

INCREASING PRODUCTIVITY WITH LIFE  
PLANNER MOBILE APPLICATION USING  
IONIC FRAMEWORK

AHMAD NAQUIB BIN ALI

Bachelor of Computer Science (Software  
Engineering) with Honors

UNIVERSITI MALAYSIA PAHANG

# UNIVERSITI MALAYSIA PAHANG

## DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : AHMAD NAQUIB BIN ALI

Date of Birth

Title : INCREASING PRODUCTIVITY WITH LIFE PLANNER  
MOBILE APPLICATION USING IONIC FRAMEWORK

Academic Session : SEM 2 21/22

I declare that this thesis is classified as:

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

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

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

Certified by:

\_\_\_\_\_  
(Student's Signature)

\_\_\_\_\_  
New IC/Passport Number  
Date:

\_\_\_\_\_  
(Supervisor's Signature)

DR. MOHD FAIZAL BIN AB RAZAK  
\_\_\_\_\_  
Name of Supervisor  
Date: 10/2/2023

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

# THESIS DECLARATION LETTER (OPTIONAL)

Librarian,  
*Perpustakaan Universiti Malaysia Pahang,*  
Universiti Malaysia Pahang,  
Lebuhraya Tun Razak,  
26300, Gambang, Kuantan.

Dear Sir,

## CLASSIFICATION OF THESIS AS RESTRICTED

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

Ahmad Naquib  
bin Ali

INCREASING  
PRODUCTIVITY  
WITH LIFE  
PLANNER  
MOBILE  
APPLICATION  
USING IONIC  
FRAMEWORK

- |         |       |
|---------|-------|
| Reasons | (i)   |
|         | (ii)  |
|         | (iii) |

Thank you.

Yours faithfully,

  
DR. MOHD. FAIZAL BIN AB RAZAK  
SENIOR LECTURER  
FACULTY OF COMPUTER SYSTEMS & SOFTWARE ENGINEERING  
UNIVERSITI MALAYSIA PAHANG  
LEBUHRAYA TUN RAZAK, 26300 GAMBANG, KUANTAN  
PAHANG DARUL MAKMUR  
TEL : 09-549 2217 FAX : 09-549 2144

---

(Supervisor's Signature)

Date: 10/2/2023

Stamp:

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

## SUPERVISOR'S DECLARATION

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



DR. MOHD FAIZAL BIN AB RAZAK  
SENIOR LECTURER  
FACULTY OF COMPUTER SYSTEMS & SOFTWARE ENGINEERING  
UNIVERSITI MALAYSIA PAHANG  
LEBUHRAYA TUN RAZAK, 26300 GAMBANG, KUANTAN  
PAHANG DARUL MAKMUR  
TEL: 09-549 2217 FAX: 09-549 2144

---

(Supervisor's Signature)

Full Name : DR. MOHD FAIZAL BIN AB RAZAK  
Position : SENIOR LECTURER  
Date : 10/2/2023

---

(Co-supervisor's Signature)

Full Name :  
Position :  
Date :



## STUDENT'S DECLARATION

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

A handwritten signature in black ink, appearing to read 'Ahmad Naquib Bin Ali', is written over a horizontal line.

---

(Student's Signature)

Full Name : AHMAD NAQUIB BIN ALI

ID Number : CB19140

Date : 10/2/2023

INCREASING PRODUCTIVITY WITH LIFE PLANNER MOBILE APPLICATION  
USING IONIC FRAMEWORK

AHMAD NAQUIB BIN ALI

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Computer Science (Software Engineering) With Honors

Faculty of Computing  
UNIVERSITI MALAYSIA PAHANG

## **ACKNOWLEDGEMENTS**

First and foremost, I would like to express sincere gratitude to all those who have supported me in building this application and completing this project. Not to forget I would like to thank my project supervisor Dr. Mohd Faizal b. Ab Razak in providing his valuable guidance, support, and encouragement throughout completing the project. Your feedback helps the project to improve greatly. I would like also to extend my thank to everyone who participated in the studies in building the Life Planner application. I would like to thank my colleagues and friends that provide me with their valuable feedback as well as encouraged me in completing the project. Last but not least, I would like to thank my family for their unwavering support and understanding during the course of the project. I hope that by doing this project, it will contribute to the development of mobile applications, and I am looking forward to other opportunities that will arrive in the near future.



## **ABSTRAK**

Aplikasi mudah alih ialah sejenis program perisian yang dicipta untuk dijalankan pada peranti mudah alih, seperti telefon pintar atau tablet, dan sering dirujuk sebagai aplikasi sahaja. Projek ini bertujuan untuk membangunkan aplikasi mudah alih yang boleh membantu pengguna dalam merancang tugas harian mereka serta menguruskan simpanan mereka. Ini berhubung masalah yang ditemui iaitu pengguna sukar mengingat tugas dan menguruskan kewangan. Objektif projek ini adalah untuk mereka bentuk dan membangunkan aplikasi iaitu Life Planner yang merangkumi ciri seperti senarai tugas, alatan perancangan, perancangan kewangan, serta alatan pengurusan masa. Disebabkan itu, pendekatan Rapid Application Development (RAD) telah dipilih dan RAD mempunyai 4 fasa iaitu perancangan keperluan, reka bentuk pengguna, pembinaan dan cutover. RAD memfokuskan pada penghantaran tepat pada masanya berdasarkan maklum balas pengguna dan interaksi pengguna aktif sepanjang proses pembangunan. Hasilnya ialah aplikasi mudah alih dengan keupayaan termasuk senarai tugas, alat pengurusan masa, pemantauan perbelanjaan, alat perancangan kewangan, alat belanjawan dan kalendar yang dapat memenuhi masalah permulaan yang telah ditemui. Kesimpulannya, projek ini berjaya membangunkan aplikasi mudah alih (Life Planner) yang membantu pengguna dalam kehidupan seharian mereka dan aplikasi ini juga boleh ditambah baik dengan penambahbaikan yang lebih baik. Selain itu, model RAD juga berjaya dilaksanakan kerana prototaip awal dapat dibangunkan dan perubahan selanjutnya boleh dilakukan secara awal.

# **ABSTRACT**

A mobile application is a sort of software program created to run on a mobile device, such as a smartphone or tablet, and is frequently referred to as simply an app. This project aims on developing a mobile application that can help users in planning their daily task as well as managing their saving. This is regarding the problems that were found which is users is having a hard time remembering their tasks and managing their finance. The objective of this project is to design and develop an app namely Life Planner that include features such as task list, planning tools, financial planning, as well as time management tools. Due to that, Rapid Application Development (RAD) approach was selected and RAD has 4 phases which are requirements planning, user design, construction, and cutover. RAD focus on timely delivery based on user feedback and active user interaction throughout the development process. The result was a mobile application with capabilities including to-do lists, time management tools, expense monitoring, financial planning tools, budgeting tools, and calendars that were able to satisfy with the starting problems that have been found. In conclusion, the project was able to successfully develop a mobile application (Life Planner) that assists user in their daily life and this application also can be improved with a better enhancement. Additionally, RAD model was also successfully implemented as an early prototype was able to be developed and further changes can be done in an early manner.

# TABLE OF CONTENT

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF SYMBOLS</b>	<b>x</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xi</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>12</b>
1.1 INTRODUCTION	12
1.2 PROBLEM STATEMENTS	13
1.3 OBJECTIVE	14
1.4 SCOPE	14
1.5 SIGNIFICANT	14
1.6 GANTT CHART	15
1.7 THESIS ORGANIZATION	15
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>17</b>
2.1 INTRODUCTION	17
2.2 MOBILE APPLICATION	17

2.3	REVIEW OF EXISTING SYSTEMS	18
2.3.1	ColorNote	18
2.3.2	Planner Pro	19
2.3.3	Money Manager: Expense Tracker	19
2.4	SUMMARY OF COMPARISONS OF THREE EXISTING SYSTEM	20
<b>CHAPTER 3 METHODOLOGY</b>		<b>22</b>
3.1	INTRODUCTION	22
3.2	PROJECT MANAGEMENT FRAMEWORK	22
3.3	PROJECT REQUIREMENT	23
3.3.1	FUNCTIONAL REQUIREMENT	24
3.3.2	NON-FUNCTIONAL REQUIREMENT	24
3.3.3	CONSTRAINTS & LIMITATIONS	24
3.4	PROJECT DESIGN	24
3.4.1	Context Diagram	25
3.4.2	Use Case Diagram	25
3.4.3	Activity Diagram	26
3.5	Data Design	27
3.5.1	Entity Relationship Diagram	27
3.5.2	Database Dictionary	28
3.6	Initial Concept	32
3.6.1	Prototype Design	32
3.6.2	Potential Use	36
3.7	Testing Plan	37
<b>CHAPTER 4 RESULTS AND DISCUSSION</b>		<b>39</b>

4.1	4.1 Introduction	39
4.2	4.2 Implementation Process	39
	4.2.1 4.2.1 Tools and Technology	39
	<b>CHAPTER 5 CONCLUSION</b>	<b>66</b>
5.1	Introduction	66
	<b>REFERENCES</b>	<b>68</b>

## LIST OF TABLES

Table 1.1	Problem statements in the project	13
Table 1.2	Comparison Chapter	15
Table 2.1	Comparison between existing application	20
Table 3.1	Users Table	28
Table 3.2	Notes Table	29
Table 3.3	Transaction table	29
Table 3.4	UserDetails Table	30
Table 3.5	Events Table	30
Table 3.6	Testing Plan	37

# LIST OF FIGURES

Figure 1.1	Gantt Chart	15
Figure 2.1	ColorNote Interface	18
Figure 2.2	Planner Pro Interface	19
Figure 2.3	MoneyManager Interface	20
Figure 3.1	RAD Phases	23
Figure 3.2	Context Diagram	25
Figure 3.3	Use Case Diagram	26
Figure 3.4	Activity Diagram	27
Figure 3.5	ERD Diagram	28
Figure 3.6	Login Page	32
Figure 3.7	Sign Up Page	33
Figure 3.8	Home Page	34
Figure 3.9	Add-Note Modal	35
Figure 3.10	Finance Page	36
Figure 4.1	Visual Studio Code	39
Figure 4.2	Firebase	40
Figure 4.3	Ionic	40
Figure 4.4	Structure of Life Planner Application	41
Figure 4.5	tab1.page.html (Note Page)	42
Figure 4.6	Structure on tab1	42
Figure 4.7	Firestore Database	43
Figure 4.8	Realtime Database	43
Figure 4.9	List of Pages for Note Function	45
Figure 4.10	Profile Page	46
Figure 4.11	List of Pages for Finance Function	48
Figure 4.12	List of Pages for Event Function	50
Figure 4.13	About Us Page	52
Figure 4.14	Admin Page	54

# LIST OF SYMBOLS



## **LIST OF ABBREVIATIONS**

LP	Life Planner
App	Application
RAD	Rapid Application Development
PC	Personal Computer
UAT	User Acceptance Test

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 INTRODUCTION**

A mobile application, often known as an app, is software that runs on a mobile device such as a smartphone or even a tablet computer. The mobile app frequently provides users with utility or functions that are close to what is available on PCs. Generally, apps on mobile phones are small hence the functions that are provided to the users are usually limited. Mobile applications even though their functions are restricted, this software is still a leap from the usually integrated software that is commonly found on PCs. Each app will be able to provide a specific function such as games, a life planner, or a social media application.

Life planning is not a simple task to do as it can vary from person to person. Things such as how can a person manage their times, plan their days, as well as how they manage their money throughout the month or year. Mobile application is one of the things that can help the users in organizing all of that. A mobile application has its categories as the topic is huge and it has been expanding throughout the decades. One of the categories is a productivity mobile app. As the name implies, it may appear to be one of the more mundane categories of mobile apps, it is a hugely popular one. These apps will be able to assist their users in completing a task swiftly and efficiently, making what can tedious activities easier and more enjoyable.

For this project, this mobile app is built specifically on that category as it will be able to help their users in planning their daily life as well as plan their savings throughout the month. Planning is crucial in our daily life as it will help users to organize, and it helps to remember important events in our daily life. This app will also have the potential to

motivate as well as engage the users in enhancing their planning and organizing their daily life to a better version.

## 1.2 PROBLEM STATEMENTS

Table 1.1 Problem statements in the project

No	Problem	Description	Effect
1	Difficulties in remembering the list of things that are needed to do.	Users will have a hard time remembering a list of things that they are supposed to do as the day progresses due to other things that will emerge.	Depending on the things that are supposed to do by the user. It will have different effects on them. Users will need to reschedule their time if it is an important thing, or they will lose the opportunity all at once to do the thing as they have lost the opportunity to do it.
2	Users need to keep track of their expenses throughout the month for better finance.	Every month, users' expenses will vary as they are important things that they must purchase. For working people that have assets, emergencies such as a house or car repairs eventually will arise. Due to that, monthly expenses, as well as savings, will differ each month.	Finance is a hugely beneficial thing if it is handled properly. On the flip side, if it's not, users will have difficulties in buying things that they are supposed to do. Other than that, if an emergency occurs users will not have any ways to cover it as their finances have not been properly planned out for the future.
3	Hard to schedule some of the things that are due according to the time given.	To fit everything into our schedule. Some things will incidentally be forgotten to do according to their time. Hence, making users have difficulties in doing them in their respective time.	Time is a valuable thing to be taken care of. Some things cannot be done anytime throughout the day. For example, if one had forgotten their appointment, they would have to do the hassle of making another appointment in the future. Next, time-sensitive things are important to be taken care of in due time as some things cannot be done at another time.

