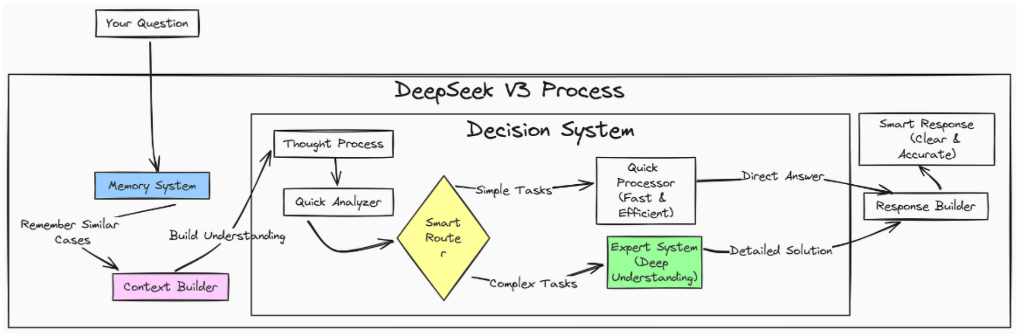


Figure 2.4.3.2.1 Figure of how DeepSeek works.

DeepSeek is an AI-powered language model designed for deep learning applications which specialize in data analysis, problem-solving, and knowledge research. According to Peng, Chen and Shih (2025) we can know that deepseek uses machine learning algorithms and natural language processing to extract information from large datasets, making it extremeful useful in research, finance, and decision-making systems. DeepSeek operates by analyzing structured or unstructured data, identifying patterns, and generating meaningful interpretations which is helpful for supporting business and academic applications.

DeepSeek excels in data analysis, pattern recognition, and decision-making, making it highly valuable for business intelligence, financial analysis, research applications and calculation. It effectively processes both structured and unstructured data to generate meaningful insights for the user. Its strength lies in handling large datasets and recognizing trends that support strategic planning for businesses. However, DeepSeek requires a high amount of data input for the best performance, and it lacks the to fluently conversate with the user and creativity that models like ChatGPT offer, making it less effective for interactive dialogue with user or content generation.

2.4.3.3 JasperAI

Figure 2.4.3.3.1 Logo of Jasper

JasperAI is an AI-driven content generation platform used exclusively for marketing, advertising, and creative writing. Like DeepSeek, it leverages natural language processing and machine learning to create high-quality text messages including blog posts, product descriptions, or even social media content, it works by understanding user-provided prompts query, generating contextually relevant content, and allowing adaptation customization to align with the user's needs.

JasperAI is a powerful AI tool for marketing and content creation, Businesses can efficiently generate high-quality text messages such as blogs, product descriptions, social media posts, and advertisements by using Jasper. It offers customizable writing styles to align with the user's brand messaging and helps improve overall content productivity. However, The downside of JasperAI is in general reasoning and conversational abilities, making it less useful for tasks beyond content marketing for example customer service and coding assitance. Additionally, it regularly requires human oversight to ensure originality, coherence, and adherence to the user's brand guidelines, this is because AI-generated content may sometimes need refinement from the user to make it meet their needs.

2.4.3.4 Copilot

Figure 2.4.3.4.1 Logo of Copilot

According to Porter and Zingaro (2024), GitHub Copilot is a AI-powered tool developed by the collaboration between Github and OpenAI. Copilot acts as a intelligent programmer partner for software developer which are able to offer real-time suggestions as developers writes the code. Copilot is trained with the public code from Github and other sources making it being able to predict and generate whole lines and blocks of code based on the developer's needs. Copilot aims to help developers to speed up coding tasks and explore unfamiliar libraries. Additionally, it is available as a tool at Visual Studio Code to assists begginers in their coding environment.

The main advantage of using Copilot is its ability to boost developers productivity by generating codes and offering creative solutions to their problems. Additionally, it can help developers to learn new libraries and also guide begginers in their coding journey. However, the disadvantages that comes within Copilot is it will sometimes produces code that is syntactically correct but is flawed logicall. On top of that, Copilot may suggest outdated coding style and method because it is trained on publicy available code. Furthermore, Copilot does not fully understand the context of the developer's project, thus developer wll need to review the codes given by copilot before applying it.

2.4.3.5 Gemini

Figure 2.4.3.5.1 Logo of Gemini AI

(Imran and Almusharraf, 2024) stated that Google Gemini is developed by Google DeepMind and is an advanced artificial intelligence system designed to integrate reasoning, multilingual and multimodal capabilities. As a successor to earlier large language models, Gemini is developed to customize enhanced understanding, problem-solving and educational support. In terms of education. Gemini can assist in both students and teachers by giving exaplnation, tutoring help and interactive learning experiences. Its ability to process text, image, and video allows it to adapt to multiple educational needs, thus improving the whole learning experience.

The advantages of Gemini is its deep analytical capabilities, ability to process with multiple types of media and its response time for complex educational queries. Additionally, it can support multiple instruction and are able to offer real-time feedback making the user experience more better. However, Gemini can sometimes produce generalized answers for difficult academic topics where expert knowledge is important. There is also a risk where students will be over-reliance on AI which will redice their independent problem solving skills. On top of that, issues like algorithm bias requires supervision to ensure that the information given is accurate.

2.4.3.6 Comparison

Table 2.4.3.6.1 Comparison Table for Artificial Intelligence

AI Model	Advantages	Disadvantanges

ChatGPT	• High capability for various application such as customer service and content creation. • Able to generate human-like text and adapt to user's needs • Quick response can enable user to enchance productivity • Highly efficient in problem solving	• Risk of generating inaccurate and biased information • Lack of highly technical or specialized content that requires high factual accuracy.
DeepSeek	• Great choice for data analysis, pattern recognition and knowledge research. • Able to extract information from large dataset for businesses. • Effectively processes the information extracted to meaningful information.	• Requires large and precise dataset to get optimal result. • Lacks human-like generation to conversate • Less effective for content generation and interactive dialogue.
JasperAI	• Great choice for marketing and content creation with rich-text information. • Able to customize writing style to align with the user's needs	• Limited general information and conversational abilities • Less effective for task above content making such as coding and customer service

	• Enhance productivity by generating high-qualkity marketing tools efficiently.	• Requires human supervision to make sure AI-generated for picture perfect generation.
Github Copilot	• Helps user to boost programming productivity • Helps user to learn new libraries and coding style quickly. • Suitable for generating code efficiently.	• May produce incorrect or outdated code. • Lacks deep understanding of project-specific context. • Requires supervision and testing for its outputs.
Google Gemini	• Highly adaptive educational support • Multimodal support such as text, image, video • Excel in multilanguage	• Context may be overly complex for user. • Potential biases when training data. • Will require supervision to ensure accuracy.

ChatGPT, DeepSeek, and JasperAI, Github Copilot, and Google Gemini each serve different purposes for the user based on their advantage and disadvantage. ChatGPT is a highly versatile AI model suitable for general conversations, customer support, coding assistance, and content creation thus making it a great choice for users who need an interactive AI with quick reply. However, it may have a possibility of generating inaccurate or biased responses and also struggle with highly deep technical topics. Other than that, DeepSeek excels in data analysis, pattern recognition, and business intelligence, thus suitable for research, financial analysis, and decision-making queries. But it depend on large datasets and lack of conversational ability with the user make it less effective for creative tasks. Next, JasperAI is designed specifically for marketing and content generation, providing high-quality text content such as blog posts, product descriptions, and social media content, making it perfect for businesses to focus on their branding and advertising. However, it lacks general information and requires human supervision for content perfection. Github Copilot aims to assits programmers by suggesting real-time code generation and solution making it perfect as a coding companion. But, it it will sometimes give the developers outdated, flawed or insecure code snippets to use thus critical supervision is required. Finally, Google Gemini is a powerful AI educational assistant which is capable of handling multimodal and multilingual but will still require human oversight to avoid potential biases and inaccuracies.

To summarize, ChatGPT is best for general AI assistance for problem solving, DeepSeek for analytical and research-based tasks, and JasperAI for rich content marketing and creative writing. Github Copilot to assist with coding and Google Gemini to be used as a tool for education.

2.4.4 Optical Character Recognition

2.4.4.1 Tesseract

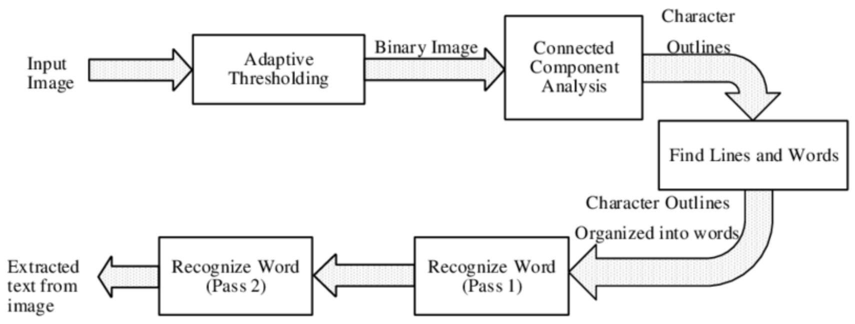


Figure 2.4.4.1.1 Figure showing how Tesseract works

Ogochukwu and Amaechi (2024) explains that Tesseract is an open-source, free-to-use OCR engine with strong language support, wide compatibility and flexibility. It’s highly customizable and efficient to use for printed text recognition, making it a solid choice for many OCR applications out in the market. Tesseract works by analyzing an image and identifying regions of interest such as text using image preprocessing techniques. After that, It cuts the image into individual characters and words and running recognition using deep learning-based classifiers. It can detect printed text in different type of fonts and languages, and it is highly customizable with its ability to train new languages, and text types. However, Tesseract struggles with certain condition such as handwritten text, poor image quality, or confusing layouts. It also faces challenges scanning with multi-oriented or curved text, requiring additional preprocessing or extra improvement steps.

2.4.4.2 Convolutional Recurrent Neural Network (CRNN)

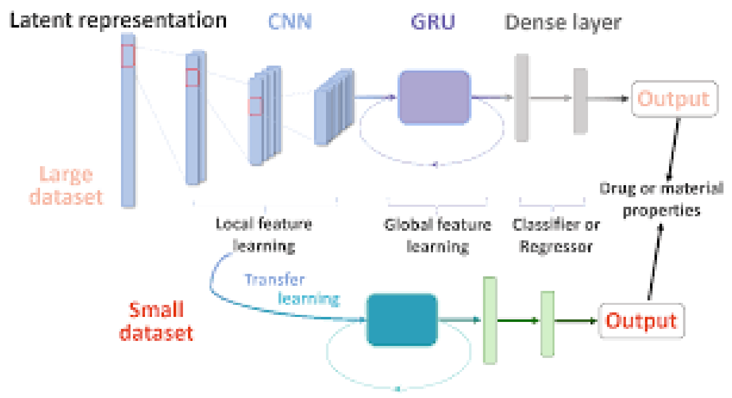


Figure 2.4.4.2.1 Figure of how CRNN works

CRNN is a deep learning-based OCR model that is a mixture of convolutional neural networks (CNN) for feature extraction and recurrent neural networks (RNN) for the sake of sequence modelling which makes it effective for recognizing text in natural scenes or in handwritten forms. Firstly, Ismail et al. (2024) explains that CRNN works by extracting information from the input image and then using the recurrent network to capture sequential dependencies in between each character. This allows it to recognize text in a continuously, making it suitable for tasks like handwriting or scene text. CRNN provides flexibility in text recognition, exclusively for continuous and sequential text, such as handwriting or curved text. It is exclusively effective for recognizing complex and diverse text styles. The model's ability to handle text in different orientations and formats makes it ideal for real-world applications. Although CRNN excels in flexibility, it is resource-intensive, requiring powerful enough hardware and software system to support it. Additionally, CRNN performance can be slower compared to traditional OCR methods, especially when processing simpler such as printed text with sparse or short segments, CRNN may have some problem to maintain its accuracy.

2.4.4.3 Efficient and Accurate Scene Text Detector (EAST)

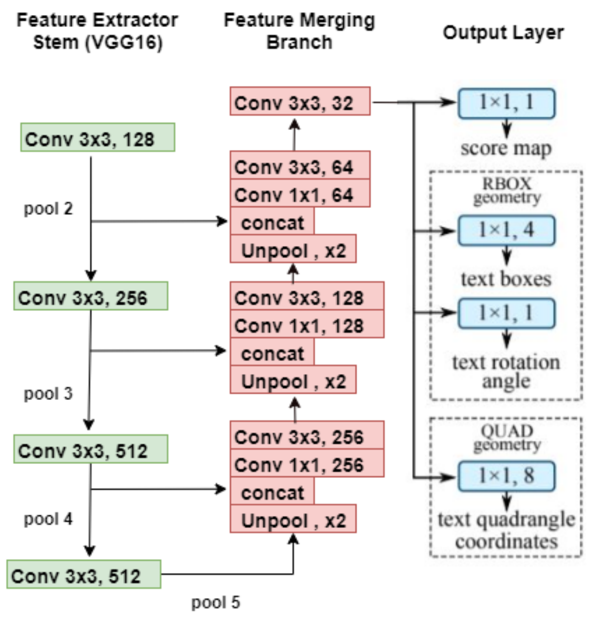


Figure 2.4.4.3.1 Figure of how EAST works

Mahajan and Rani (2024, pp.97–119) explains in their article that EAST is a text detection model based on a fully convolutional network that works to detect text regions in images fast and accurately, especially for scene text with different orientations and distortions. Unlike traditional OCR, EAST only focuses on text detection and does not include recognition. The model uses a pixel-level segmentation strategy to identify text areas thus making it capable to detect multi-oriented and curved text. EAST works well in real-time text detection due to its speed and efficiency. It works well in detecting text in multiplex backgrounds and different orientations of texts, which is incredibly useful for scene text recognition in applications like scanning a vehicle license plate or even street signs. It is highly efficient, giving a quick processing time to the system. EAST only performs text detection and does not provide text recognition, meaning it needs to be paired with another OCR engine to fully extract meaningful text to have optimal performance. Its performance may go down in cases where the text is too small, heavily distorted, or too noisy.

2.4.4.4 Comparison

Table 2.4.4.4.1 Comparison Table for Optical Character Recognition

OCR	Advantage	Disadvantage
Tesseract	• Free and open-source • Customizable and able to train new language and fonts • Super efficient for printed text recognition	• Not efficient against handwritten text, poor image quality and confusing layouts • Requires extra steps for multi-oriented and curved text
CRNN	• Efficient for recognizing handwritten and scene text. • Able to handle sequential and curved text well. • Works for different type of orientation and text formats.	• Very resource intensive which requires high CPU power. • Performance is slower than traditional OCR method especially for short messages.
EAST	• Fast and efficient in real-world scenarios. • Works well for multi-oreiented and disorted text. • Useful for application of text format such as street sign and license plate.	• Detect text only but does not recognize it • Needs to be paired with another OCR • Struggle with small, noisy, or distorted text

The comparison of Tesseract, CRNN, and EAST highlights their strengths and weaknesses in different OCR applications. Tesseract is a free and open-source OCR engine. It is also known for its strong language support and customization capabilities being able to customize and train new language, making it ideal for printed text recognition. However, it struggles with handwritten text, poor image quality, and complex layouts. CRNN excels in recognizing sequential and handwritten text because of its deep learning-based approach, making it suitable for many real-world applications. Despite this, it is resource-intensive, slower than traditional OCR for simple tasks, and may experience accuracy issues with sparse and simple text. EAST, on the other hand, is optimized for fast and efficient text detection, especially for multi-oriented and distorted text in real-world scenarios such as street signs or license plates. However, it only detects text and does not recognize it. Thus, EAST must be paired with another OCR engine for full functionality. Each model has its own strong point. Tesseract is best for printed text, CRNN for sequential and handwritten text, and EAST for fast and efficient real-time text detection.

2.5 Conclusion

Many aspects of the application for the development of the bill splitting applications are made to ensure that the most suitable tools are used to complete the project. Many existing systems are compared to understand their current strength and weaknesses with places to be improved. Additionally, methology are researched to ensure the project’s scope are able to be done in a mannered way. Tools such as framework, database, AI, and OCR framework are deeply compared to ensure the most suitable tools are used to develop the application with ease.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

The chapter will discuss the project planning and the software development methodology that will be used in the project. The content that will be included in is the Work Breakdown Structure (WBS), Project Plan, Gantt Chart, together with discussion throughout the development tools that will be used for developing the system.

3.2 Software Development Methodology

After research is done, Rapid Application Development (RAD) is chosen application development methodology. RAD is chosen as the Software Development Methodology because it is the most suitable for this project after in-depth research and comparison with other methodologies. This is because RAD is suitable for quick system development with its prototyping method. Additionally, the bill splitting application can be well developed with high user involvement during the development process and also does not require high scalability.

3.2.1 Rapid Application Development

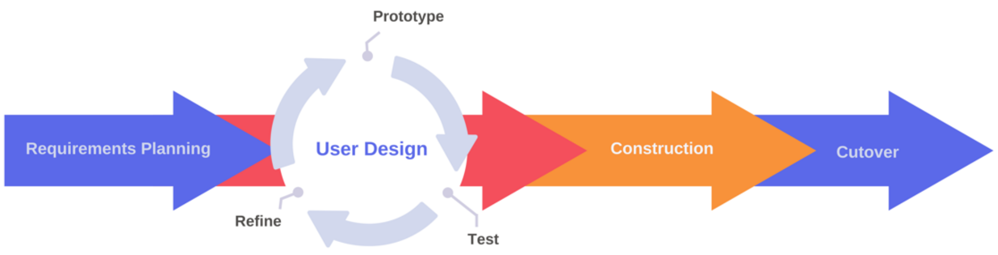


Figure 3.2.1.1 Process of RAD

Rapid Application Development (RAD) is a software development methodology that focuses on being able to deliver functional application quickly through prototyping and user involvement to improve the requirements of the application. According to Figure 3.2.1.1, RAD consists of six phases which is requirements planning, prototyping, test, refine, construction, and cutover. RAD's key advantages is having a fast development cycle with strong user involvement while being able to adapt to changes needed by the user thoughout the development process.

3.2.1.1 Requirements Planning

The first phase that will be done in RAD is requirements planning. The requirements are gathered through comprehensive research on multiple existing bill-splitting system. Their advantages and disadvantages are analyzed and the disadvantages found in the existing system are to be taken carefuly into consideration for enhancement for the new system. During this phase, the functional and non functional requirements are also documented which includes the key functionalties such as receipt scanning, participants and event management. User's expectations will be reviewed and include in the planning to ensure the system are able to solve the problems that occurring in the existing system.

3.2.1.2 Refine

After the initial requirements are gathered, the next phase is the refine phase. In this phase, all the collected functional and non-functional requirements are analyzed and carefully selected. This phase involves reviewing each requirements to ensure that it is clear, achievable, and align with the objective of the application. The requirements will be assessed based on the avaible resource to ensure that the application is feasible. Main objective of the refine phase is to ensure that no misunderstanding regarding the initial requirements arise and to make usre the system design is based on realistic and structured requirements.

3.2.1.3 Prototyping

After the requirements are carefully refined, it will enter the prototyping phase, Axure RP 9 will be used to design an interactive prototype for the user to interact with. The prototype will include all the pages and functionalities of the proposed system. Additionally, the prototpy will be based on the functional and non-functional requirements that are refined during the refine phase. Each function and page will be created to give a clear and interactive understanding on how the system will operate, allowing the user to have a realistic expectation of the final product.

3.2.1.4 Test

After the prototype is developed, the prototype will be tested through multiple testing including unit testing to verify that each component aligns with the end product expectation. Users will interact with the prototype as if it is an actual application, allowing them to understand the application actual flow and functionality. All the test result and feedback will be documented in details with areas and components that needs changes or improvements. The user feedback will then be a benchmark on how to further improve the system's usability. This then cycles back to the to refine phase until all stakeholders are satisfied with the prototype.

3.2.1.5 Construction

Once the prototype is approved by all stakeholders, construction phase will start, this phase explains the actual development of the application with the tools used to make it happen, the tools to be used has been researched properly to ensure the best tools with the best attributes are used to develop the system. The mobile application will use the Flutter framework with Visual Studio Code as the code editor. The reason why Flutter framework is used because after further research during the literature review, it offers cross-platform capability for both Android and iOS while also supporting hot reloading for rapid updates. Additionally, Flutter are able to support both Artificial Intelligence(AI) and Opctical Character Recognition(OCR) integration which are the requirements for the bill-splitting application. The database that will be implemented for the system is MySQL due to its strong performance and read/write operation discussed during the literature review. Tesseract OCR will also be used as the system's optical character recognition framwework as it's function is to accurately scan and extract text from printed receipt that will be given by the user which is one of the feature of the application. Tesseract OCR is selected because it is the most suitable OCR programmes to use after further comparison with other OCR programmes. Additionally, ChatGPT AI will be integrated within the system to help interpret the information scanned on the receipt using

OCR. ChatGPT is used because of its quick response and highly reliable. Testing will also be done at this phase such as system testing and unit testing to ensure that the end product works flawlessly and within the user's expectation.

3.2.1.6 Cutover

Finally is the cutover phase, the completed system will be deployed to a live environment for actual user to use, this phase includes migrating any necessary data fom the development and providing user guides to ensure seamless deployment. Issues and bugs that arise are also closely monitored to ensure the problems can be solved fast and efficiently.

3.3 Project Plan

3.3.1 Work Breakdown Structure (WBS)

1.0 Requirements Planning
- 1.1 Conduct Research
  - 1.1.1 Pleminary Planning
    - 1.1.1.1 Study existing apps
    - 1.1.1.2 Understand problem in existing system
    - 1.1.1.3 Define problem statements
    - 1.1.1.4 Define project objectives
    - 1.1.1.5 Define project scopes
  - 1.1.2 Literature Review
    - 1.1.2.1 Review software development methodology
    - 1.1.2.2 Review project tools
- 1.2 Methodology and Work Plan
  - 1.2.1 Identify suitable software development methodology
  - 1.2.2 Create Work Breakdown Structure
  - 1.2.3 Create Gantt Chart
  - 1.2.4 Identify suitable software development tools
- 1.3 Implementary and Deliverables
  - 1.3.1 Identify functional requirements
  - 1.3.2 Identify non functional requirements

- 1.3.3 Develop use case diagram
- 1.3.4 Develop use case description
- 1.3.5 Develop ERD diagram
- 1.3.6 Develop data dictionary

2.0 Refine
- 2.1 Review functional requirements list
  - 2.1.1 Filter duplicates
  - 2.1.2 Confirm with stakeholders
- 2.2 Evaluate feasibility
  - 2.2.1 Check tool compatibility
  - 2.2.2 Estimate completion time
- 2.3 Update requirements document

3.0 Prototyping
- 3.1 Build Interface
  - 3.1.1 Design prototype interface page
  - 3.1.2 Implement prototype interaction component
- 3.2 Integrate design with interaction

4.0 Test
- 4.1 Unit testing
- 4.2 System usability testing
- 4.3 Collect feedback from user

5.0 2nd Prototype Iteration

6.0 3rd Prototype Iteration

7.0 Contruction
- 7.1 Frontend Development
  - 7.1.1 Setup React Native
  - 7.1.2 Design interface page
  - 7.1.3 Add interaction component
  - 7.1.4 Integrate component and navigation
- 7.2 Backend and Database
  - 7.2.1 Create database tables
  - 7.2.2 Create and add dummy data to tables
  - 7.2.3 Integrate database to application
- 7.3 Additional Tools

7.3.1 API

7.3.1.1 Generate API token and key

7.3.1.2 Integrate API to application

7.3.2 OCR

7.3.2.1 Develop camera function for application

7.3.2.2 Integrate OCR into application

7.4 Testing

7.4.1 Unit testing

7.4.2 Integration Testing

7.4.3 Feature Testing

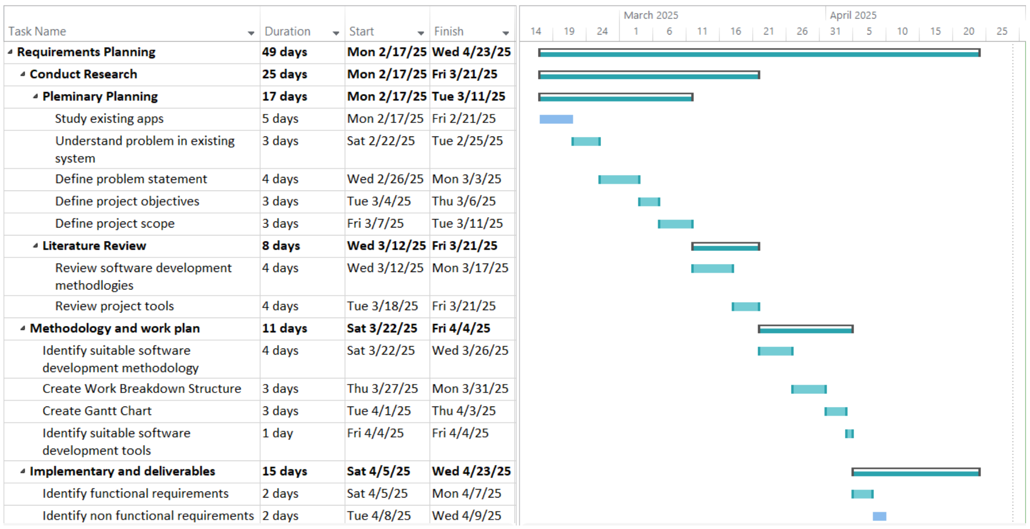
7.4.4 Black Box Testing

8.0 Cutover

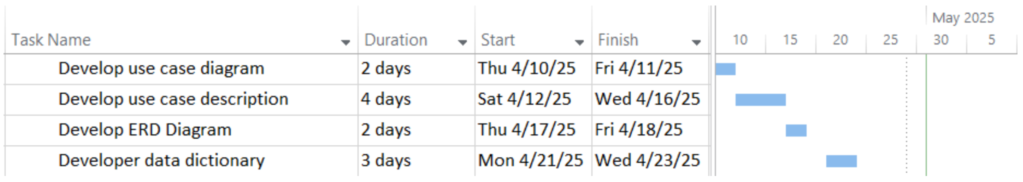
8.1 System development

8.2 Provide user training

3.3.2 Gantt Chart

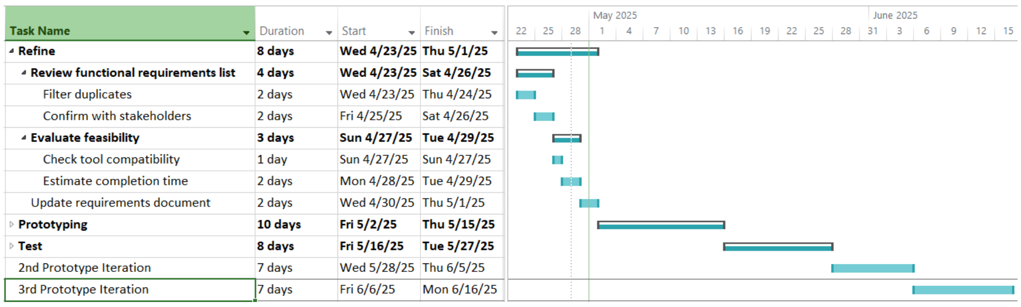


Task Name	Duration	Start	Finish
Requirements Planning	**49 days**	**Mon 2/17/25**	**Wed 4/23/25**
Conduct Research	**25 days**	**Mon 2/17/25**	**Fri 3/21/25**
Pleminary Planning	**17 days**	**Mon 2/17/25**	**Tue 3/11/25**
Study existing apps	5 days	Mon 2/17/25	Fri 2/21/25
Understand problem in existing system	3 days	Sat 2/22/25	Tue 2/25/25
Define problem statement	4 days	Wed 2/26/25	Mon 3/3/25
Define project objectives	3 days	Tue 3/4/25	Thu 3/6/25
Define project scope	3 days	Fri 3/7/25	Tue 3/11/25
Literature Review	**8 days**	**Wed 3/12/25**	**Fri 3/21/25**
Review software development methodlogies	4 days	Wed 3/12/25	Mon 3/17/25
Review project tools	4 days	Tue 3/18/25	Fri 3/21/25
Methodology and work plan	**11 days**	**Sat 3/22/25**	**Fri 4/4/25**
Identify suitable software development methodology	4 days	Sat 3/22/25	Wed 3/26/25
Create Work Breakdown Structure	3 days	Thu 3/27/25	Mon 3/31/25
Create Gantt Chart	3 days	Tue 4/1/25	Thu 4/3/25
Identify suitable software development tools	1 day	Fri 4/4/25	Fri 4/4/25
Implementary and deliverables	**15 days**	**Sat 4/5/25**	**Wed 4/23/25**
Identify functional requirements	2 days	Sat 4/5/25	Mon 4/7/25
Identify non functional requirements	2 days	Tue 4/8/25	Wed 4/9/25



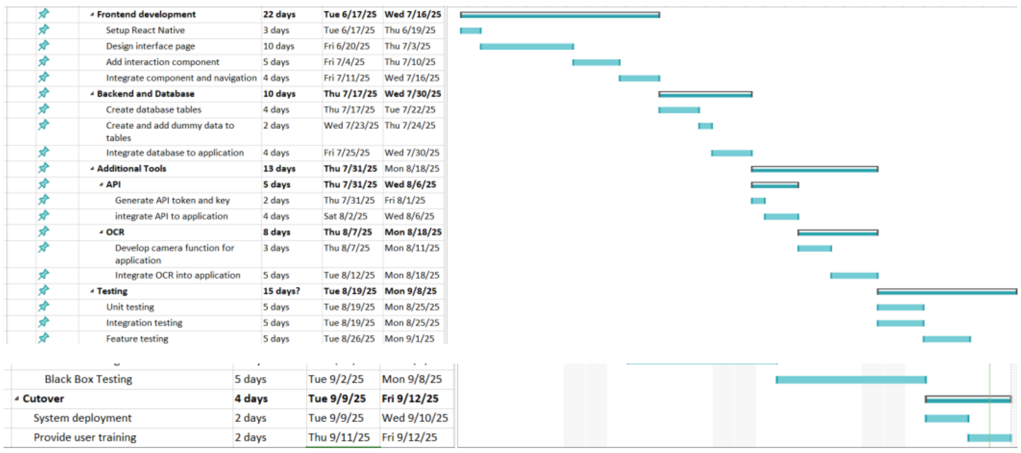
Task Name	Duration	Start	Finish
Develop use case diagram	2 days	Thu 4/10/25	Fri 4/11/25
Develop use case description	4 days	Sat 4/12/25	Wed 4/16/25
Develop ERD Diagram	2 days	Thu 4/17/25	Fri 4/18/25
Developer data dictionary	3 days	Mon 4/21/25	Wed 4/23/25

Figure 3.3.2.1: Gantt Chart for Requirements Planning phase



Task Name	Duration	Start	Finish
Refine	**8 days**	**Wed 4/23/25**	**Thu 5/1/25**
Review functional requirements list	**4 days**	**Wed 4/23/25**	**Sat 4/26/25**
Filter duplicates	2 days	Wed 4/23/25	Thu 4/24/25
Confirm with stakeholders	2 days	Fri 4/25/25	Sat 4/26/25
Evaluate feasibility	**3 days**	**Sun 4/27/25**	**Tue 4/29/25**
Check tool compatibility	1 day	Sun 4/27/25	Sun 4/27/25
Estimate completion time	2 days	Mon 4/28/25	Tue 4/29/25
Update requirements document	2 days	Wed 4/30/25	Thu 5/1/25
Prototyping	**10 days**	**Fri 5/2/25**	**Thu 5/15/25**
Test	**8 days**	**Fri 5/16/25**	**Tue 5/27/25**
2nd Prototype Iteration	7 days	Wed 5/28/25	Thu 6/5/25
3rd Prototype Iteration	7 days	Fri 6/6/25	Mon 6/16/25

Figure 3.3.2.2: Gantt Chart for Refine, Prototyping and Test cycle phase



Frontend development	**22 days**	**Tue 6/17/25**	**Wed 7/16/25**
Setup React Native	3 days	Tue 6/17/25	Thu 6/19/25
Design interface page	10 days	Fri 6/20/25	Thu 7/3/25
Add interaction component	5 days	Fri 7/4/25	Thu 7/10/25
Integrate component and navigation	4 days	Fri 7/11/25	Wed 7/16/25
Backend and Database	**10 days**	**Thu 7/17/25**	**Wed 7/30/25**
Create database tables	4 days	Thu 7/17/25	Tue 7/22/25
Create and add dummy data to tables	2 days	Wed 7/23/25	Thu 7/24/25
Integrate database to application	4 days	Fri 7/25/25	Wed 7/30/25
Additional Tools	**13 days**	**Thu 7/31/25**	Mon 8/18/25
API	**5 days**	**Thu 7/31/25**	**Wed 8/6/25**
Generate API token and key	2 days	Thu 7/31/25	Fri 8/1/25
integrate API to application	4 days	Sat 8/2/25	Wed 8/6/25
OCR	**8 days**	**Thu 8/7/25**	**Mon 8/18/25**
Develop camera function for application	3 days	Thu 8/7/25	Mon 8/11/25
Integrate OCR into application	5 days	Tue 8/12/25	Mon 8/18/25
Testing	**15 days?**	**Tue 8/19/25**	**Mon 9/8/25**
Unit testing	5 days	Tue 8/19/25	Mon 8/25/25
Integration testing	5 days	Tue 8/19/25	Mon 8/25/25
Feature testing	5 days	Tue 8/26/25	Mon 9/1/25
Black Box Testing	5 days	Tue 9/2/25	Mon 9/8/25
Cutover	**4 days**	**Tue 9/9/25**	**Fri 9/12/25**
System deployment	2 days	Tue 9/9/25	Wed 9/10/25
Provide user training	2 days	Thu 9/11/25	Fri 9/12/25

Figure 3.3.2.3: Gantt Chart for Construction and Cutover phase

CHAPTER 4
SYSTEM SPECIFICATION

4.1 Introduction

This chapter mainly focuses on the specification of the bill-splitting application which includes the functional and non functional requirements of the application. The requirements are properly coded to make it more structured and ensure standards are followed. The use case diagram and description are also provided to explain the functions of the application. Lastly, an ERD diagram and data dictionary are provided to provide better understanding of how the database will work around the application.

4.2 Requirements Specification

Here provides the two tables for explaining the functional and non functional requirements of the project are further requirements gathering.

4.2.1 Functional Requirements

Table 4.2.1.1 Functional Requirements for Bill Splitting Application

Code	Description
FR01	The mobile application shall allow the user to log in into the application with a username and password.
FR02	The mobile application shall allow the user to register a new account in the application with their unique email.
FR03	The mobile application shall allow the user to add a new event for bill splitting
FR04	The mobile application shall allow the user to edit previous event for changes and history purposes.
FR05	The mobile application shall allow the user to join other user's event as a participants to manage the event together.
FR06	The mobile application shall allow the user to manage the bills within an event for further calculation.
FR07	The mobile application shall allow user to take picture of the receipt when managing the bills to allow the use of optical character recognition.

FR08	The mobile application shall generate a new summary for the user regarding the event when requested from user

4.2.2 Non-Functional Requirements

Table 4.2.2.1 Non-Functional Requirements for Bill Splitting Application

Code	Description
NFR01	The mobile application response time should be within 1 second when the user interact with the system.
NFR02	The mobile application should validate user's input and prompting error message to prevent data error in the database.
NFR03	The mobile application should be simple and easy to navigate to allow user to easily understand.
NFR04	The mobile application should be able to be accessed by multiple user at the same time with the latest information.
NFR05	The mobile application should have access to the mobile's camera to use the optical character recognition function.

4.3 Use Cases

This part will include the use case diagram and the use case description for each use for further understanding on how the functions in the system will work.