According to table 1.1 above, there are a few problems that are being stated for the project. One of the problems is the users will have a hard time remembering the things that they are supposed to do. Eventually, as the day progress, users will have a few things emerge adding on one by one hence making it difficult to remember each of them. Other than that, expenses for the users can vary for each user. There are also emergency expenses that will emerge from time to time. Making it become a problem

to track all their expenses throughout the month. Lastly, users' schedules can become quite busy depending on their occupation of the user. Depending on the user, it can become hard to schedule everything and do things in the given period.

### **1.3 OBJECTIVE**

Based on the problem statements, the objectives of the project are:

- i) To study the existing productivity mobile application for a better mobile application
- ii) To design and develop a system that allows the users to improve their daily user's life.
- iii) To evaluate the effectiveness and functionality of the developed productivity mobile application.

### **1.4 SCOPE**

User scope, system scope, and development scope are the three types of scopes. The app's user scope describes the app's intended audience. This app is intended for a mature audience which is for people who is working as the app is also intended to record the user's savings as they have a stable income. System scope is the required system in the game. This application will be able to record any user's to-do list as well as remind the user based on the given time, it can also record the expenses of the users and show them in a graph table. The development scope describes the software and element that is used to develop the application. The software that is used to develop this application is using ionic framework, visual studio code as well as command prompt. In terms of hardware, a mobile phone is used to test the application that will be developed.

### **1.5 SIGNIFICANT**

- i) Allow users to be reminded or record the things that they are supposed to do during that time.

ii) Let users view or plan their expenses throughout the month.

## 1.6 GANTT CHART

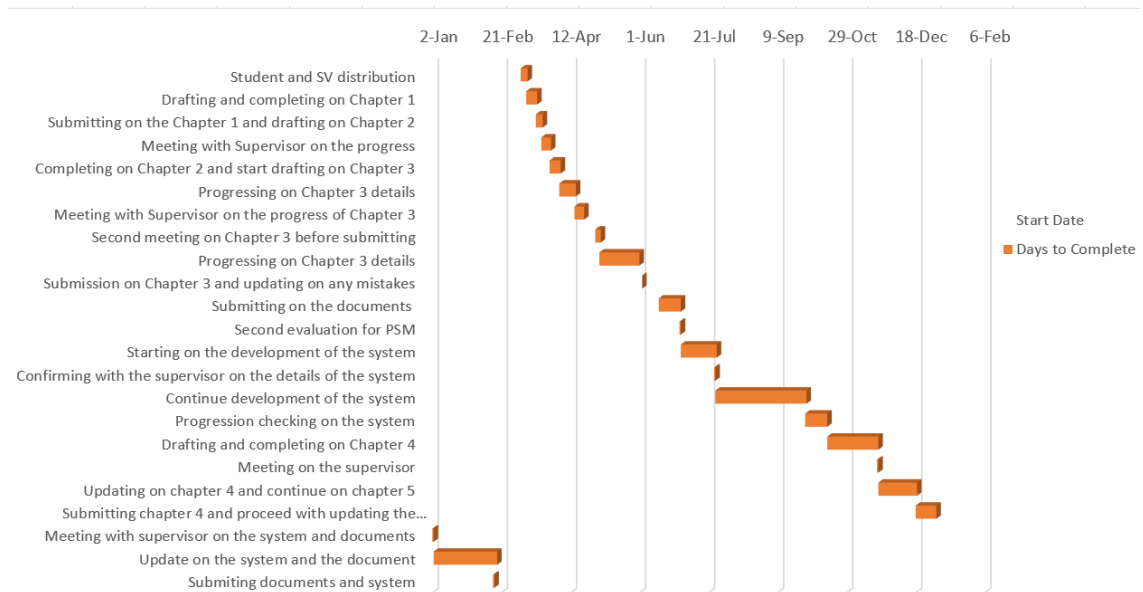


Figure 1.1 Gantt Chart

## 1.7 THESIS ORGANIZATION

Table 1.2 Comparison Chapter

	<b>Title</b>	<b>Explanation</b>	<b>Significance</b>
<b>Chapter 1</b>	Introduction	Discussed the project's introduction, which included problem statements, objectives, the scope and significance of the project, and thesis organization.	Give an overview of what this thesis is about.
<b>Chapter 2</b>	Literature Review	Briefly presents the literature review of the three existing mobile applications, descriptions, and comparisons of available mobile applications that are specific to productivity applications.	Provide an analysis of different application that is like the application that will be built.
<b>Chapter 3</b>	Methodology	Describe and justify the method employed during the project's development. In addition, chapter 3 covered the project's technique, software, and hardware.	Provide a detailed plan on how the application will be built based on the figures, interfaces, and model that will be used.
<b>Chapter 4</b>	Results and	Describe the creation and	Provide the testing

	Discussion	implementation of the system as well as the testing, result from discussion, and user manual in the application.	result on the test application that had been made to the user. Hence, making it easier to update the highlighted part.
<b>Chapter 5</b>	Conclusion	Conclusion of the project findings, as well as a discussion of potential enhancements that could be done to improve the project's requirements.	Summary of the project and state the potential upgrade that can be done to the application.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION**

Productivity is one of the most important features in one's life as it can improve their quality of life. Thus, the purpose of this chapter is to explain more in-depth the three current systems in the mobile application that is almost similar to the Life Planner mobile application and compare them. These three applications are more focused on their Graphical User Interface (GUI), Mobile Operating System (OS), language support, connection type, target audience, the topic covered in the application, main function, advantages as well as disadvantages of the application. This comparison of the existing application suggests the strength and efficacy of the present application, allowing this project to build a better version of the application.

#### **2.2 MOBILE APPLICATION**

A mobile application (also known as a mobile app) is a type of software that runs on a mobile device such as a smartphone or tablet computer. Apps, despite being little software units with limited functionality, manage to provide consumers with high-quality services and experiences.

Mobile apps, in contrast to desktop applications, are a departure from the traditional integrated software systems seen on PCs. Instead, each app offers a single, limited feature, such as a game, calculator, or mobile web browsing. Although early mobile devices' restricted hardware resources prevented programs from multitasking, their uniqueness is now part of their appeal because it allows consumers to choose what their devices can and cannot accomplish.

Mobile apps avoid multi-functionality due to the restricted hardware resources of early mobile devices. Even though today's smartphones are significantly more advanced, mobile apps remain limited in their functionality.

## 2.3 REVIEW OF EXISTING SYSTEMS

This section will explain about three existing applications that emphasize the productivity of the user. The three applications are ColorNote, Planner Pro, as well as Money Manager mobile application.

### 2.3.1 ColorNote

ColorNote is a versatile notepad mobile application that is developed by Notes. This application's main function is to let the user do a quick and simple notepad editing experience whenever the user wants to write notes for their to-do lists, memos, messages, and shopping lists. From the main function, it derives into two basic note-taking formats, a lined-paper styled text option and a checklist option. The text option, which acts as a rudimentary word processor, allows the user to write as many characters as they like. Users can edit, share, set a reminder, and check off their notes once they've been saved. Other than that, users also can organize their schedules by notes in the calendar [2].

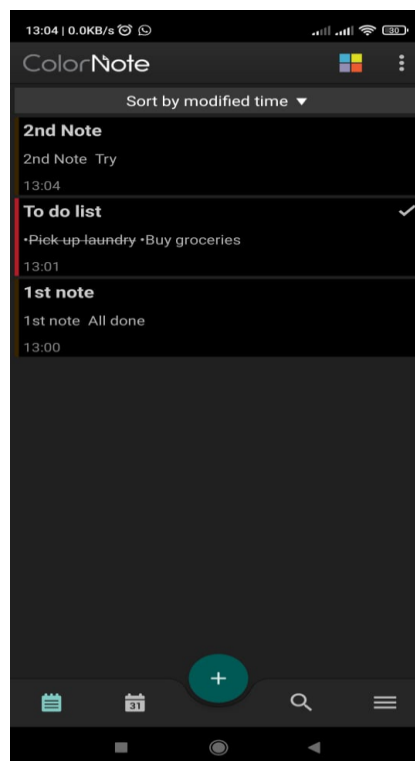


Figure 2.1 ColorNote Interface



### 2.3.2 Planner Pro

Planner Pro is an application that is developed by Beesoft Apps. This application combines different kinds of events, tasks, and notes in one place. It can also be used as a day, weekly and monthly planner. Events such as syncing with google calendar and events, showing or hiding calendars as the user want, and full records searching including events, tasks, and notes are what makes the application unique from the other two applications [3]. Other key features of this application are, that it has a well-designed interface as well as the day, week, month, and tasks module for better management.

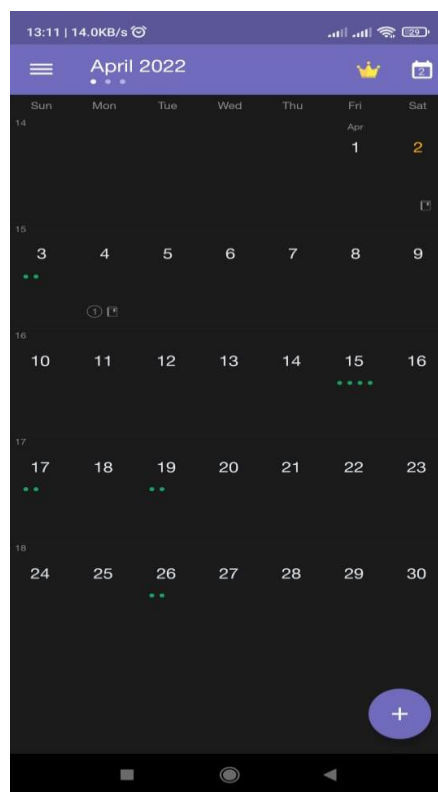


Figure 2.2 Planner Pro Interface

### 2.3.3 Money Manager: Expense Tracker

This application will assist users in budgeting and income planning. Users can use it to keep track of their costs and revenue daily. Any user's financial well-being depends on their ability to organize their budget. Keeping a budget will allow the user to see how much money they are making and where it is going. They can form their income in a few months and identify the income that is prevalent in it (wages, income from

entrepreneurship, hobbies, or what you borrowed from your parents). They will also be given detailed information on your expenses [4]. Users can discover how they will be able to properly manage all their income that is coming in and out.

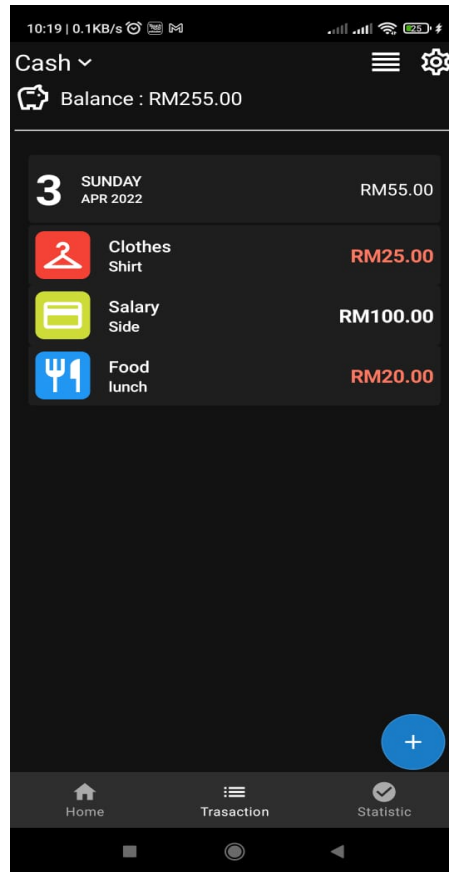


Figure 2.3 MoneyManager Interface

## 2.4 SUMMARY OF COMPARISONS OF THREE EXISTING SYSTEM

Table 2.1 Comparison between existing application

Application	ColorNote	Planner Pro	Money Manager: Expense Tracker
Current Version	Varies with device	5.4	1.1.6
Requires Android	Varies with device	5.0 and above	5.0 and above
Features	- Provide two basic note-taking formats	- Sync with google calendar - Add, edit, or	- Add income of the 9 categories for the day

	<ul style="list-style-type: none"> <li>- Organize notes by color</li> <li>- Sticky note memo widget on the home screen</li> <li>- Secured backup notes to SD storage</li> </ul>	<ul style="list-style-type: none"> <li>- delete notes</li> <li>- System reminder for each task</li> <li>- Unlimited number of notes for the user</li> </ul>	<ul style="list-style-type: none"> <li>- Add expense for the 19 expense categories</li> <li>- Provide statistics for any wallet</li> <li>- Able to track the percentage of user's income and expenses</li> </ul>
Advantages	<ul style="list-style-type: none"> <li>- Provide multiple functions to ease user</li> </ul>	<ul style="list-style-type: none"> <li>- Able to connect with email</li> <li>- Shows available holidays on the calendar</li> </ul>	<ul style="list-style-type: none"> <li>- Simple app to use, easy to understand</li> <li>- Able to provide details on the expenses and income</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>- Complex software to use, take a bit of time to get used to</li> </ul>	<ul style="list-style-type: none"> <li>- The user interface is lacking in certain parts</li> </ul>	<ul style="list-style-type: none"> <li>- Lots of settings to get used to</li> </ul>

## 2.5 Conclusion

In this chapter, the existing literature review on mobile application, development, and user development were reviewed. Numerous models and strategies have been put out and applied in the past and determined each one's advantages and disadvantages. This chapter has also recognized the gap in the research that is needed to be fill. Through this chapter also provide a greater understanding of the state of the subject as well as it evelop the project's goals and research questions.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 INTRODUCTION**

This chapter will go to the process that was used to create the Life Planner application. SDM (System Development Methodology) is a phrase or framework for structuring, planning, and controlling the development of an information system. Choosing an appropriate methodology is critical since it gives a step-by-step guide to the process that occurs during project development to meet system requirements and effectively identify the project's success or failure since there are different process models such as Rapid Application Development (RAD), agile, and waterfall. Each of these models has its approach, sequence, and steps. This chapter also covers the system's overall architecture and the software that was utilized to create it.

#### **3.2 PROJECT MANAGEMENT FRAMEWORK**

For this project, the most suitable process model that will be chosen is Rapid Application Development (RAD) due to various reasons. Rapid Application Development (RAD) is a model that emphasizes a quick delivery based on the user's feedback instead of strict planning and a continuous requirement gathering from the user.

Quick project delivery can be done by employing RAD since a working element of a system that may be "good enough for now" can be produced as a prototype and shown to the client in a short amount of time. After the first delivery, the client will review and provide suggestions for improvements and changes to the interface or functionality. The client will be actively involved in the project to ensure that the entire development meets their vision and needs.

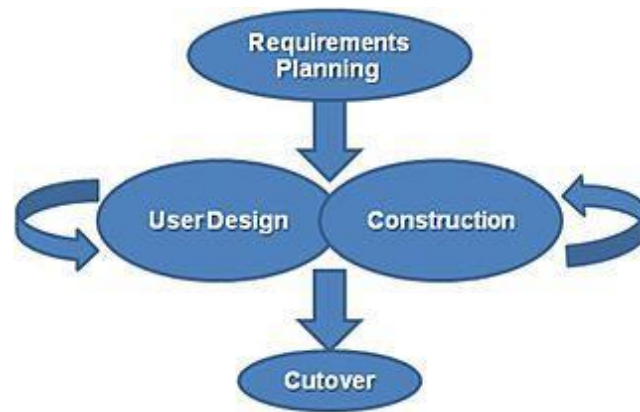


Figure 3.1 RAD Phases

According to figure 3.1, the RAD model is separated into four phases. The first one is requirement planning which is the collection of data that will be used as the confirmation for the title of the project. This phase is also used to collect the requirements from the clients to fit into the project.

The second stage is user design in which the necessary diagrams are made to have a better understanding of the project. This phase is connected to the construction phase where the developer can cycle back to the second phase. The third phase or construction phase is repeated as many as needed to fulfill the new component or alterations that have been made by the developer.

The last phase or cutover phase is where the finished product is ready to use. During this stage, the developer must plan on how to test the product and test its users in using the system [5].

### 3.3 PROJECT REQUIREMENT

The developer first brainstorms the concept and research similar applications to determine the prerequisites for building the application during this phase. From the brainstorming session, all ideas are jotted down as the initial requirement before it is filtered down to fit the project and the time given. The proposed project requirements are then brought up for discussion between the developer and the project supervisor. The development process will move to the User Design phase if the developer and project supervisor agree on the requirement.

### **3.3.1 FUNCTIONAL REQUIREMENT**

The main function of the Life Planner application is the application can take note of what the user wishes to store in the application. Users can fill in the details on the note with items such as date, title, and details. Next, users can also input an event based on the calendar interface that will be provided in the Life Planner application.

Other than that, users are also able to manage their finance in the application. Users can input their initial balance into the wallet. They can then input either the expenses or income in the application within the month. The app also can generate a report on the list of expenses and income based on the month.

### **3.3.2 NON-FUNCTIONAL REQUIREMENT**

In terms of performance, the application can run smoothly without any delay in terms of response time. Other than that, reliability is also one of the factors that are needed to be taken care of. Reliability in terms of its functionality is all the functions can work properly without any errors to the user.

Compatibility is also one of the main points as users will have different kinds of mobile as well as operating systems. For this application, it will only be compatible with the android system and android version 5.0 or above [6].

### **3.3.3 CONSTRAINTS & LIMITATIONS**

A few constraints will apply to the application. First, this application is only available for the android operating system and android version 5.0 or above. Other than that, this application function is a must for the internet to be functioning properly. For the limitations, the login function is only available with sign-up, and the google log-in method only.

## **3.4 PROJECT DESIGN**

For this section, it will discuss on the basic flow as well as the proposed design for this project.

### 3.4.1 Context Diagram

A context diagram or a level 0 Data Flow Diagram (DFD) is a chart that shows the input as well as the output to show how the data is processed [7]. The overall system data flow is only represented by one process to show how the data is interacted. Based on figure 3.1 below, there will be a data interaction between the user and the system.

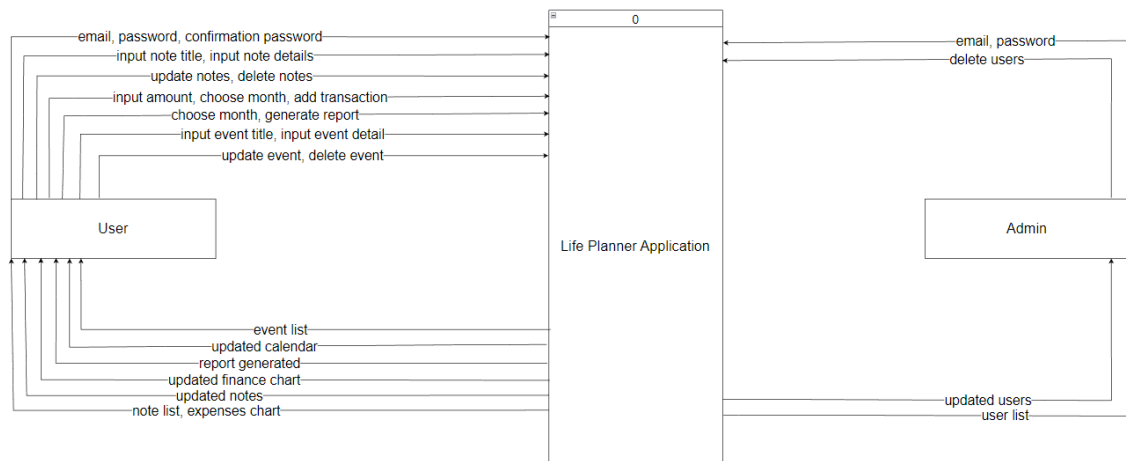


Figure 3.2 Context Diagram

### 3.4.2 Use Case Diagram

A use case diagram is one of the Unified Modelling Language (UML) which is a simple diagram to show the interaction between the possible actors and the system [8]. From figure 3.3 below, there will be 1 actor who is the User who will interact with the system based on the functions that are provided in the figure below.

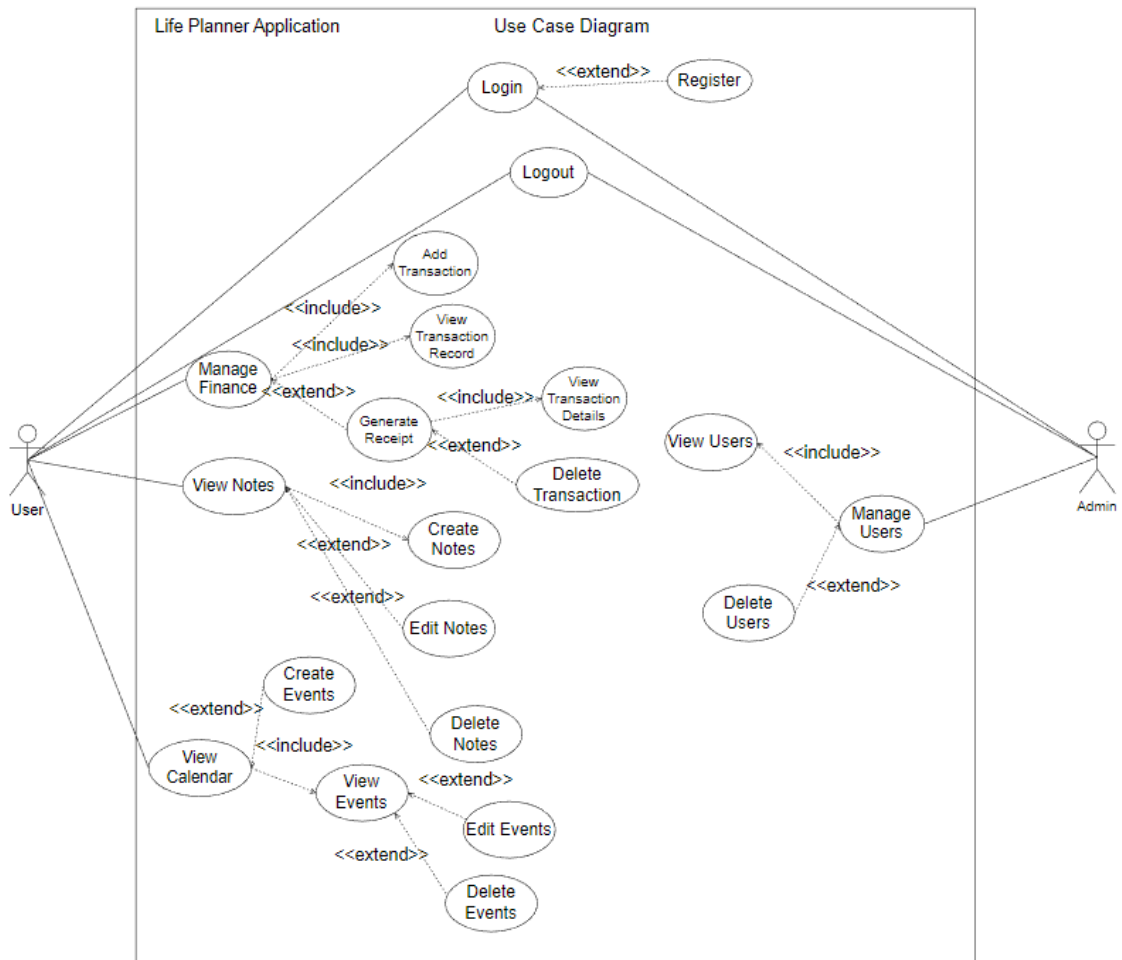


Figure 3.3 Use Case Diagram

### 3.4.3 Activity Diagram

Similar to a flowchart, activity diagram is used to represent the series of actions or flow control in a system based on the actor that is available in the system as will be show in the Figure 3.4 below [9].



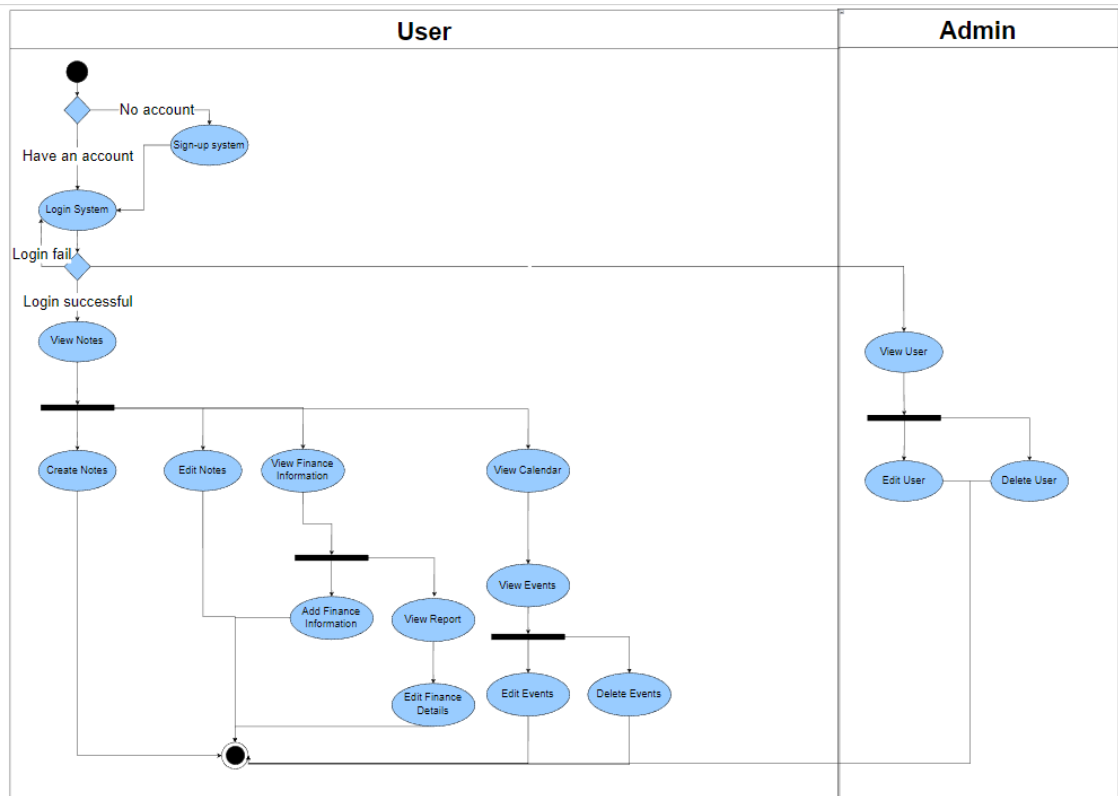


Figure 3.4 Activity Diagram

### 3.5 Data Design

In this section, the organization of the data will be explained according to the data model that will be implemented into the system [10]. Data design will give a clear view on what data type will be used, the connection between them and how will the data be organized together.