4.3.1 Use Case Diagram

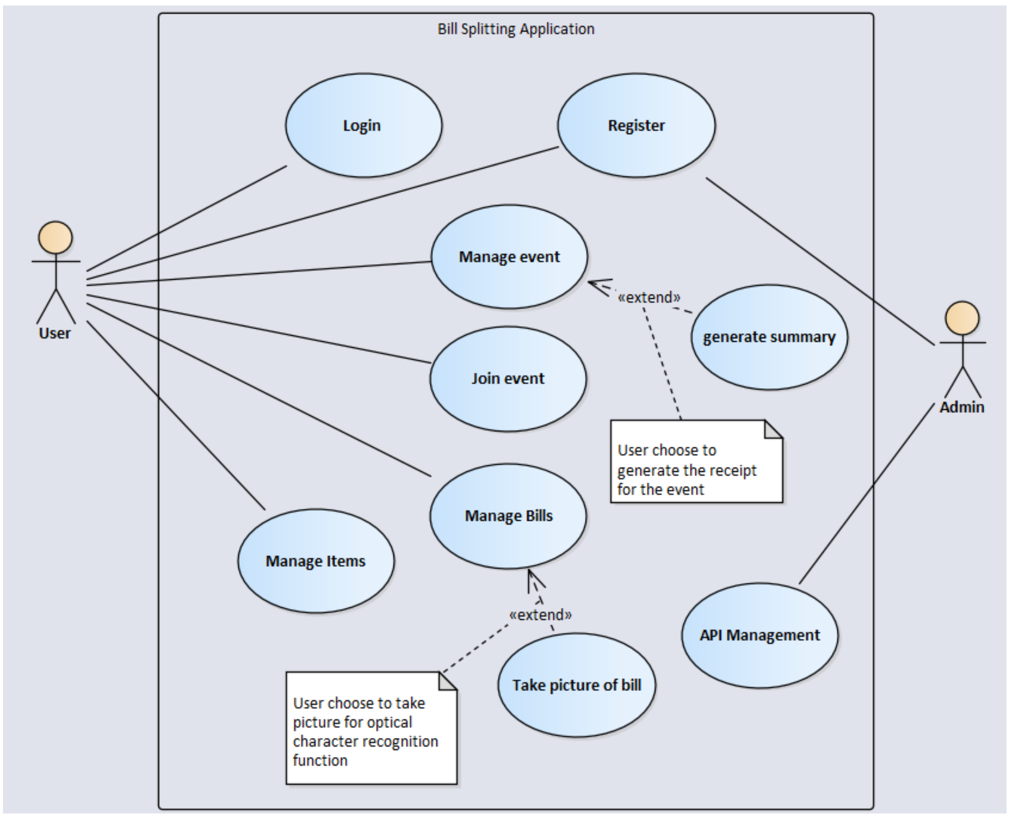

Bill Splitting Application
Login
Register
User
Manage event
«extend»
generate summary
Join event
Admin
User choose to generate the receipt for the event
Manage Bills
Manage Items
«extend»
API Management
Take picture of bill
User choose to take picture for optical character recognition function

Figure 4.3.1.1: Use Case Diagram of Bill Splitting Application

4.3.2 Use Case Description

Table 4.3.2.1 Use Case Description for Login Use Case

Use Case Name: Login	ID: UC1	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – want to login into the application		
Brief description: This use case describe how the user login into the application before using it		
Trigger: User opens the application		
Relationship: Association: User		

Include: N/A Extend: N/A Generalisation: N/A
Normal Flow of Events: 1. User opens up the application. 2. System will display login page with email and password input for user to enter. 3. User inputs its email and password to log in. 4. System checks the email and password in the database. Continue flow S-1. 5. User logs into the system successfully.
Sub-flows: S-1: Perform flow 4.1 or 4.2 4.1. If email and password match with database, continue flow no.5. 4.2. If email and password does not match with database, display error message and continue flow no.2.
Alternate/Exceptional Flows: N/A

Table 4.3.2.2 Use Case Description for Register Use Case

Use Case Name: Register	ID: UC2	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – want to register as a new user Admin – want to accept new user's application		
Brief description: This use case describe how the user register a new account into the system and admin who will need to accept the user's registration application.		
Trigger: User intends to register a new account at the log in page. Admin wants to accept user's registration application.		
Relationship: Association: User, Admin		

Include: N/A Extend: N/A Generalisation: N/A
Normal Flow of Events: User Flow: 1. User opens the application. 2. The system display the login page with a registration button. 3. User selects the option to register a new account 4. System displays the registration page with email, name, password and confirm password input for the user to enter. 5. User enters the details needed to register a new account. 6. System checks the email against the database. Continue S-1. 7. System displays registration success message. 8. User is led to the main page of the system. Admin Flow: 1. Admin heads to the application management page. 2. System will display all pending registration application. 3. Admin can select an application and either accept or reject. 4. The choice will be updated in the system.
Sub-flows: User Flow: S-1: Perform flow 6.1 or 6.2 6.1. If email exist in database OR password and confirm password does not match, display error message and continue flow no.4. 6.2. If email does not exist in database AND password and confirm password match, continue flow no.7.
Alternate/Exceptional Flows: N/A

Table 4.3.2.3 Use Case Description for Manage Event Use Case

Use Case Name: Manage Event	ID: UC3	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	

Stakeholders and Interests: User – user want to manage their events
Brief description: This use case describes how user can manage their events such as adding, editing and deleting the events.
Trigger: User wishes to manage their events
Relationship: Association: User Include: Generate Summary Extend: N/A Generalisation: N/A
Normal Flow of Events: 1. The system displays the user home page. 2. System will display current existing events' information. 3. User may select an existing event, press "Add Event" button, or press "Join Event" button.
Sub-flows: Add Event: 1. User presses the "Add Event Button" 2. System will display inputs for the title and the participants of the event. 3. User may inputs the title of the event and add in the participants of the event's name. 4. System will save the input into the database of the system. Edit Event: 1. User selects an existing event 2. User enters a page with the event's information 3. User may edit the participants and title of the event. 4. System will save the changed information to the database. Delete Event:

1. User selects an existing event 2. User presses delete event button 3. System will save the information to the database.
Alternate/Exceptional Flows: Alternate Flow 1: If user presses the “Join Event” Button, continue UC5. Alternate Flow 2: If user presses the “Generate Summary” button, continue UC4

Table 4.3.2.4 Use Case Description for Generate Summary Use Case

Use Case Name: Generate Summary	ID: UC4	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – user want to generate a summary of the event		
Brief description: This use case describe how the user can generate all the information for the an event.		
Trigger: User chooses “Previous Event” button in main page.		
Relationship: Association: User Include: Generate Summary Extend: N/A Generalisation: N/A		
Normal Flow of Events: 1. The system displays the user home page. 2. The user chooses option to manage previous event. 3. The system will search the database for event that is hosted by the user and event that is joined by the user. 4. The system will display all of the event searched. 5. The user selects one of the event.		

6. The system will retrieve the information of the event and display it. 7. The system will display three buttons, “Manage Bills”, “Delete Event” and “Summary”. Continue S-1
Sub-flows: S-1: Perform flow 7.1 or 7.2 or 7.3 7.1 If user chooses “Manage Bills” button, execute UC6. 7.2 If user chooses “Delete Event” button, a confirmation button will appear for the user, if user agrees, current event will be deleted from the database and and execute UC4. 7.3 If user chooses “Summary”, execute UC08.
Alternate/Exceptional Flows: N/A

Table 4.3.2.5 Use Case Description for Join Event Use Case

Use Case Name: Join event	ID: UC5	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – user want to join other’s user hosted event		
Brief description: This use case describe how a user wants to join other user’s event to participate in the bill splitting progress		
Trigger: User chooses the “Join Others Event” button in main page.		
Relationship: Association: User Include: N/A Extend: N/A Generalisation: N/A		
Normal Flow of Events: 1. The system displays the user home page. 2. User chooses option to join an existing event. 3. The system will display a event ID input for user to input. 4. The user will enter the event ID of the event that they wish to join. Continue S-1.		

5. System checks the event ID input against the database. Continue S-1. 6. System display success message, and update the database accordingly.
Sub-flows: S-1: Perform flow 5.1 or 5.2 5.1. If email exist in database OR password and confirm password does not match, display error message and continue flow no.3. 5.2. If email does not exist in database AND password and confirm password match, continue flow no.6.
Alternate/Exceptional Flows: N/A

Table 4.3.2.6 Use Case Description for Manage Bills Use Case

Use Case Name: Manage bills	ID: UC6	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – user want to view, add, update, delete bills in an event.		
Brief description: This use case describe when a user wants to view, add, update and delete bills in an event for calculation.		
Trigger: User wishes to manage their event's bills		
Relationship: Association: User Include: Take Picture of Bill Extend: N/A Generalisation: N/A		
Normal Flow of Events: 1. The system will display the bills within an event. 2. User are able to edit existing bill by selecting it, pressing the "Add Bill" button to add a new bill, or the hamburger icon for more functions.		
Sub-flows: Add Bill: 1. User presses the "Add Bill Button"		

2. Continue UC7. 3. System will go through optical character recognition process to retreieve items from the receipt into the system. 4. System will display title input for the use to enter. 5. User enters the title of the bill, if left empty, it will be "Untitled" by default. Edit Bill: 6. User selects an bill. 7. System will display all information about the existing event. 8. User may edit information such as discount, tax, and distribution method. 9. System will save the changed information to the database. Delete Bill: 10. User selects an existing event 11. System will display all information about the existing event. 12. User selects the "Delete Button" 13. System will save the information to the database.
Alternate/Exceptional Flows:

Table 4.3.2.7 Use Case Description for Take Picture of Bill Use Case

Use Case Name: Take picture of bills	ID: UC7	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – user want to take picture of the receipt		
Brief description: This use case describe how a user wants to take picture of the receipt to apply optical character recognition function.		
Trigger: User selects to "Add New Bill" button		
Relationship:		

Association: N/A Include: N/A Extend: N/A Generalisation: N/A
Normal Flow of Events: 1. The system will open up the device's camera 2. User will be able to take a picture of the receipt. 3. User can retake the picture if needed 4. The system will process the receipt for optical character recognition to retrieve information from the receipt. 5. System will append the information retrieved to the database.
Sub-flows: N/A
Alternate/Exceptional Flows: N/A

Table 4.3.2.8 Use Case Description for Manage Item Use Case

Use Case Name: Manage Item	ID: UC8	Importance Level: High
Primary Actor: User	Use Case Type: Detailed, Essential	
Stakeholders and Interests: User – user want to manage item information in a bill.		
Brief description: This use case describe when user wants to manage item in a bill		
Trigger: User chooses to add,edit or delete an item within a bill		
Relationship: Association: User Include: N/A Extend: N/A Generalisation: N/A		
Normal Flow of Events: 1. The system will display the items within an bill.		

2. User are able to edit existing item by selecting it, pressing the "Add Itm" button to add a new item, pressing the "Delete Item" button to delete an item.
Sub-flows: Add Item: 1. User presses the "Add Item Button" 2. System will display "Item Description", "Item Price", and "Particpants" inputs for the user to enter. 3. User enters the item description, price and participants involved. 4. System will save the information to the database. Edit Bill: 1. User selects an existing item 2. System will display the existing information regarding the bill. 3. User can change the item description, price and also append the participant involved. 4. System will save the information to the database. Delete Bill: 1. User selects an existing item and presses the "Delete Item" button. 2. System will delete the item from the database and save it.
Alternate/Exceptional Flows: N/A

Table 4.3.2.9 Use Case Description for API Management Use Case

Use Case Name: API Management	ID: UC9	Importance Level: High
Primary Actor: Admin	Use Case Type: Detailed, Essential	
Stakeholders and Interests: Admin – want to manage the API for aritificial intelligeince integration		
Brief description: This use case describe how a admin can manage the API for the AI in the system.		
Trigger:		

User wants to change the API of AI integration in the system
Relationship: Association: Admin Include: N/A Extend: N/A Generalisation: N/A
Normal Flow of Events: 1. Admin logs into the system. 2. System displays the current API information in the system. 3. Admin presses the “Edit” button in the page. 4. Admin edit the existing API or change to a new API. 5. System saves the API information in the database.
Sub-flows: N/A
Alternate/Exceptional Flows: N/A

4.4 Prototyping

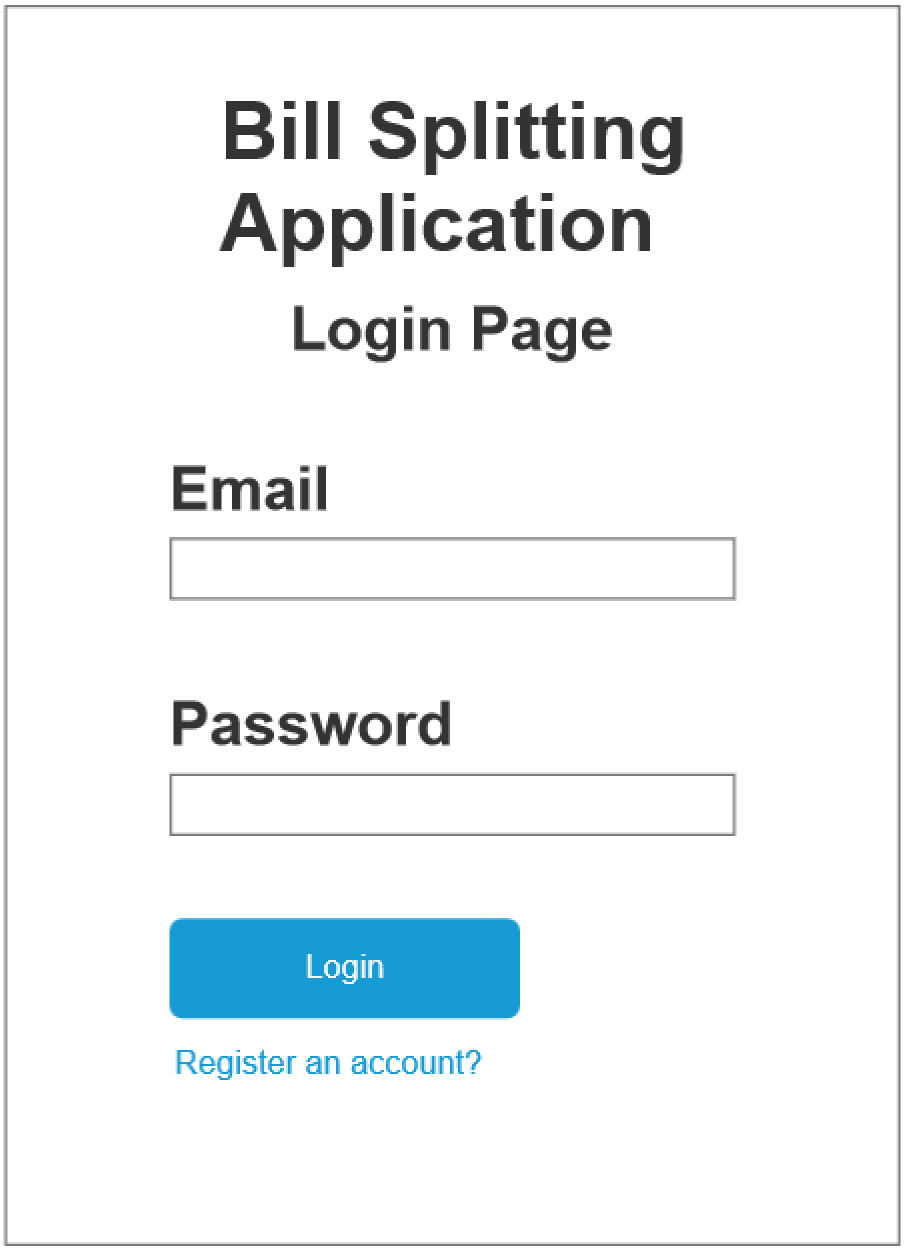


Figure 4.4.1 Login Page

The login page will have two inputs for the user to enter which is email and password. User will need to enter their registered email and password to enter the application. If user is not registered, there is a hyperlink to register an account.

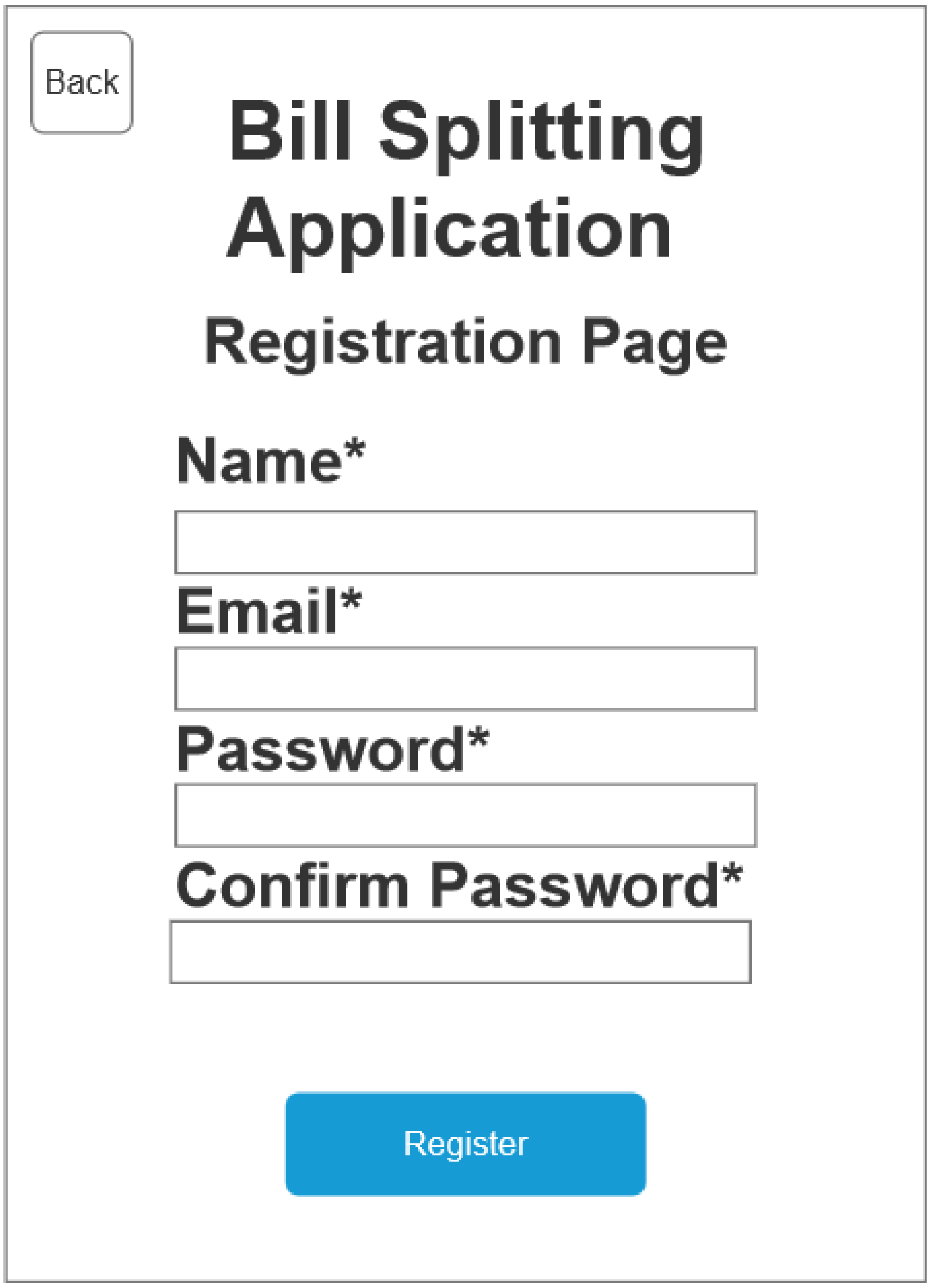


Figure 4.4.2 User Registration Page

The registration page will help the user to register an account in the system. User will need to enter their name, email, and password to register an account and will need to wait for admin to approve the information.

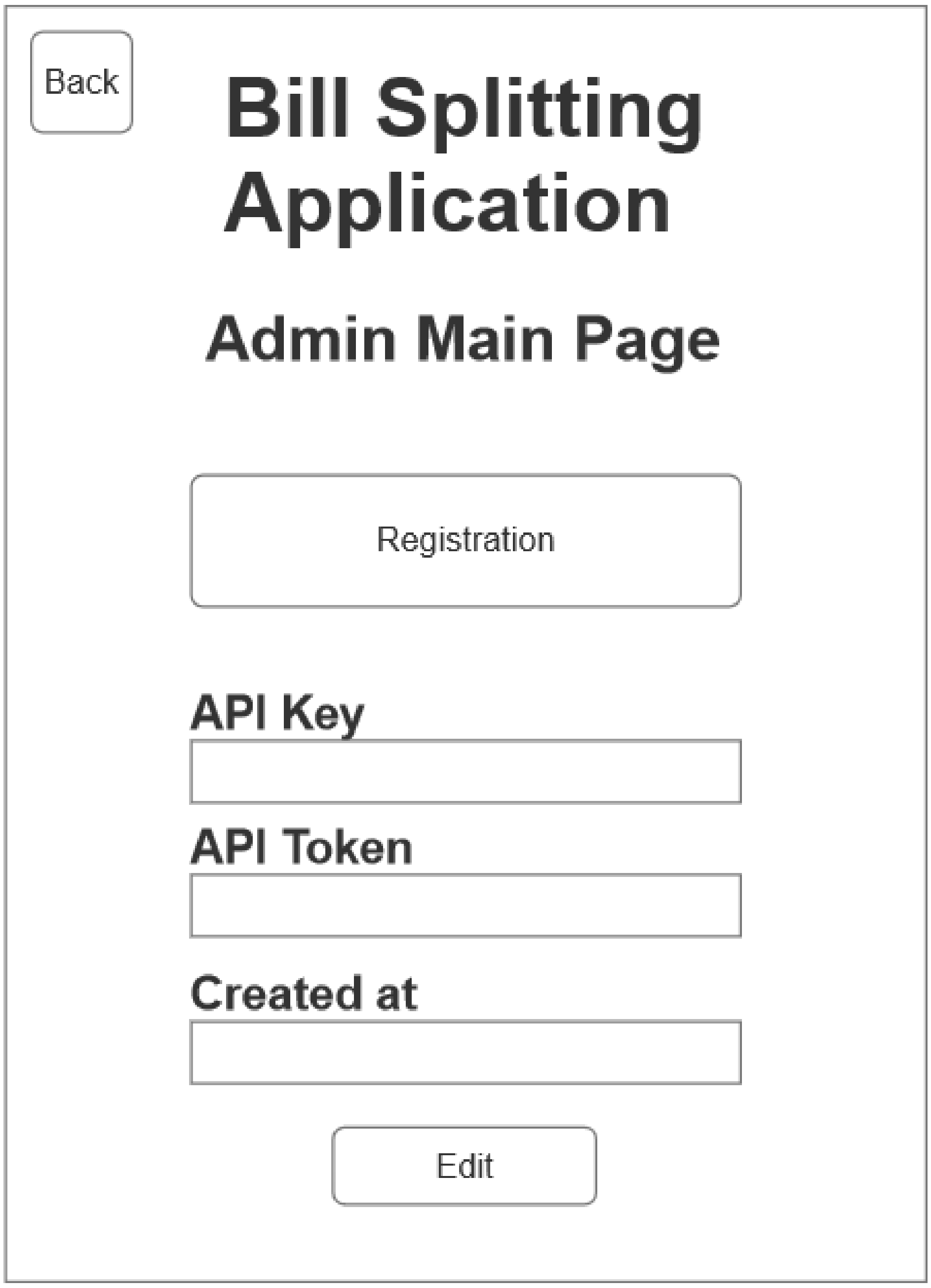


Figure 4.4.3 Admin Main Page

The admin main page will display the existing API information and two buttons for the admin to navigate to for managing registration and edit the existing API. Admin can edit the API by pressing the “Edit” button.

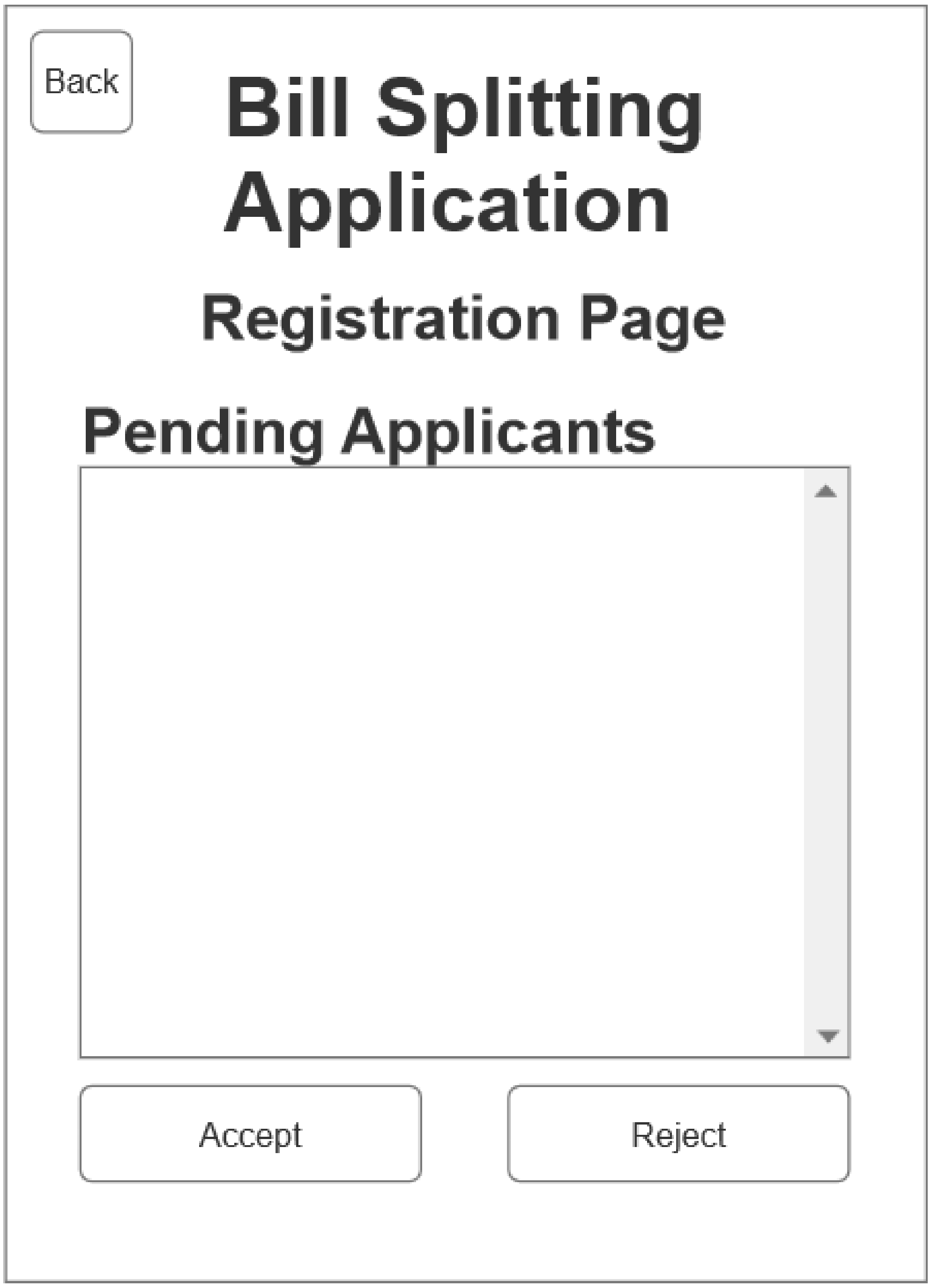


Figure 4.4.4 Registration Acceptance Page

In the registration acceptance page, a list of new applicant will appear for the admin to accept or reject. If accepted, the applicant's information will be saved in the system and they are able to access to the application.

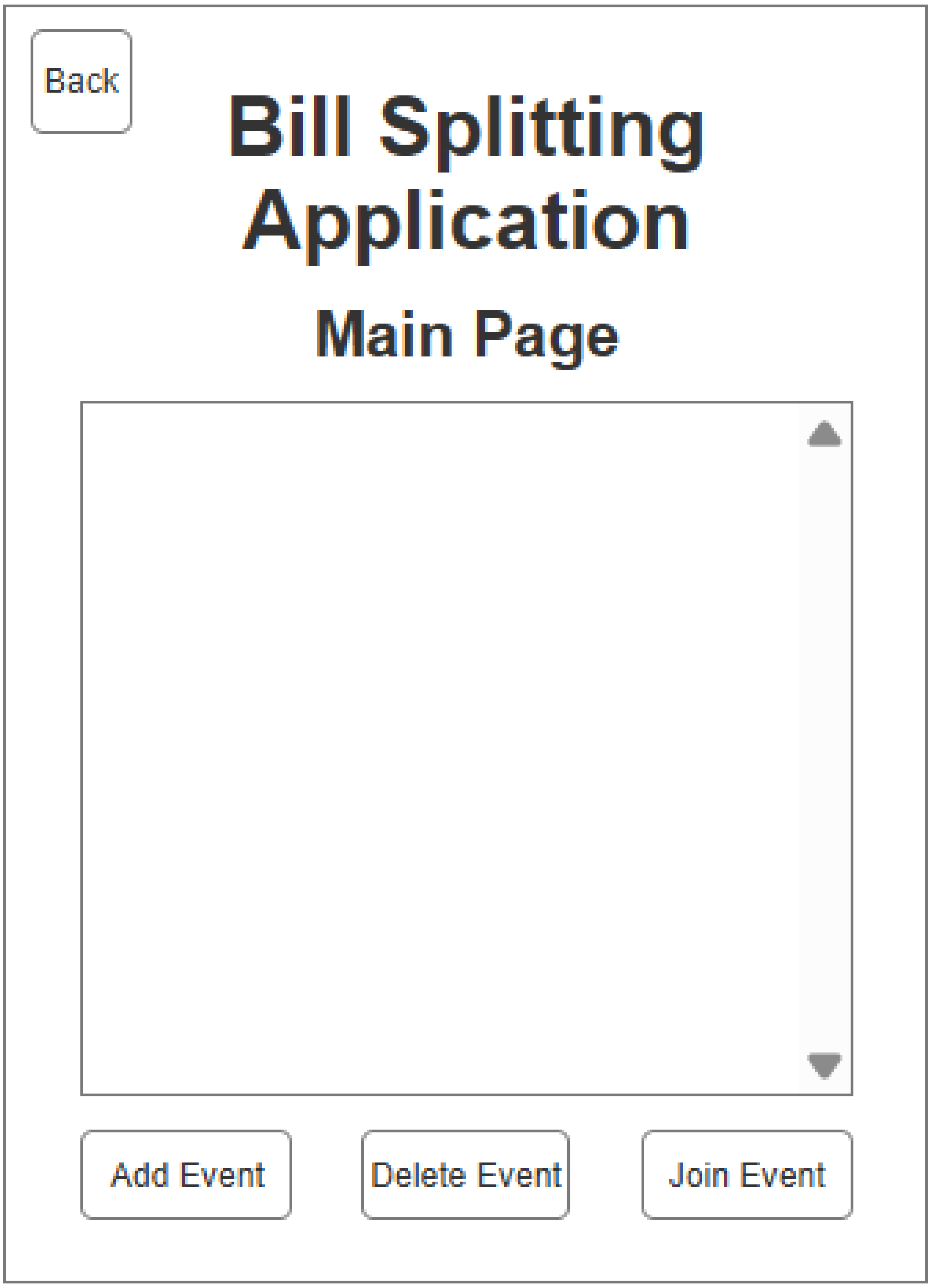


Figure 4.4.5 User Main Page

The user main page will display out all the event's title that is related to the user. The user will also have the option to add a new event, delete and event, and join other user's event. Additionally, the user will be led to manage event page when a event in the table is selected and clicked.

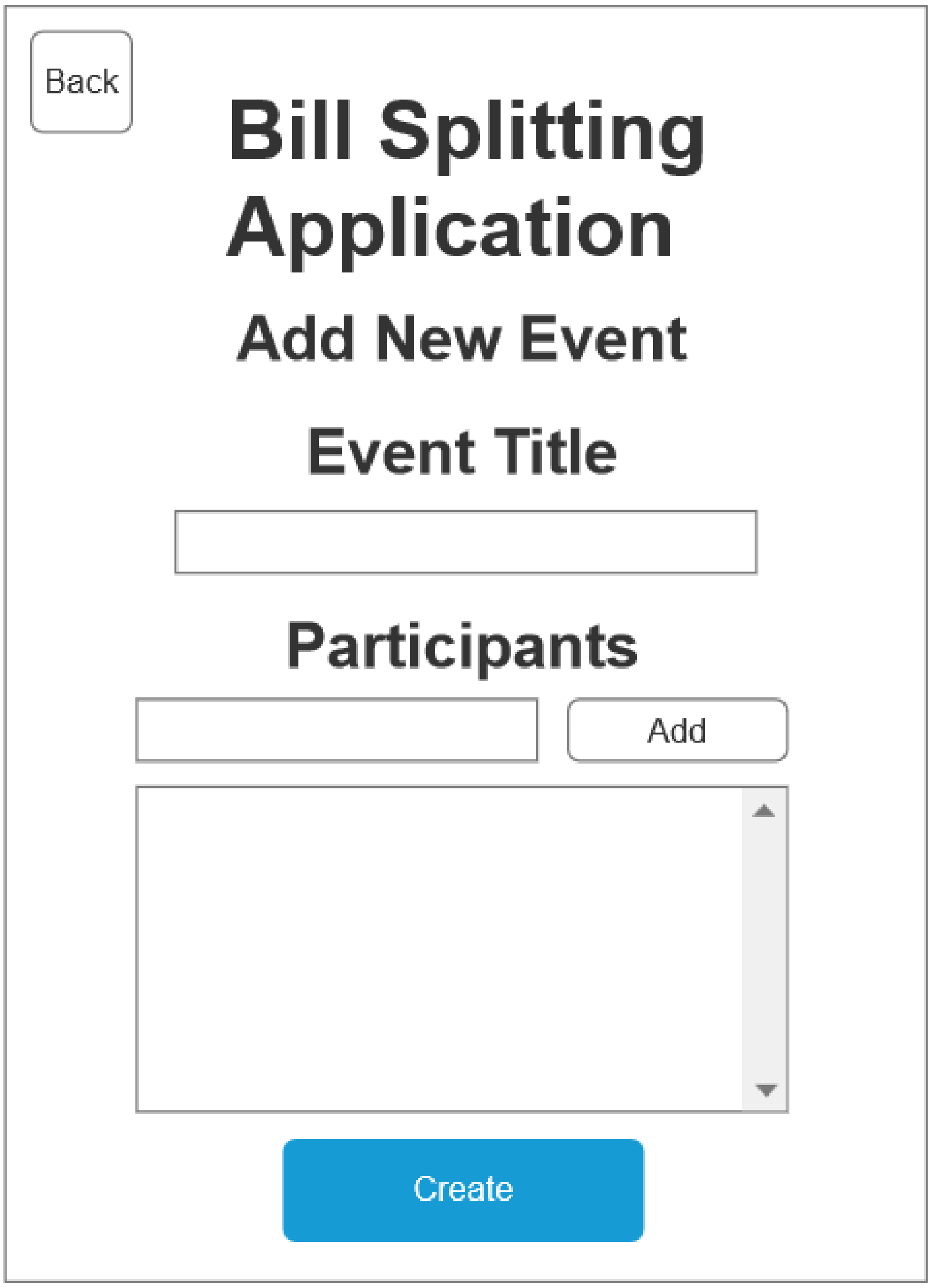


Figure 4.4.6 Add Event Page

The add event page allow the user to add a new event. User will have to enter the event’s title, it will be left with “Untitled” if left empty, participants of the event also can be added.

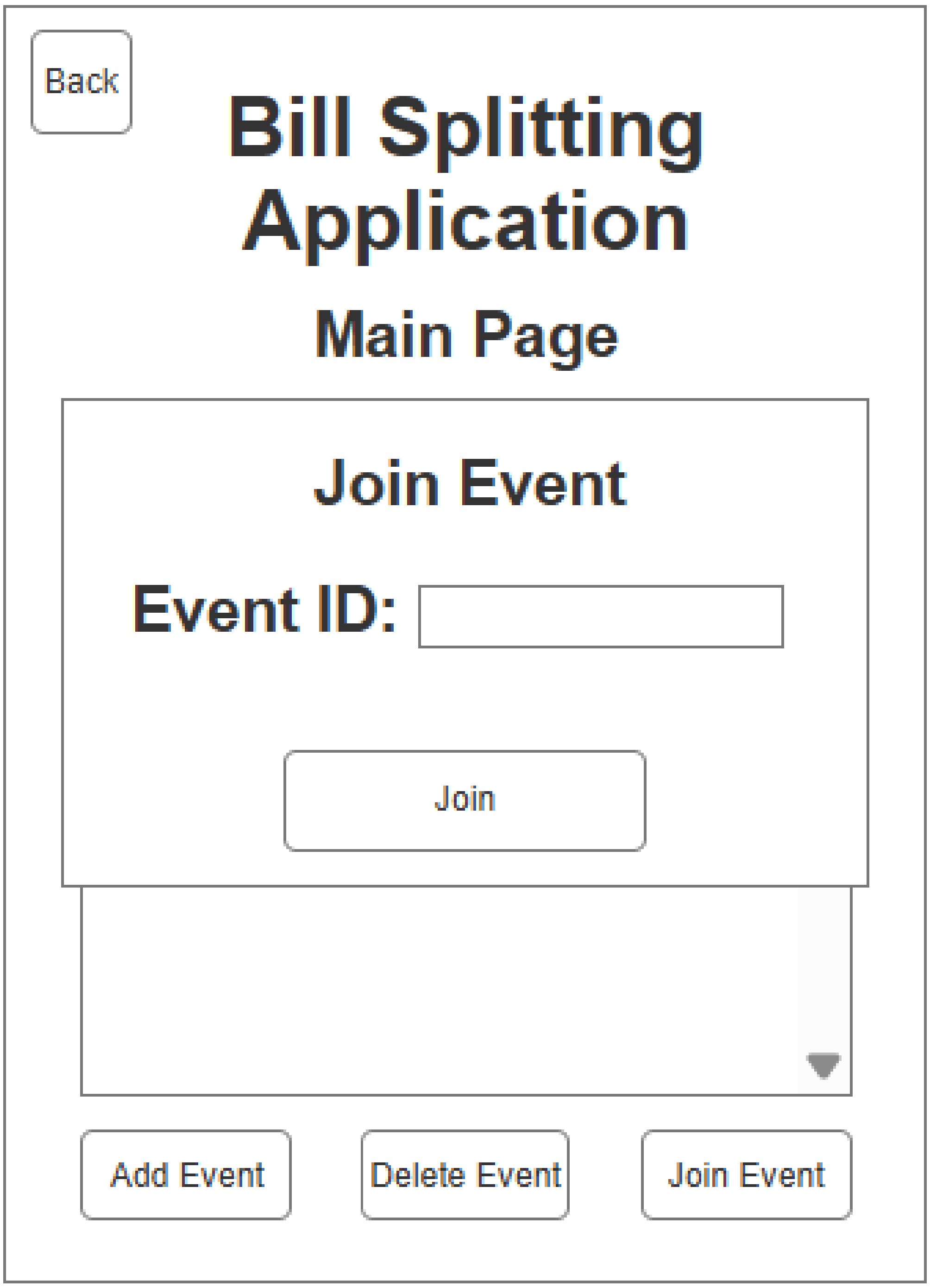


Figure 4.4.7 Join Event Interface

The join event container interface will show when the "Join Event" button is pressed. For the user to join other user's event, they will need to enter the host's event id to join. The event id can be retrieve from the host.