#### 3.5.1 Entity Relationship Diagram

One type of form of flowchart that shows how "entities" within a system, such as people, things, or concepts, interact with one another is an Entity Relationship Diagram (ERD) as shown as the figure below [11]. Figure 3.5 shows how the connection of database will happen in the application.

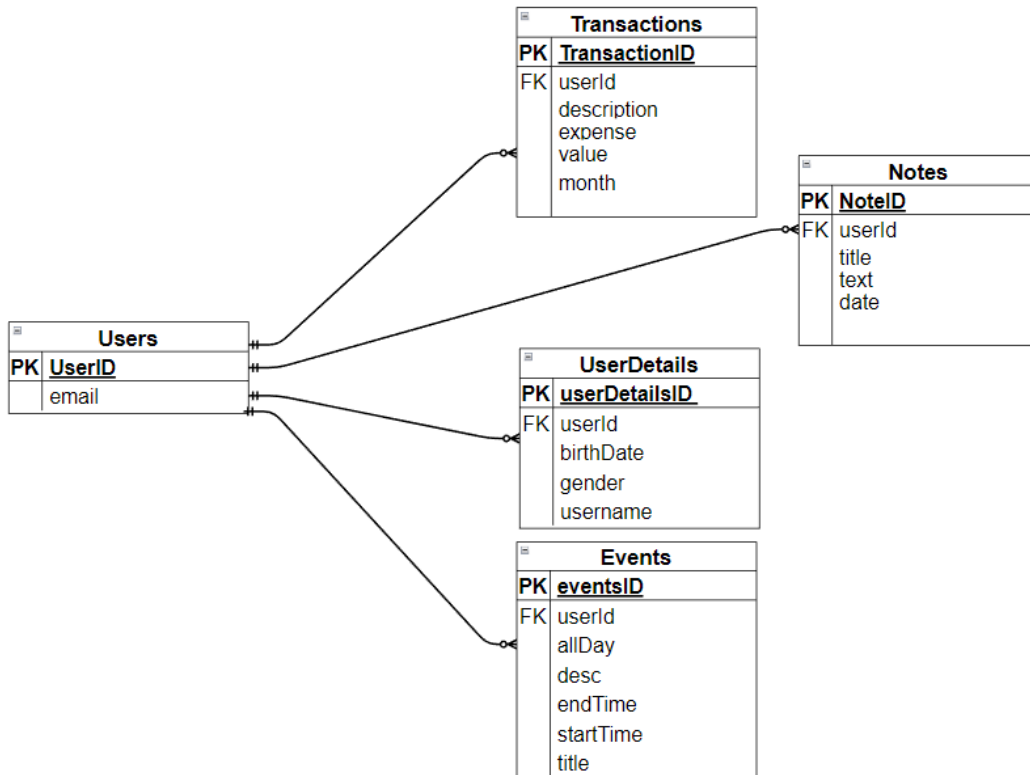


Figure 3.5 ERD Diagram

### 3.5.2 Database Dictionary

A Data Dictionary is a list of names, definitions, and properties for data elements in a database, information system, or research project as shown from the list table below [12].

#### 3.5.2.1 Users Table

Table 3.1 Users Table

Field Name	Description	Data Type	Constraint
UserID	User ID	String	PK
email	User's email for login or signup	String	-

### 3.5.2.2 Notes Table

Table 3.2 Notes Table

Field Name	Description	Data Type	Constraint
NoteID	Note's ID	String	PK
userID	User's ID	String	FK
title	Main title for a specific note	String	
text	Description for the specific note	String	
date	Date when the note is created	Date	

### 3.5.2.3 Transactions Table

Table 3.3 Transaction table

Field Name	Description	Data Type	Constraint
financeID	Finance's Id	String	PK
userId	User's Id	String	FK
description	Description for the details of the transaction that has been added by the user	String	
expense	Condition if the income is expenses	String	

	or not		
month	Month of the income that has been selected	Integer	
value	Value of the month that will be inputted	Integer	

### 3.5.2.4 UserDetails Table

Table 3.4 UserDetails Table

Field Name	Description	Data Type	Constraint
userDetailsID	User Details's ID	String	PK
userID	User Table's ID	String	FK
birthDate	Birth date for the specific user	Date	
gender	Gender of the user	String	
username	Username of the user	String	

### 3.5.2.5 Events Table

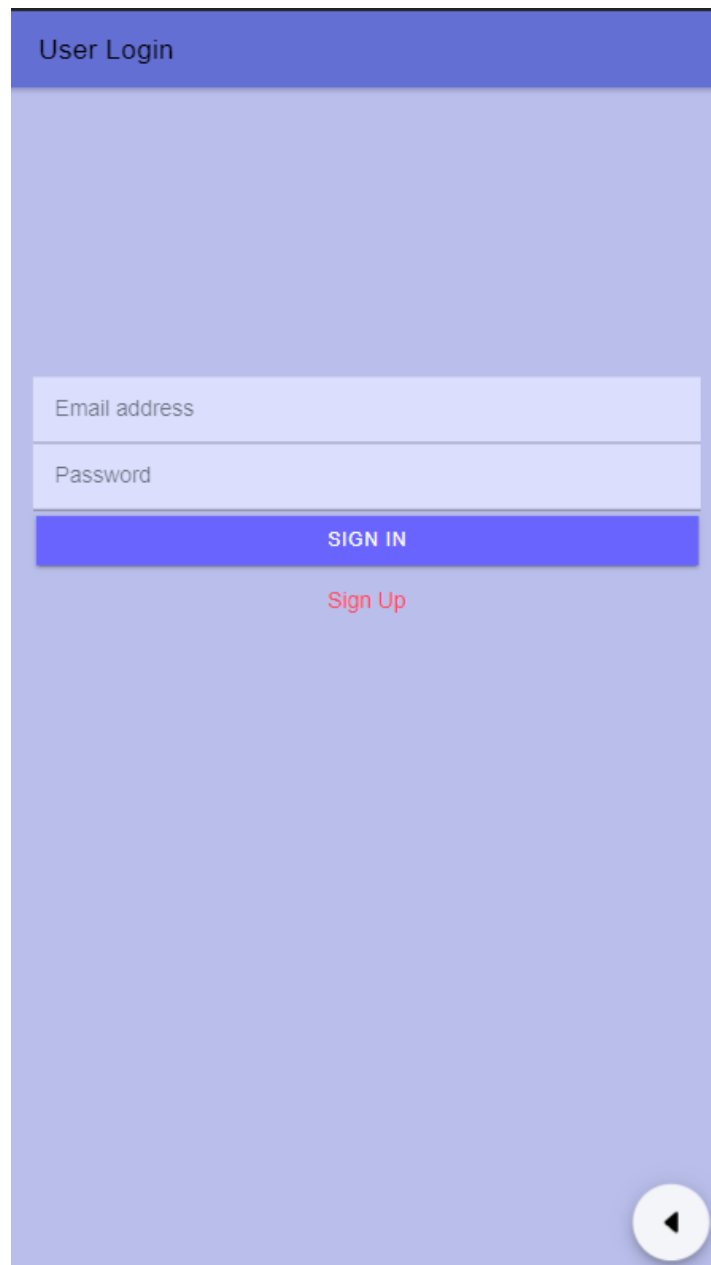
Table 3.5 Events Table

Field Name	Description	Data Type	Constraint
eventsID	Event's ID	String	PK

userID	User Table's ID	String	FK
allDay	Condition if the user's time of the event will be held all day or not	Boolean	
desc	Description of the event	String	
endTime	End time for the event		
startTime	Start time for the event	String	
title	Title for the event	String	

## 3.6 Initial Concept

### 3.6.1 Prototype Design



The image shows a mobile application prototype for a user login page. At the top, there is a dark blue header with the text "User Login" in white. Below the header is a light blue gradient background. The form consists of two input fields: "Email address" and "Password", both with light blue borders and placeholder text. Below the "Password" field is a dark blue button with the text "SIGN IN" in white. Underneath the button is a red link that says "Sign Up". In the bottom right corner, there is a white circular button with a black left-pointing arrow, likely a back button.

Figure 3.6 Login Page

The first page for the application when the system start is the login page. The user will need to input their email as well as their password. If the credential is correct, then the

user will be directed to the home page. If not, then the user will receive an error message regarding their errors. On the bottom is the widget if the user wants to login using any other social media such as Google or Facebook.

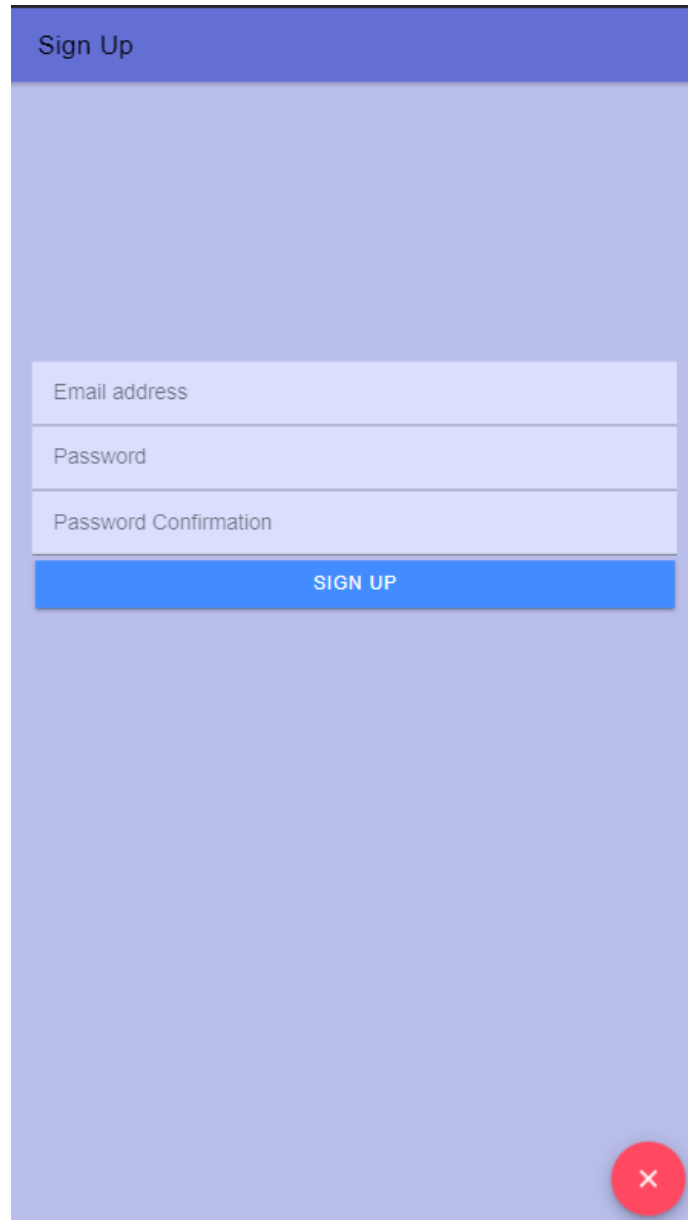
The image shows a mobile application sign-up screen. At the top, there is a dark blue header with the text "Sign Up" in white. Below the header is a light blue gradient background. In the center, there are three stacked input fields: "Email address", "Password", and "Password Confirmation". Below these fields is a blue button with the text "SIGN UP" in white. In the bottom right corner, there is a red circular button with a white "X" inside, which serves as a cancel or close button.

Figure 3.7 Sign Up Page

Figure 3.7 is the sign-up page where any user who does not have an account will be able to register. The credential that is needed is a valid email address as well as password which 6 is the minimum character that is needed to be input. If the user already has an account, then user can go back to the login page via the cancel button

below. If the user successfully entered the correct credential, then user will be directed to the login page where the user will be needed to login again.