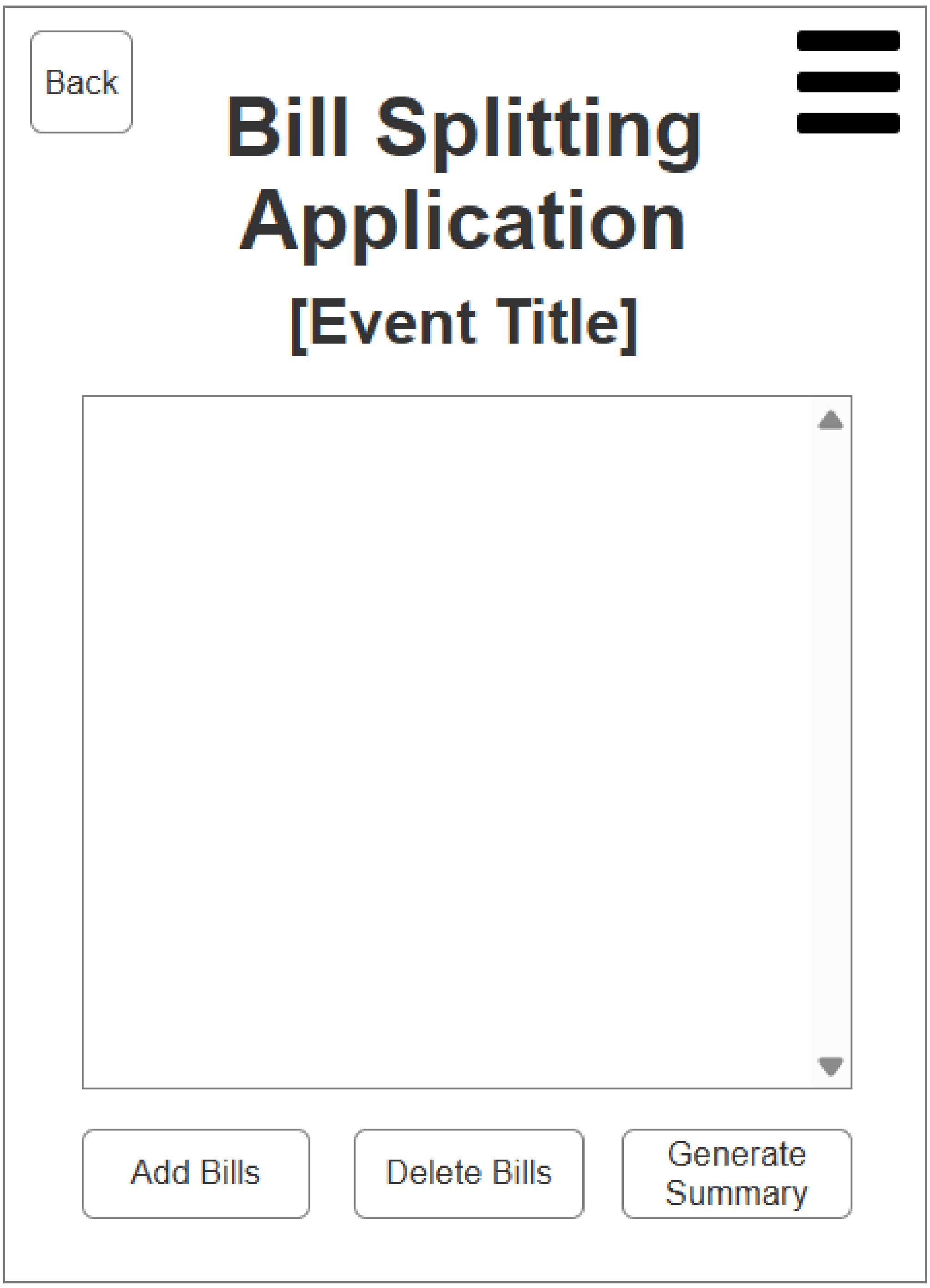


Figure 4.4.8 Manage Event Page

The manage event page allows the user to see the overall information of the event which includes the event's title, and the list of all the bills included in the events, on the top right corner of the interface, a "hamburger" icon that will allow other functions to happen.

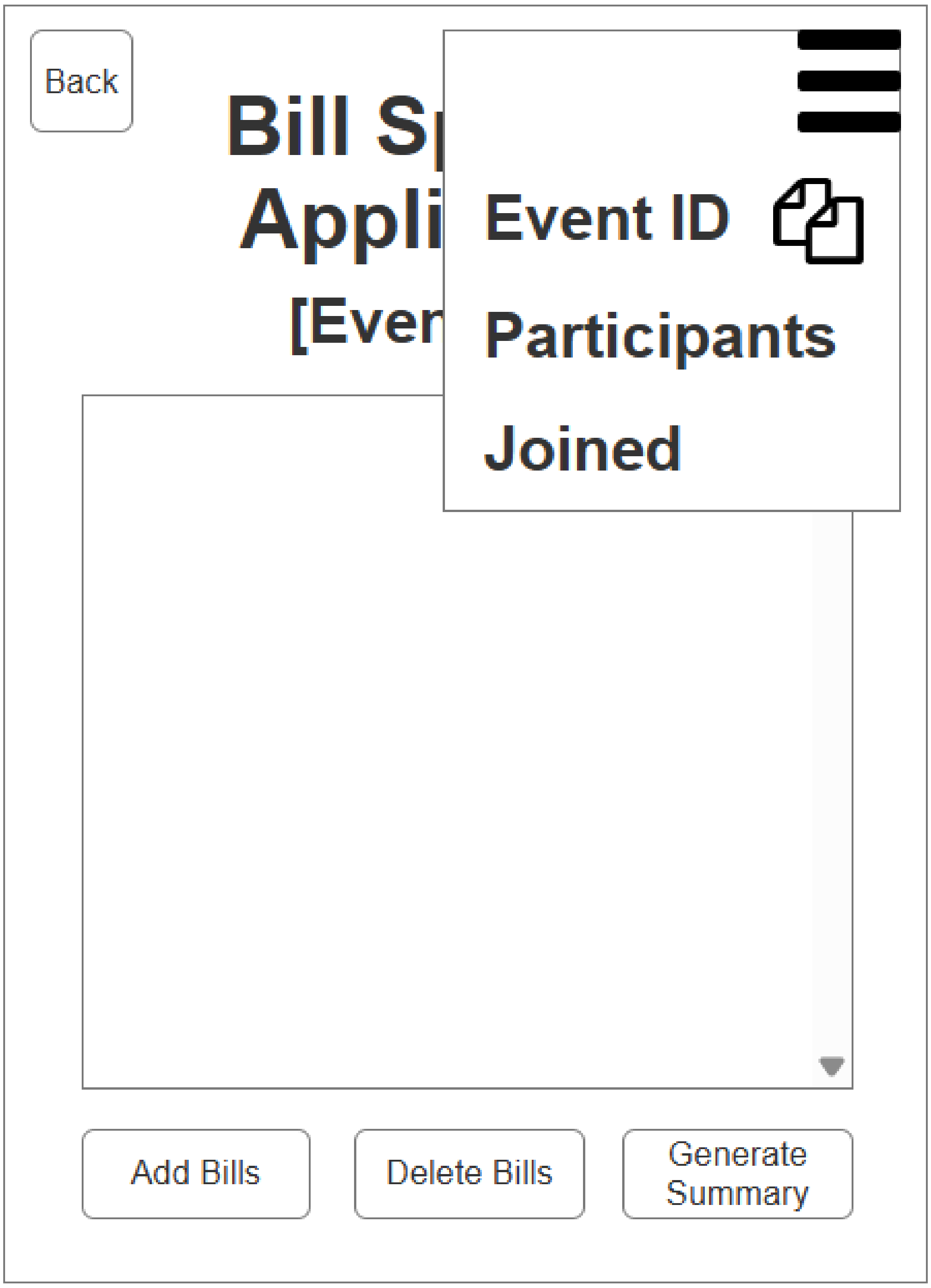


Figure 4.4.9 Hamburger Icon Functions for Manage Event Page

When the hamburger icon is pressed, a small container appears which allows the user to copy the event id, edit the participants, view who joined the event, and delete the event.

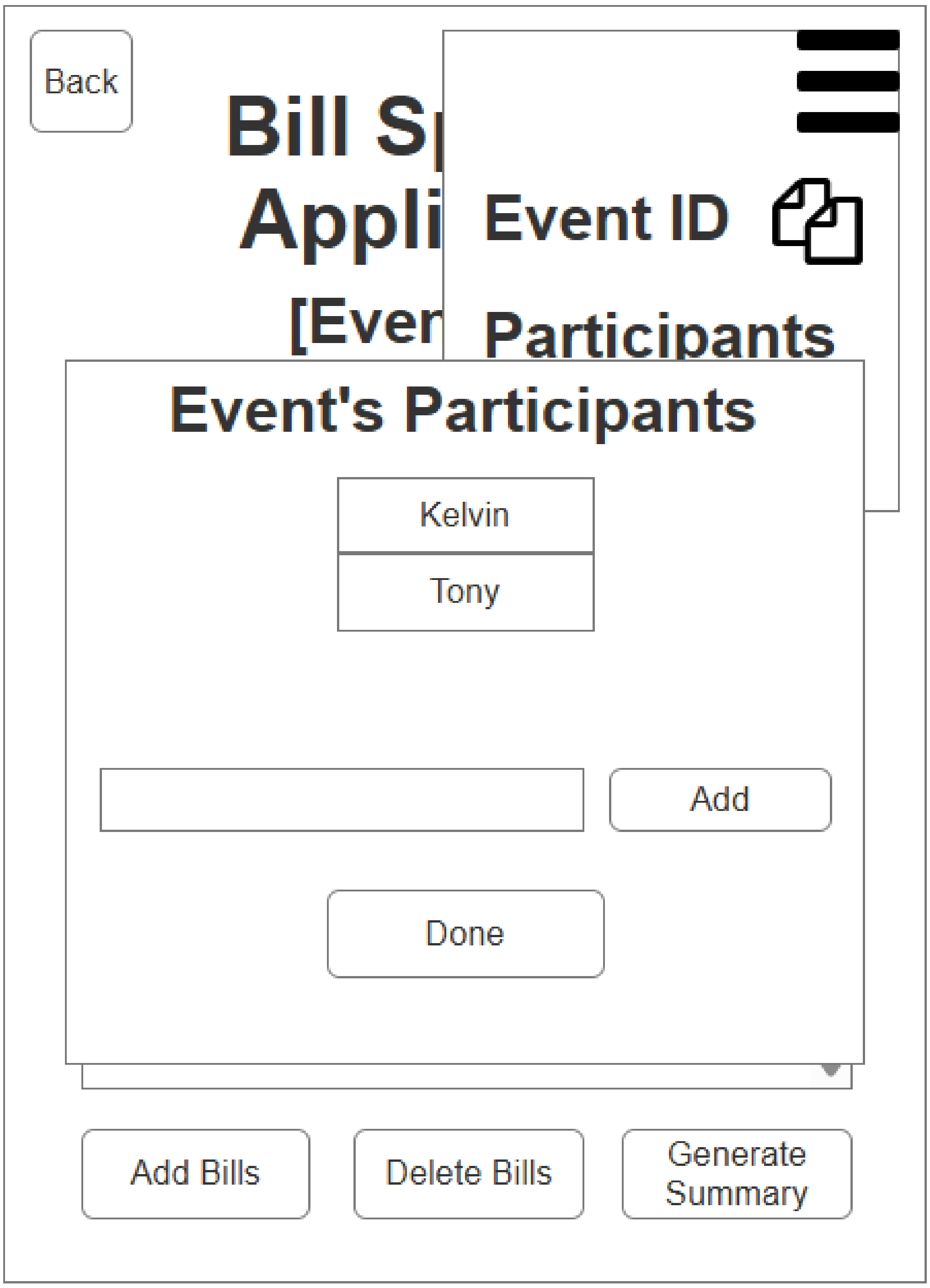


Figure 4.4.10 Managing Event’s Participants Interface

When the participants hyperlink is pressed in the hamburger, user can edit the participants of the event, they can add or remove participants.

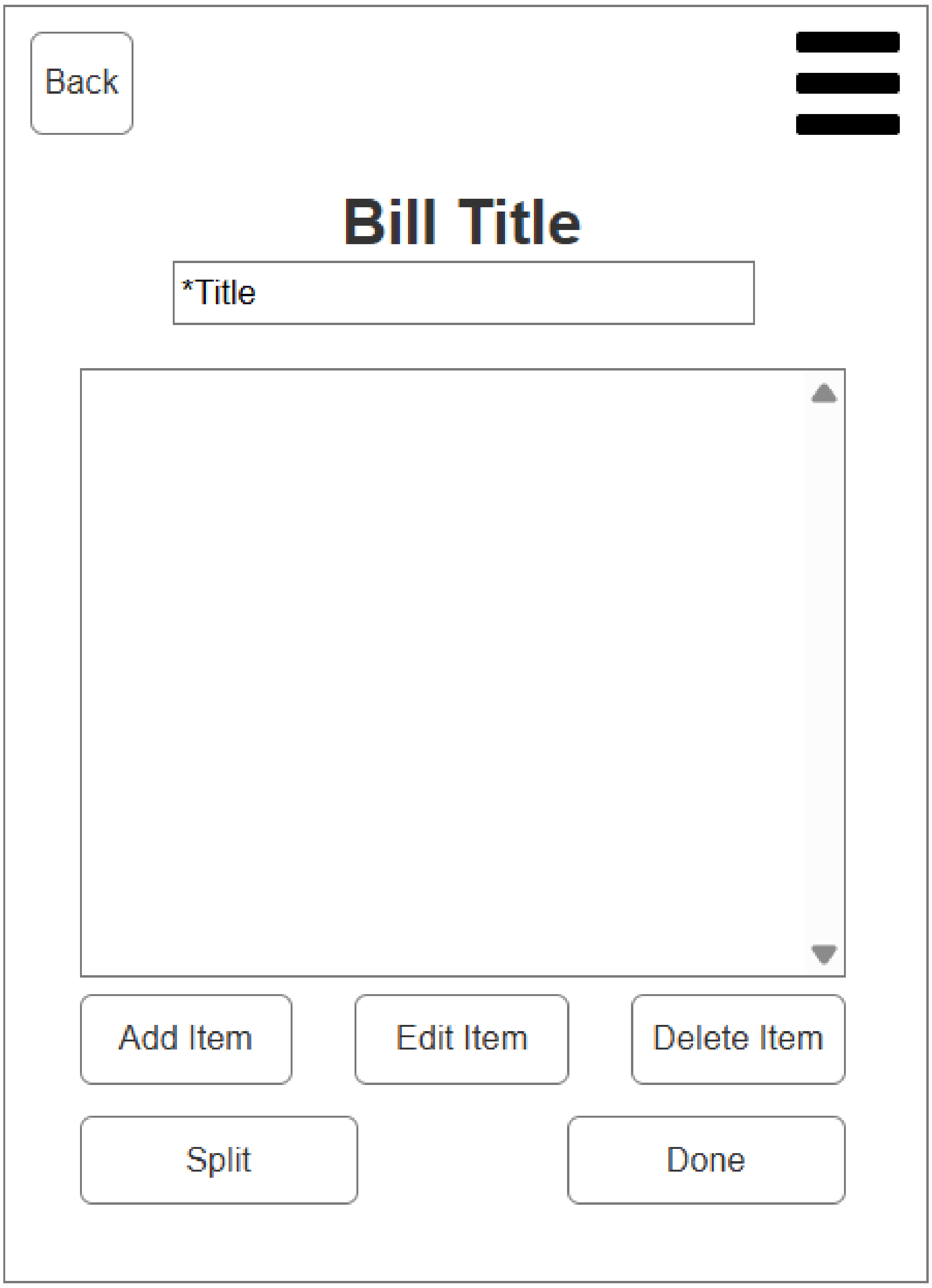


Figure 4.4.11 Add and Edit Bill Page

When the “Add Bill” button is pressed or a bill is selected at the Manage Event Page. This page will appear, it will display the title all the bills that is related to the event if it is a existing event, if it’s a new event, all will be left empty.

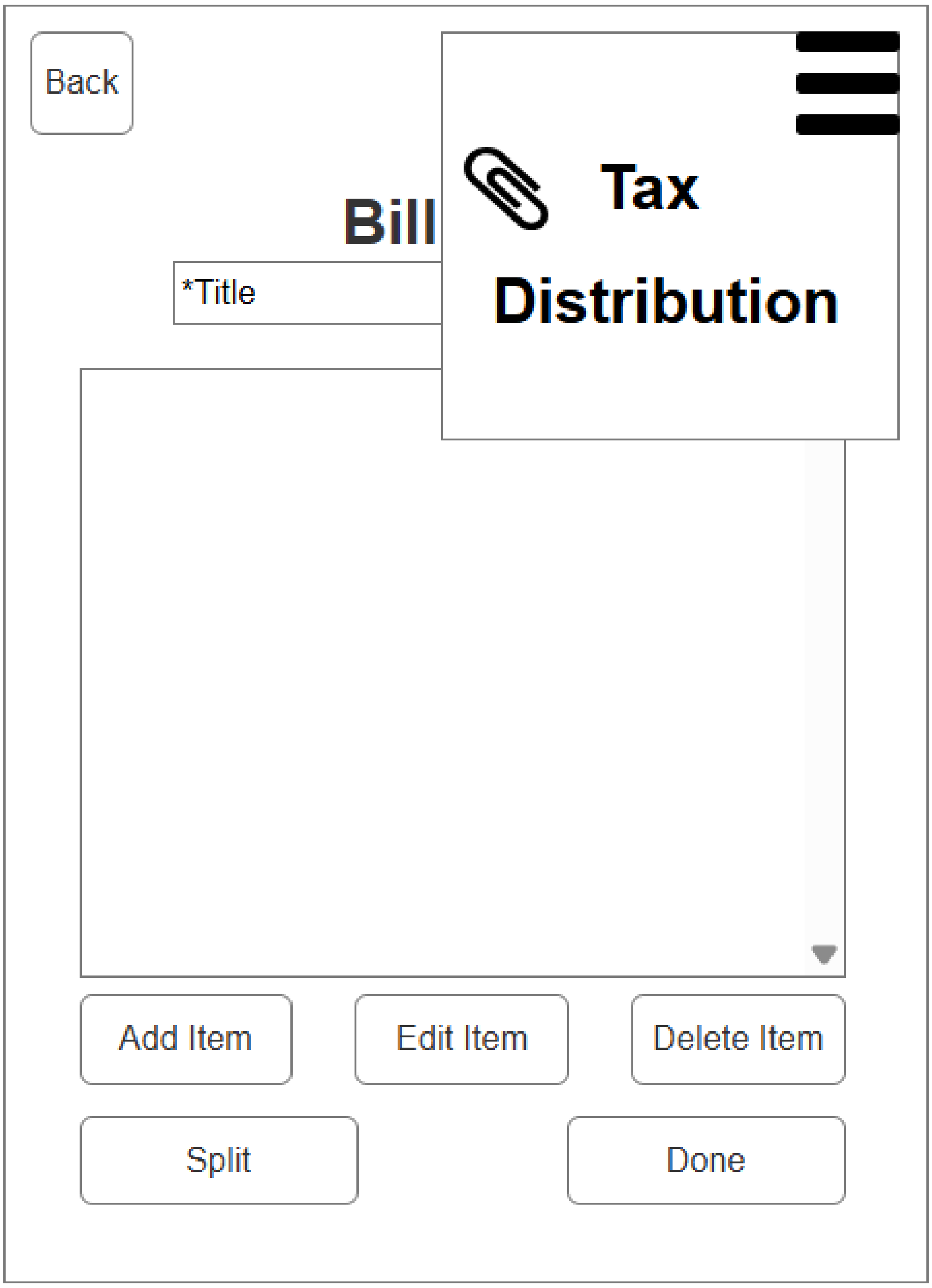


Figure 4.4.12 Hamburger Icon Functions for Manage Bill Page

When the hamburger icon is pressed on manage bill page, a small container appears which allows the user to append the tax, distribution method and also delete the bill

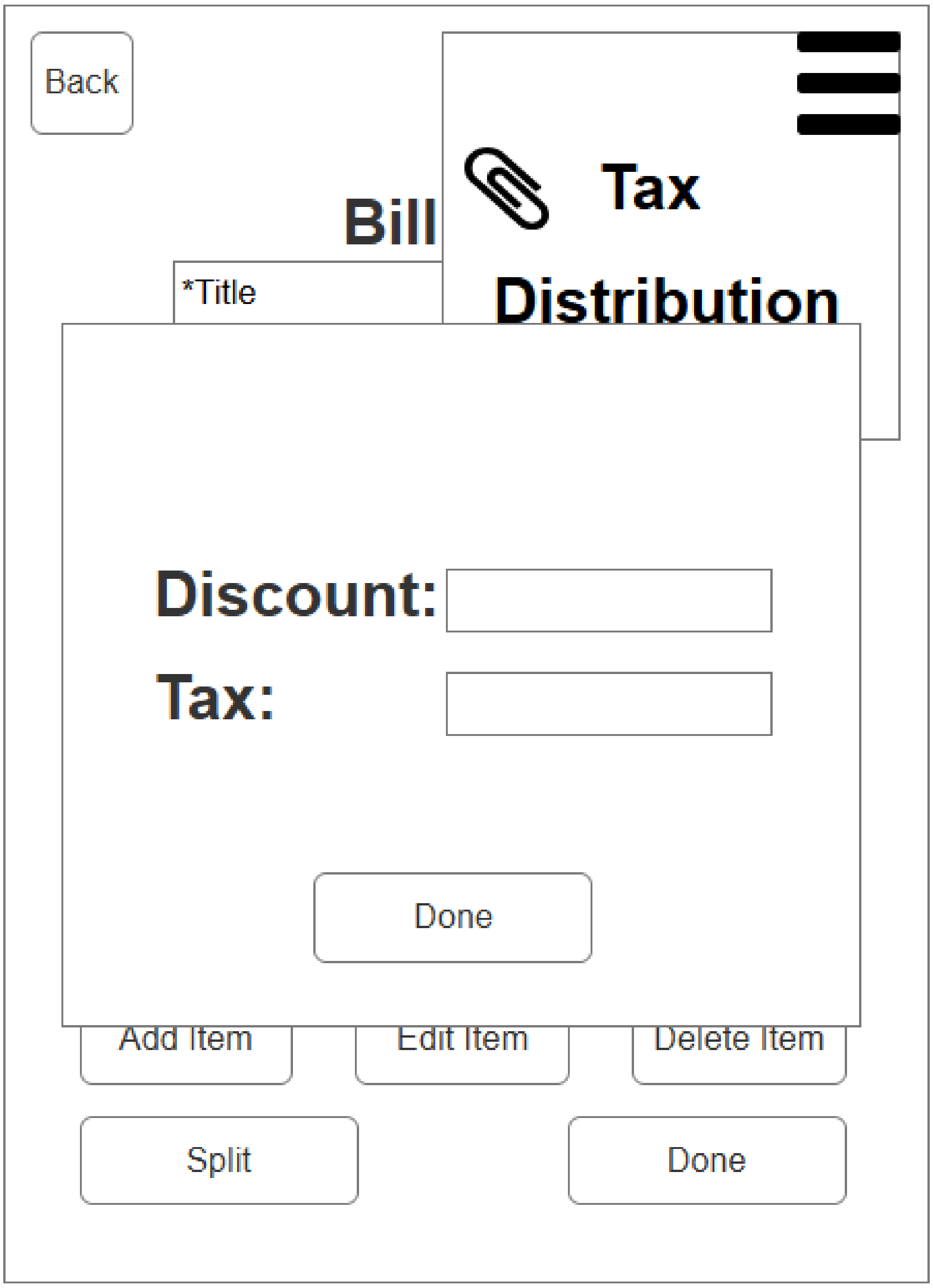


Figure 4.4.13 Appending Tax to Bill Interface

When the tax hyperlink is pressed in the hamburger, user can enter the discount and tax applied on the bills for future calculation

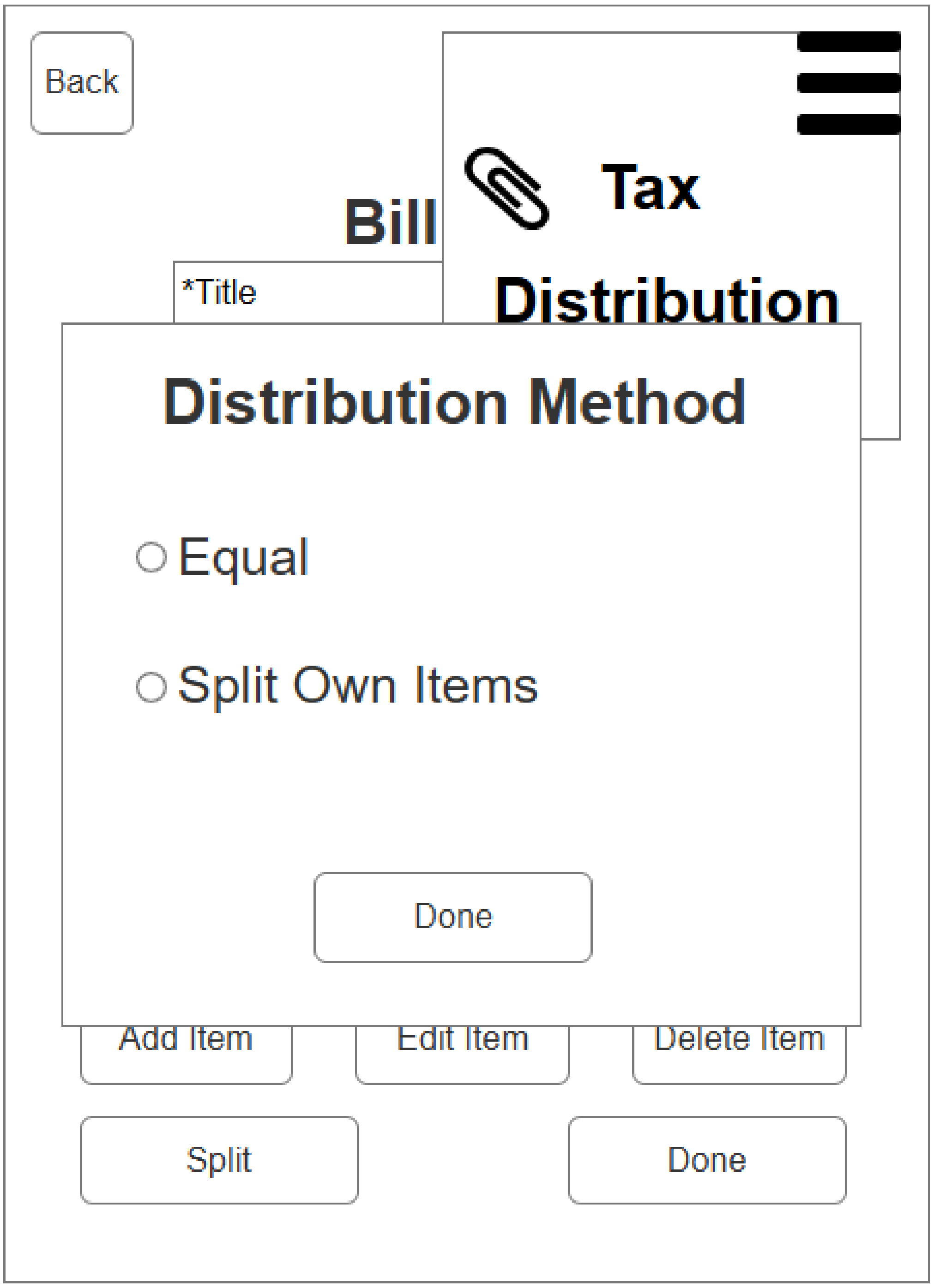


Figure 4.4.14 Appending Distribution to Bill Interface

When the distribution hyperlink is pressed in the hamburger, user can choose what type of cost distribution method to use, by default it will be split own items.

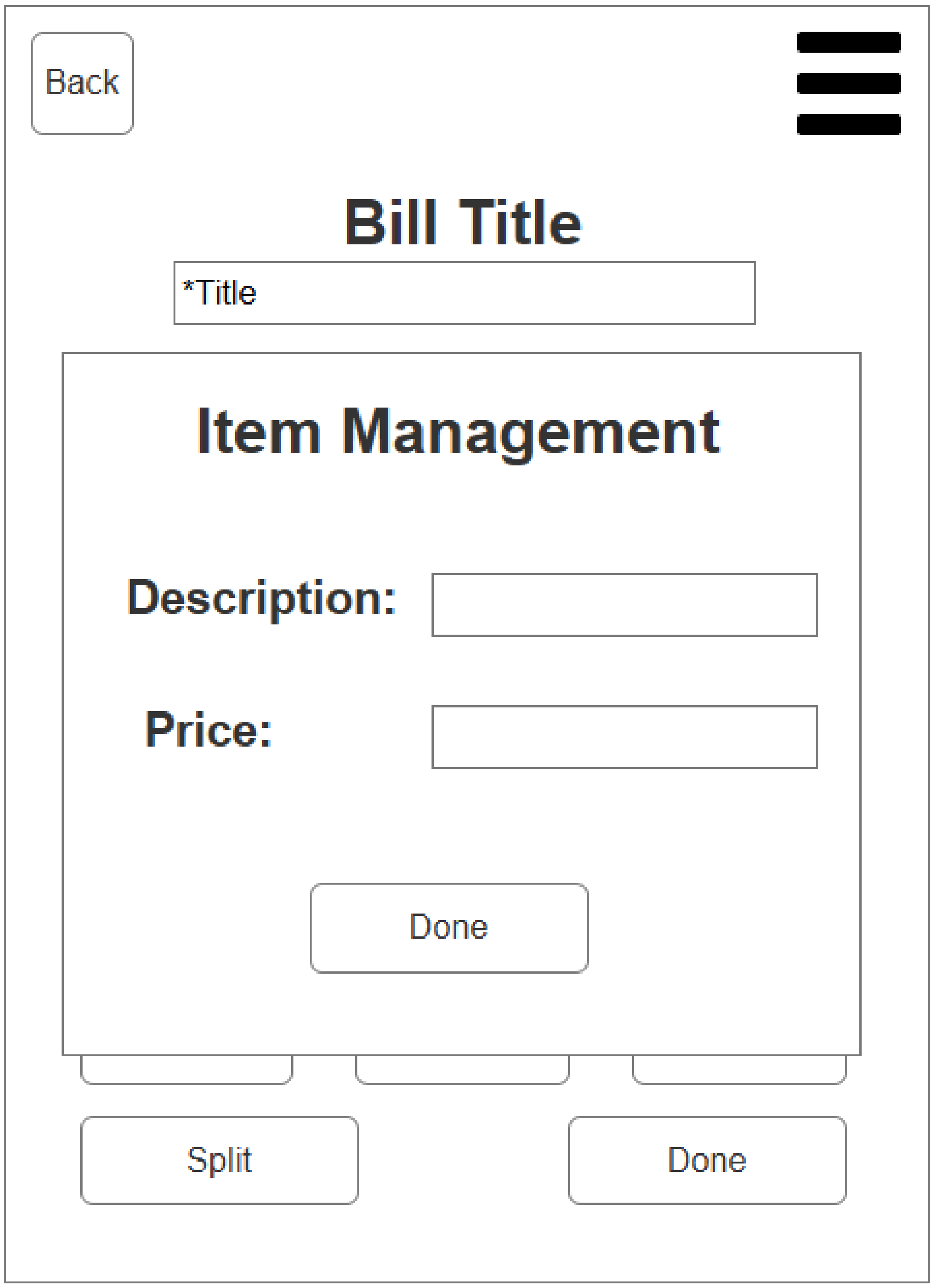


Figure 4.4.15 Item Management Page

This page allows users to enter the item description, item price and participants involved in this item. A question mark will appear at the top right corner to help guide the user through this process.

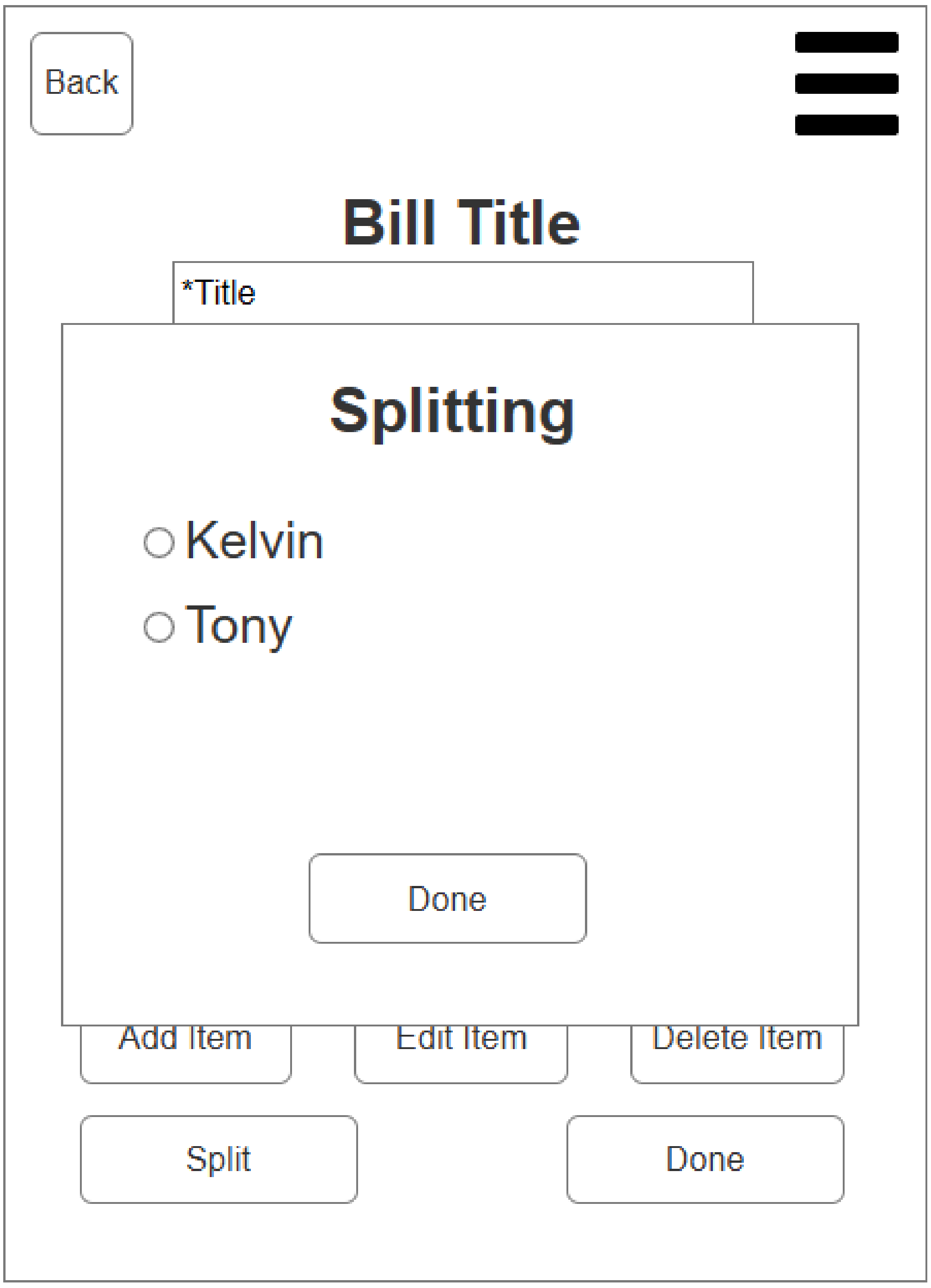


Figure 4.4.16 Bill Splitting Container

This container allows user to choose one or many participants indicating that the items are used or consumed by these participants. This part is the core of the application therefore will need to be care strictly.

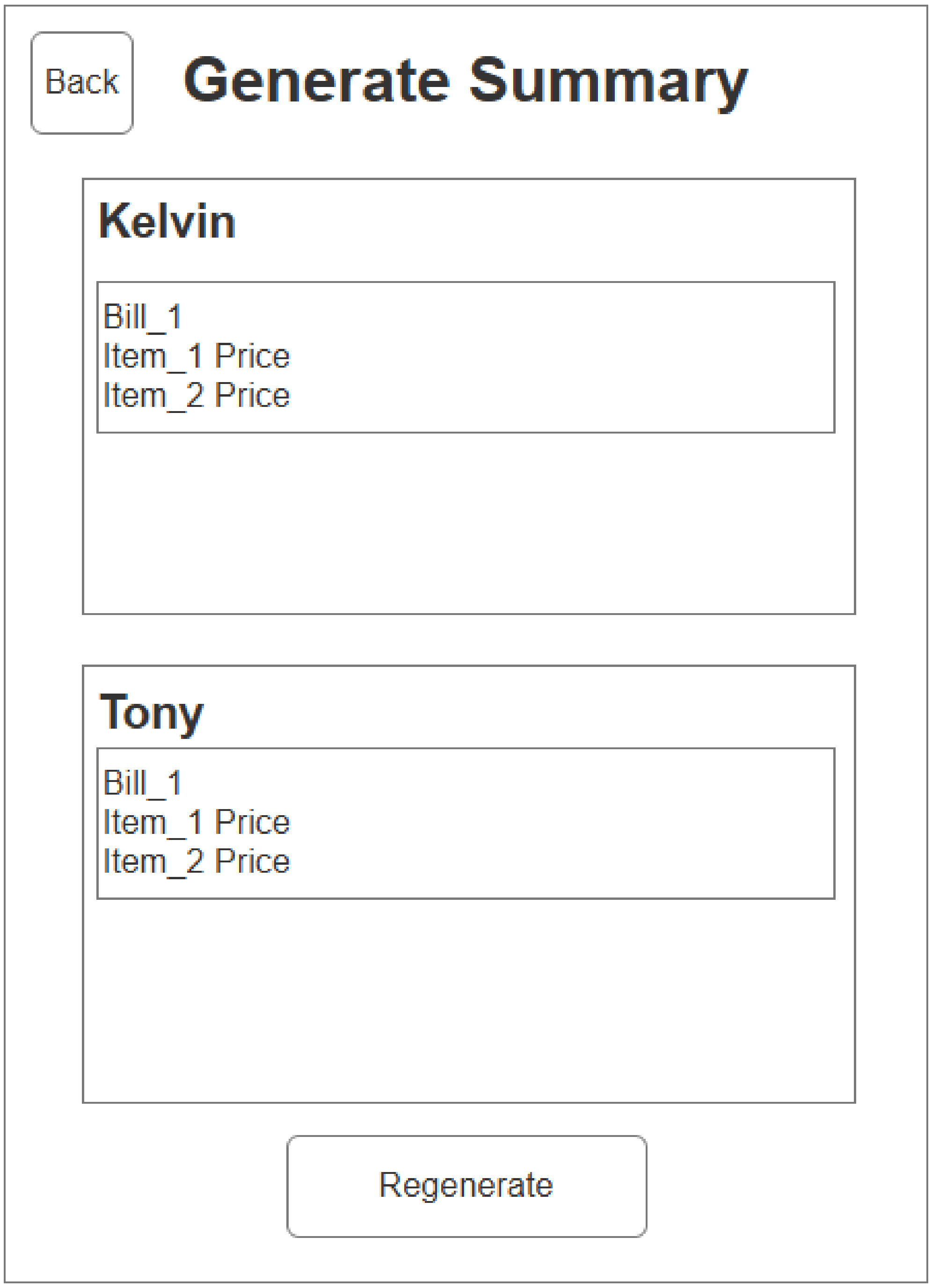


Figure 4.4.17 Generate Summary Page

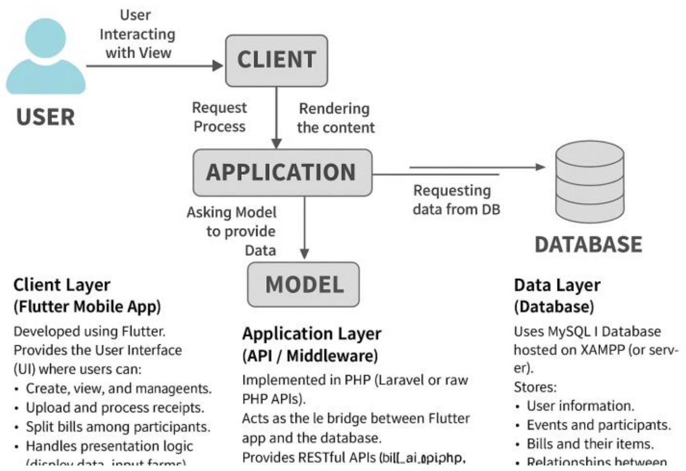
This page will generate the event's summary using artificial intelligence, it will show all the bills and its cost in a table and will generate a summary at the bottom of the screen, if changes are made to the event's bill, user can press the regenerate button to regenerate a new summary.

CHAPTER 5
SYSTEM DESIGN

5.1 Introduction

This chapter will focus on the project application's system design. There will be three sections in this chapter to focus on. Section 5.2 will explain about the system architecture design of the application, this will give a clear understanding of how the structure of the system will look like. Other than that, section 5.3 gives us the perspective of all the UML Diagram designed for this project. Finally, section 5.4 will structure out the the database design of the application giving us an understanding of how data is stored and interact with each other

5.2 System Architecture Design



Client Layer
(Flutter Mobile App)
Developed using Flutter.
Provides the User Interface (UI) where users can:
- Create, view, and manageents.
- Upload and process receipts.
- Split bills among participants.
- Handles presentation logic (display data, input farms)

Application Layer
(API / Middleware)
Implemented in PHP (Laravel or raw PHP APIs).
Acts as the le bridge between Flutter app and the database.
Provides RESTful APIs (bill_ai_api.php,

Data Layer
(Database)
Uses MySQL I Database hosted on XAMPP (or server).
Stores:
- User information.
- Events and participants.
- Bills and their items.
- Relationships between

Figure 5.2.1: Figure of application's system architecture design

The figure above shows the structure of the system's architecture, the client layer is developed using Flutter and it provides the user interface. Users can do tons of functions in this layer such as create, view, and managing events. The Client layer is responsible for handling the presentation logic and user interactions and help communicate with the backend of the application using HTTP requests.

The next layer to be discussed is the application layer, this layer acts as the bridge between the flutter app and the database and it is implemented using PHP. This layer consists of RESTful APIs for example bill_api.php, event_api.php, process_receipt.php to help manage bills, itemsd and also help perform the OCR and AI integration.

The final layer in the data layer which is basically the database of the application. MySQL is used in the application for easy and quick integration. Next, this layer stores all sort of information such as user information, events information, bill information and api configuration.

5.3 Designed UML

5.3.1 ERD Diagram

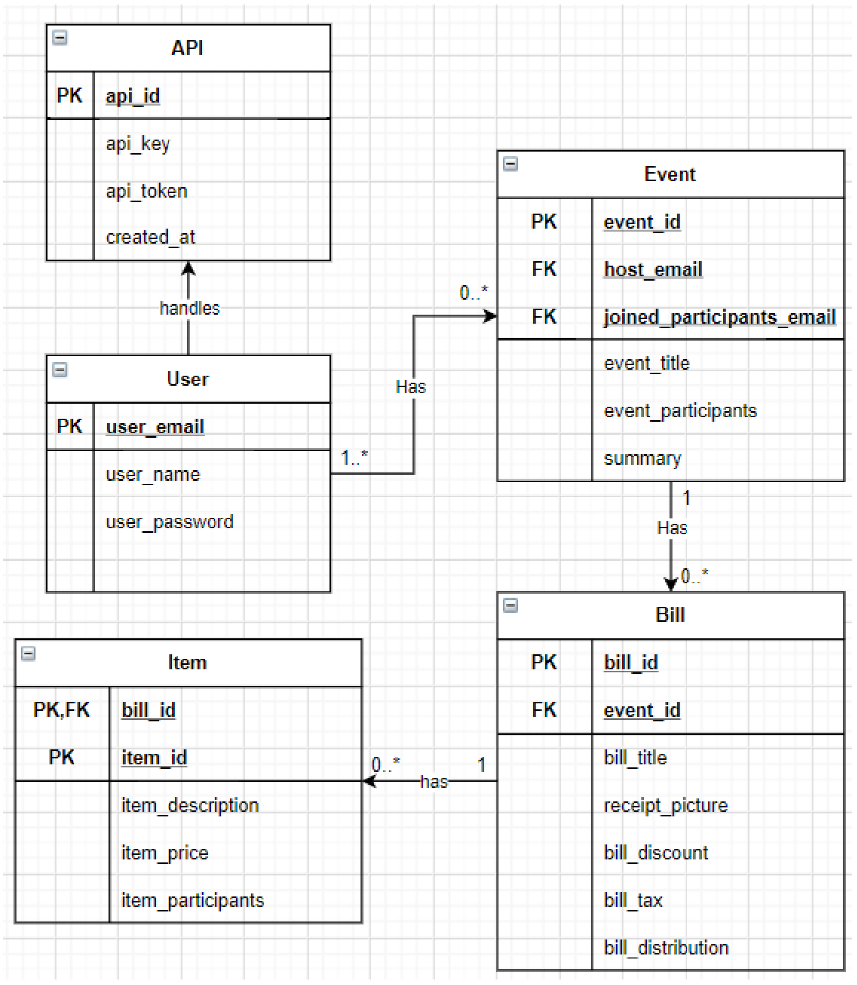


Figure 5.3.1.1: ERD diagram 1

5.3.2 Activity Diagram

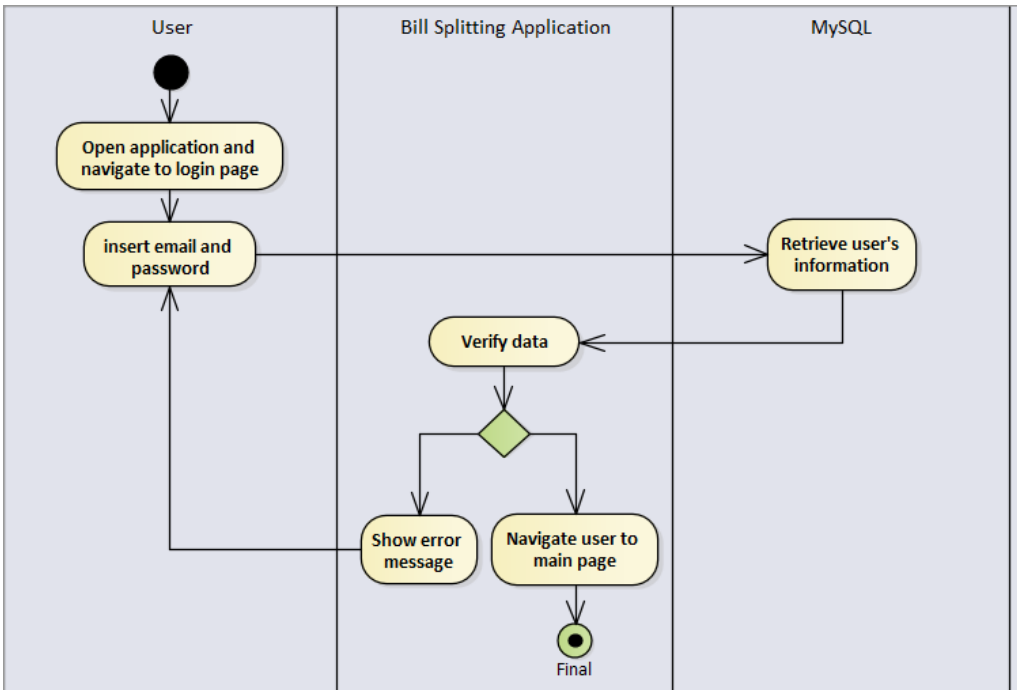

User
Bill Splitting Application
MySQL
Open application and navigate to login page
insert email and password
Retrieve user's information
Verify data
Show error message
Navigate user to main page
Final

Figure 5.3.2.1: Activity diagram of login feature

After user opens the application, user will be opted to enter the application by entering their email address and password. Once inserted, the database will scan if the user's credential exists in it. The application will process the validation, if user is not found in the system, they will need to enter again their credentials, otherwise they will be navigated to the user main page.

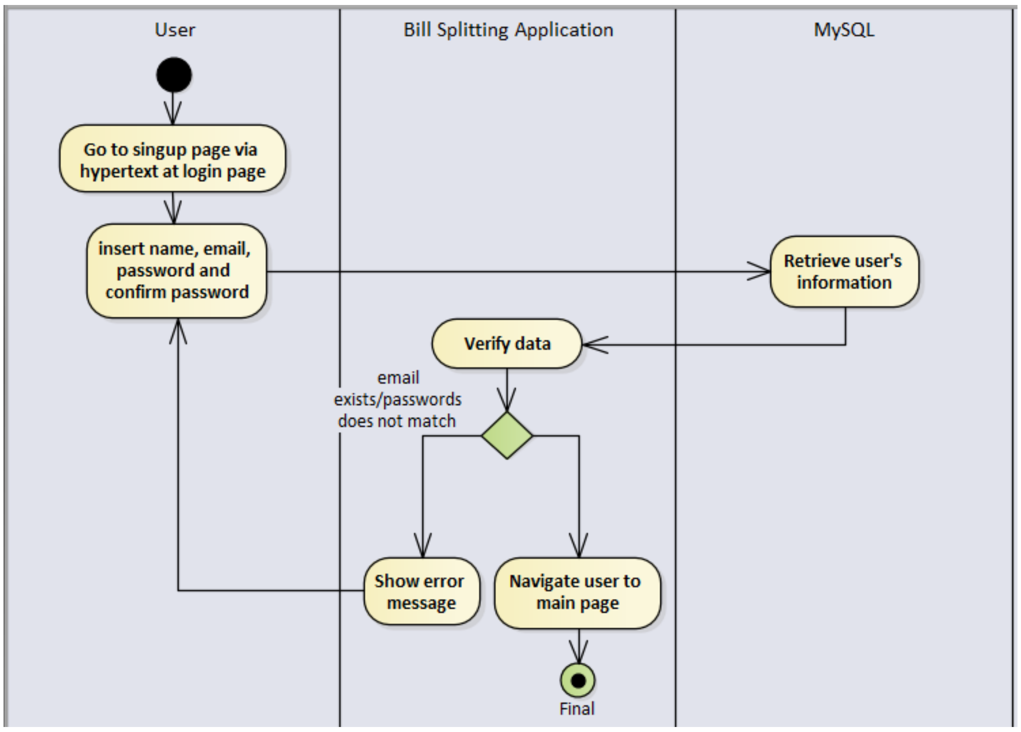

User
Bill Splitting Application
MySQL
Go to singup page via hypertext at login page
insert name, email, password and confirm password
Retrieve user's information
Verify data
email exists/passwords does not match
Show error message
Navigate user to main page
Final

Figure 5.3.2.2: Activity diagram of signup feature

If the user does not have an account in the application, they can head to the sign-up page to register an account via the hypertext in login page. The user will be able to enter their name, email, password and another password confirmation. After they entered the information, the database will retrieve all the users from the user table. The application will then make sure that the email given by the user does not exist in the database and the password is same with the password confirmation. If valid, they will pop up a success message and redirected to the login page. Otherwise, it will show an error message and will opt the user to enter the inputs again.

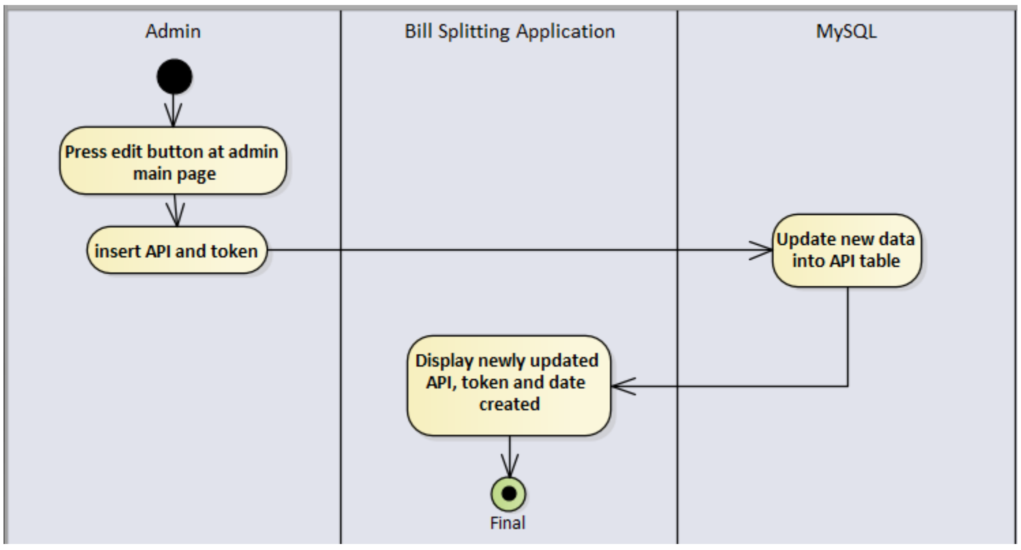

Admin
Bill Splitting Application
MySQL
Press edit button at admin main page
insert API and token
Update new data into API table
Display newly updated API, token and date created
Final

Figure 5.3.2.3: Activity diagram of API management

Admins are allowed to edit the AI API token and key through their admin main page. This can be done by pressing the edit button on the bottom of the screen, the admins then can freely edit the key and token as they wish. Once they are done, the database will save the new key and token given by the admin.

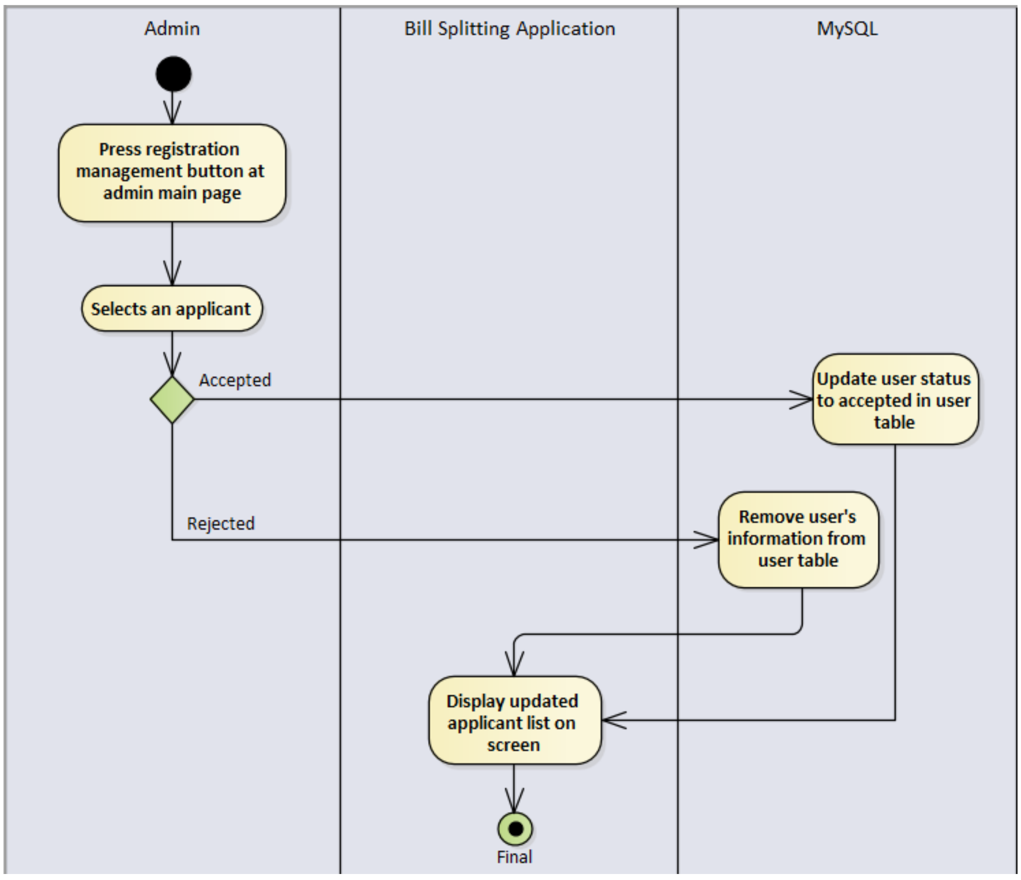

Admin
Bill Splitting Application
MySQL
Press registration management button at admin main page
Selects an applicant
Accepted
Update user status to accepted in user table
Rejected
Remove user's information from user table
Display updated applicant list on screen
Final

Figure 5.3.2.4: Activity diagram of registration management feature

For each new registration from new user, they will need to be accepted from one of the admin to access the application. Agter the admin presses the “Registration List” button at the admin main page. It will display a list of pending applicants ready to be accepted or rejected. The admin can choose one of the applicants and choose either accept or decline. If accepted, the applicant’s user status will change to ‘1’ which is accepted. Otherwise, the applicant’s data will e removed from the database.

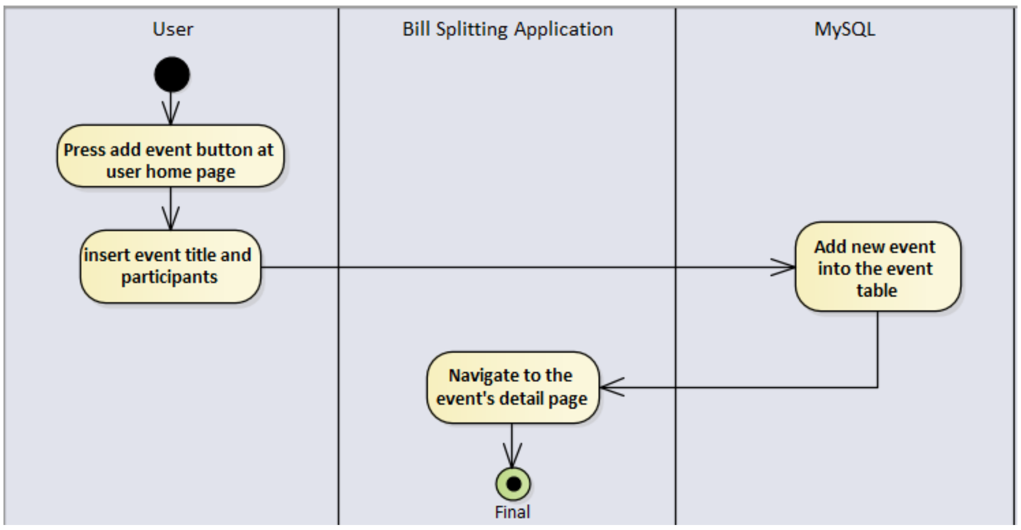

User
Bill Splitting Application
MySQL
Press add event button at user home page
insert event title and participants
Add new event into the event table
Navigate to the event's detail page
Final

Figure 5.3.2.5: Activity diagram of add event feature

User can start a new event by pressing the “Add Event” button on their main page. The new page will require users to enter a title and add in participants that will be participating in the event. If title is left blank, a default “Untitled” will be replaced as the title. The database will create a new event based on the title and participants and the user will be redirected back to the main page.

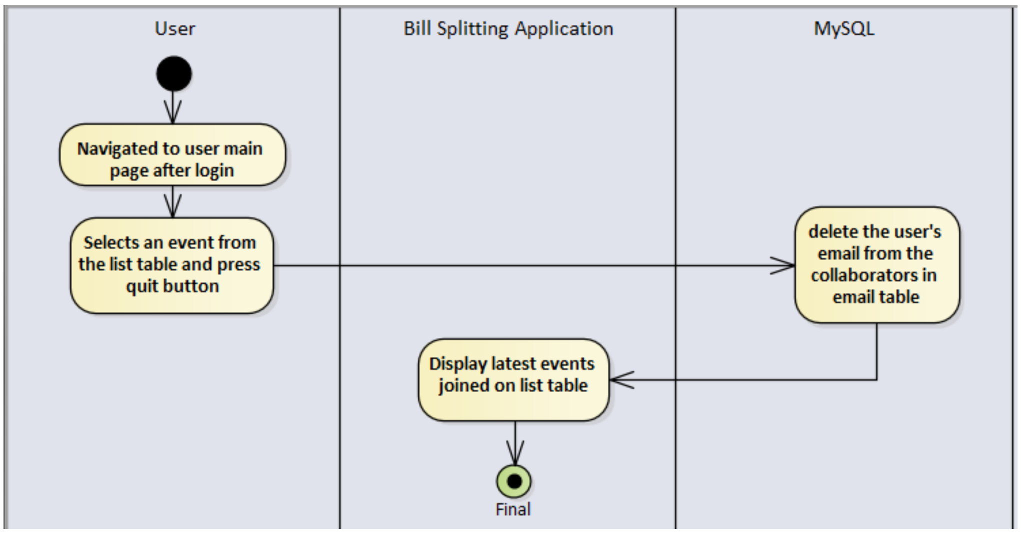

User
Bill Splitting Application
MySQL
Navigated to user main page after login
Selects an event from the list table and press quit button
delete the user's email from the collaborators in email table
Display latest events joined on list table
Final

Figure 5.3.2.6: Activity diagram of quit event feature

User that would like to quit an event will have to select an event from their main page and press the "Quit" button. At the database, it will remove the user's email from the event's collaborators_email column. The list will display the latest information with the exited event being missing.

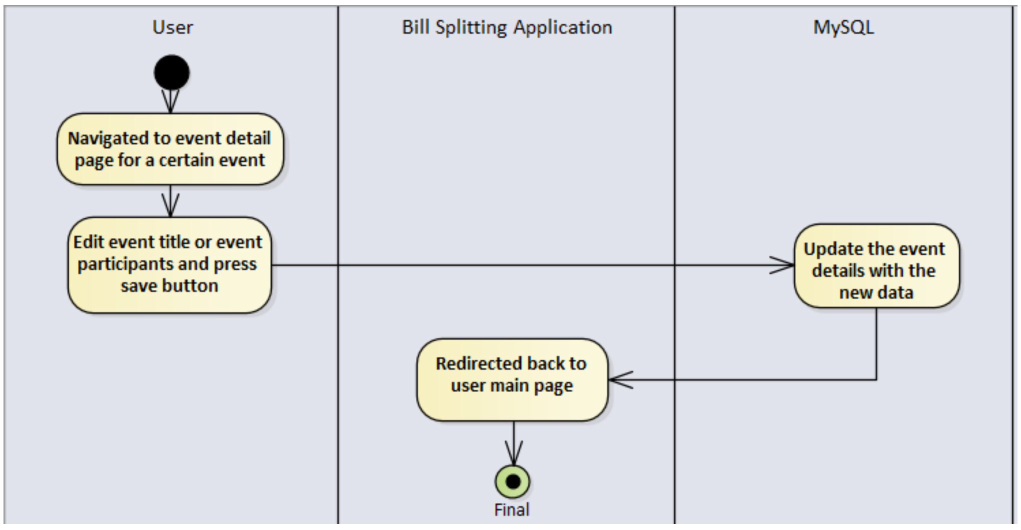

User
Bill Splitting Application
MySQL
Navigated to event detail page for a certain event
Edit event title or event participants and press save button
Update the event details with the new data
Redirected back to user main page
Final

Figure 5.3.2.7: Activity diagram of edit event feature

In this feature, the user will want to edit information in the event such as the titles or participants. To do so, they will need to select an event form the user main page which will direct them to the event detail page, here user can make changes to the title or the participants and press save, the database will get the updated data and update accordingly.

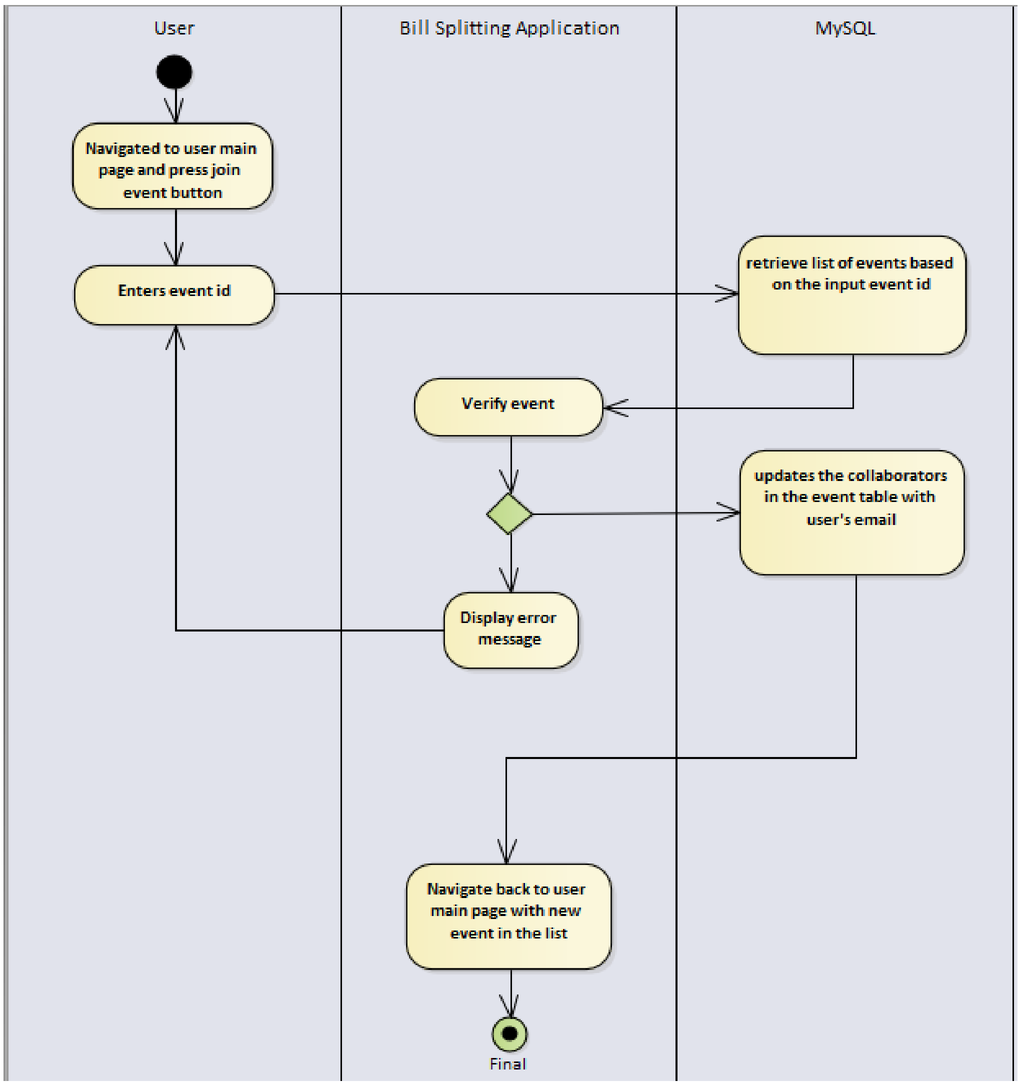

User
Bill Splitting Application
MySQL
Navigated to user main page and press join event button
Enters event id
retrieve list of events based on the input event id
Verify event
updates the collaborators in the event table with user's email
Display error message
Navigate back to user main page with new event in the list
Final

Figure 5.3.2.8: Activity diagram of join event feature

For user who wishes to collaborate an event hosted by other users, they can do so by joining their event. In the user main page, user can press the "Join Event" button and enter the event id of the event they wish to join. After entering the event id and pressing join, the database will a list of events and the application will do verification. If the event does not exist, user will be needed to enter another event id. Otherwise, the user's email will be added into the event's collaborator_email column in the database. The user will then be redirected back to the main page.

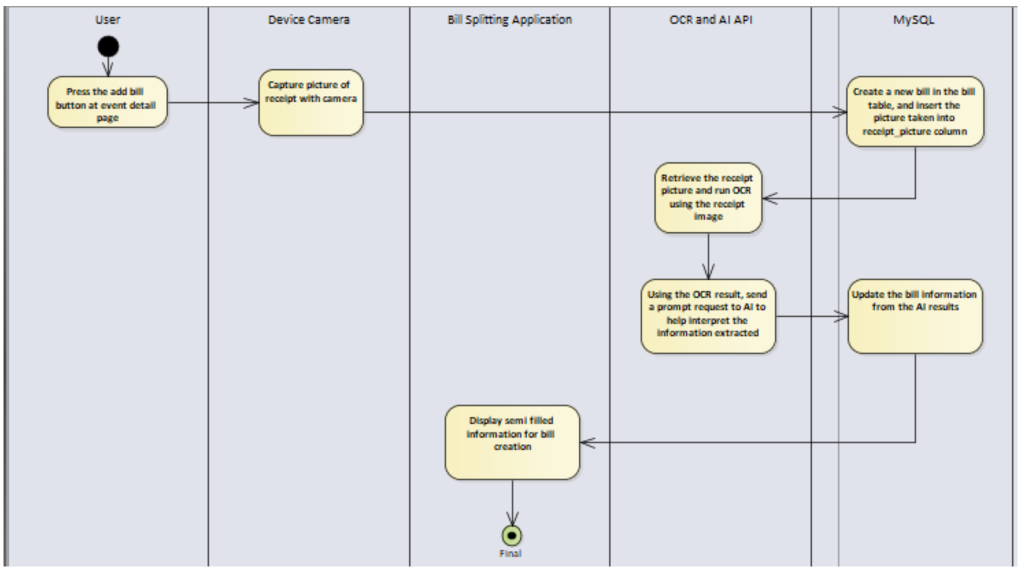

User
Device Camera
Bill Splitting Application
OCR and AI API
MySQL
Press the add bill button at event detail page
Capture picture of receipt with camera
Create a new bill in the bill table, and insert the picture taken into receipt_picture column
Retrieve the receipt picture and run OCR using the receipt image
Using the OCR result, send a prompt request to AI to help interpret the information extracted
Update the bill information from the AI results
Display semi filled information for bill creation
Final

Figure 5.3.2.9: Activity diagram of creating bill via camera feature

To create a bill using the camera functions. User will press the “Add Bill” button in the event details page. Automatically, the camera will try to open up, if the permission is not granted, the application will ask the user’s device permission to access the device’s camera, user then can take picture of the receipt and crop it to only fit the items and price of the receipt to increase accuracy. The receipt image will first be saved into the database before sending it to process OCR and AI, the interpreted information which contains the item’s description and price will then be added into the bill’s items. The user will then be directed to bill’s details page with the information extracted.

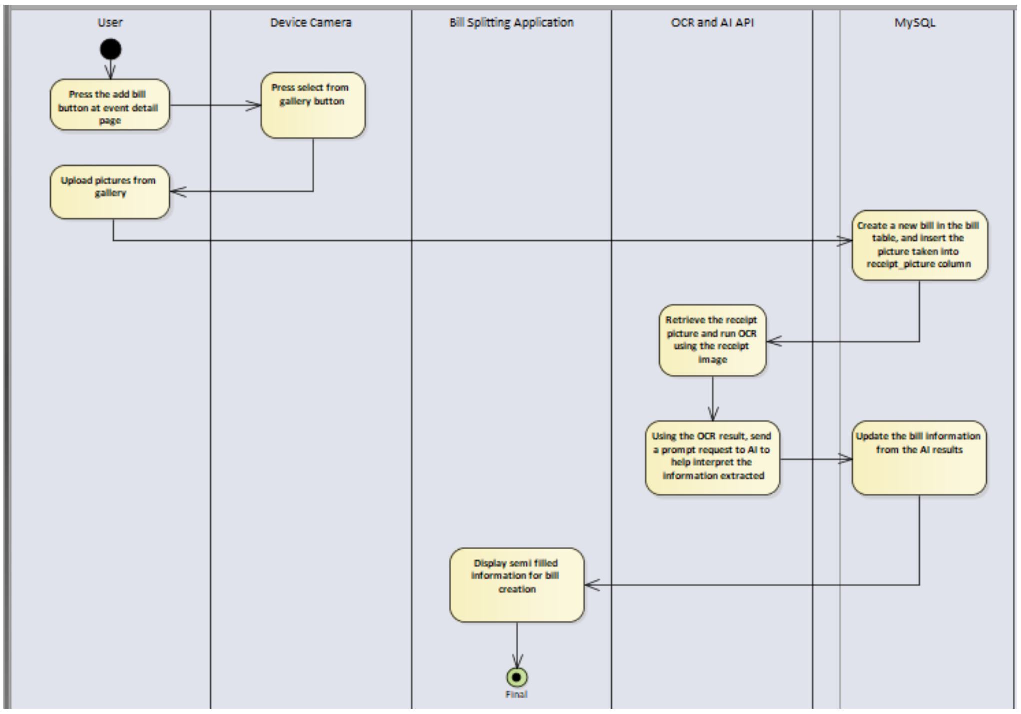

User
Device Camera
Bill Splitting Application
OCR and AI API
MySQL
Press the add bill button at event detail page
Press select from gallery button
Upload pictures from gallery
Create a new bill in the bill table, and insert the picture taken into receipt_picture column
Retrieve the receipt picture and run OCR using the receipt image
Using the OCR result, send a prompt request to AI to help interpret the information extracted
Update the bill information from the AI results
Display semi filled information for bill creation
Final

Figure 5.3.2.10: Activity diagram of creating bill via gallery feature

To create a bill using the gallery functions. After pressing the "Add Bill" button in the event details page. Automatically, the camera will try to open up, on the buttom of the screen, there will be abutton where the user can opt to choose pictures from their gallery instead of taking picture. After user presses "Choose from gallery", if permission is not granted, the application will ask the user's device permission to access the device's gallery, user then can select a picture of the receipt and crop it to only fit the items and price of the receipt. First, the receipt's image will first be saved into the database before sending it to process OCR and AI, the interpreted information which contains the item's description and price will then be added into the bill's items. The user will then be directed to bill's details page with the information extracted.