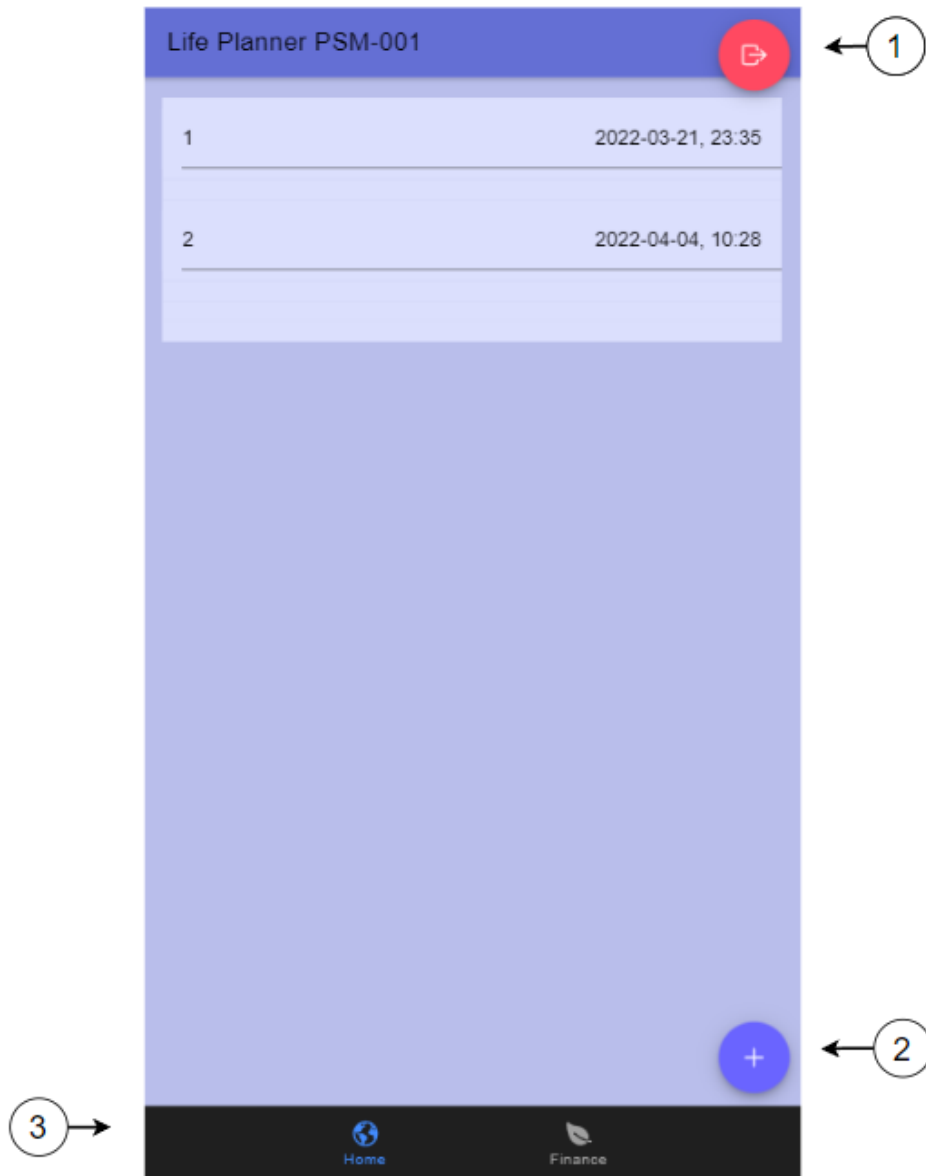


Figure 3.8 Home Page

Figure 3.8 is the home page where the user will be directed when the user successfully entered the correct credential in the login page. Home page is where all the notes will be listed in according to their time created. According to the figure above, 1 is the logout button where the user can logout and back to the login page. No 2 is for when



the user wants to create a new note by filling in the details of the note. No 3 is where all the tabs of the application, on the left is the home page and on the right is the Finance page.

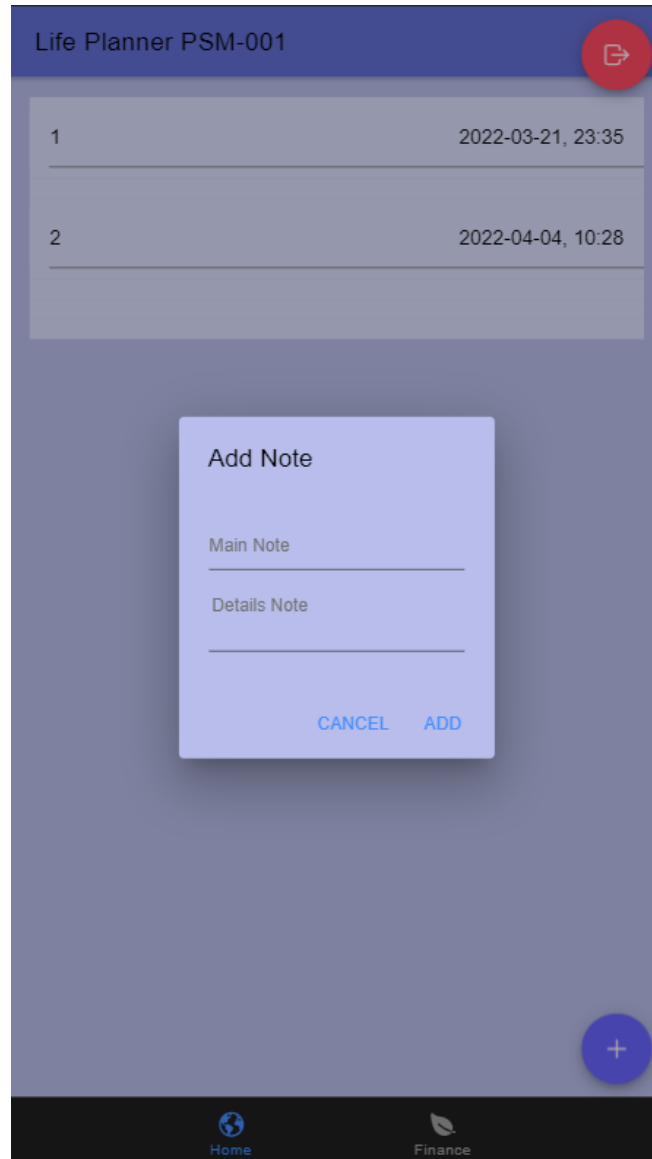


Figure 3.9 Add-Note Modal

Figure 3.9 is the Add Note modal where the user able input the new notes with their main note as well as the details of the note.

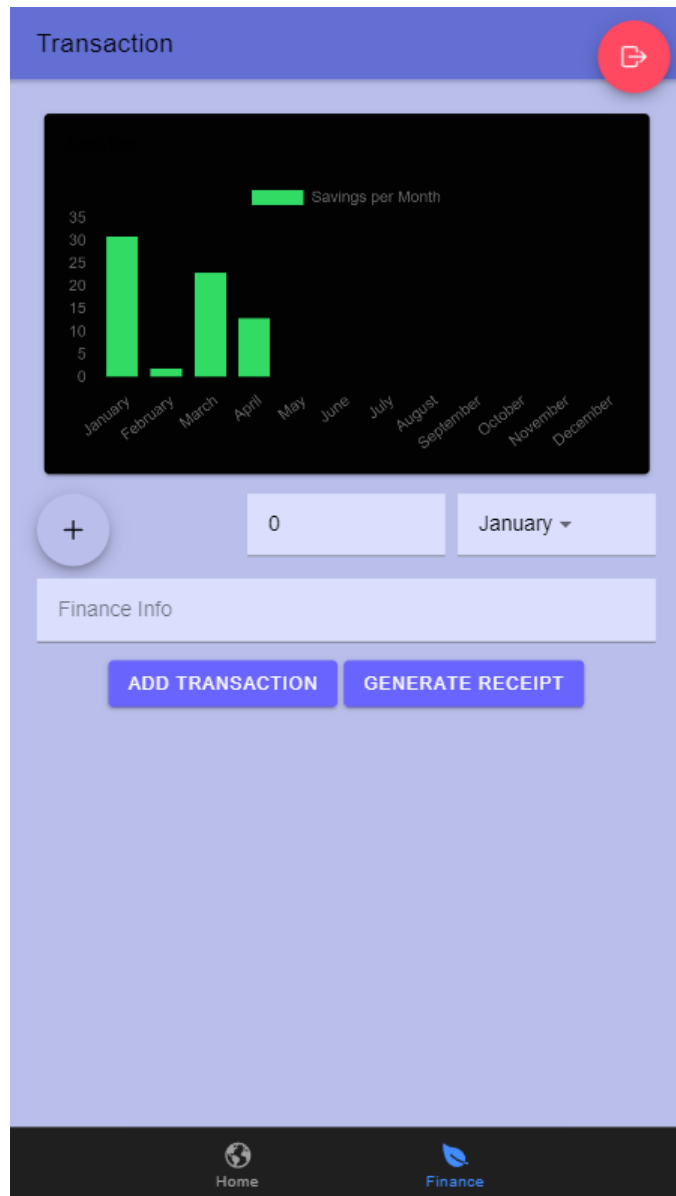


Figure 3.10 Finance Page

Figure 3.10 is the Finance Page where the user will be able to view records of the transaction throughout the whole year. Users can also input a new transaction either it is a saving or a withdraw money. Receipt can also be generated with their financial info.

### 3.6.2 Potential Use

1. Reminds user on the list of things that the users are needed to do that is sorted based on the time created.

2. The app will also be able to notify the users based on the time that had been set if the users choose for the application to notify them when they are creating or updating the system.
3. Able to track the user's finance information throughout the year with the chart that is provided in the application.
4. Users will be able to see their financial details on the particular month if the users choose to insert the details.
5. Users will be able to plan their financial income better as they will be able to see any shortage or extras in their financial life.

### 3.7 Testing Plan

Table 3.6 Testing Plan

Function	Expected Result	Pass/Fail	Comment
User registration	User with no account be able to create a new account		
Login with correct identification	User is able to access the home page with the correct credential		
Login with wrong password	User is not able to access the home page and will be alerted		
View list of notes if existed	User can view list of notes that had been stored before		
Create new notes	User can create a new note and view it in the home page		
Edit the current notes	User can edit a specific note and view it back in home page		
Delete any notes	User can delete any notes in the home page		
Add finance information on any month	User can input any amount on the finance page		

View finance chart	User can view the finance chart in according to the amount given		
View finance report on the selected month	System can generate the proper data based on what had been inputted		

This test has been performed by:

Name:

Signature:

Date:

### 3.8 Conclusion

In this chapter, the methodology that will be used were identified. The project requirements (functional and non-functional), as well as the constraints and the limitation for the application were also discussed. The project design were also reviewed in terms of UML diagram were also presented such as Context diagram, Use Case diagram, and Activity diagram. Overall, RAD methodology allowed the project to analyze the data and help the project to run smoothly.

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 Introduction

This chapter will discuss the implementation, and results or findings of the Life Planner Mobile Application. This chapter contains the result of the finding based on the testing that has been done. It also includes an explanation of the discussion that shows the objectives of the project is fulfilled.

#### 4.2 Implementation Process

##### 4.2.1 Tools and Technology



Figure 4.1 Visual Studio Code

For this project, the main editor that will be used in Visual Studio Code as the Integrated Development Environment (IDE). The apps are developed using the Ionic framework that was built on top of Angular JS. Other than that, it also combines TypeScript language that was developed and maintained by Microsoft.



Figure 4.2      Firebase

Other than that, Firebase has also been used as the database for the system. Firebase offered a collection of hosting services that may host any kind of application. It provides real-time and NoSQL database hosting, content, social authentication, and notifications, as well as other services including a real-time communication server. One of the services that have been used is their Realtime Database. Firebase Realtime Database is an API that keeps application data in the cloud and synchronizes it across iOS, Android, and Web devices. The tool supports the creation of collaborative, real-time applications by software developers.



Figure 4.3      Ionic

Using cutting-edge web development techniques and technologies including CSS, HTML5, and Sass, Ionic offers tools and services for creating hybrid mobile, desktop, and progressive web apps. By employing Cordova or Capacitor, mobile apps may be created with these Web technologies and then published through native app stores to be loaded on devices.

## 4.2.2 Development of Application

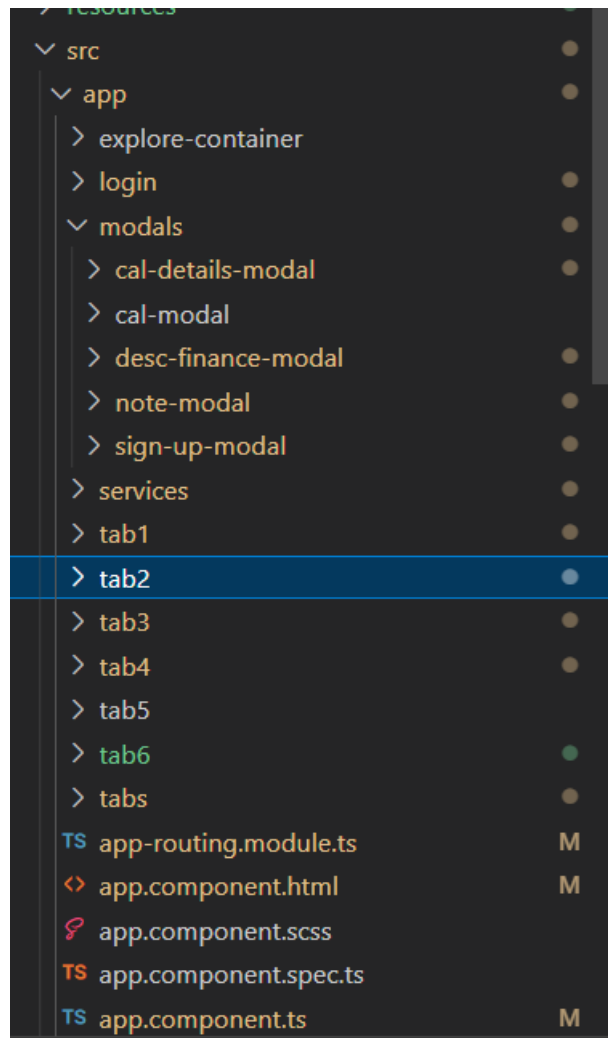


Figure 4.4 Structure of Life Planner Application

As shown in figure 4.4, these will be the structure of folders and files that will be implemented in the Life Planner application. From the login page which is the login page for the user, the list of modals which is sub-page from the main page. Tab 1 is the home page or the note page, tab 2 is the profile page, tab 3 is the finance page, tab 4 is the calendar page, tab 5 is the About Us page, and tab 6 is the admin page.

```

src > app > tab1 > < tab1.page.html > ion-header
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-buttons slot="start">
4       <ion-menu-button></ion-menu-button>
5     </ion-buttons>
6     <ion-title>
7       Note List
8     </ion-title>
9   </ion-toolbar>
10 </ion-header>
11
12 <ion-content [fullscreen]="true">
13   <ion-header collapse="condense">
14     <ion-toolbar>
15       <ion-title size="large">Note List</ion-title>
16     </ion-toolbar>
17   </ion-header>
18   <ion-list *ngFor="let note of notes">
19     <div *ngIf="(userId == note.userId)">
20       <ion-item (click)="openNote(note)">
21         <ion-label>
22           {{ note.title }}
23         </ion-label>
24         <span>{{ note.date | date:' YYYY-MM-dd, HH:mm' }}</span>
25       </ion-item>
26     </div>
27

```

Figure 4.5 tab1.page.html (Note Page)

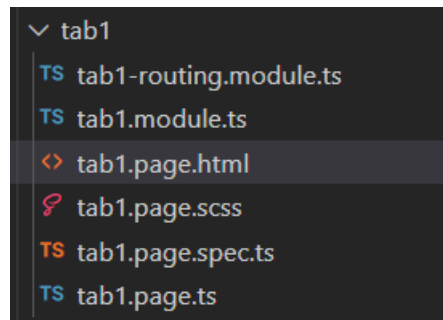


Figure 4.6 Structure on tab1

As shown in figure 4.4, Tab1 will be the home page that is implemented in the visual studio code. The home page will be the base page for the system as it will show the main functionality of the system which is the note list functionality. Next, edit and delete the notes is also implemented on the tab 1 page.