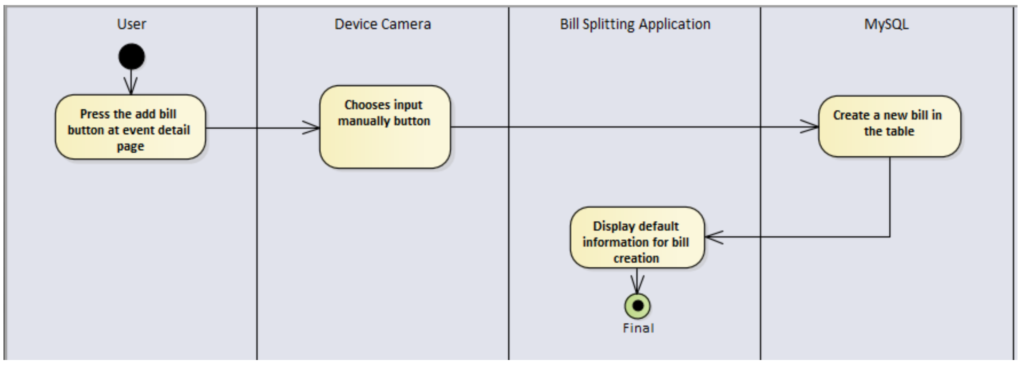

User
Device Camera
Bill Splitting Application
MySQL
Press the add bill button at event detail page
Chooses input manually button
Create a new bill in the table
Display default information for bill creation
Final

Figure 5.3.2.11: Activity diagram of creating bill manually feature

To create a bill manually. After pressing the "Add Bill" button in the event details page, there will be a button at the bottom of the screen "Manually input". Pressed, database will create a new bill with no items whatsoever for the user to enter as they wish.

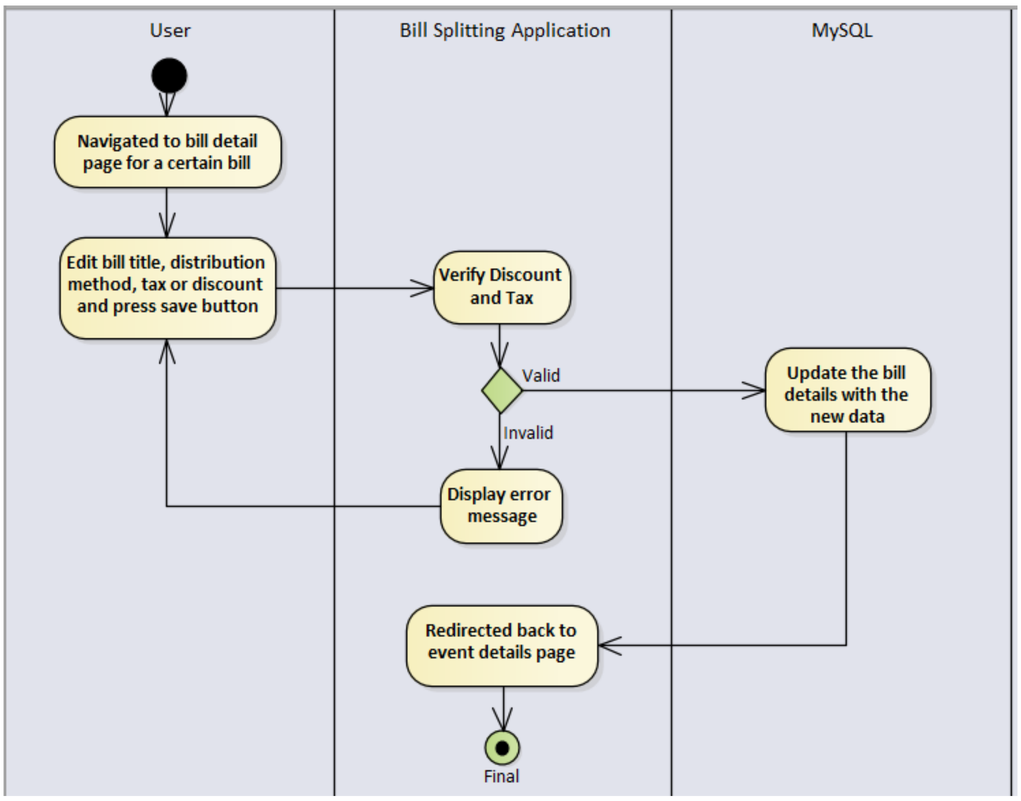

User
Bill Splitting Application
MySQL
Navigated to bill detail page for a certain bill
Edit bill title, distribution method, tax or discount and press save button
Verify Discount and Tax
Valid
Invalid
Update the bill details with the new data
Display error message
Redirected back to event details page
Final

Figure 5.3.2.12: Activity diagram of edit bill feature

In this feature, the user will want to edit information for the bill such as the titles, distribution method, tax or discount. To do so, they will need to select a bill form the event detail page which will direct them to the bill detail page, here user can make changes. The application will verify the discount and tax before sending the data to the database, if no error occurs, the database will update with the given input.

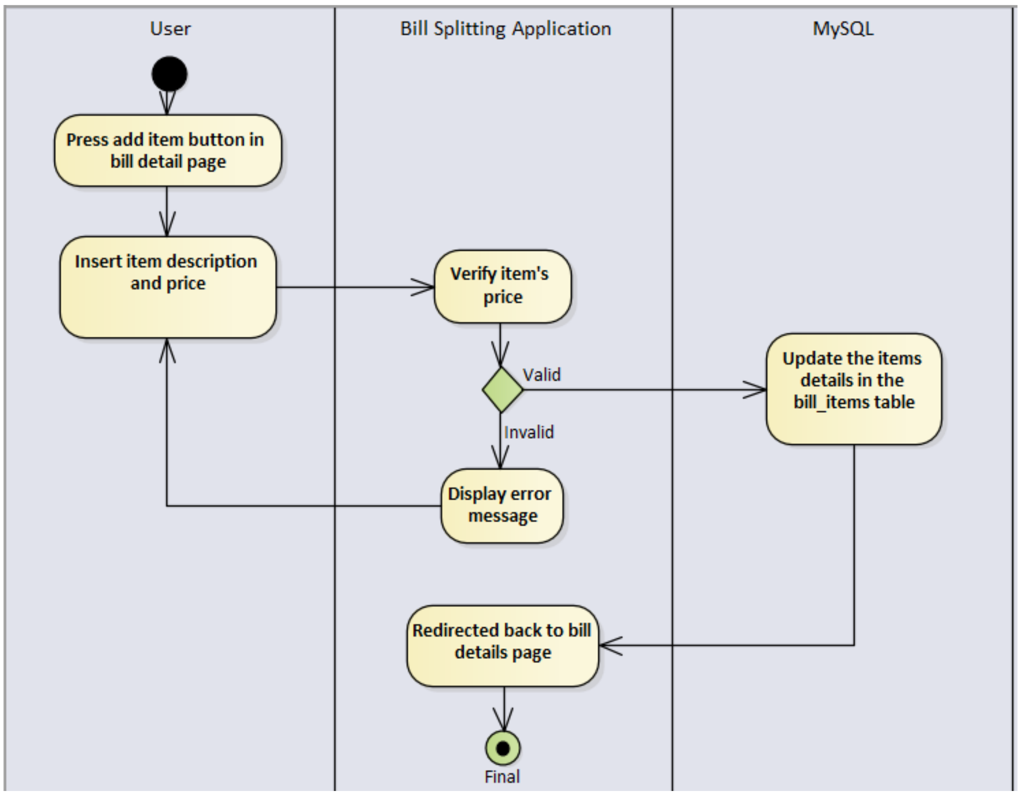

User
Bill Splitting Application
MySQL
Press add item button in bill detail page
Insert item description and price
Verify item's price
Valid
Invalid
Update the items details in the bill_items table
Display error message
Redirected back to bill details page
Final

Figure 5.3.2.13: Activity diagram of add items feature

At the bill details page, after pressing the "Add Items" button, a container will come out and the user will be opt to enter the item's description and price, the application will first verify the item's price if it is a verified double data type. If valid, the item's information will be added in the database. Otherwise, an error message will display out.

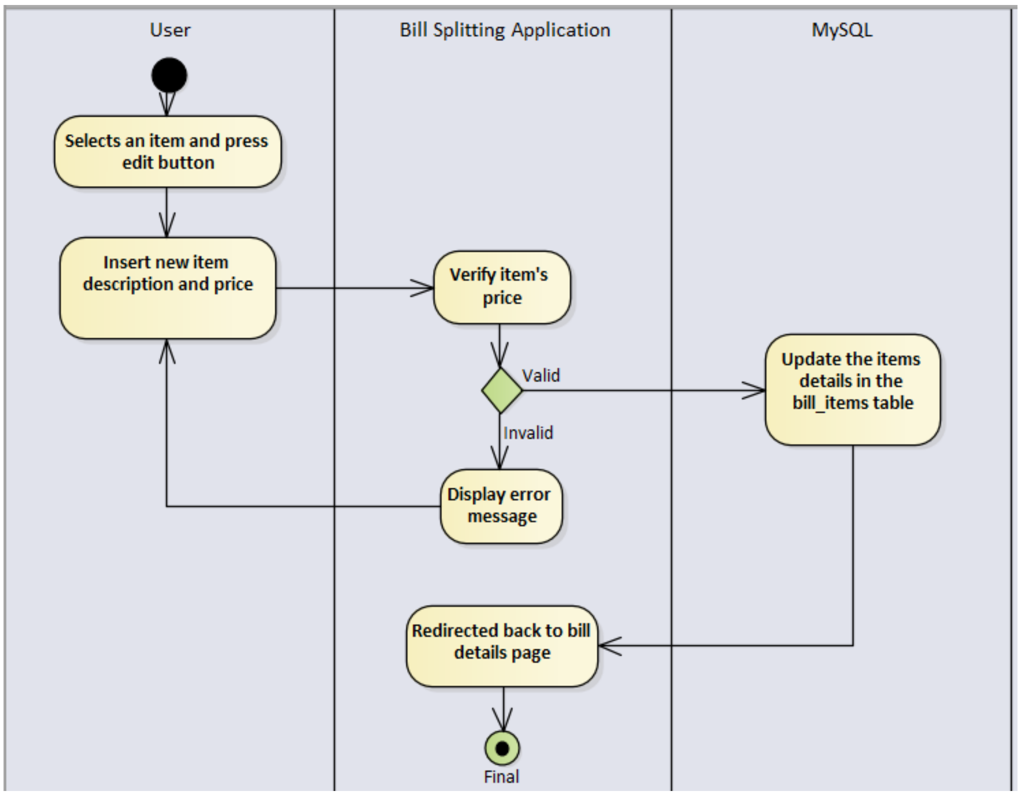

User
Bill Splitting Application
MySQL
Selects an item and press edit button
Insert new item description and price
Verify item's price
Valid
Invalid
Update the items details in the bill_items table
Display error message
Redirected back to bill details page
Final

Figure 5.3.2.14: Activity diagram of edit items feature

This feature start with user selecting an item from the bill detail page, the existing item's information will show in the input space. Users are free to edit it with the new data. Validation will be made for the price's data type to ensure no error occurs. If all is good, the new data will be updated into the database.

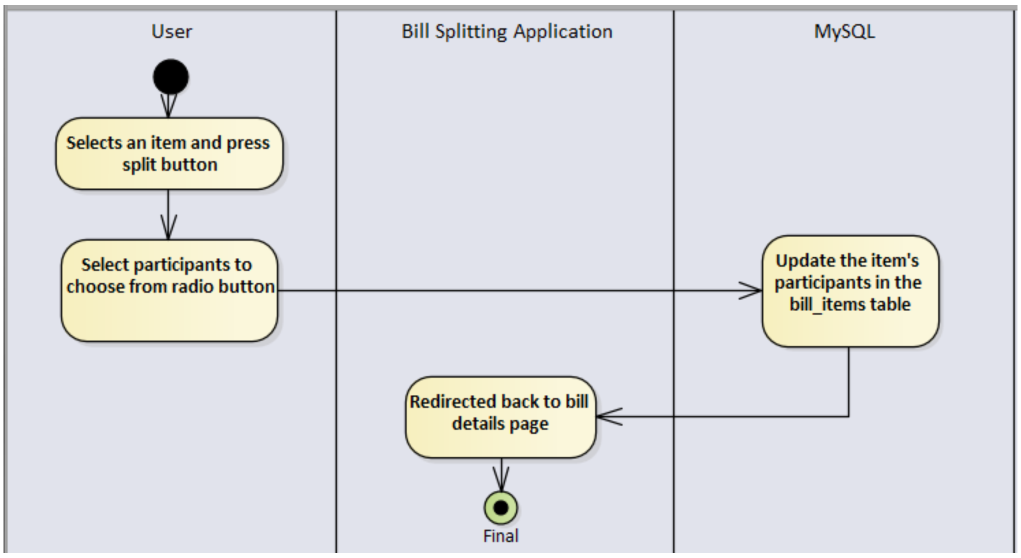

User
Bill Splitting Application
MySQL
Selects an item and press split button
Select participants to choose from radio button
Update the item's participants in the bill_items table
Redirected back to bill details page
Final

Figure 5.3.2.15: Activity diagram of split items feature

The user can select an item and press the "Split" button. This feature is crucial as it will affect the core of the application. A container will be displayed and a list of the event’s participants will be displayed via radio button. The user can choose the radio button if the participant is responsible for the item. After the “Done” button is pressed, the information will be saved in the database.

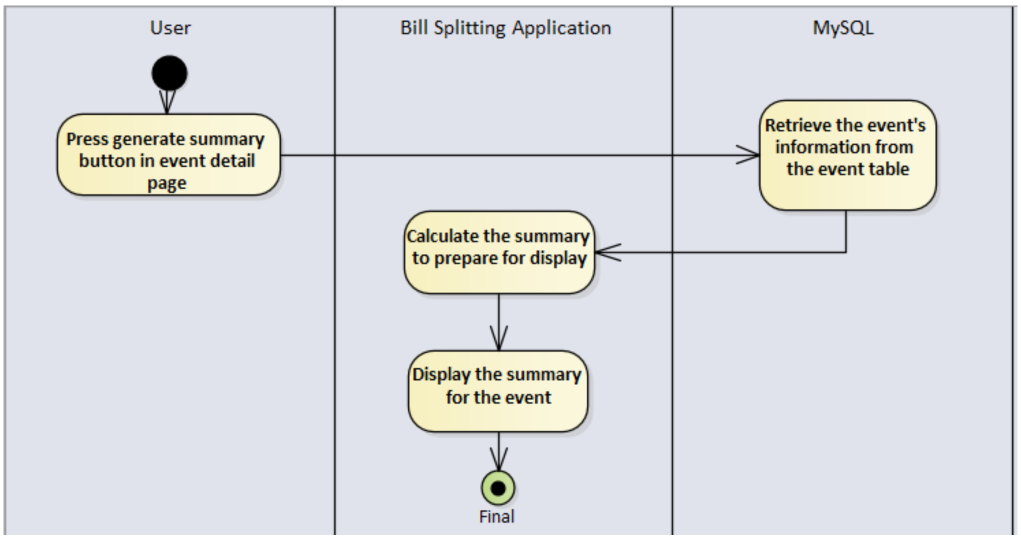

User
Bill Splitting Application
MySQL
Press generate summary button in event detail page
Retrieve the event's information from the event table
Calculate the summary to prepare for display
Display the summary for the event
Final

Figure 5.3.2.16: Activity diagram of generate summary feature

At the event details page, a button “Generate Summary” will lead the user to this function. In the generate summary page, if there is previous existing summary, it will display out. On the bottom of the screen is a button “Regenerate Summary”. After pressing it, the database will retrieve the information needed to calculate the summary to the application. The application will then summarize the whole event through a algorithm and it will be displayed on the screen.

5.4 Database Design

5.4.1 Data Dictionary

The tables below will be the data dictionary for the bill splitting application. It will be stored in MySQL with four tables being used which are titled as User, Event, Bill, and Item Description.

Table 5.4.1.1 Data dictionary of the User table

Attribute	Data Type	Key	Description
user_email	VARCHAR(255)	Primary Key	Unique key for each user.
user_name	VARCHAR(255)		Name of the user

user_password	VARCHAR(255)		Password created by the user for user authentication.
user_status	BOOLEAN		Status of registration application

Table 5.4.1.2 Data dictionary of the Event Table

Attribute	Data Type	Key	Description
event_id	INT AUTO_INCREMENT	Primary Key	Unique key for each event
host_email	VARCHAR(255)	Foreign Key	Reference user_email in User table as the event's host
collaborators_ email	VARCHAR(255)	Foreign Key	Reference user-email in User table as the event's participants
event_title	VARCHAR(255)		Title of the event
event_ participants	TEXT		List of participants of the event.
Summary	TEXT		Summary text of the event

Table 5.4.1.3 Data dictionary of the Bill Table

Attribute	Data Type	Key	Description
bill_id	INT AUTO_INCREMENT	Primary Key	Unique key for each bill.
event_id	INT	Foreign Key	References event_id in Event table as event selected.
bill_title	VARCHAR(255)		Name or tile of the bill

receipt_picture	BLOB		Image or the file of the receipt picture
bill_discount	DECIMAL(10,2)		Discount given on the bill
bill_tax	DECIMAL(10,2)		Tax occurred on the bill
bill_distribution	VARCHAR(255)		The method on how this bill is split between participants

Table 5.4.1.4 Data dictionary of the Item Description Table.

Attribute	Data Type	Key	Description
bill_id	INT	Primary Key Foreign Key	Reference bill_id in Bill table as the bill selected.
item_id	INT AUTO_INCREMENT	Primary Key	Unique key for each item
item_ description	VARCHAR(255)		Description of the item.
item_price	DECIMAL(10,2)		Price of the item
item_ pariticipants	TEXT		Participants who are involved with the item

Table 5.4.1.5 Data dictionary of API Table.

Attribute	Data Type	Key	Description
api_id	INT AUTO_INCREMENT	Primary Key	Unique key for each API
api_key	VARCHAR(255)		Shared API key
api_token	VARCHAR(255)		Shared token

created_at	DATETIME		When the key was issued

CHAPTER 6
SYSTEM IMPLEMENTATION

6.1 Introduction

This chapter will focus on how the system is implemented from start to finish. Section 6.2 will talk about the software tools that were used to develop the finish application. Also, Section 6.3 talks about how frameworks and tools were used to support the application which includes the system's and also database. Next, Section 6.4 will showcase the visual demonstration for the main feature of the system to give a better understanding of of the application. Finally, Section 6.5 will discuss about the problem that was encountered and how it was solved during the process of developing the application.

6.2 Software Setup

To successfully develop the application, a few essential software applications were needed to be installed and started. Such software applications are Visual Studio Code, XAMPP, and Android Studio. These applications were crutial to edit, test, and also run the application locally.

6.2.1 Visual Studio Code

Visual Studio Code is one and only code editor software that were used to develop the application. One of the reason why is because it can support many kinds of programming language such as Dart and PHP which are the core language for this application to work. Also, no special terminal will need used as there is a built-in terminal ready to be used in the app which is convenient for developers. Visual Studio Code can be found and downloaded online or via the link: https://code.visualstudio.com/download

6.2.2 XAMPP

XAMPP is a local server software used to help run and test the application locally. The reason why it is used is because it is a complete package that includes Apache as the server and also it supports MySQL which is the database framework that is used in this project. XAMPP can be found and downloaded online or via the link: https://www.apachefriends.org/download.html

6.2.3 Android Studio

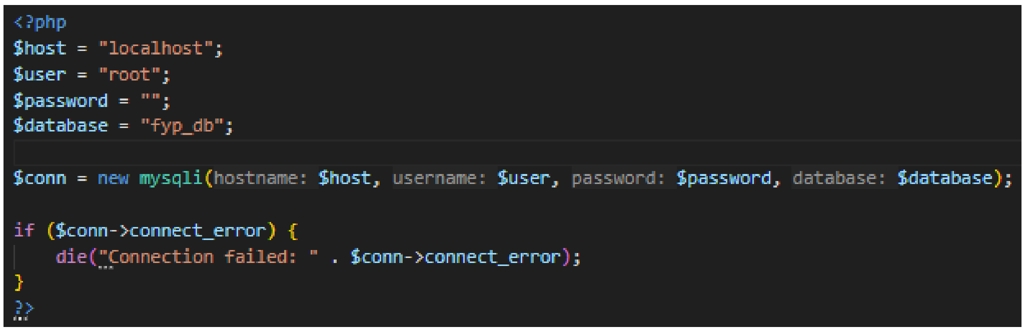
Android Studio is the main integrated development environment (IDE) used to help view and test the mobile application. It provides this project with an emulator to help view the it. Also, it supports Flutter's integration which makes it easier to design, develop and run. Android Studio can be found and downloaded online or via the link: https://developer.android.com/studio

6.3 Setting and Configuration

6.3.1 System Configuration

The "yaml" file is set-up properly to ensure that all the required dependencies, libraries, and plugins are defined and installed when the project is built. This file is crucial to maintain consistency across different environments.