Other than that, the structures of the home page will be as shown in figure 4.5. There will be 5 different file types the first one is for the routing; the second one is for



the module; the third one is for the HTML file; the fourth one spec.ts file which is for testing purposes; the last one is .ts which is TypeScript file, a superset of JavaScript that helps in building a robust component as it offers classes, modules, and interfaces.

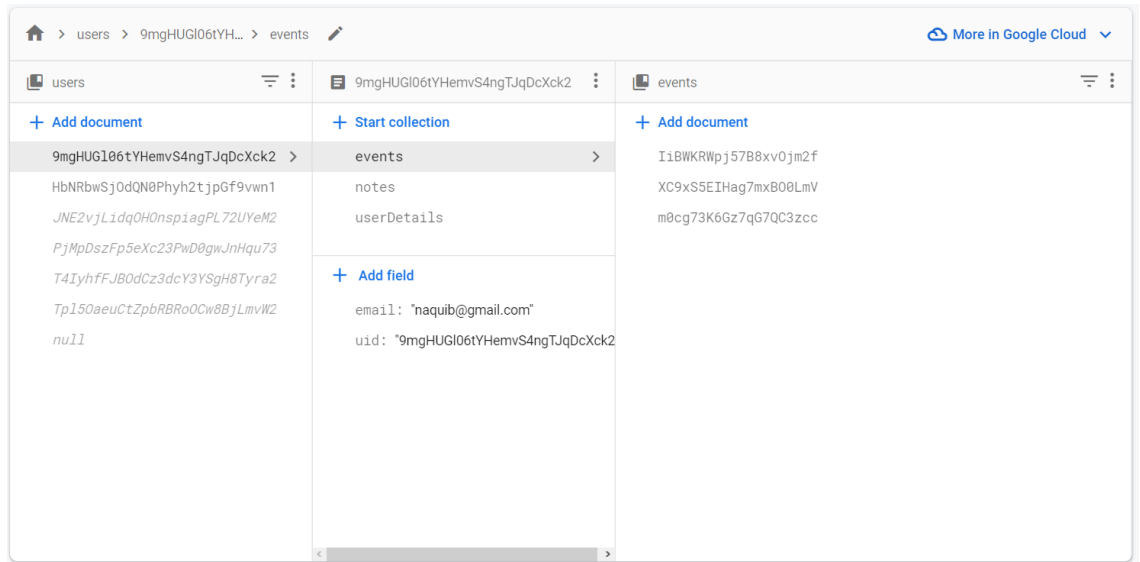


Figure 4.7 Firestore Database

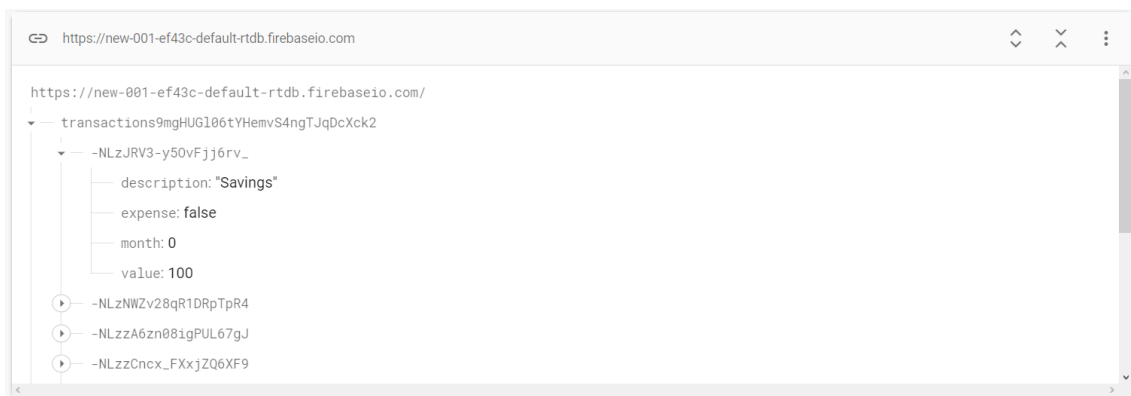


Figure 4.8 Realtime Database

The database implementation will be implemented using the service provided by Firebase as shown in figures 4.6 and 4.7. The first one that will be used is Firestore as it will store all the user data, including from events, user's details, as well as user's notes. The NoSQL data model of Cloud Firestore dictates that you store data in documents with fields mapped to values. These files are kept in collections, which are storage spaces for your files that you may use to arrange your information and create queries. Documents can include a wide range of data kinds, from straightforward characters and numbers to intricate, nested objects. Additionally, it can also develop hierarchical data structures that scale as your database expands and subcollections within documents.

Whatever data structure is most effective for your app is supported by the Firestore Database data architecture [13].

For Realtime Database will be used to store the data that is used for managing user's finance which will show their data on a graph in the application. Firebase Realtime Database is a cloud-based database where data is synchronized in real-time to every connected client and stored in JSON format. It allows for cross-platform app development by sharing one database instance among all clients using our JavaScript, Android, and Apple platform SDKs, ensuring that all clients always have the most recent data available [14].

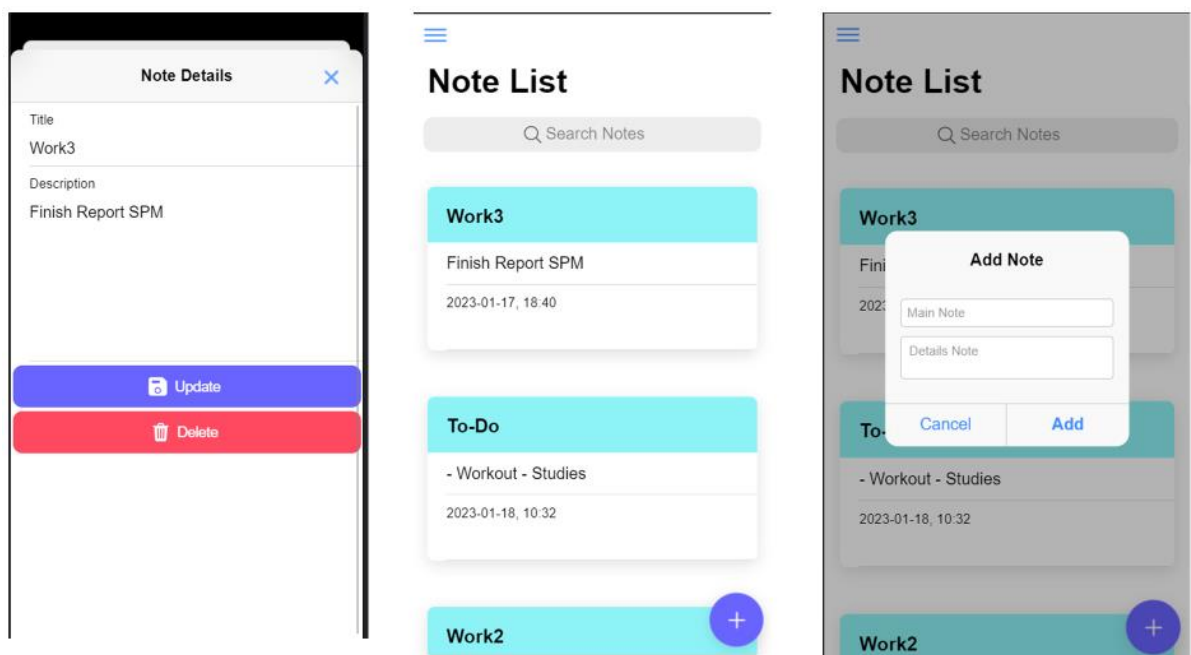
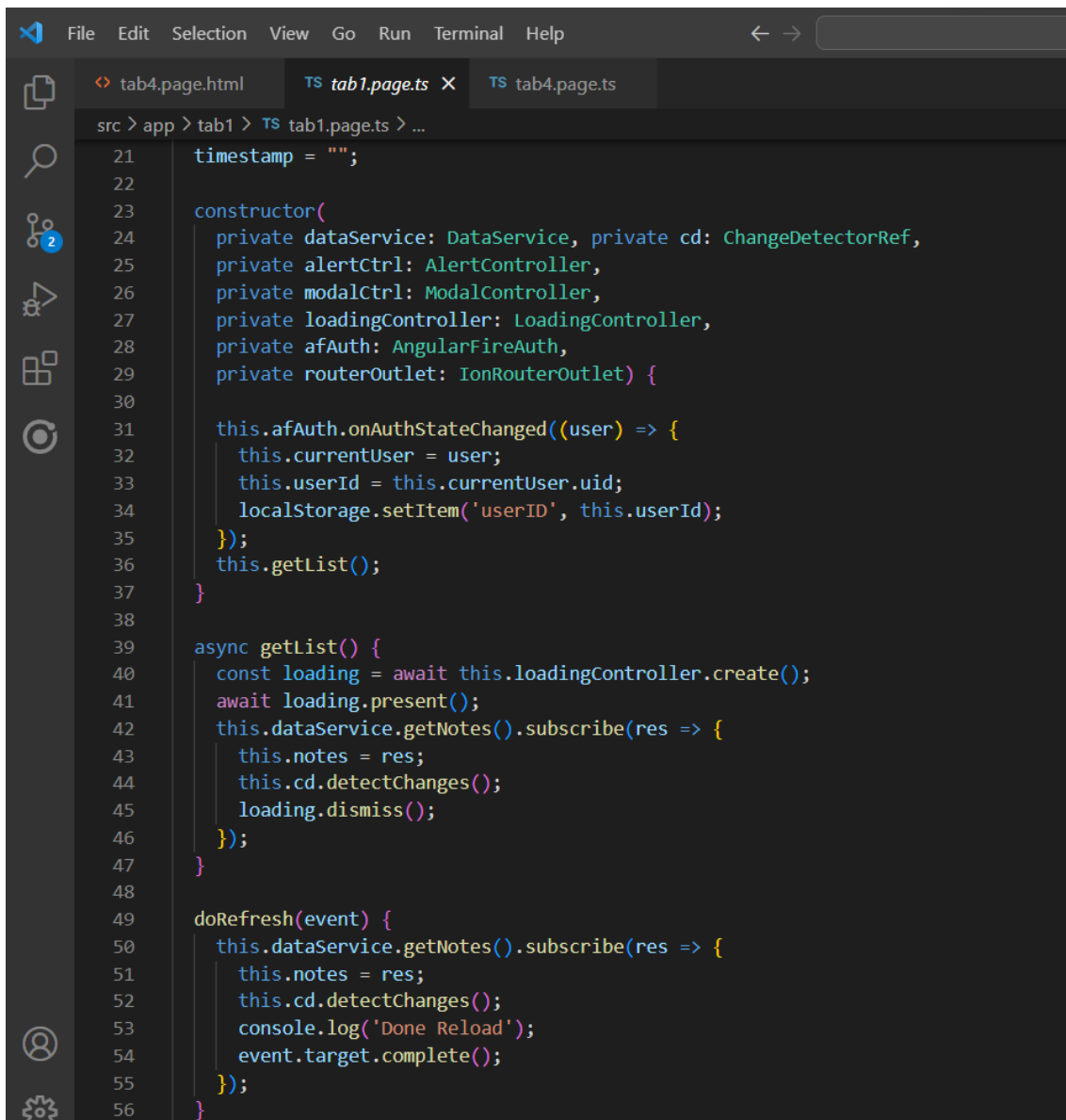


Figure 4.9 List of Pages for Note Function



```
src > app > tab1 > TS tab1.page.ts > ...
21     timestamp = "";
22
23     constructor(
24         private dataService: DataService, private cd: ChangeDetectorRef,
25         private alertCtrl: AlertController,
26         private modalCtrl: ModalController,
27         private loadingController: LoadingController,
28         private afAuth: AngularFireAuth,
29         private routerOutlet: IonRouterOutlet) {
30
31         this.afAuth.onAuthStateChanged((user) => {
32             this.currentUser = user;
33             this.userId = this.currentUser.uid;
34             localStorage.setItem('userId', this.userId);
35         });
36         this.getList();
37     }
38
39     async getList() {
40         const loading = await this.loadingController.create();
41         await loading.present();
42         this.dataService.getNotes().subscribe(res => {
43             this.notes = res;
44             this.cd.detectChanges();
45             loading.dismiss();
46         });
47     }
48
49     doRefresh(event) {
50         this.dataService.getNotes().subscribe(res => {
51             this.notes = res;
52             this.cd.detectChanges();
53             console.log('Done Reload');
54             event.target.complete();
55         });
56     }
```

Figure 4.10 Coding on Note Page

As shown in figure 4.9 and 4.10, these are the list of pages and modals that will be used to implement the note function. It is implemented in Tab 1, Note Modal, as well as the Data Service. List of functionalities for these pages are, view, create, and edit notes. Search or filter notes function is also implemented into the page as it will ease the user in finding the specific noted that is required by the user.

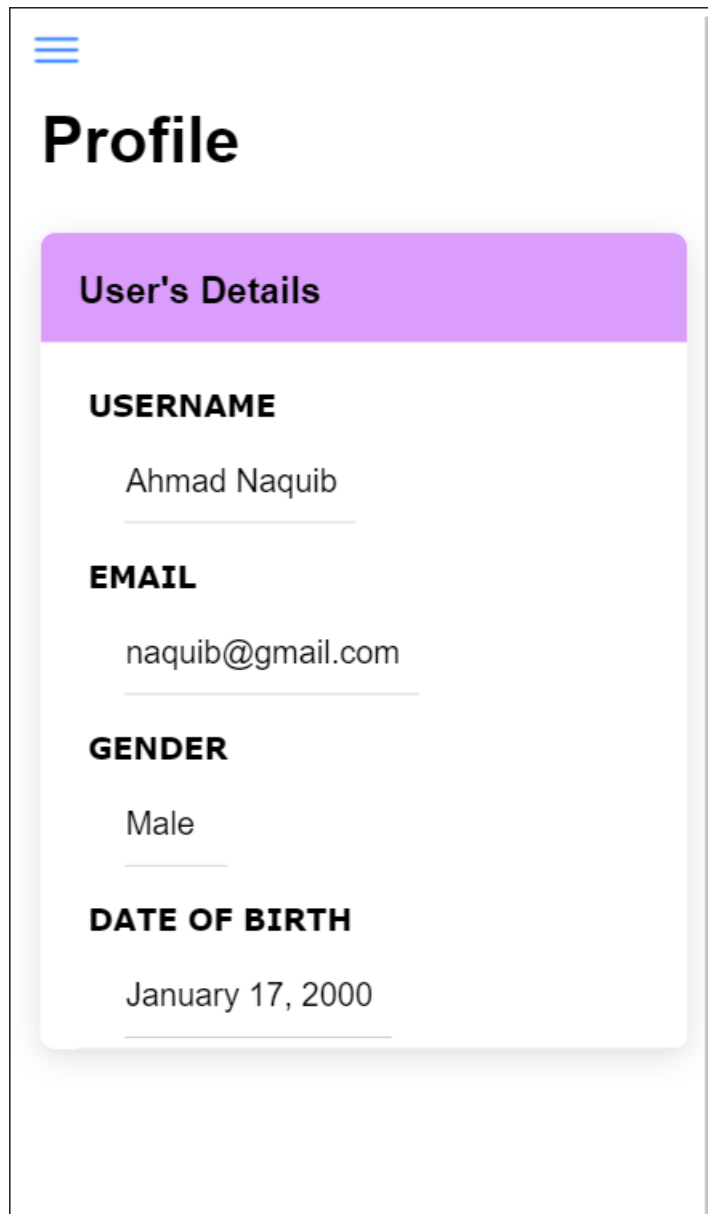


Figure 4.11 Profile Page

```
File Edit Selection View Go Run Terminal Help PSM_1
tab4.page.html TS tab2.page.ts X TS tab4.page.ts
src > app > tab2 > TS tab2.page.ts > ...
14   currentUser: User = null;
15   userDetails = [];
16   public userId: string = null;
17
18   constructor(
19     private userDataService: UserDataService,
20     private userService: AuthFirebaseService,
21     private loadingController: LoadingController,
22     private db: AngularFirestore,
23   ) {
24   }
25
26   ngOnInit() {
27     this.userId = localStorage.getItem('userID');
28     this.userDataService.getUserById(this.userId).subscribe(res => {
29       this.currentUser = res;
30       console.log('user: '+this.currentUser);
31     });
32     this.getUser();
33   }
34
35   async getUser(){
36     const loading = await this.loadingController.create();
37     await loading.present();
38
39     this.db.collection(`users/${this.userId}/userDetails`).snapshotChanges().subscribe(colSnap => {
40       colSnap.forEach(snap => {
41         let event: any = snap.payload.doc.data();
42         event.id = snap.payload.doc.id;
43         event.gender = event.gender;
44         event.birthDate = event.birthDate;
45         this.userDetails.push(event);
46       });
47       loading.dismiss();
48     });
49   }
```

Figure 4.12 Coding on Profile Page

As shown in figure 4.11 and 4.12, this is the page where the user will be able to see their personal details after the user has successfully sign-up and login into the system.

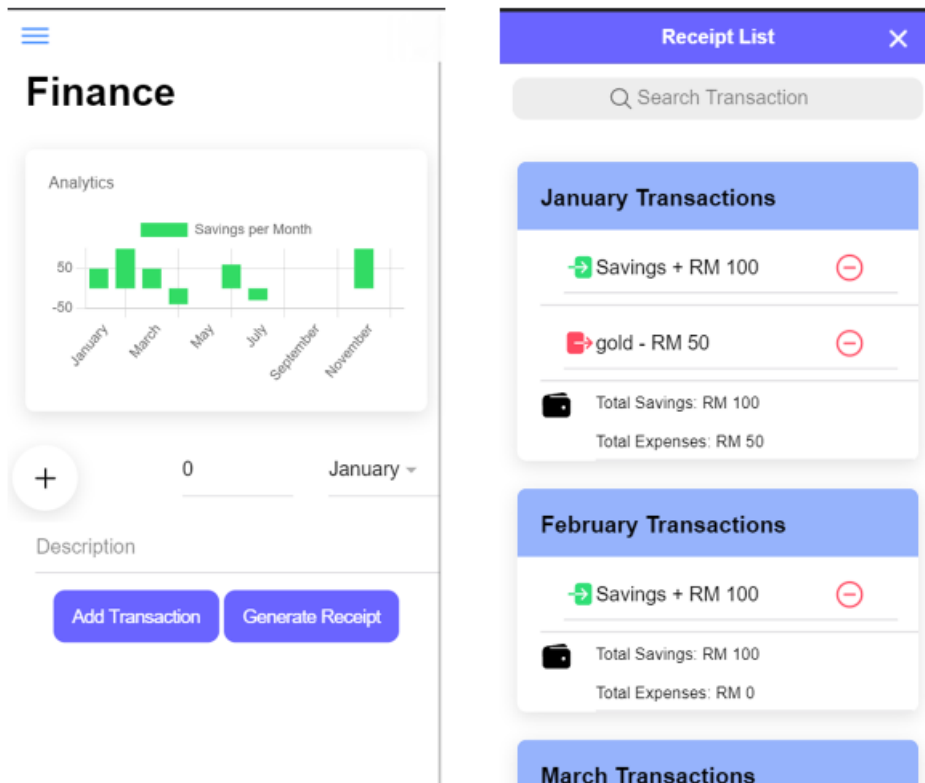


Figure 4.13 List of Pages for Finance Function

```

src > app > tab3 > TS tab3.page.ts > ...
54   private modalCtrl: ModalController
55   ) { }
56
57   ngAfterViewInit() {
58     let userId = localStorage.getItem('userID');
59
60     // Reference to our Firebase List
61     this.ref = this.db.list('transactions' + userId, ref => ref.orderByChild('month'));
62
63     // Catch any update to draw the Chart
64     this.ref.valueChanges().subscribe(result => {
65       if (this.chartData) {
66         this.updateCharts(result)
67       } else {
68         this.createCharts(result)
69       }
70     })
71
72     this.isHidden = true;
73     let date = Date.now();
74     let formattedDate = (moment(date)).format('MMMM DD, YYYY HH:MM');
75     this.today = formattedDate;
76     console.log("today: "+this.today);
77   }
78
79   async addTransaction() {
80     this.ref.push(this.transaction).then(async () => {
81       this.transaction = {
82         value: 0,
83         month: 0,
84         expense: false,
85         description: null,
86         date: null
87       };
88       let toast = this.toastCtrl.create({
89         message: 'New Transaction added',
90         duration: 3000

```

#### Figure 4.14 Coding for Finance Page

As shown in figure 4.13 & 4.14, these are the interface that will be implement into the Life Planner application. It is implemented in Tab 3 and DescFinance modal. This does not have any service as the function is directly implemented in the .ts file. The list of function that is implemented are view chart transaction, add transaction, generate receipt, view list of transactions, search for a specific transaction, and delete a specific transaction.

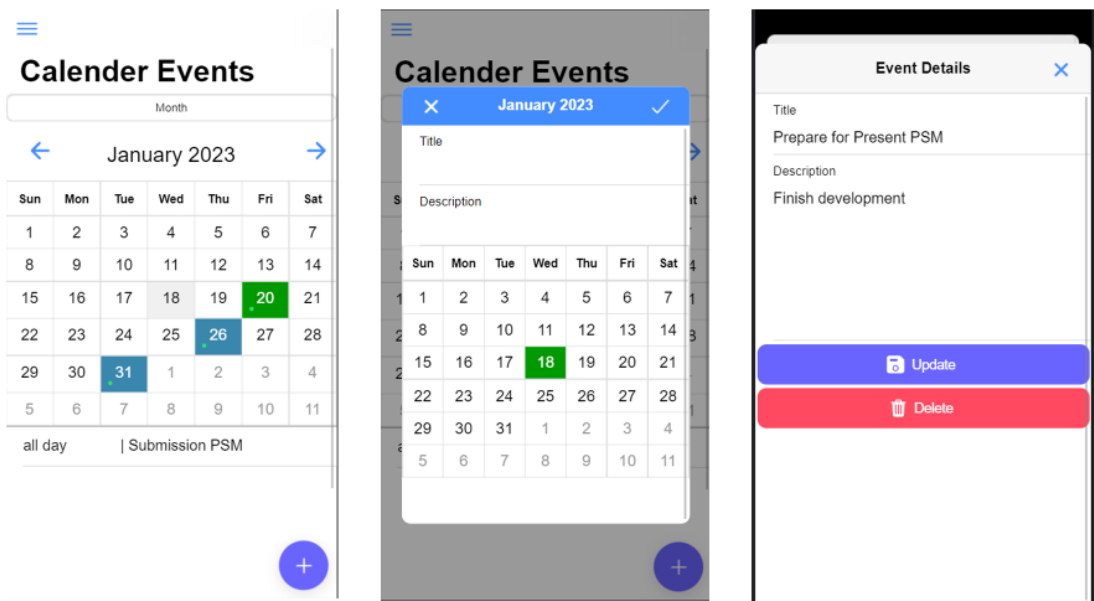


Figure 4.15 List of Pages for Event Function

```

src > app > tab4 > TS tab4.page.ts > ...
24
25 @ViewChild(CalendarComponent) myCal: CalendarComponent;
26
27 constructor(
28   private db: AngularFirestore,
29   private loadingController: LoadingController,
30   private modalCtrl: ModalController,
31   private routerOutlet: IonRouterOutlet,
32 ) {
33   this.getEvents();
34 }
35
36 async getEvents() {
37   const loading = await this.loadingController.create();
38   await loading.present();
39
40   this.db.collection(`users/${this.uid}/events`).snapshotChanges().subscribe(colSnap => {
41     this.eventSource = [];
42     colSnap.forEach(snap => {
43       let event: any = snap.payload.doc.data();
44       event.id = snap.payload.doc.id;
45       event.startTime = event.startTime.toDate();
46       event.endTime = event.endTime.toDate();
47       this.eventSource.push(event);
48     });
49     loading.dismiss();
50   });
51 }
52
53 next() {
54   this.myCal.slideNext();
55 }
56
57 back() {
58   this.myCal.slidePrev();
59 }
60

```

Figure 4.16 Coding for Event Page



As shown in figure 4.15 & 4.16, these are the interfaces for the event functionality. It is implemented in the Tab 4, Cal Modal, CalDetails Modal, as well as the CalData Service. The list of functionality that is implemented is the view calendar as well as the event, view list of events that is in the specific date, add event, edit event, and delete event.



# About Us

## Life Planner

### DESCRIPTION

Life planning is not a simple task to do as it can vary from person to person. Things such as how can a person manage their times, plan their days, as well as how they manage their money throughout the month or year. Mobile application is one of the things that can help the users in organizing all of that. A mobile application has its categories as the topic is huge and it has been expanding throughout the decades.