6.3.2 Database Configuration



```php
<?php
$host = "localhost";
$user = "root";
$password = "";
$database = "fyp_db";

$conn = new mysqli(hostname: $host, username: $user, password: $password, database: $database);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

Figure 6.3.2.1: Code snippet connecting application to database

This line of php code is written in db.php is to help the application to connect with the local database that is ran with XAMPP, the name of the database is "fyp_db" and the default database credentials are used.

6.3.3 APIs Configuration

The API key and token for AI integration is kept inside the database itself under a table named api_keys, there is only one key and the key is used to connect the system with AI promt integration via online to help with the OCR.

6.4 System Operations' Screenshots

6.4.1 Application Login Page

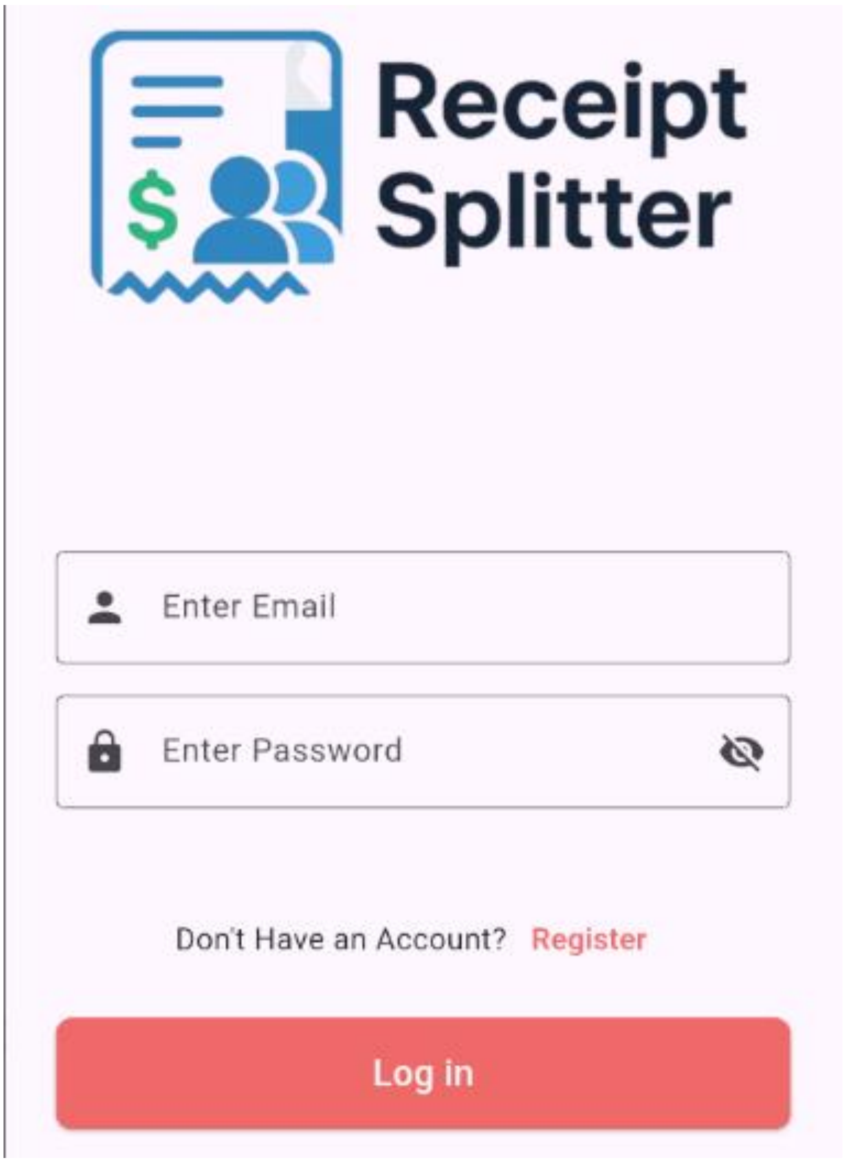


Figure 6.4.1.1: Login Page

User are directed to the Main Page when the application is started. This login page is for admin or user with accepted status to enter the application's main page. Users with account created and already accepted by admin can enter their credential to login, same with admin. If user does not have an account, they can click the hypertext below the page to register an account.

6.4.2 Sign Up page

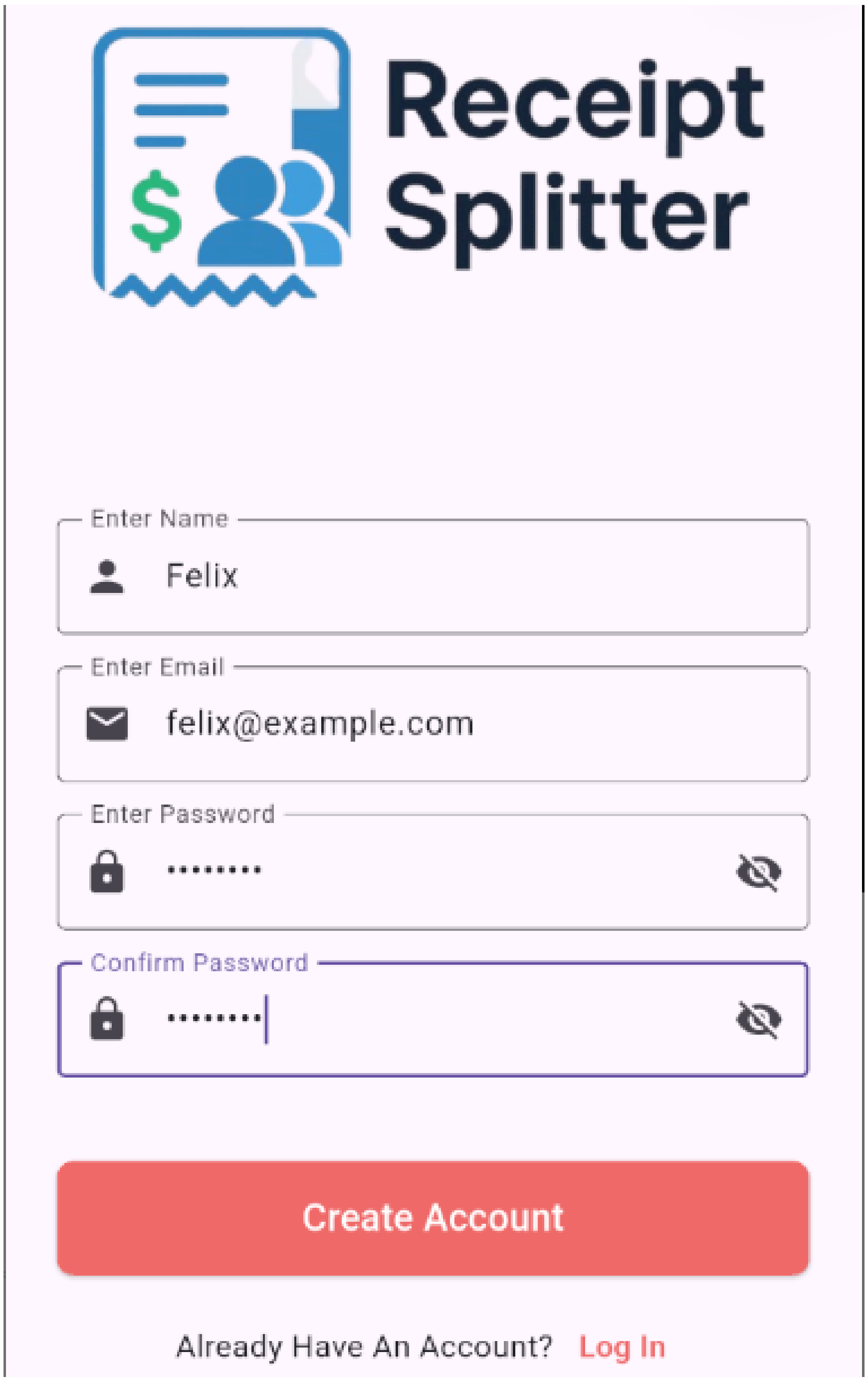
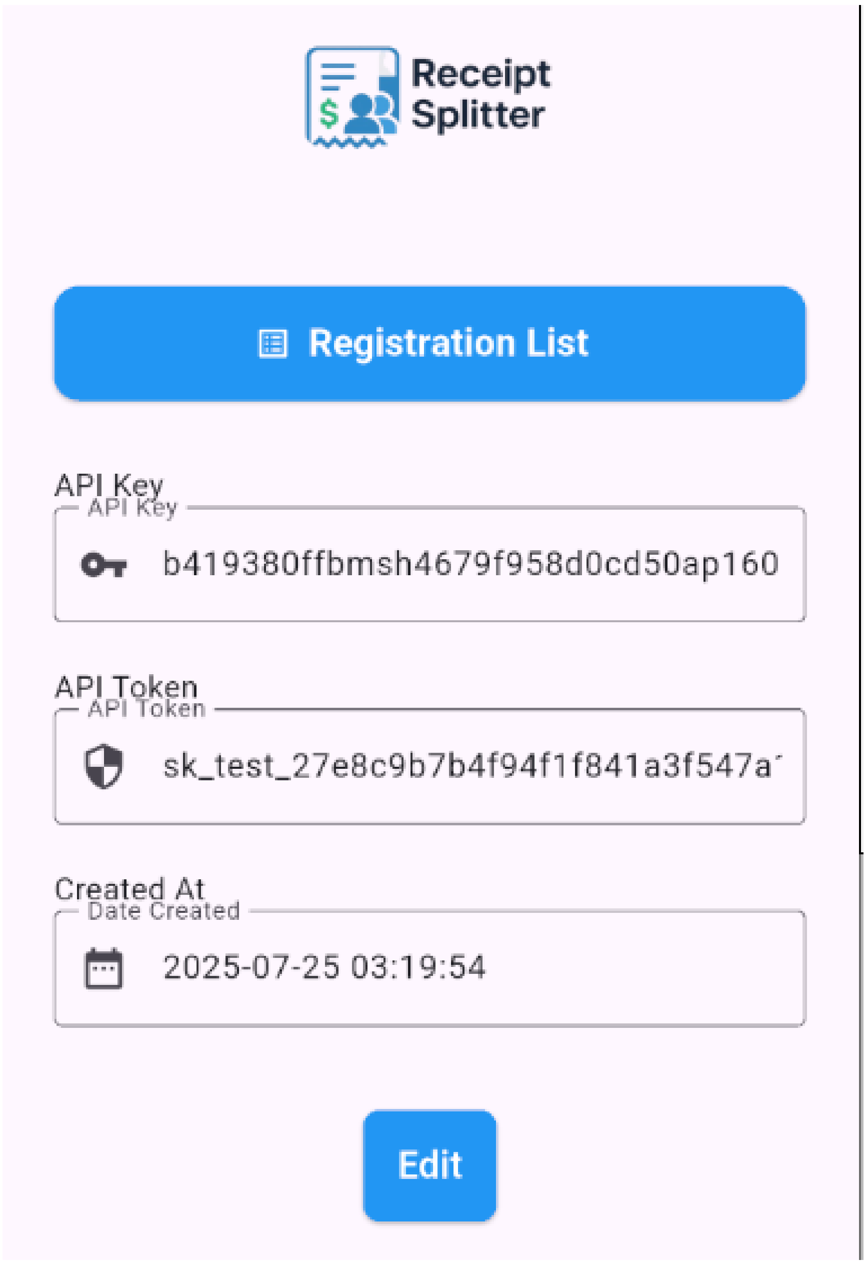


Figure 6.4.2.1: Sign-up page

User can register a new account in the login page, user are required to input their name, email address, and password. The password needs to be typed twice to make sure there is no typo. There are two conditions where the account cannot be created. First condition is same email address detected from database, second is if the password confirmation does not match.

6.4.3 Admin Panel

6.4.3.1 Home Page



API Key
API Key
b419380ffbmsh4679f958d0cd50ap160

API Token
API Token
sk_test_27e8c9b7b4f94f1f841a3f547a

Created At
Date Created
2025-07-25 03:19:54

Figure 6.4.3.1.1: Admin Home Page

Admin home page will be directed when an admin is logged into the application. The most important feature is the API key and token control. Admin are allowed to edit the API key and token if needed, when edit button is pressed, the textfield will be editable for the admin to change. The date will be updated when a new API key or token is updated. There will be a button "Registration List" which will direct to the Registration Acceptance Page as in Figure 6.4.3.2.1.

6.4.3.2 Registration Acceptance Page

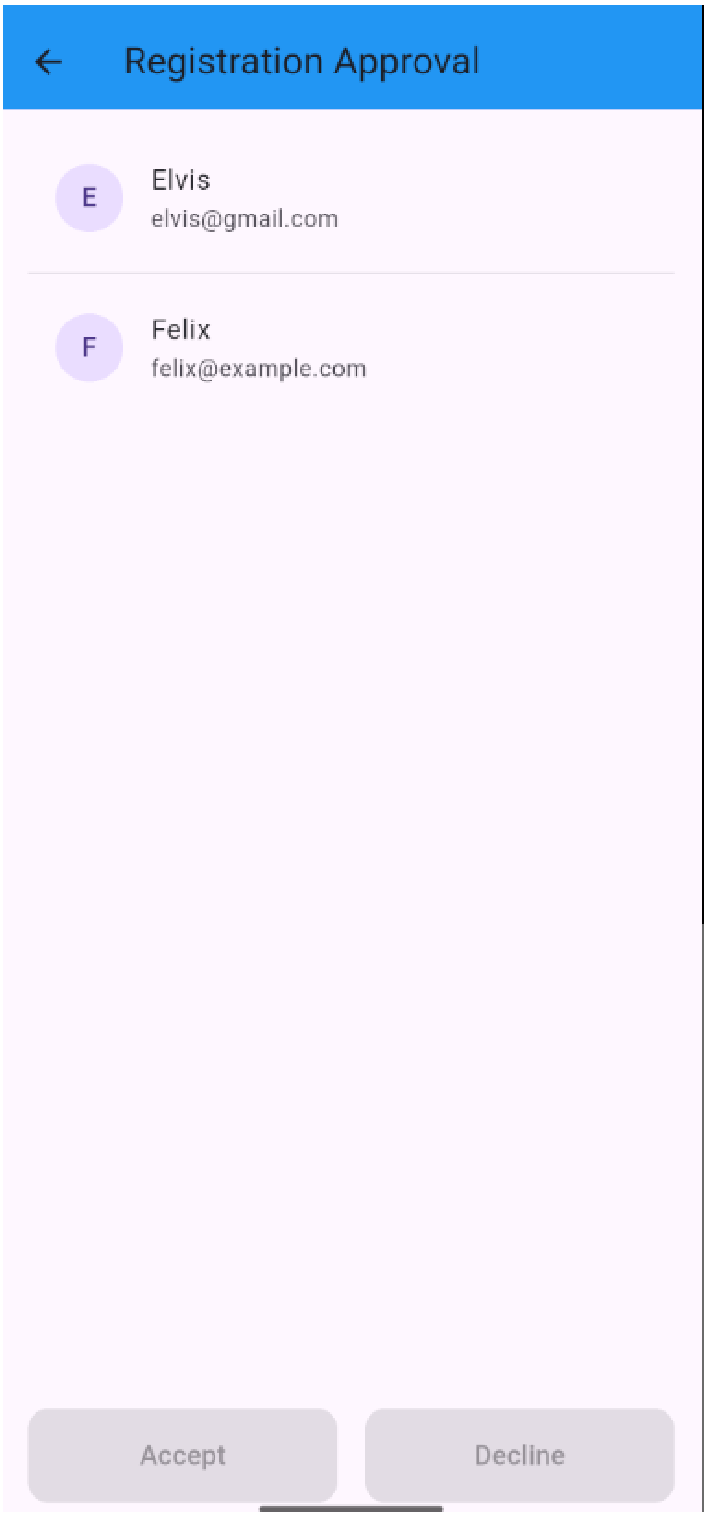
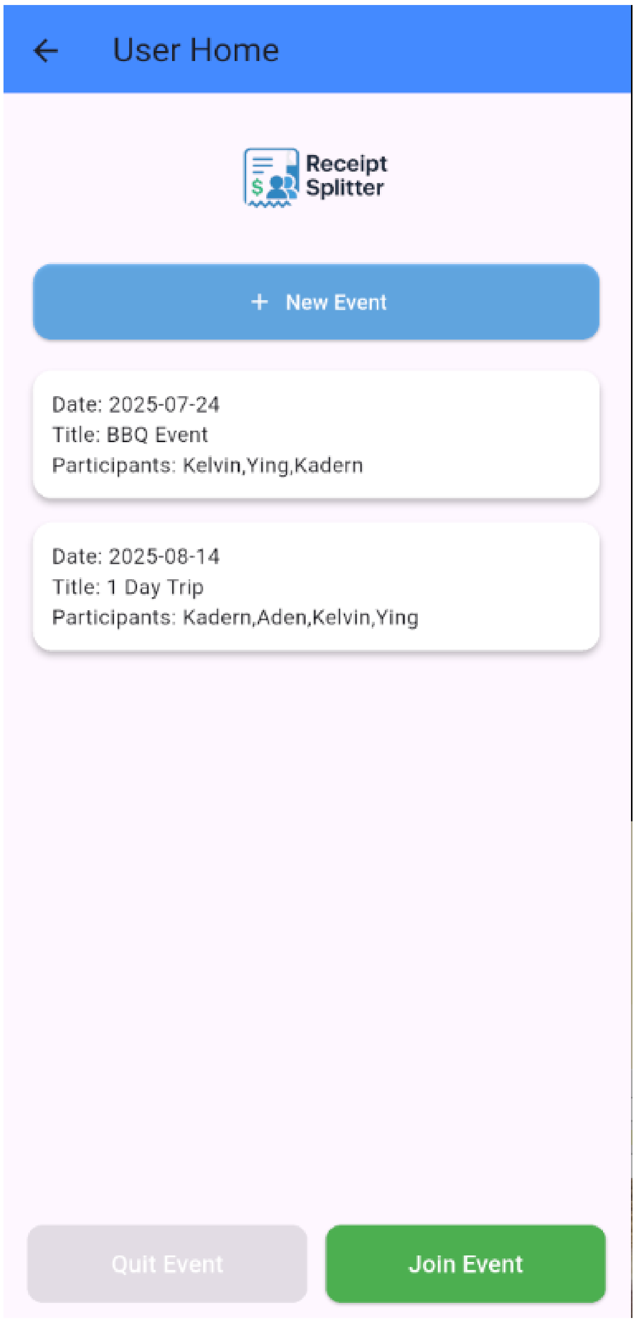


Figure 6.4.3.2.1: Registration Management Page

This page functions for the admin to accept new user's application, user can only login when the admin accepted their profile. Admin can either accept or decline the application. When a user is accepted or declined, the profile will be removed from the list.

6.4.4 User Panel

6.4.4.1 Main Page



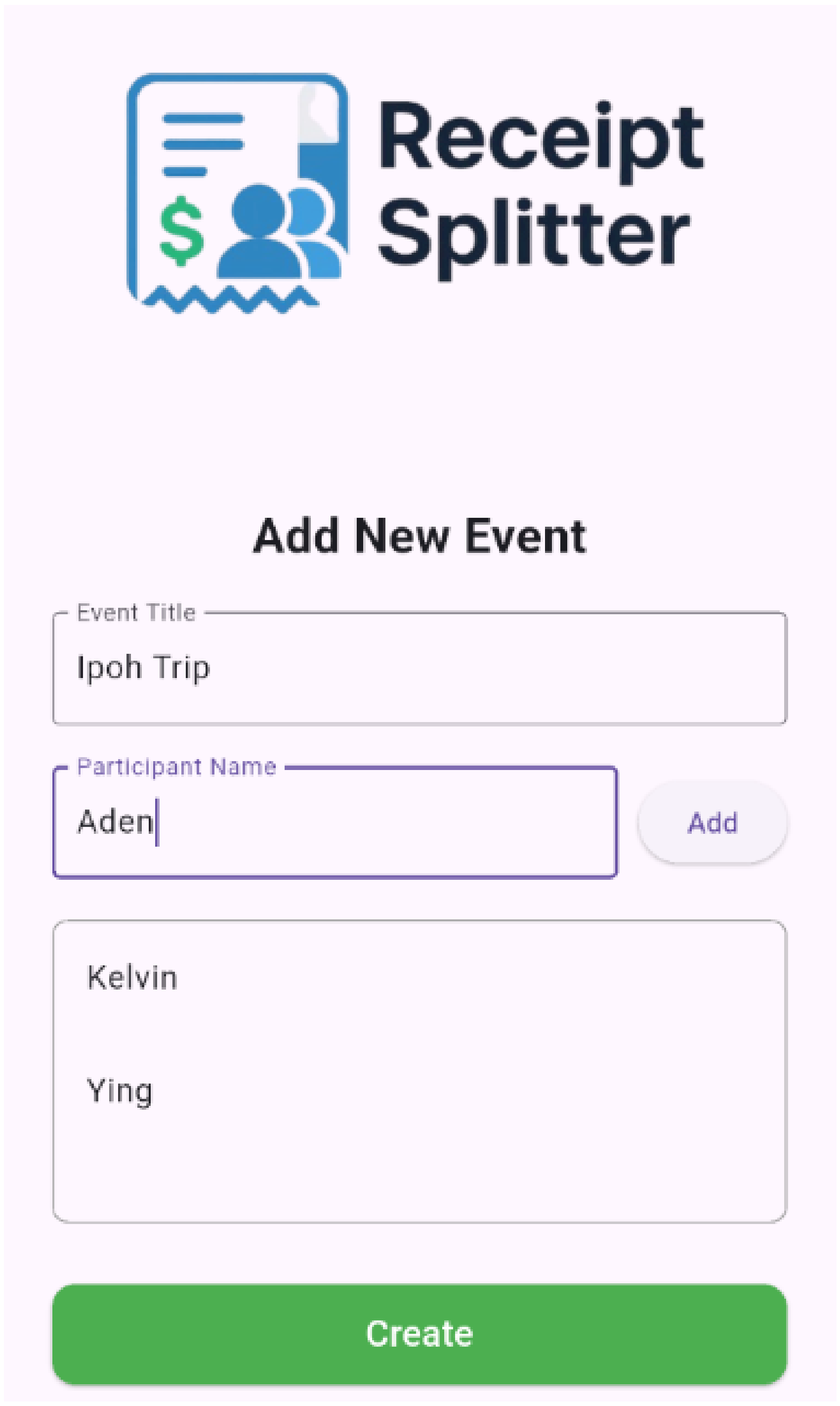
Date: 2025-07-24
Title: BBQ Event
Participants: Kelvin,Ying,Kadern

Date: 2025-08-14
Title: 1 Day Trip
Participants: Kadern,Aden,Kelvin,Ying

Figure 6.4.4.1.1: User Main Page

This page will be directed when an accepted user is logged into the application. It will display all Event that is joined and created by the user. In each event column, it will display the date is was created, the title and the participant that is included in the event. There are multiple function in this page which includes, Creating a new event, viewing an event's detail, quitting an event and also joining an event.

6.4.5 Add New Event



Event Title: Ipoh Trip

Participant Name: Aden

Kelvin

Ying

Figure 6.4.5.1: Add Event Page

The add event page will be displayed after the "+ New Event" is pressed during the main page. The creation will need an event title, if no written any, it will be labelled "Untititled Event". Participants can also be added, participants in the event means the people that is included in the event for bill splitting calculations.

6.4.6 Join Event Page

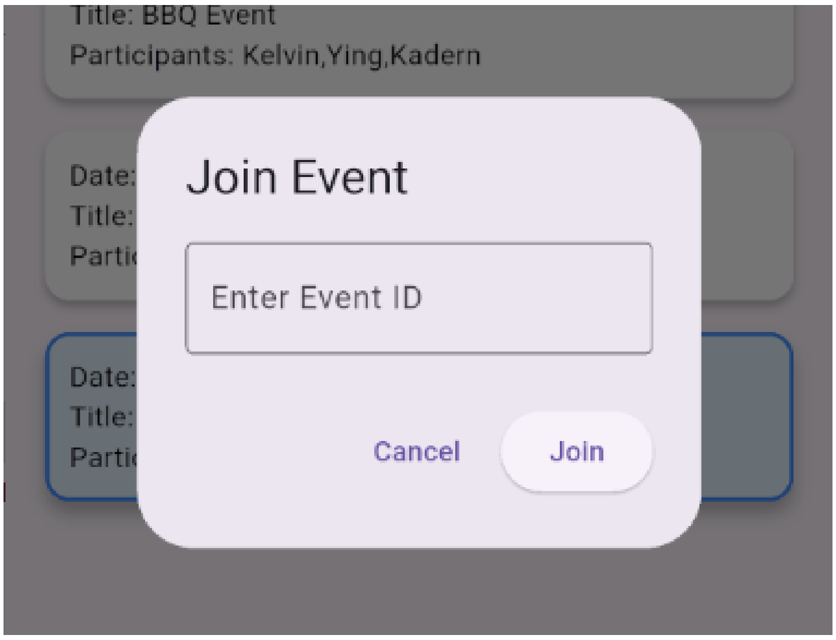


Figure 6.4.6.1: Join Event Container

"Join Event" functions allows users to collaborate together in an event, user can retrieve and enter the event id from the host user to join in the event. By joining in the event, users are now seen as collaborators. They are able to view, add and delete anything related to the event.

6.4.7 Event Detail Page

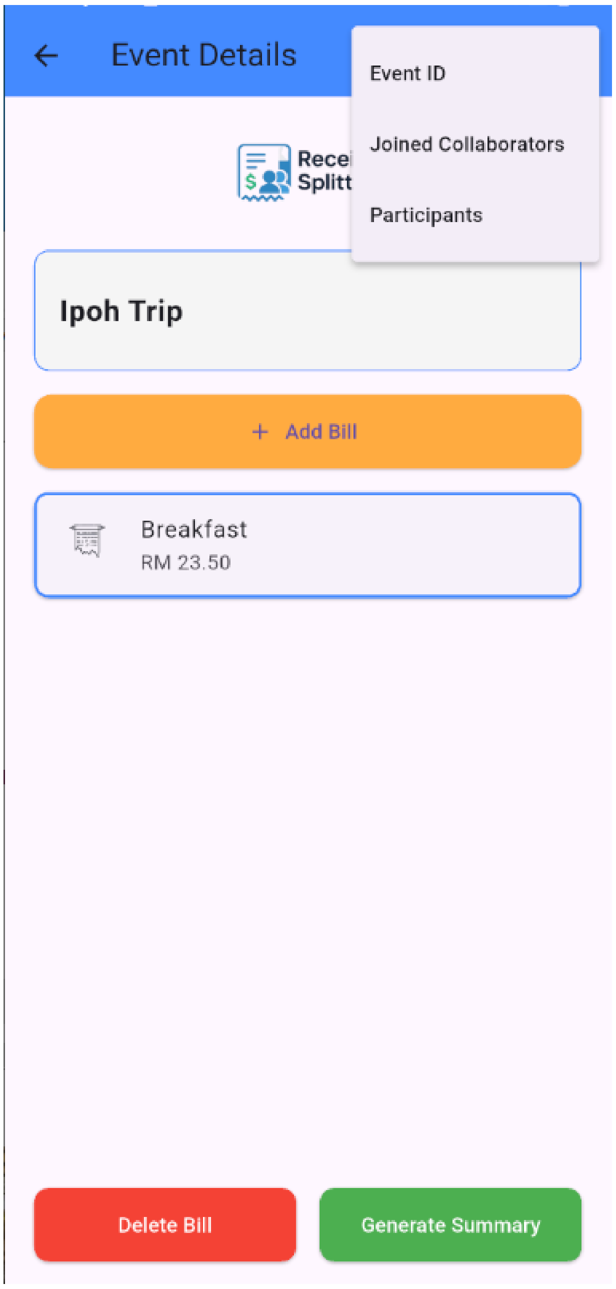


Figure 6.4.7.1: Event Detail Page

This is the page which will display information of an event when entered from the Home Page. On the top right side of the application, a container will appear when a hamburger icon is clicked, the container contains the event's id, the ability to view the collaboratos who joined in the event, and also participants of the event. The page also reveals the event's title which can be easilty edited. Also, users can view, add and delete bills of the event. Finally, on the bottom right corner is a generate summary button which will redirect the user to the generate summary page.

6.4.8 Adding Participants Into Event Container

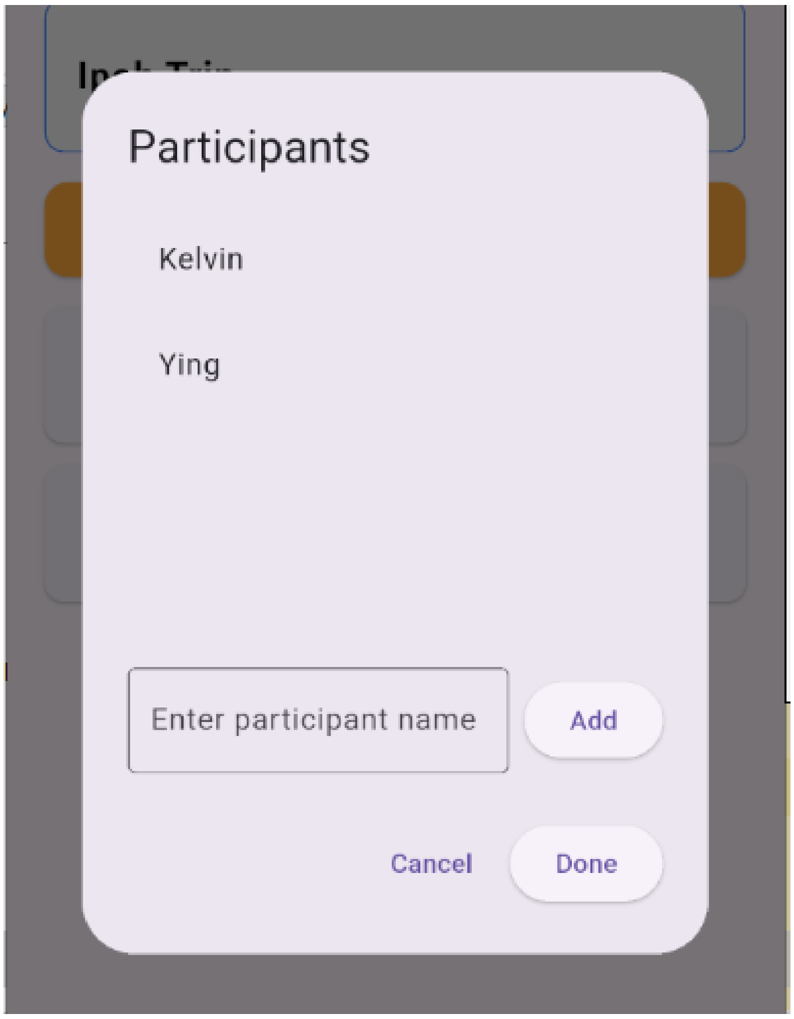


Figure 6.4.8.1: Adding Participants Container

When the "Participants" hypertext is clicked in the hamburger icon at event details page, users can freely add participants into the list as preferred.

6.4.9 Creating A New Bill

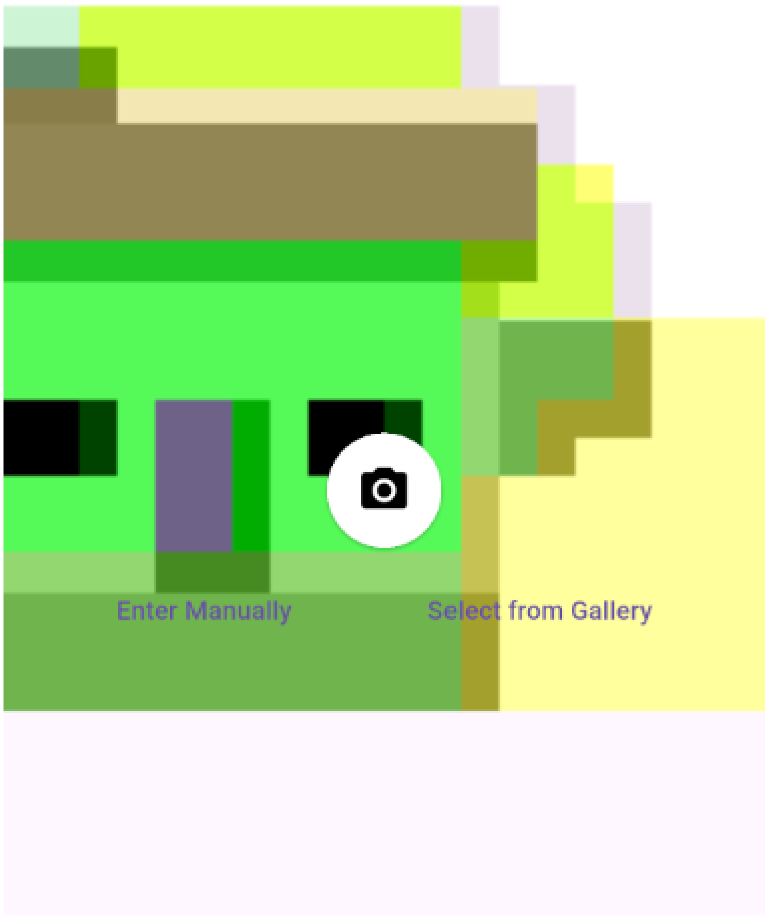
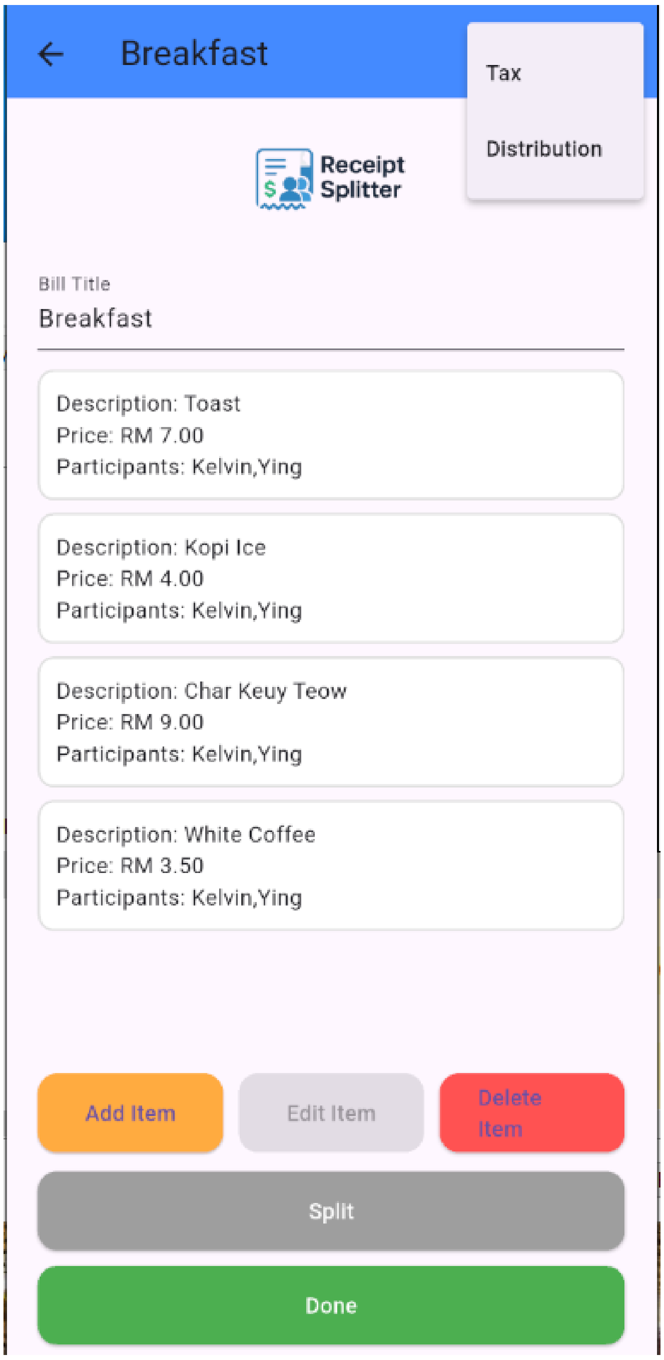


Figure 6.4.9.1: Camera Interface

After "Add Bill" button is pressed in event details page, the user's camera will pop up, it the user is using this function for the first time, the application will ask for the user's permission to use the camera and also access the gallery of the device. User have three choices to create a new bill. They can on-the-spot take a picture of the receipt, select a receipt image from the gallery or create it manually. If user takes a picture or selects from the gallery, the image will uploaded to the database and it will run OCR for the receipt.

6.4.10 Bill Detail Page



Bill Title
Breakfast

Description: Toast
Price: RM 7.00
Participants: Kelvin,Ying

Description: Kopi Ice
Price: RM 4.00
Participants: Kelvin,Ying

Description: Char Keuy Teow
Price: RM 9.00
Participants: Kelvin,Ying

Description: White Coffee
Price: RM 3.50
Participants: Kelvin,Ying

Figure 6.4.10.1: Bill Detail Page

The bill details page will be directed when an existing bill is selected from the event details page or when a new bill is created. If there is existing items from creating a bill, the participants for the items will be empoty by default. Firstly, on the top right corner of the the page, there will be a hamburger icon which will provide some function for the user to do. Users can also view, add, and edit items in the bill if they wish to add.

6.4.11 Add Item Container

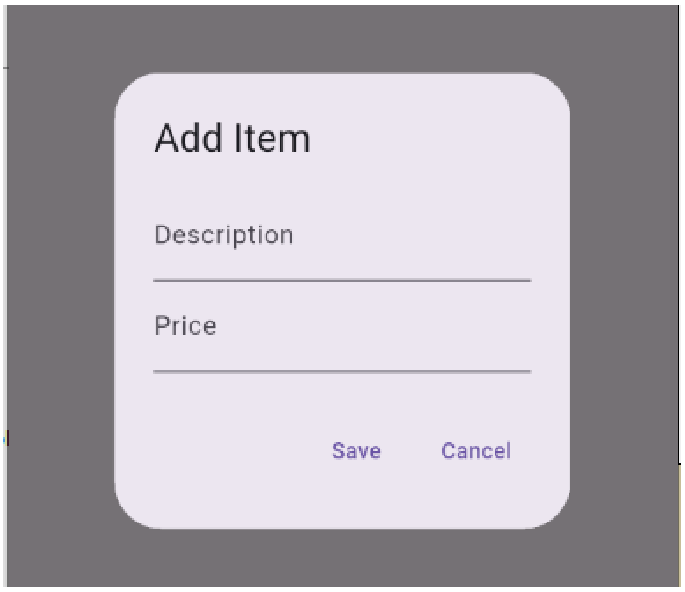


Figure 6.4.11.1: Add Item Container

When the "Add Item" button is pressed at bill details, this container will come up for the user to add an item, the user can enter the item's description and price and the information will be saved to the database when "Save" is pressed.

6.4.12 Edit Item Container

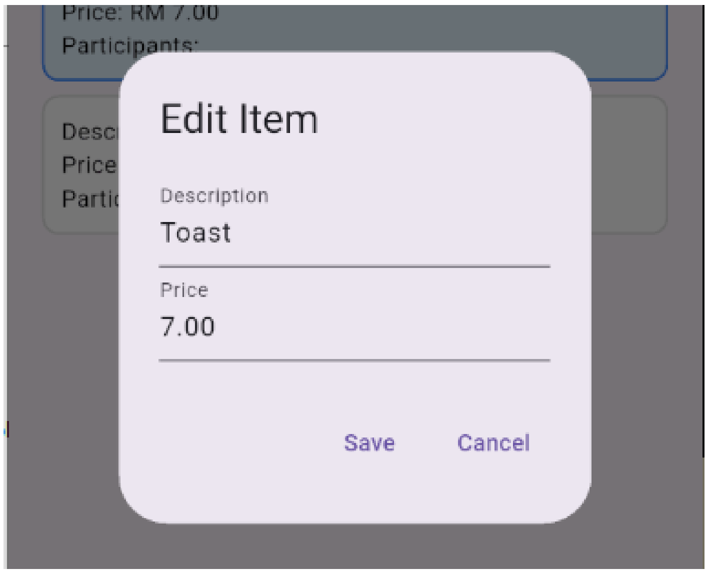


Figure 6.4.12.1: Edit Item Container

When an item is selected and the "Edit Item" button is pressed at bill details, this container will come up for the user to edit an item, the user can edit the item's description or price and the information will be saved to the database when "Save" is pressed.

6.4.13 Assign Items To Participants Container

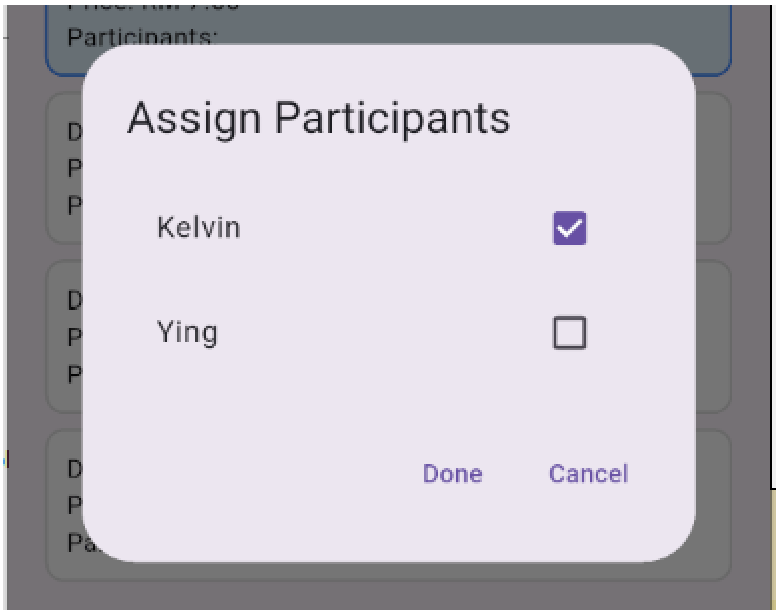


Figure 6.4.13.1: Split Bill Container

When an item is selected and the "Split" Button is pressed in the bill details page, this container will come up for the user to tick. Basically, user can assign the which the item is assigned to, this functions plays a big role to calculate who has what item and will affect the total owed by each participants, the participants that can be selected are from the list of participants from the event details page. The information will be saved when "Done" is pressed.

6.4.14 Tax & Discount Container

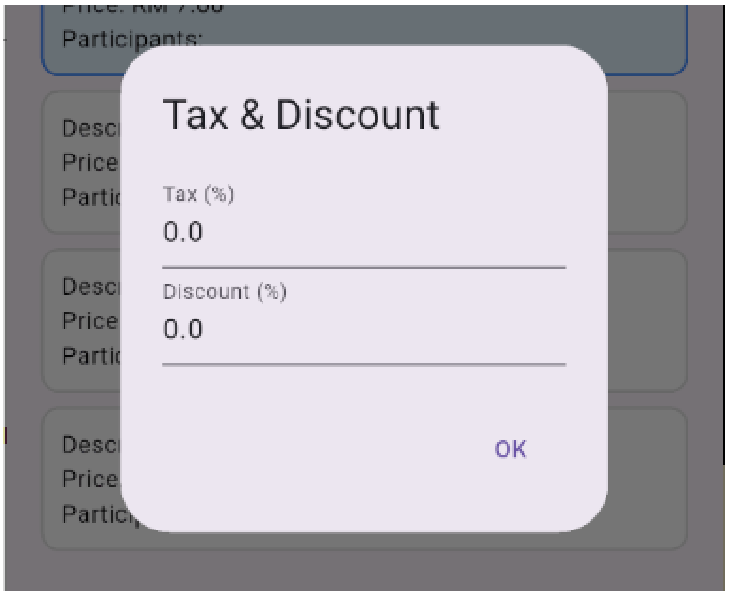
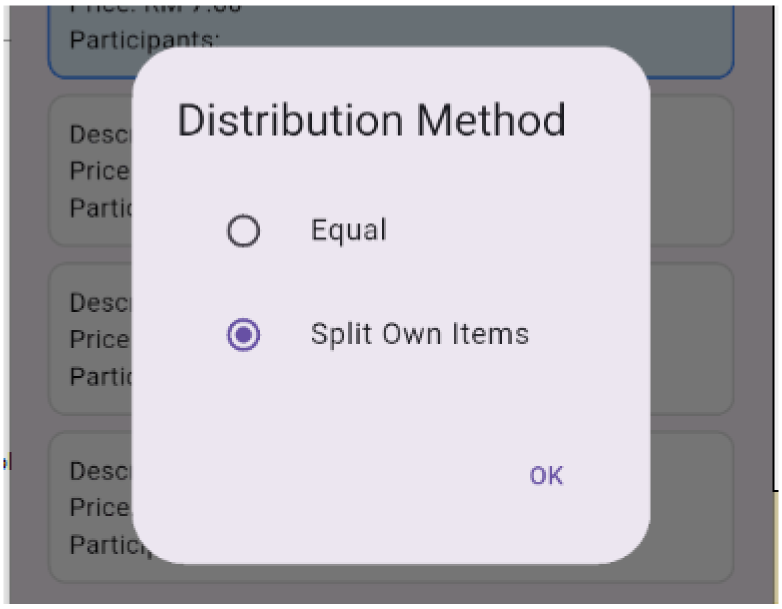


Figure 6.4.14.1: Tax and Discount Container

This container will appear when the user clicks on the "Tax" hypertext in the bill details page's hamburger container. In this container, user can enter the tax and discount if needed given by the receipt to help with the calculation of the bill. User are not required to use this container if the items are already in included with the item's price.

6.4.15 Distribution Types



Distribution Method

( ) Equal

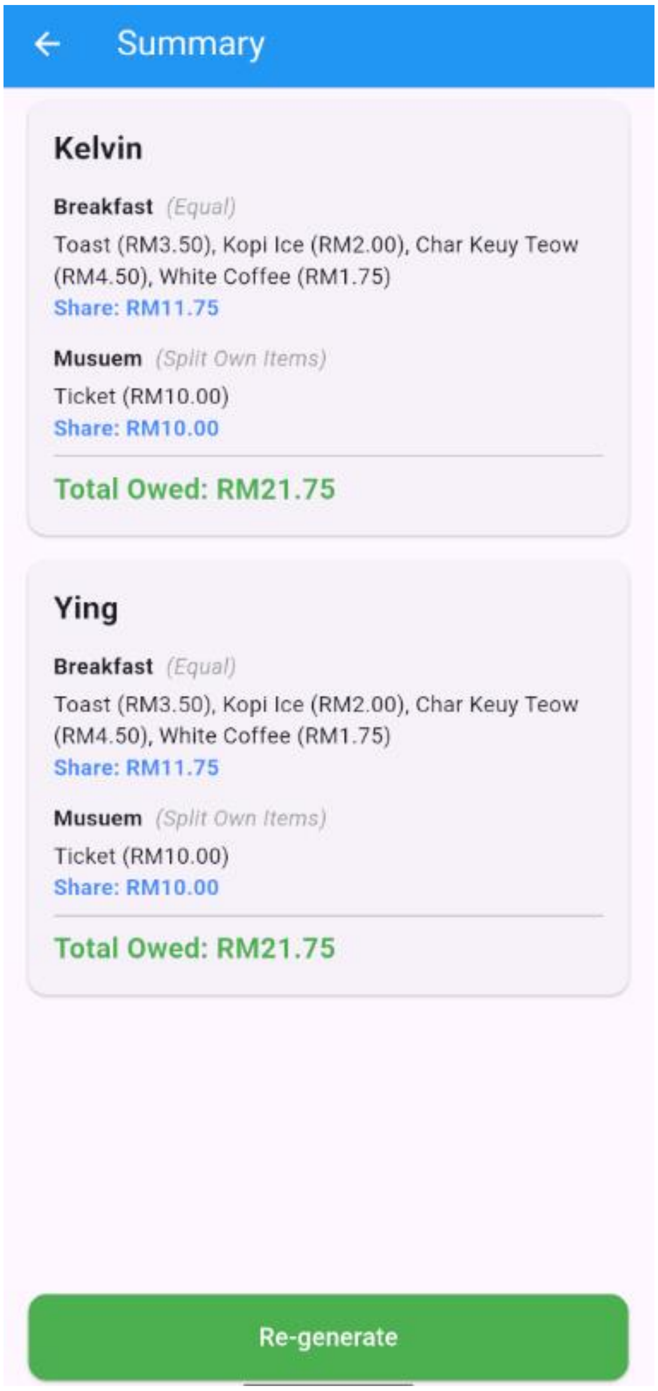
(●) Split Own Items

OK

Figure 6.4.15.1: Edit Distribution Method Container

The distribution method container will appear when the "Distribution Method" hypertext is clicked on the bill details page's hamburger container. In this container, user can pick how the bill should be distributed. By default, the bill will be "Split Own Items" where user can freely assign particpants for each item. If the "Equal" method are selected, All participants are assigned to all items indicating that all the items are fairly distributed to all participants.

6.4.16 Generate Summary Page


Summary

Kelvin

Breakfast (Equal)

Toast (RM3.50), Kopi Ice (RM2.00), Char Keuy Teow (RM4.50), White Coffee (RM1.75)

Share: RM11.75

Musuem (Split Own Items)

Ticket (RM10.00)

Share: RM10.00

Total Owed: RM21.75

Ying

Breakfast (Equal)

Toast (RM3.50), Kopi Ice (RM2.00), Char Keuy Teow (RM4.50), White Coffee (RM1.75)

Share: RM11.75

Musuem (Split Own Items)

Ticket (RM10.00)

Share: RM10.00

Total Owed: RM21.75

Re-generate

Figure 6.4.16.1: Generate Summary Page

The generate summary page is the most important part of the application. This page shows the summary of how much each participants used in the trip. All the calculations are made based on the each items in each bills are distributed to each participants. The calculations are made through a backend complex algorithm. If there any changes in the bills or items, users can press the “Re-generate” button to give a new summary based on the changes.

6.5 Implementation Problems and Challenges

During the development of this project, I faced multiple challenges along the way. The first issue that I would like to highlight is one of the functions in the application which "Split Bill". Before the actual implementation, a user has spoken out about the potential inefficiencies of the function and the changes that can be made to increase usability. During testing, the function infact is time consuming and can be improved. It was straightforward but takes a long time to achieve the result wanted as there was a lot of steps involved. Therefore, a new function was born to improve the previous one making it both straightforward and user friendly. The new function is easy to use and can also help the user paint a better picture of what the functions actually does.

Next, the second challenge that is faced is getting unnecessary information from the receipt during OCR functions. The implementation of OCR is to retrieve information from the receipt such as item's description and price. However, OCR will tend to capture some unnecessary information such as the company's address, the footer and also card information. This has strongly affected the effectiveness of the whole process as the AI that is worked with the OCR will interpret the unnecessary information as item which will need to be deleted by the user. Thus, the functions of taking pictures or uploading from gallery has been improved by allowing the user to crop the pictures before sending it to OCR for scanning. By cropping only the necessary information, it allows the OCR and AI interpretation to be more accurate and only send back accurate information which will overall increase the efficiency of the project.

The last challenge that I will be highlighting is understanding that AI does not work well for every functions. The "Generate Summary" function previously uses AI to help with the calculation to capitalize with the API functions. However, the results that is given back during testing was not consistent with the expected results which. Many attempts of trying to train the prompt were made to achieve the expected result. However, it was still not consistent despite the improvement made. Therefore, the usage of AI for this function was changed to hardcoded complex algorithms in the backend. In effect, it brought highly accurate and consistent results which increased the application's efficiency and usability.

CHAPTER 7
SYSTEM TESTING

7.1 Introduction

This chapter will focus on explaining the testing results for the Bill Splitting Application. The test that is involved are unit testing, integration testing, feature testing, and also black box testing. The reason testing is important is to make sure the app works as intended and the system can meet the requirements.

7.2 Types of Testing

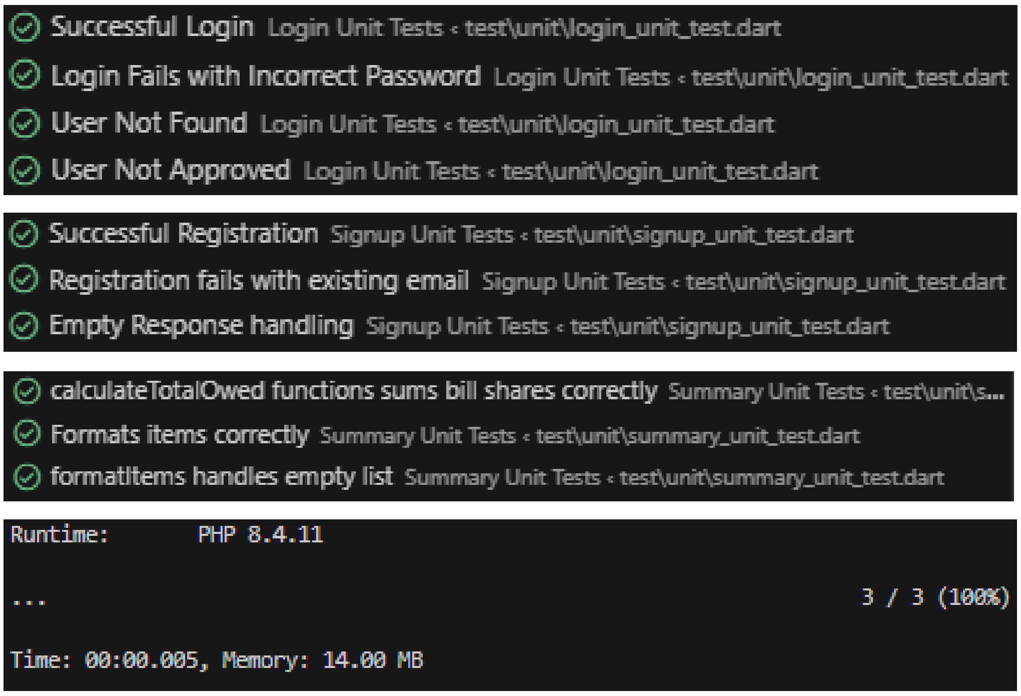
After the project is done developed and ready to use, it undergoes testing in which four types of testing were used in this project. On top of that, manual feature testing is done to ensure the features of the application works as intended by trying a function from start to end. Finally, black-box testing are done to evaluate the overall validation functionality of the application without the need to consider internal code.

7.2.1 Unit Testing

Unit testing is done to validate some small and independent components in the application's logic without the need to go live API and database. Dummy JSON data is used to tests the functions to ensure that the condition handling and validation is working correctly. With unit testing, it is able to detect errors early which can increase the overall reliability of the application. The codes for unit testing is shown at appendix A.

Table 7.2.1.1: Result table of unit testing

Unit Test Module	Total unit test cases in module	Number of passess
LoginTest	4	4
SingupTest	3	3
SummaryTest	3	3
CalculateSummaryTest	3	3
Total	13	13



```
✓ Successful Login  Login Unit Tests ‹ test\unit\login_unit_test.dart
✓ Login Fails with Incorrect Password  Login Unit Tests ‹ test\unit\login_unit_test.dart
✓ User Not Found  Login Unit Tests ‹ test\unit\login_unit_test.dart
✓ User Not Approved  Login Unit Tests ‹ test\unit\login_unit_test.dart

✓ Successful Registration  Signup Unit Tests ‹ test\unit\signup_unit_test.dart
✓ Registration fails with existing email  Signup Unit Tests ‹ test\unit\signup_unit_test.dart
✓ Empty Response handling  Signup Unit Tests ‹ test\unit\signup_unit_test.dart

✓ calculateTotalOwed functions sums bill shares correctly  Summary Unit Tests ‹ test\unit\s...
✓ Formats items correctly  Summary Unit Tests ‹ test\unit\summary_unit_test.dart
✓ formatItems handles empty list  Summary Unit Tests ‹ test\unit\summary_unit_test.dart

Runtime:       PHP 8.4.11

...                                                          3 / 3 (100%)

Time: 00:00.005, Memory: 14.00 MB
```

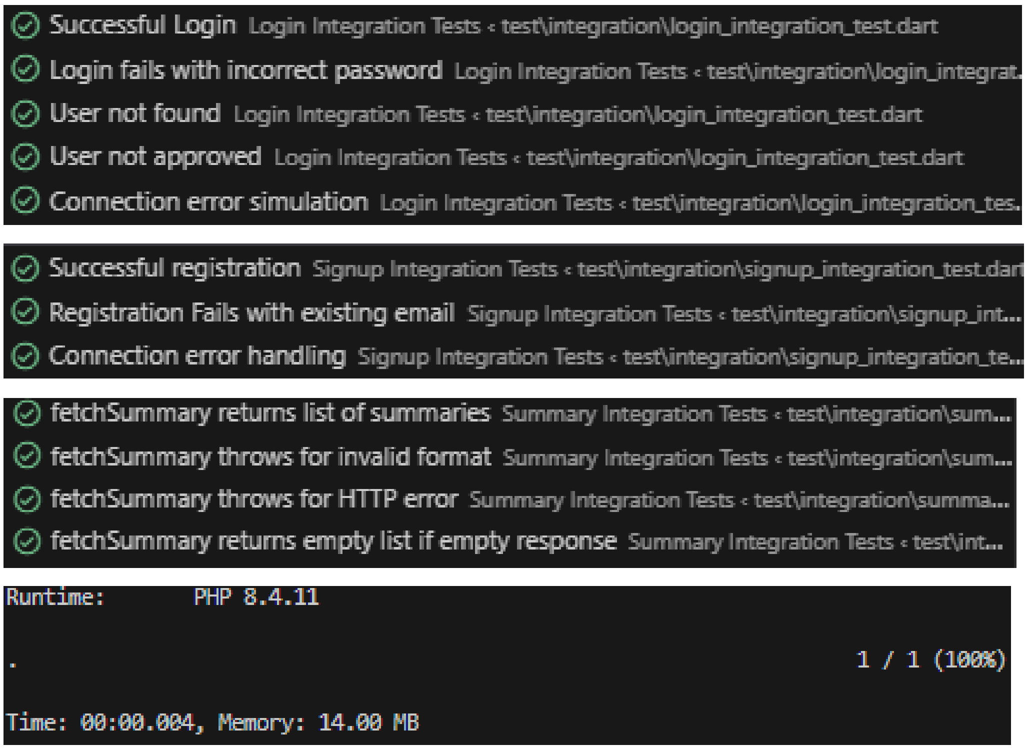
Figure 7.2.1.1: Figure of Unit Testing ran

7.2.2 Integration Testing

Integration testing focus on making sure that different parts of the application are communicated properly, for example, Flutter frontend with the PHP API and database. Functions such as login flow by checking if valid credentials will return correct user data and if the request will show proper error message when given an invalid request. These tests are to prove that pieces of the application can work together smoothly that will significantly reduce the runtime failures when user interact with the systems. The codes for integration testing is shown at appendix B.

Table 7.2.2.1: Result table of integration testing

Integration Test Module	Total unit test cases in module	Number of passess
LoginTest	5	5
SignupTest	3	3
SummaryTest	4	4
SummaryCalculationTest	1	1
Total	13	13



```
✓ Successful Login  Login Integration Tests ‹ test\integration\login_integration_test.dart
✓ Login fails with incorrect password  Login Integration Tests ‹ test\integration\login_integrat
✓ User not found  Login Integration Tests ‹ test\integration\login_integration_test.dart
✓ User not approved  Login Integration Tests ‹ test\integration\login_integration_test.dart
✓ Connection error simulation  Login Integration Tests ‹ test\integration\login_integration_tes.

✓ Successful registration  Signup Integration Tests ‹ test\integration\signup_integration_test.dar
✓ Registration Fails with existing email  Signup Integration Tests ‹ test\integration\signup_int...
✓ Connection error handling  Signup Integration Tests ‹ test\integration\signup_integration_te...

✓ fetchSummary returns list of summaries  Summary Integration Tests ‹ test\integration\sum...
✓ fetchSummary throws for invalid format  Summary Integration Tests ‹ test\integration\sum...
✓ fetchSummary throws for HTTP error  Summary Integration Tests ‹ test\integration\summa...
✓ fetchSummary returns empty list if empty response  Summary Integration Tests ‹ test\int...

Runtime:       PHP 8.4.11

.                                                            1 / 1 (100%)

Time: 00:00.004, Memory: 14.00 MB
```

Figure 7.2.2.1: Test Results for Integration Testing

7.2.3 Feature Testing

Feature testing is this project is done manually to stimulate actual user behaviour and confirm that the main features in the project can match the project's aim and objectives. This includes singing up a new account, logging in, creating new event and splitting the bills. The functions were tested to check if it produced expected outcome such as showing alerts for incorrect login credentials. This testing ensures that the main functionality of the application is usable to end users. A total of 16 testing cases were made for feature testing.

Table 7.2.3.1: Result table of feature testing

Test Case Id	Feature Tested	Test Scenario	Steps	Expected Behaviour	P/F Pass or Fail
FT01	Login#	Login with valid credentials	Enter valid email and password. Press login button	User navigated to user home page	P

FT02	Registration#	Register with unique email	Enter unique email, password Press signup button	New account is created and user navigated to login page	P
FT03	Edit API#	Admin updates API key	Press edit button Enter new key and token Press save button	New API key and token are updated in database and new date is shown	P
FT04	Registration Management#	Admin accepts and declines user application	Select a applicant Press accept/decline	User account's status in database are changes to '1', if decline, delete user's record	P
FT05	Add Event#	User creates new event	Press add event button Enter title and participants Press Save	User directed to event page. Event is stored in database	P
FT06	Quit Event#	User quits event	Select event Press quit button	User's name is removed from the event's collaborators in the database.	P

FT07	Edit Event	User edits existing events	Select event Edit title/participants Press save button	Updated event data is shown and updated in database.	P
FT08	Join Event	User joins existing events	Enter valid event id Press join button	User's name is added into the event's collaborator column	P
FT09	Add Bill via Camera	Users scans receipt with camera	Press add bill button Take picture of receipt Observe	User directed to bill detail page with semi-filled form.	P
FT10	Add Bill via Gallery	User selects receipt from gallery	Press add bill button Select receipt from gallery Observe	User directed to bill detail page with semi-filled form.	P
FT11	Add Bill via Manual Input	User manually adds bills	Press add bill button Observe	User directed to bill detail page with empty form.	P
FT12	Edit Bill	User edits existing bills	Select bill Edit distribution method, tax and discount Press save	Updated bill data is shown and updated in database.	P
FT13	Add Items	User adds item in bill	Press add items Enter description and price	New item is shown on bill screen and	P

			Press Save	saved in database.	
FT14	Edit Items	User edits existing item in bill	Select item Edit description and price Press save	Updated item data is shown and updated in database.	P
FT15	Splitting Items	User splits items among participants	Select item and press split Select participants from list Press save	The participants column in item's table is updated and displayed in bill detail.	P
FT16	Generate Summary	Generate event summary	Press generate summary button	Correct calculation and allocation of event displayed on screen	P

7.2.4 Black Box Testing

Black box testing can evaluate the application's functionality without the need to investigate internal code. To process this testing, the app is interacted like how a typical user will which includes inputting data, submitting forms and reporting the outputs. For example, the login functions are tested with the right and wrong credentials to ensure that the responses aligned with the expected results. This testing helped to ensure the system is efficient enough from a user's perspective. A total of 19 testing cases were made for black box testing.

Table 7.2.4.1: Result table of feature testing

Test Case Id	Feature	Input	Expected Output	Actual Output	P/F Pass or Fail
BB01	Login	Valid email and password	User logs into user home page	Expected Output	P
BB02	Login	Invalid email and password	Error message: “Invalid credentials”	Expected Output	P
BB03	Login	Empty email and password	Error message: “Email and password required”	Expected Output	P
BB04	Register	Valid email, name, and password	New account created and user redirected to login page	Expected Output	P
BB05	Register	Existing email, name and password	Error message: Email already exists	Expected Output	P
BB06	Register	Existing email, name, however, password and confirm password are different.	Error message: Passwords does not match	Expected Output	P
BB07	API Management	New API and Token	API and token edited and stored successfully	Expected Output	P
BB08	Registration Management	Select applicant and accept	Applicant status updated to ‘1’	Expected Output	P

BB09	Registration Management	Select applicant and reject	Applicant ‘s data is removed	Expected Output	P
BB10	Registration Management	No applicant selected and accept.	Error message: No applicant selected	Expected Output	P
BB11	Join Event	Valid event id	User joins into the event’s collaborator	Expected Output	P
BB12	Join Event	Invalid event id	Error message: Event not found	Expected Output	P
BB13	Edit Bill	Equal distribution selected	Bill’s participants include all participants	Expected Output	P
BB14	Edit Bill	Valid float number entered in tax	Tax saved successfully	Expected Output	P
BB15	Edit Bill	Invalid float number entered in tax	Error message: “Invalid tax value”	Expected Output	P
BB16	Edit Bill	Valid float number entered in discount	Discount saved successfully	Expected Output	P
BB17	Edit Bill	Invalid float number entered in discount	Error message: “Invalid discount value”	Expected Output	P
BB18	Add Item	Description and valid float number entered	Item added to bill with correct amount	Expected Output	P
BB19	Add Item	Description and invalid float number entered	Error message: “Invalid price”	Expected Output	P

CHAPTER 8
CONCLUSION AND RECOMMENDATION

8.1 Conclusion

In conclusions, all the objective that is specified in Chapter 1 Section 1.3 has been successfully achieved. The objectives that were achieved are as follows:

1. To employ camera function for Optical Character Recognition to avoid unessacary manually key in and increase efficiency.
2. To incorporate multiple type of calculation option for bill splitting to ensure fairness and user control.
3. To integrate Artificial Intelligence into the application for increased accurate and consistent receipt reading.
4. To facilitate the option to invite other participants to join with the calculation for efficient collaboration.

The first objective is to employ camera function to enable the use of Optical Character Recognition (OCR), was achieved by enabling the application to take picture of the receipt directly through the device's camera. In results, this eliminates the need for users to key in the data manually which can reduce human error as well as the application's overall efficiency. The scanned data are stored in the system's database and sent over for AI to interpret the data onto useful information.

Next, the second objective is to incorporate multiple types of calculation option for bill splitting which includes custom distribution, allowing flexibility in how the bill is shared. User can choose whether the bill is splitted fairly between each participants or splitted equally between everyone. This objective gives users full control over how the bill is splitted which enhances usability.

The third objective explains to integrate Artificial Intelligence (AI) into the application for accurate and consistent receipt data interpretation, was achieved through the implementation of AI-enhanced OCR in the application. Through this, the receipt data was extracted and interpreted accurately ensuring that item's description, and price are captured more reliably. By reducing the errors at data extraction. The AI integration improved the system's overall consistency and user are given a more trustworthy result.

Finally, the fourth objective is to provide the option to invite other participants to join in the calculation process, was accomplished by implementing a collaboration feature in the application. This allows other users to input an event id to join in the event's calculation, where each collaborators can edit the bills and items in the event. This feature can ensure a smoother and efficient group experience and make the application practical for real-world social event experience.

8.2 Limitation

Here are some of the limitations for the developed application

1. Restricted OCR Language Interpretation. The OCR function is limited to interpreting the receipts written in English only. Multilingual receipt will return inaccurate results for the OCR. This reduces the application's usability in multilingual contexts.
2. No Offline Functionality. Because of the usage of AI in the application, it requires and active Wi-Fi connection to fully optimize the feature. Users without internet access are not able to use the application which can reduce the usability in certain conditions such as no internet connection at outstation or oversea.
3. Limited Currency Support. The system OCR only supports Malaysian Ringgit (MYR). Any receipt that is printed out in a different currency will not be interpreted accurately by the AI. This reduces the application's felixbility, especially for frequent travelers.
4. Inaccuracy of OCR and AI. Error may still occurs to the receipt data interpretation during OCR and AI, this may due to blurry receipts, or receipts that are purely in poor condition which will affect the accuracy of the bill-splitting process.
5. Device Compatibility. Every user may have different device performance depending on the user's device specification. For example, having a device with bad camera quality will affect the end result of the OCR.

8.3 Recomendation

Based on the limitations on Section 8.2, here are the recommendations and enhancements that can be done for future work

1. Integrate Multi-Language OCR and Automatic Language Detection. With the implementation of multi-language OCR and auto language detection, the application will be able to process receipts no matter the language which will directly increase the usability in multilingual contexts.
2. Offline Mode Function. Offline Mode in this application can be made where the mode only allows manual input and will store the data locally first. Once internet connection is detected, the new data can be updated with the database to ensure synchronization, this way, users can still use the app in areas without the need of Wi-Fi access.
3. Auto Currency Detection and Conversion. By implementing automatic currency detection and conversion features, the application can convert the item's price into local currency using regularly updated exchange rate. With this feature done, it will be more efficient and flexible for regular travelers.
4. Confidence Scoring. This feature is where a confidence score from low to high are highlighted for the picture taken by the user, a low score indicate that the quality of the picture is considered low quality and may fetch inaccurate information, vice versa. This feature can also guide user on how to capture clearer receipt images.
5. Device Compatibility and Camera Quality Detection. The application can detect the user device's specification and camera quality and give alternatives such as manual input if limitations are found. The application can also provide tips for capturing a clearer image and developers should clearly state the minimum recommended device specifications before installation.

REFERENCES

Alsaqqa, S., Sawalha, S. and Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies*, 14(11), pp.246–270. doi:https://doi.org/10.3991/ijim.v14i11.13269.

Chandran, K. and Das Aundhe, M. (2021). Agile or waterfall development: The Clementon Company dilemma. *Journal of Information Technology Teaching Cases*, 12(1), p.204388691987054. doi:https://doi.org/10.1177/2043886919870544.

Gneezy, U., Haruvy, E. and Yafe, H. (2004). The Inefficiency of Splitting the Bill. *The Economic Journal*, 114(495), pp.265–280. doi:https://doi.org/10.1111/j.1468-0297.2004.00209.x.

Hummert, C. and Pawlaszczyk, D. (2022). *Mobile forensics -- the file format handbook : common file formats and file systems used in mobile devices*. Cham, Switzerland: Springer, pp.129–155.

Imran, M. and Almusharraf, N. (2024). Google Gemini as a next generation AI educational tool: a review of emerging educational technology. *Smart learning environments*, 11(1). doi:https://doi.org/10.1186/s40561-024-00310-z.

Ismail, M.A., Mangshor, N.N.A., Fadzil, A.F.A. and Ibrahim, S. (2024). Automated Receipt Scanning Using Convolutional Recurrent Neural Network (CRNN). *2024 IEEE 22nd Student Conference on Research and Development (SCOReD)*, pp.494–497. doi:https://doi.org/10.1109/scored64708.2024.10872626.

Mahajan, S. and Rani, R. (2024). *International Journal on Document Analysis and Recognition (IDJAR)*. IDJAR, pp.97–119.

Makris, A., Tserpes, K., Spiliopoulos, G., Zissis, D. and Anagnostopoulos, D. (2020). Correction to: MongoDB Vs PostgreSQL: a comparative study on performance aspects. *GeoInformatica*, 25. doi:https://doi.org/10.1007/s10707-020-00424-9.

Neng, H.Z. (2022). *Automated Scanned Receipt Processing with Optical Character Recognition and Machine Learning*.

Nixon, R., (2021). *Learning PHP, MySQL & JavaScript: a step-by-step guide to creating dynamic websites*. " O'Reilly Media, Inc.".

None Mikita Piastou (2023). Comprehensive Performance and Scalability Assessment of Front-End Frameworks: React, Angular, and Vue.js. *World Journal of Advanced Engineering Technology and Sciences*, 9(2), pp.366–376. doi:https://doi.org/10.30574/wjaets.2023.9.2.0153.

Ogochukwu, O. and Amaechi (2024). Enhanced Text Recognition in Images Using Tesseract OCR within the Laravel Framework. *Asian Journal of Research in Computer Science*, 17(9), pp.58–69. doi:https://doi.org/10.9734/ajrcos/2024/v17i9499.

Ong, E. (2016). *Making Splitting Bills Less Painful - Emily Ong - Medium*. Medium. Available at: https://medium.com/@uxlime/making-splitting-bills-less-painful-c22b16f00440#3963 [Accessed 13 Mar. 2025].

Peng, Y., Chen, Q. and Shih, G. (2025). DeepSeek is open-access and the next AI disrupter for radiology. *Radiology Advances*, 2(1). doi:https://doi.org/10.1093/radadv/umaf009.

Porter, L. and Zingaro, D., (2024). *Learn AI-Assisted Python Programming: With Github Copilot and ChatGPT*. Simon and Schuster.
Pradana, M.K., Andrianto, A. and Auliya, Y.A. (2022). Pengembangan Sistem Informasi Desa Terpadu Menggunakan Metode Rapid Application

Development (RAD) Studi Kasus Desa Arjasa. *INFORMAL: Informatics Journal*, 7(2), p.64. doi:https://doi.org/10.19184/isj.v7i2.25238.

Pratiwi, M., Mayola, L., Kris Hiburan Laoli, V., Ilhami Arsyah, U. and Pratiwi, N. (2022). Medical Record Information System with Rapid Application Development (RAD) Method. *Journal of Information Systems and Technology Research*, 1(2), pp.124–130. doi:https://doi.org/10.55537/jistr.v1i2.170.

Senarath, U.S., (2021). Waterfall methodology, prototyping and agile development. *Tech. Rep.*, pp.1-16.

Shevtsiv, N.A. and Striuk, A.M. (2021). Cross platform development vs native development. doi:https://doi.org/10.31812/123456789/4428.

Singh, S.K., Kumar, S. and Pawan Singh Mehra (2023). Chat GPT & Google Bard AI: A Review. *2023 International Conference on IoT, Communication and Automation Technology (ICICAT)*. doi:https://doi.org/10.1109/icicat57735.2023.10263706.

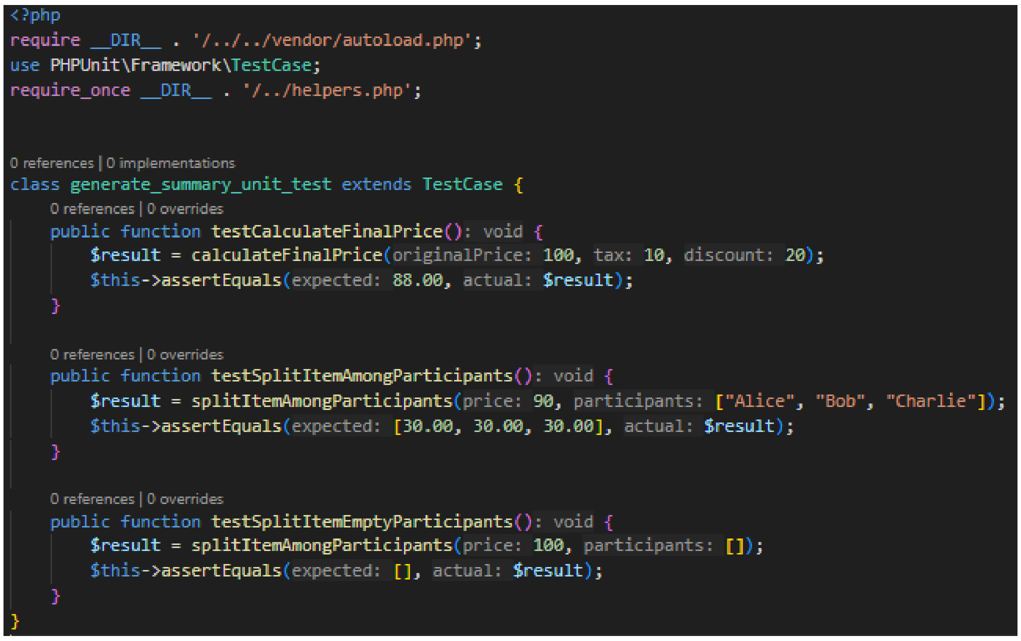
Sufyan Bin Uzayr (2022). *Mastering NativeScript*. Mastering Computer Science.
Taylor, M. (2018). *Solving the problem of splitting the bill... in an afternoon*. Medium. Available at: https://medium.com/@maxjct/solving-the-problem-of-splitting-the-bill-in-an-afternoon-7bdd3fce3211 [Accessed 13 Mar. 2025].

Vara, T. and Rakshitha, N.L. (2024). The Power of React JS for Business Applications. *Deleted Journal*, 2(05), pp.1637–1639. doi:https://doi.org/10.47392/irjaem.2024.0229.

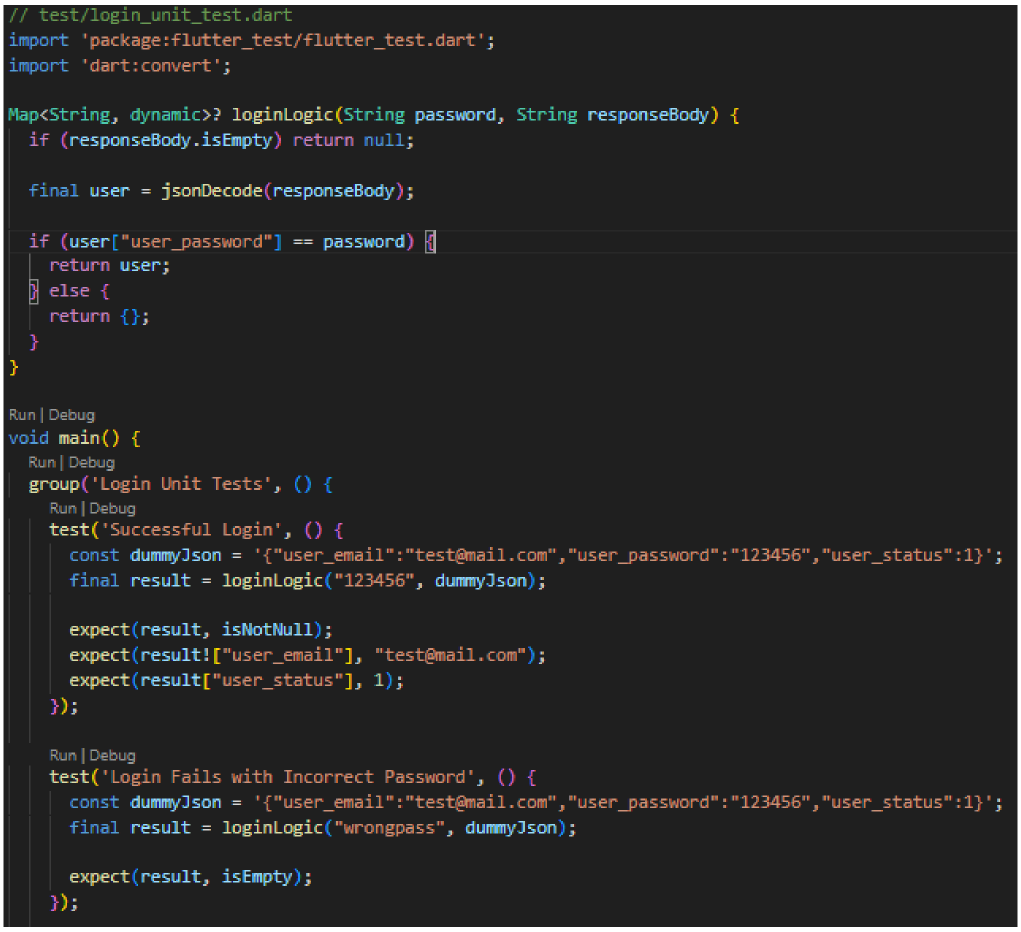
Vyas, R. (2022). Comparative Analysis on Front-End Frameworks for Web Applications. *International Journal for Research in Applied Science and Engineering Technology*, 10(7), pp.298–307. doi:https://doi.org/10.22214/ijraset.2022.45260.

APPENDIX

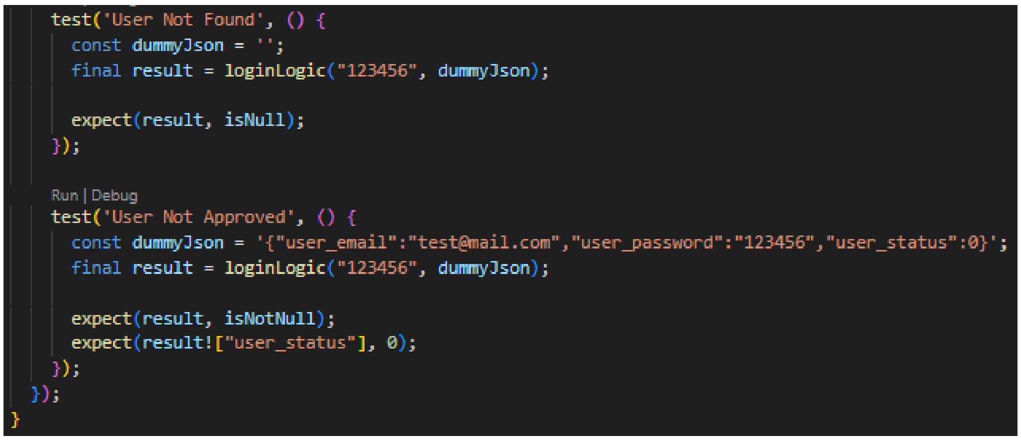
Appendix A



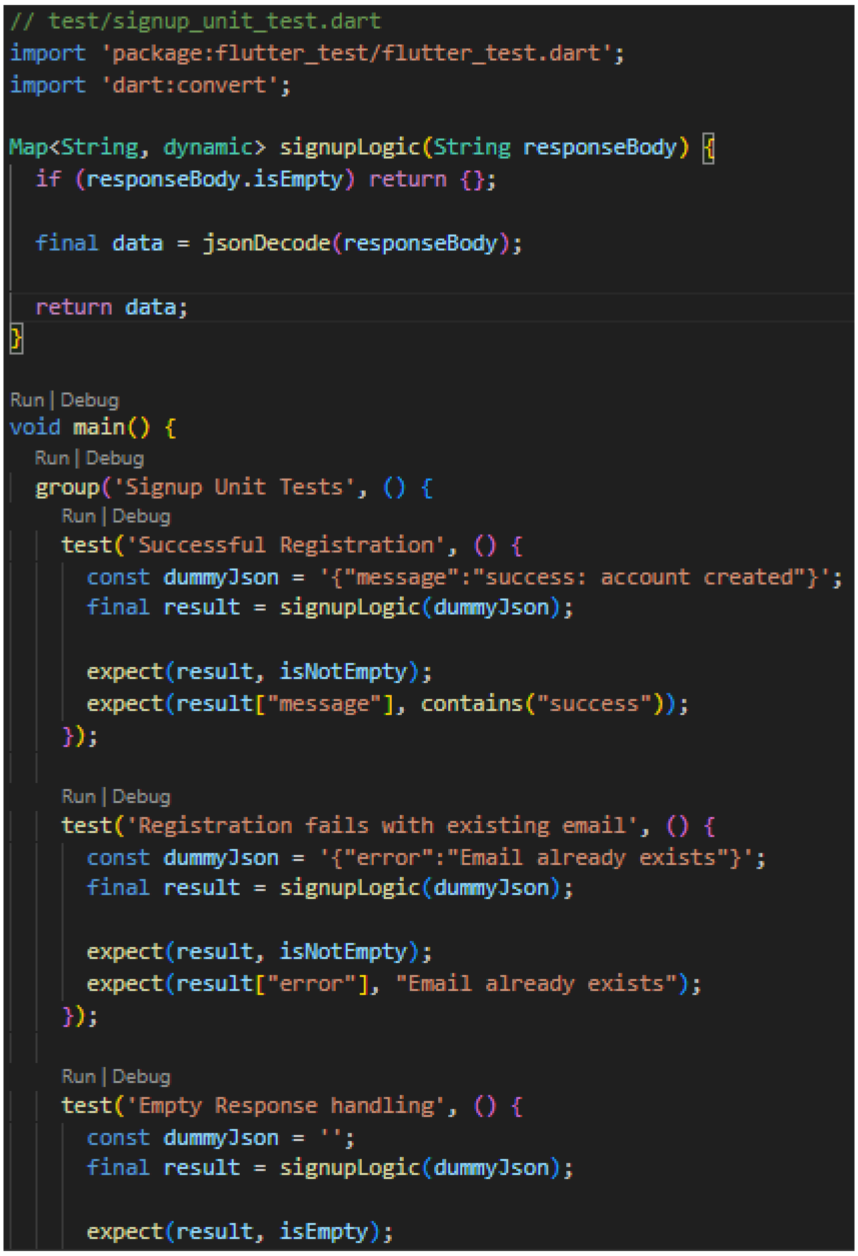
```php
<?php
require __DIR__ . '/../../vendor/autoload.php';
use PHPUnit\Framework\TestCase;
require_once __DIR__ . '/../helpers.php';


0 references | 0 implementations
class generate_summary_unit_test extends TestCase {
    0 references | 0 overrides
    public function testCalculateFinalPrice(): void {
        $result = calculateFinalPrice(originalPrice: 100, tax: 10, discount: 20);
        $this->assertEquals(expected: 88.00, actual: $result);
    }

    0 references | 0 overrides
    public function testSplitItemAmongParticipants(): void {
        $result = splitItemAmongParticipants(price: 90, participants: ["Alice", "Bob", "Charlie"]);
        $this->assertEquals(expected: [30.00, 30.00, 30.00], actual: $result);
    }

    0 references | 0 overrides
    public function testSplitItemEmptyParticipants(): void {
        $result = splitItemAmongParticipants(price: 100, participants: []);
        $this->assertEquals(expected: [], actual: $result);
    }
}
```



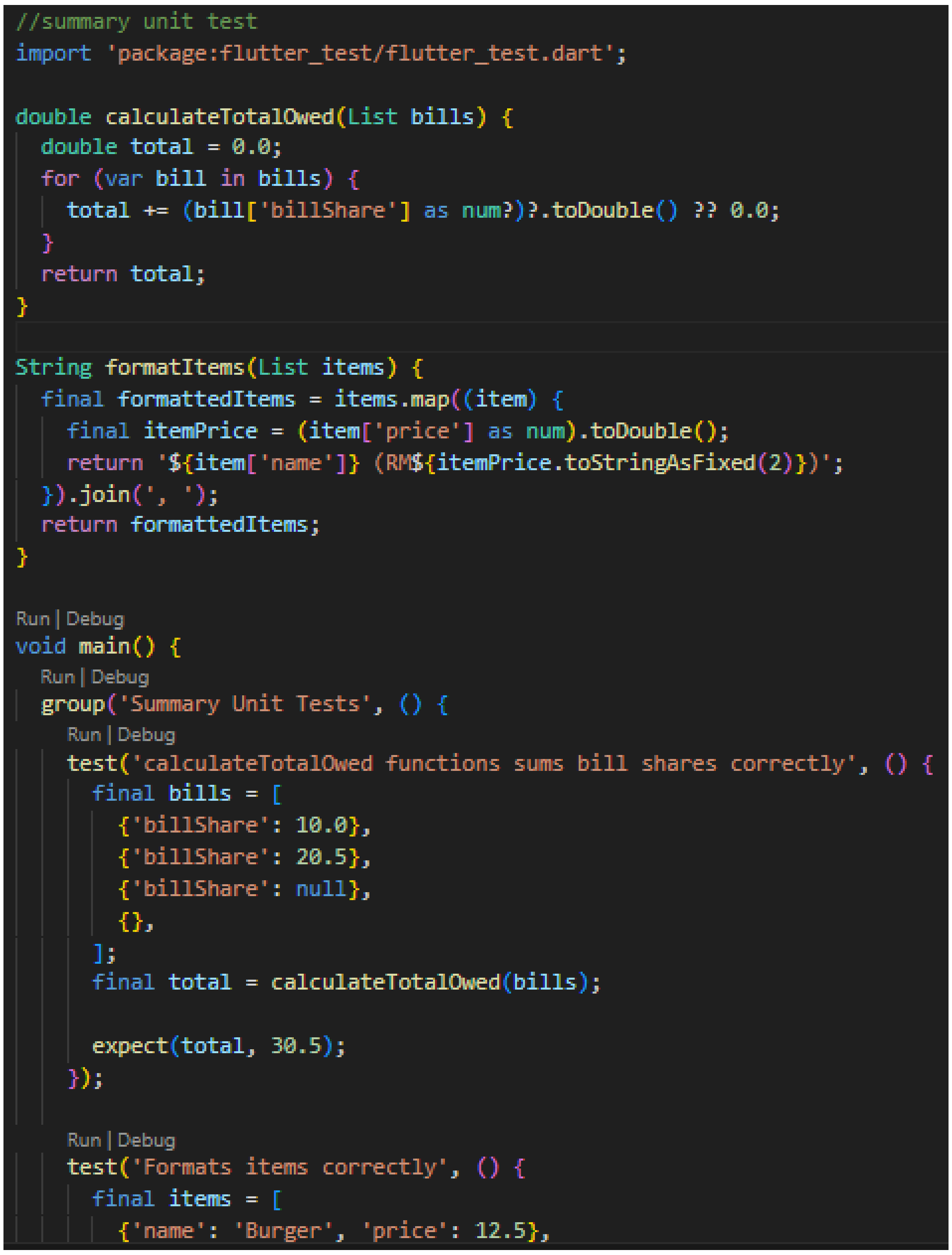
```dart
// test/login_unit_test.dart
import 'package:flutter_test/flutter_test.dart';
import 'dart:convert';

Map<String, dynamic>? loginLogic(String password, String responseBody) {
  if (responseBody.isEmpty) return null;

  final user = jsonDecode(responseBody);

  if (user["user_password"] == password) {
    return user;
  } else {
    return {};
  }
}

Run | Debug
void main() {
  Run | Debug
  group('Login Unit Tests', () {
    Run | Debug
    test('Successful Login', () {
      const dummyJson = '{"user_email":"test@mail.com","user_password":"123456","user_status":1}';
      final result = loginLogic("123456", dummyJson);

      expect(result, isNotNull);
      expect(result!["user_email"], "test@mail.com");
      expect(result["user_status"], 1);
    });

    Run | Debug
    test('Login Fails with Incorrect Password', () {
      const dummyJson = '{"user_email":"test@mail.com","user_password":"123456","user_status":1}';
      final result = loginLogic("wrongpass", dummyJson);

      expect(result, isEmpty);
    });
```



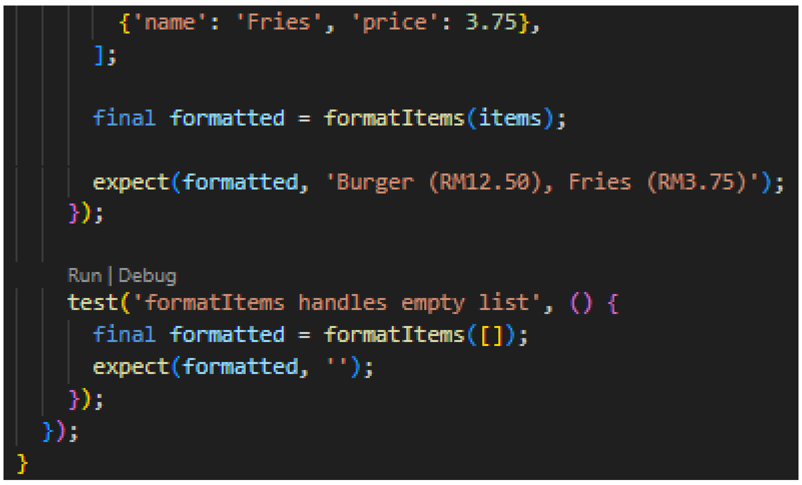
```dart
    test('User Not Found', () {
      const dummyJson = '';
      final result = loginLogic("123456", dummyJson);

      expect(result, isNull);
    });

    Run | Debug
    test('User Not Approved', () {
      const dummyJson = '{"user_email":"test@mail.com","user_password":"123456","user_status":0}';
      final result = loginLogic("123456", dummyJson);

      expect(result, isNotNull);
      expect(result!["user_status"], 0);
    });
  });
}
```



```dart
// test/signup_unit_test.dart
import 'package:flutter_test/flutter_test.dart';
import 'dart:convert';

Map<String, dynamic> signupLogic(String responseBody) {
  if (responseBody.isEmpty) return {};

  final data = jsonDecode(responseBody);

  return data;
}

Run | Debug
void main() {
  Run | Debug
  group('Signup Unit Tests', () {
    Run | Debug
    test('Successful Registration', () {
      const dummyJson = '{"message":"success: account created"}';
      final result = signupLogic(dummyJson);

      expect(result, isNotEmpty);
      expect(result["message"], contains("success"));
    });

    Run | Debug
    test('Registration fails with existing email', () {
      const dummyJson = '{"error":"Email already exists"}';
      final result = signupLogic(dummyJson);

      expect(result, isNotEmpty);
      expect(result["error"], "Email already exists");
    });

    Run | Debug
    test('Empty Response handling', () {
      const dummyJson = '';
      final result = signupLogic(dummyJson);

      expect(result, isEmpty);
```

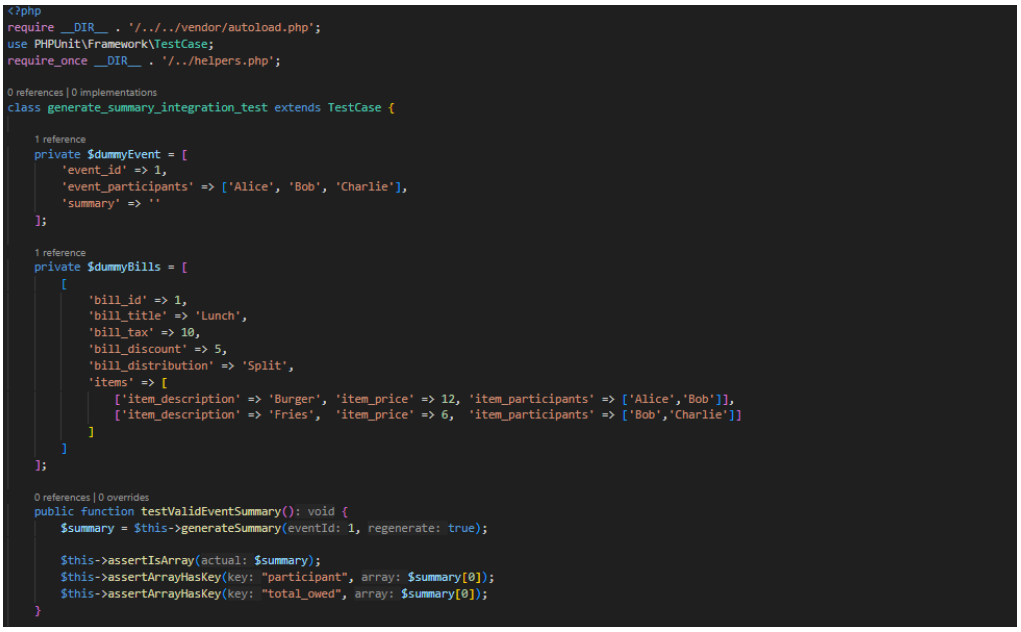


```dart
//summary unit test
import 'package:flutter_test/flutter_test.dart';

double calculateTotalOwed(List bills) {
  double total = 0.0;
  for (var bill in bills) {
    total += (bill['billShare'] as num?)?.toDouble() ?? 0.0;
  }
  return total;
}

String formatItems(List items) {
  final formattedItems = items.map((item) {
    final itemPrice = (item['price'] as num).toDouble();
    return '${item['name']} (RM${itemPrice.toStringAsFixed(2)})';
  }).join(', ');
  return formattedItems;
}

Run | Debug
void main() {
  Run | Debug
  group('Summary Unit Tests', () {
    Run | Debug
    test('calculateTotalOwed functions sums bill shares correctly', () {
      final bills = [
        {'billShare': 10.0},
        {'billShare': 20.5},
        {'billShare': null},
        {},
      ];
      final total = calculateTotalOwed(bills);

      expect(total, 30.5);
    });

    Run | Debug
    test('Formats items correctly', () {
      final items = [
        {'name': 'Burger', 'price': 12.5},
```

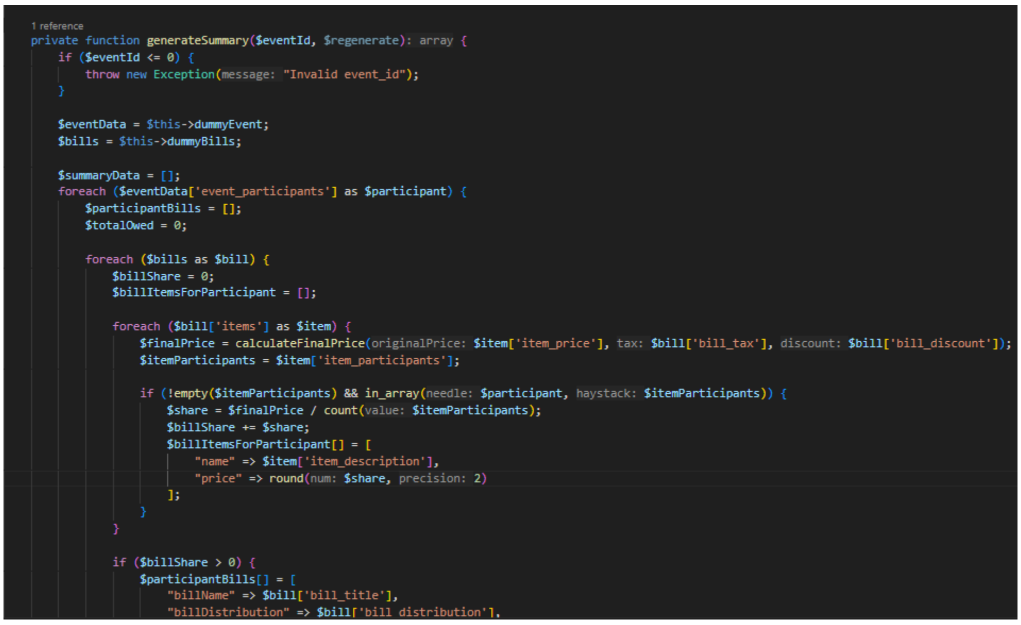


```dart
        {'name': 'Fries', 'price': 3.75},
      ];

      final formatted = formatItems(items);

      expect(formatted, 'Burger (RM12.50), Fries (RM3.75)');
    });

    Run | Debug
    test('formatItems handles empty list', () {
      final formatted = formatItems([]);
      expect(formatted, '');
    });
  });
}
```

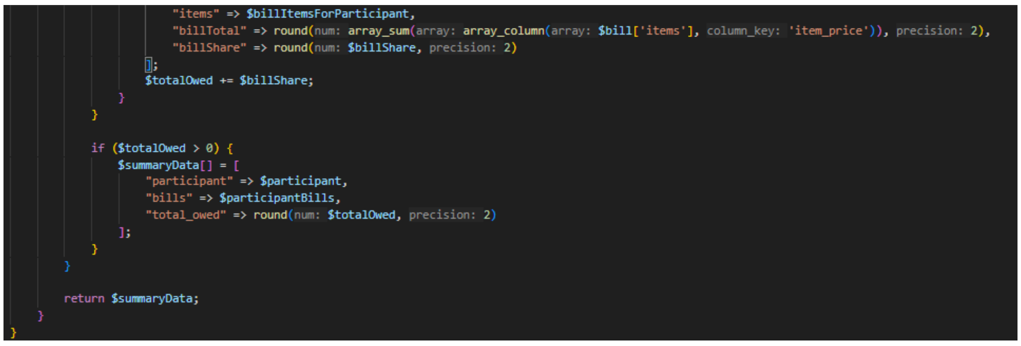
Appendix B



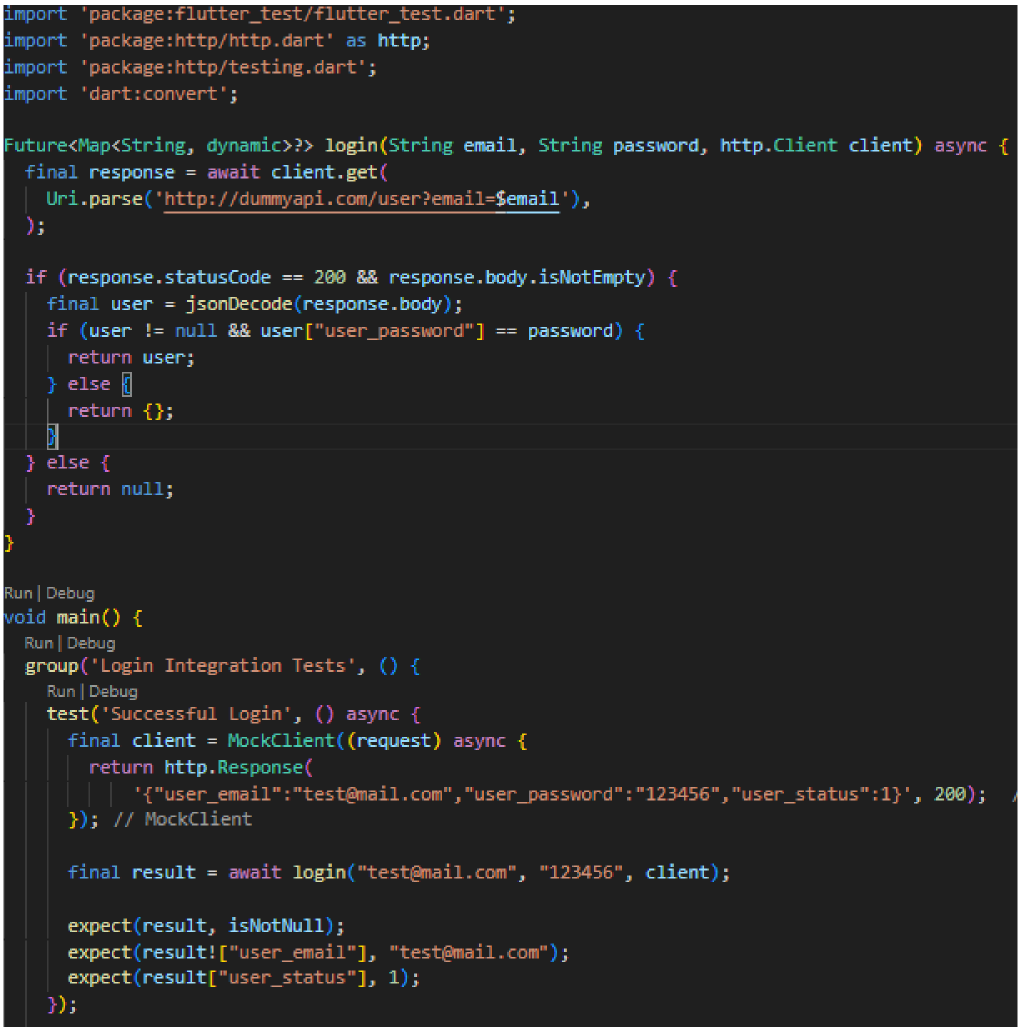
```php
<?php
require __DIR__ . '/../../vendor/autoload.php';
use PHPUnit\Framework\TestCase;
require_once __DIR__ . '/../helpers.php';

0 references | 0 implementations
class generate_summary_integration_test extends TestCase {

    1 reference
    private $dummyEvent = [
        'event_id' => 1,
        'event_participants' => ['Alice', 'Bob', 'Charlie'],
        'summary' => ''
    ];

    1 reference
    private $dummyBills = [
        [
            'bill_id' => 1,
            'bill_title' => 'Lunch',
            'bill_tax' => 10,
            'bill_discount' => 5,
            'bill_distribution' => 'Split',
            'items' => [
                ['item_description' => 'Burger', 'item_price' => 12, 'item_participants' => ['Alice','Bob']],
                ['item_description' => 'Fries',  'item_price' => 6,  'item_participants' => ['Bob','Charlie']]
            ]
        ]
    ];

    0 references | 0 overrides
    public function testValidEventSummary(): void {
        $summary = $this->generateSummary(eventId: 1, regenerate: true);

        $this->assertIsArray(actual: $summary);
        $this->assertArrayHasKey(key: "participant", array: $summary[0]);
        $this->assertArrayHasKey(key: "total_owed", array: $summary[0]);
    }
```



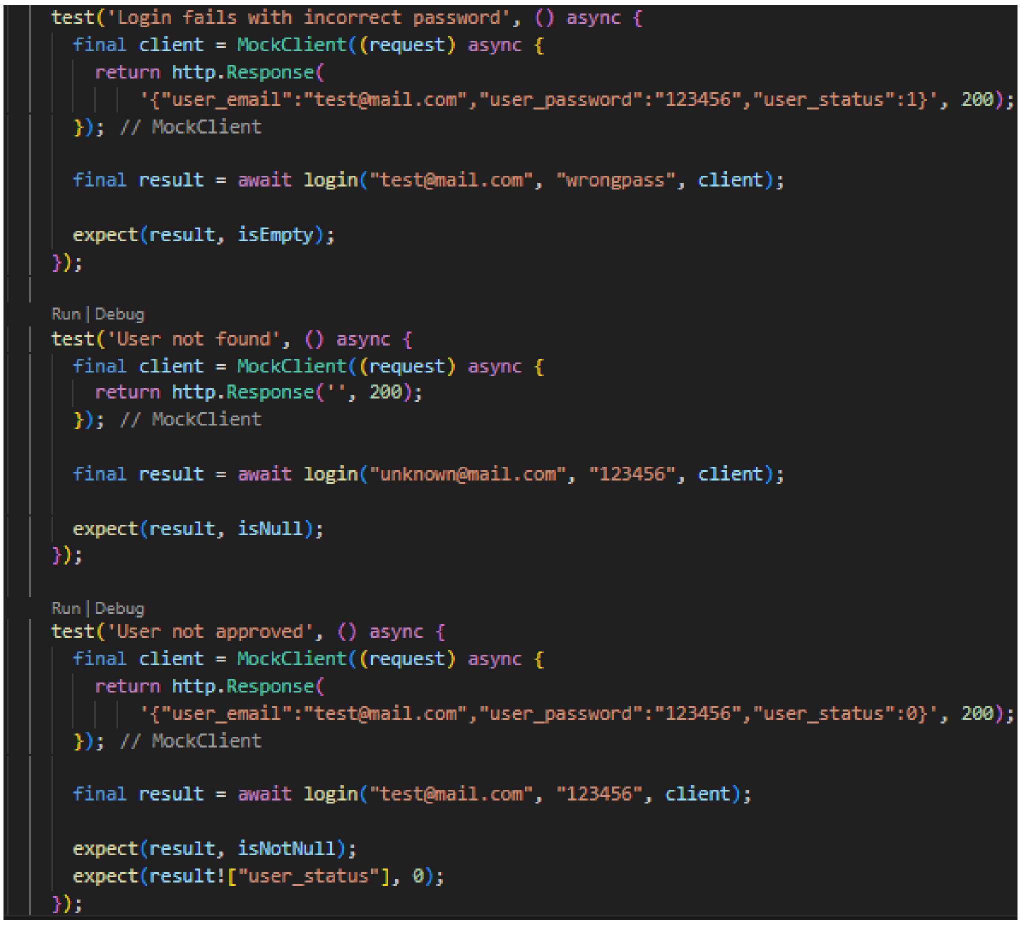
```php
    1 reference
    private function generateSummary($eventId, $regenerate): array {
        if ($eventId <= 0) {
            throw new Exception(message: "Invalid event_id");
        }

        $eventData = $this->dummyEvent;
        $bills = $this->dummyBills;

        $summaryData = [];
        foreach ($eventData['event_participants'] as $participant) {
            $participantBills = [];
            $totalOwed = 0;

            foreach ($bills as $bill) {
                $billShare = 0;
                $billItemsForParticipant = [];

                foreach ($bill['items'] as $item) {
                    $finalPrice = calculateFinalPrice(originalPrice: $item['item_price'], tax: $bill['bill_tax'], discount: $bill['bill_discount']);
                    $itemParticipants = $item['item_participants'];

                    if (!empty($itemParticipants) && in_array(needle: $participant, haystack: $itemParticipants)) {
                        $share = $finalPrice / count(value: $itemParticipants);
                        $billShare += $share;
                        $billItemsForParticipant[] = [
                            "name" => $item['item_description'],
                            "price" => round(num: $share, precision: 2)
                        ];
                    }
                }

                if ($billShare > 0) {
                    $participantBills[] = [
                        "billName" => $bill['bill_title'],
                        "billDistribution" => $bill['bill_distribution'],
```



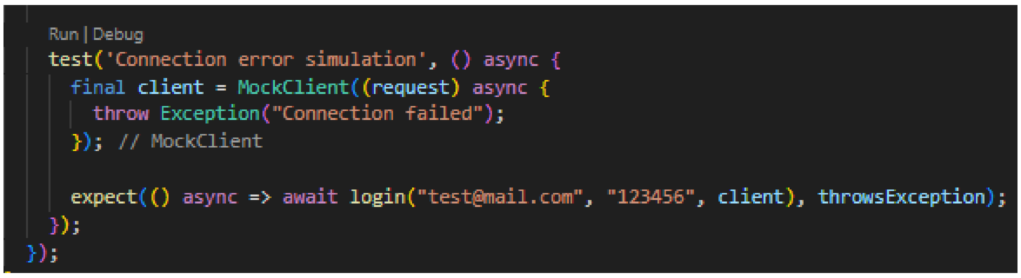
```php
                        "items" => $billItemsForParticipant,
                        "billTotal" => round(num: array_sum(array: array_column(array: $bill['items'], column_key: 'item_price')), precision: 2),
                        "billShare" => round(num: $billShare, precision: 2)
                    ];
                    $totalOwed += $billShare;
                }
            }

            if ($totalOwed > 0) {
                $summaryData[] = [
                    "participant" => $participant,
                    "bills" => $participantBills,
                    "total_owed" => round(num: $totalOwed, precision: 2)
                ];
            }
        }

        return $summaryData;
    }
}
```



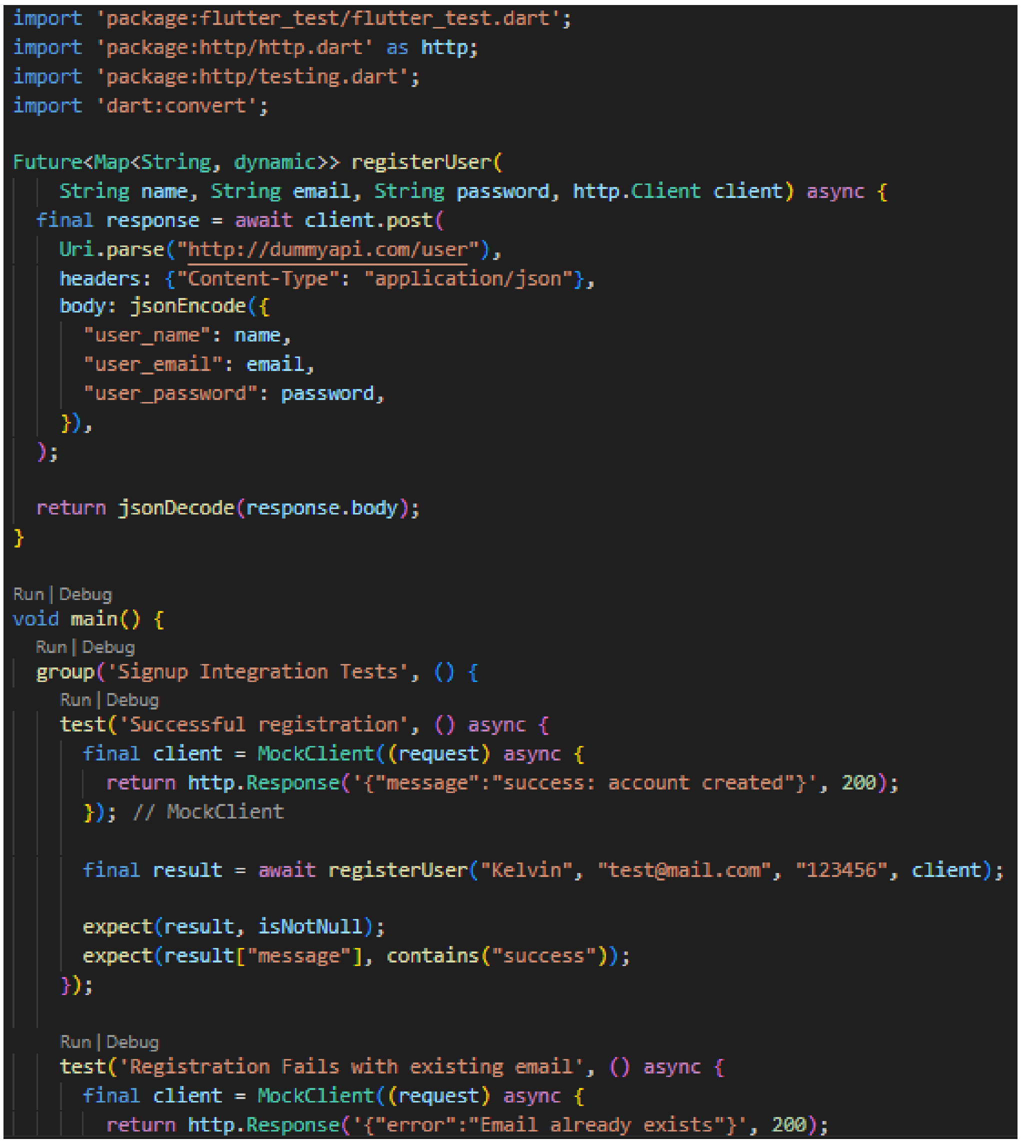
```
import 'package:flutter_test/flutter_test.dart';
import 'package:http/http.dart' as http;
import 'package:http/testing.dart';
import 'dart:convert';

Future<Map<String, dynamic>?> login(String email, String password, http.Client client) async {
  final response = await client.get(
    Uri.parse('http://dummyapi.com/user?email=$email'),
  );

  if (response.statusCode == 200 && response.body.isNotEmpty) {
    final user = jsonDecode(response.body);
    if (user != null && user["user_password"] == password) {
      return user;
    } else {
      return {};
    }
  } else {
    return null;
  }
}

Run | Debug
void main() {
  Run | Debug
  group('Login Integration Tests', () {
    Run | Debug
    test('Successful Login', () async {
      final client = MockClient((request) async {
        return http.Response(
            '{"user_email":"test@mail.com","user_password":"123456","user_status":1}', 200);  /
      }); // MockClient

      final result = await login("test@mail.com", "123456", client);

      expect(result, isNotNull);
      expect(result!["user_email"], "test@mail.com");
      expect(result["user_status"], 1);
    });
```



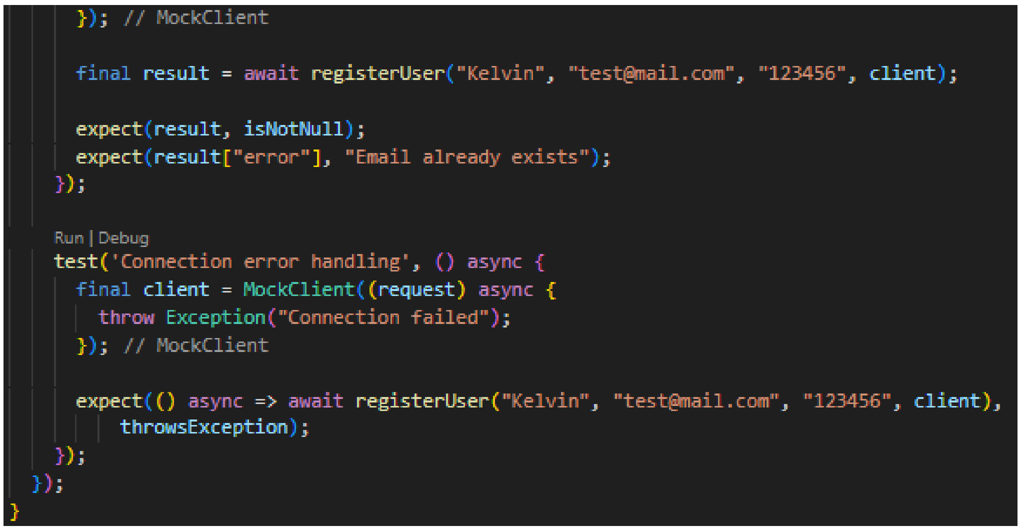
```
  test('Login fails with incorrect password', () async {
    final client = MockClient((request) async {
      return http.Response(
          '{"user_email":"test@mail.com","user_password":"123456","user_status":1}', 200);
    }); // MockClient

    final result = await login("test@mail.com", "wrongpass", client);

    expect(result, isEmpty);
  });

  Run | Debug
  test('User not found', () async {
    final client = MockClient((request) async {
      return http.Response('', 200);
    }); // MockClient

    final result = await login("unknown@mail.com", "123456", client);

    expect(result, isNull);
  });

  Run | Debug
  test('User not approved', () async {
    final client = MockClient((request) async {
      return http.Response(
          '{"user_email":"test@mail.com","user_password":"123456","user_status":0}', 200);
    }); // MockClient

    final result = await login("test@mail.com", "123456", client);

    expect(result, isNotNull);
    expect(result!["user_status"], 0);
  });
```



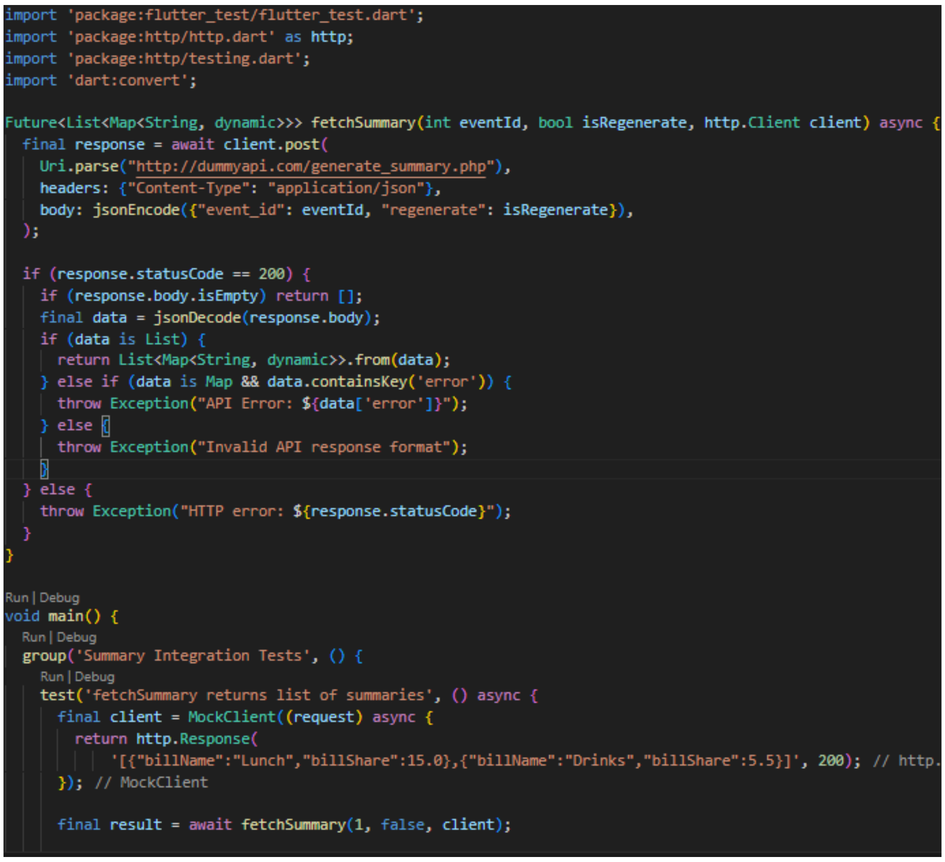
```
  Run | Debug
  test('Connection error simulation', () async {
    final client = MockClient((request) async {
      throw Exception("Connection failed");
    }); // MockClient

    expect(() async => await login("test@mail.com", "123456", client), throwsException);
  });
});
```



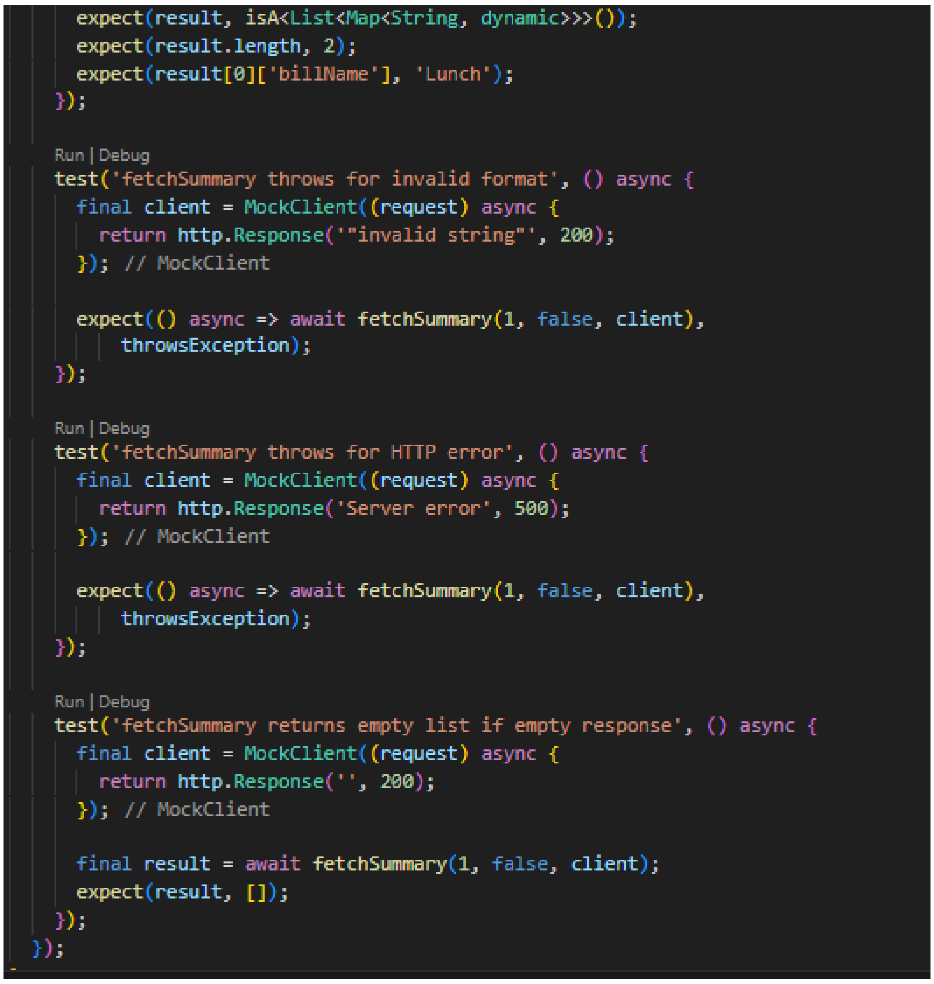
```dart
import 'package:flutter_test/flutter_test.dart';
import 'package:http/http.dart' as http;
import 'package:http/testing.dart';
import 'dart:convert';

Future<Map<String, dynamic>> registerUser(
    String name, String email, String password, http.Client client) async {
  final response = await client.post(
    Uri.parse("http://dummyapi.com/user"),
    headers: {"Content-Type": "application/json"},
    body: jsonEncode({
      "user_name": name,
      "user_email": email,
      "user_password": password,
    }),
  );

  return jsonDecode(response.body);
}

Run | Debug
void main() {
  Run | Debug
  group('Signup Integration Tests', () {
    Run | Debug
    test('Successful registration', () async {
      final client = MockClient((request) async {
        return http.Response('{"message":"success: account created"}', 200);
      }); // MockClient

      final result = await registerUser("Kelvin", "test@mail.com", "123456", client);

      expect(result, isNotNull);
      expect(result["message"], contains("success"));
    });

    Run | Debug
    test('Registration Fails with existing email', () async {
      final client = MockClient((request) async {
        return http.Response('{"error":"Email already exists"}', 200);
```



```dart
      }); // MockClient

      final result = await registerUser("Kelvin", "test@mail.com", "123456", client);

      expect(result, isNotNull);
      expect(result["error"], "Email already exists");
    });

    Run | Debug
    test('Connection error handling', () async {
      final client = MockClient((request) async {
        throw Exception("Connection failed");
      }); // MockClient

      expect(() async => await registerUser("Kelvin", "test@mail.com", "123456", client),
          throwsException);
    });
  });
}
```



```dart
import 'package:flutter_test/flutter_test.dart';
import 'package:http/http.dart' as http;
import 'package:http/testing.dart';
import 'dart:convert';

Future<List<Map<String, dynamic>>> fetchSummary(int eventId, bool isRegenerate, http.Client client) async {
  final response = await client.post(
    Uri.parse("http://dummyapi.com/generate_summary.php"),
    headers: {"Content-Type": "application/json"},
    body: jsonEncode({"event_id": eventId, "regenerate": isRegenerate}),
  );

  if (response.statusCode == 200) {
    if (response.body.isEmpty) return [];
    final data = jsonDecode(response.body);
    if (data is List) {
      return List<Map<String, dynamic>>.from(data);
    } else if (data is Map && data.containsKey('error')) {
      throw Exception("API Error: ${data['error']}");
    } else {
      throw Exception("Invalid API response format");
    }
  } else {
    throw Exception("HTTP error: ${response.statusCode}");
  }
}

Run | Debug
void main() {
  Run | Debug
  group('Summary Integration Tests', () {
    Run | Debug
    test('fetchSummary returns list of summaries', () async {
      final client = MockClient((request) async {
        return http.Response(
            '[{"billName":"Lunch","billShare":15.0},{"billName":"Drinks","billShare":5.5}]', 200); // http.
      }); // MockClient

      final result = await fetchSummary(1, false, client);
```



```dart
      expect(result, isA<List<Map<String, dynamic>>>());
      expect(result.length, 2);
      expect(result[0]['billName'], 'Lunch');
    });

    Run | Debug
    test('fetchSummary throws for invalid format', () async {
      final client = MockClient((request) async {
        return http.Response('"invalid string"', 200);
      }); // MockClient

      expect(() async => await fetchSummary(1, false, client),
          throwsException);
    });

    Run | Debug
    test('fetchSummary throws for HTTP error', () async {
      final client = MockClient((request) async {
        return http.Response('Server error', 500);
      }); // MockClient

      expect(() async => await fetchSummary(1, false, client),
          throwsException);
    });

    Run | Debug
    test('fetchSummary returns empty list if empty response', () async {
      final client = MockClient((request) async {
        return http.Response('', 200);
      }); // MockClient

      final result = await fetchSummary(1, false, client);
      expect(result, []);
    });
  });
```