One of the categories is a productivity mobile app. As the name implies, it may appear to be one of the more mundane categories of mobile apps, it is a hugely popular one. These apps will be able to assist their users in completing a task swiftly and efficiently, making what can tedious activities easier and more

Figure 4.17 About Us Page

```
src > app > tab5 > tab5.page.html > ion-header
8     </ion-title>
9     </ion-toolbar>
10  </ion-header>
11
12  <ion-content [fullscreen]="true">
13    <ion-header collapse="condense">
14      <ion-toolbar>
15        <ion-title size="large">About Us</ion-title>
16      </ion-toolbar>
17    </ion-header>
18
19    <ion-card>
20      <ion-card-header>
21        <ion-card-title>Life Planner</ion-card-title>
22        <ion-card-subtitle>Description</ion-card-subtitle>
23      </ion-card-header>
24
25      <ion-card-content>
26        Life planning is not a simple task to do as it can vary from person to person.
27        Things such as how can a person manage their times, plan their days, as well as
28        throughout the month or year.
29        Mobile application is one of the things that can help the users in organizing their
30        application has its categories as the topic is huge and it has been expanded.
31      </ion-card-content>
32
33      <ion-card-content>
34        One of the categories is a productivity mobile app. As the name implies, it is a
35        mundane categories of mobile apps, it is a hugely popular one.
36        These apps will be able to assist their users in completing a task swiftly and
37        tedious activities easier and more enjoyable.
38      </ion-card-content>
39    </ion-card>
40
41    <ion-card>
42      <ion-card-header>
43        <ion-card-title>Contact Us</ion-card-title>
44        <ion-card-subtitle>Feel free to do so :)</ion-card-subtitle>
```

Figure 4.18 Coding for Profile Page

As shown in figure 4.17 & 4.18, about us page is created to share the objectives of the application as well as what Life Planner is to the user. Other than that, this page also shares on ways to contact the developer if the user has any enquiries on the system.








 Admin	
Email: az@gmail.com	
Email: naquib@gmail.com	
Email: adam2@gmail.com	
Email: raes@gmail.com	
Email: n3@gmail.com	
Email: george@gmail.com	

Figure 4.19 Admin Page

```
src > app > tab6 > TS tab6.page.ts > Tab6Page > deleteUser > alert > buttons > handler
14 export class Tab6Page implements OnInit {
15
16   users: User[] = [];
17   public userId: string = null;
18   handlerMessage = '';
19
20   constructor(
21     private userDataService: UserDataService,
22     private toastCtrl: ToastController,
23     private alertController: AlertController,
24     private router: Router,
25     private tabService: TabBarService
26   ) {
27     this.router.events.pipe(
28       filter((event: RouterEvent) => event instanceof NavigationEnd)
29     ).subscribe((event: RouterEvent) => {
30       this.tabService.setTabVisibility(event);
31     });
32   }
33
34   ngOnInit() {
35     let userId = localStorage.getItem('userID');
36     this.userDataService.getUsers().subscribe(res => {
37       this.users = res;
38     });
39   }
40 }
41
42 async deleteUser(user) {
43   // console.log(user);
44   const alert = await this.alertController.create({
45     header: 'Delete User?',
46     buttons: [
47       {
48         text: 'Cancel',
49         role: 'cancel',
50         handler: () => {
```

Figure 4.20 Coding for Admin Page

As shown in figure 4.19 & 4.20, this will be the admin page which is only accessible by the admin itself. As the admin will be able to view all the users that is registered in the system. The admin can also delete a specific user if necessary.

### 4.3 User Manual

In general, Life Planner mobile application is divided into 4 modules which will be explained in this sub-chapter in an order of new user with the interface. There are also other functions that will be added to ease the user.

### 4.3.1 User Manual: Login & Register Module

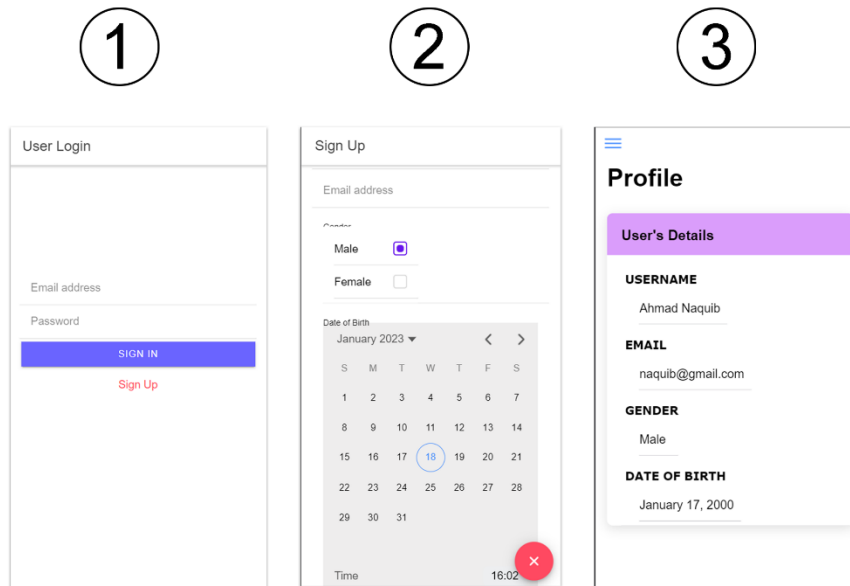


Figure 4.21 User Manual: Login & Register Module

The first module which is the Login & Register as shown from figure 4.21 is the first page which will be show to the user after the user managed to download the application.

Table 4.1 User Manual: Login & Register Module

User Manual: Login & Register Module	
Page No	Description
1	Login Page
2	Sign-up Page
3	Profile Page
Manual Instruction	

1. System will show the login page and user must input their credentials if the user already registered.
2. If the user is not registered, the user will have to click on the register button and the system will show page no 2(Sign-up page) where the user needs to fill in their credentials.
3. After successfully sign-up the user will be redirect back to page 1 and repeat step no 1.
4. After successfully sign-in, the user will be directed to the Note page as shown in figure 4.23.

### 4.3.2 User Manual: Side Navigation

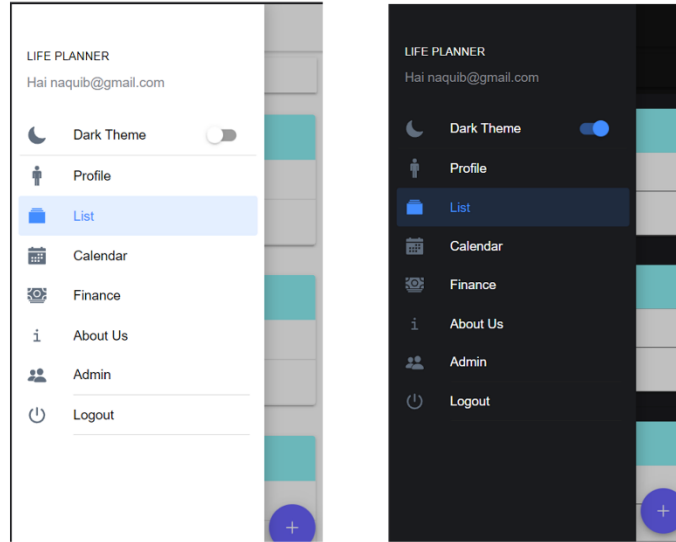


Figure 4.22 User Manual: Side Navigation

Table 4.2 User Manual: Side Navigation

User Manual: Side Navigation	
Title	Description
Dark Theme	Option for user to apply dark-theme for whole application
Profile	Direct user to profile page where the user's information will be shown as shown figure 4.21 (no 3).
List	Direct user to note page as shown in figure 4.23 (no.



	1) where all of the user's notes will be shown.
Calendar	Direct user to calendar page as shown in figure 4.24 (no. 1) where a calendar will be shown as well as user's events.
Finance	Direct user to finance page as shown in figure 4.25 (no. 1) where a chart of user's transaction will be shown.
About Us	Direct user to about us page where the application general information will be shown to the user.
Admin	Direct admin to the admin page where a list of user's email will be shown and the user will be given the option to delete a specific user.
Logout	Direct user back to the login page as shown in figure 4.21 (no. 1).

### 4.3.3 User Manual: Manage Note Module

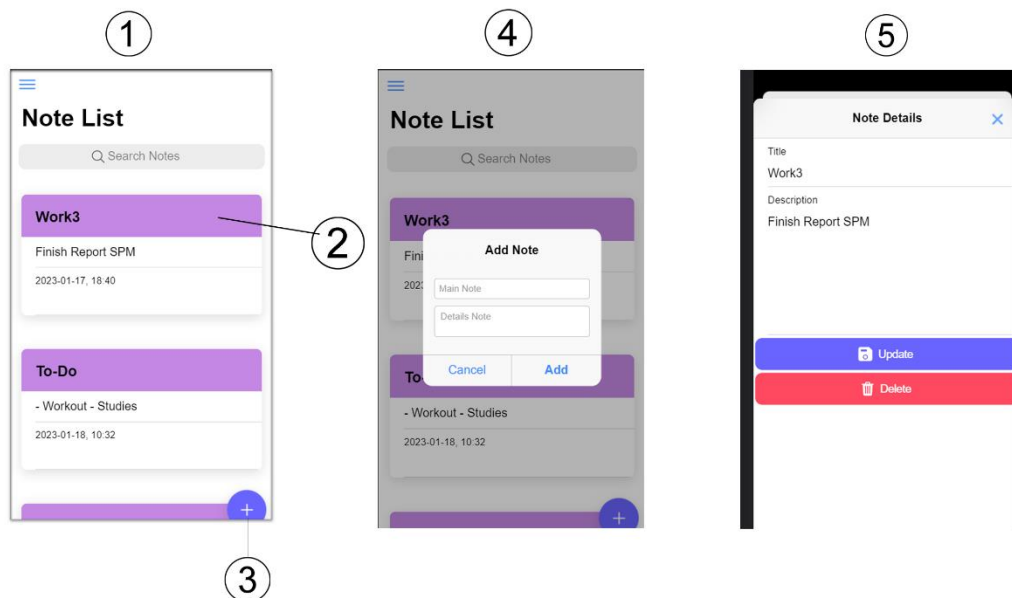


Figure 4.23 User Manual: Note Module

Table 4.3 User Manual: Note Module

User Manual: Note Module	
No	Description
1	Note page
2	User's Note
3	Add Button
4	Add note modal
5	Edit note modal
Manual Instruction	
1.	After user login into the system, system will show No.1 (Note page) where all the user's note will be shown.
2.	The user can click on No.2 (User's Note) where the user will be direct to No.5 (Edit note modal). After the user complete edit the note, user will be direct back to No.1.
3.	The user can also click on No.3 (Add note button) where the user will be direct to No.4 (Add note modal). After the user successfully add a note, user will be direct back to No.1.

### 4.3.4 User Manual: Manage Event Module

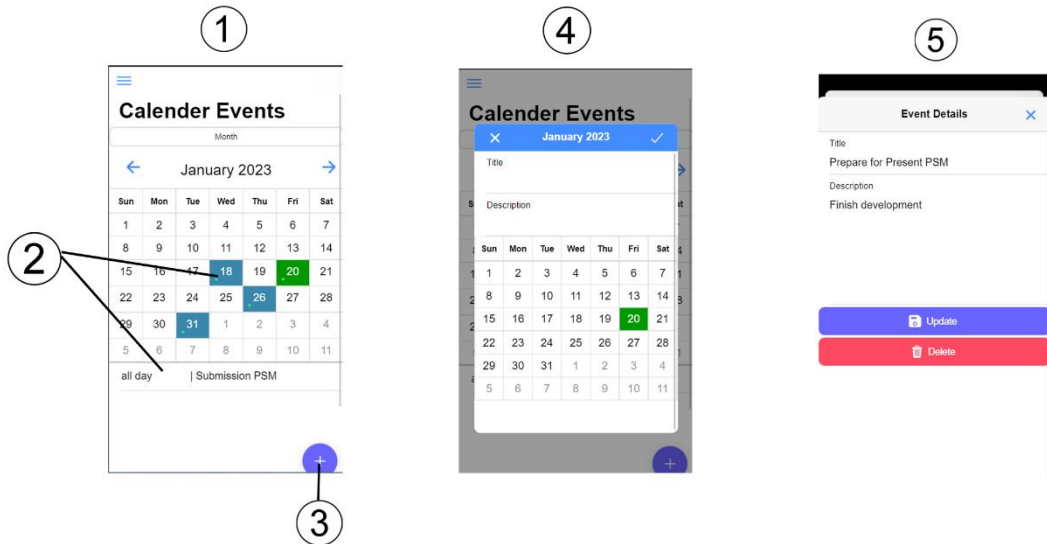


Figure 4.24 User Manual: Manage Event Module

Table 4.4 User Manual: Manage Event Module

User Manual: Manage Event Module	
No	Description
1	Event page
2	Event Details
3	Add Button
4	Add event modal
5	Edit event modal
Manual Instruction	

1. The user can access this page via side navigation on the calendar icon and the system will show Event page (No.1).
2. The user can add an event by clicking add button (No.3) and Add event modal (No.4) will be shown.
3. Users can add events through the modal and the user will redirect to event page (No.1)
4. The user also can click on the specific date on the calendar to view the details. User also can view Edit event modal (No.5) by clicking the specific event at the bottom of the page.

### 4.3.5 User Manual: Manage Finance Module

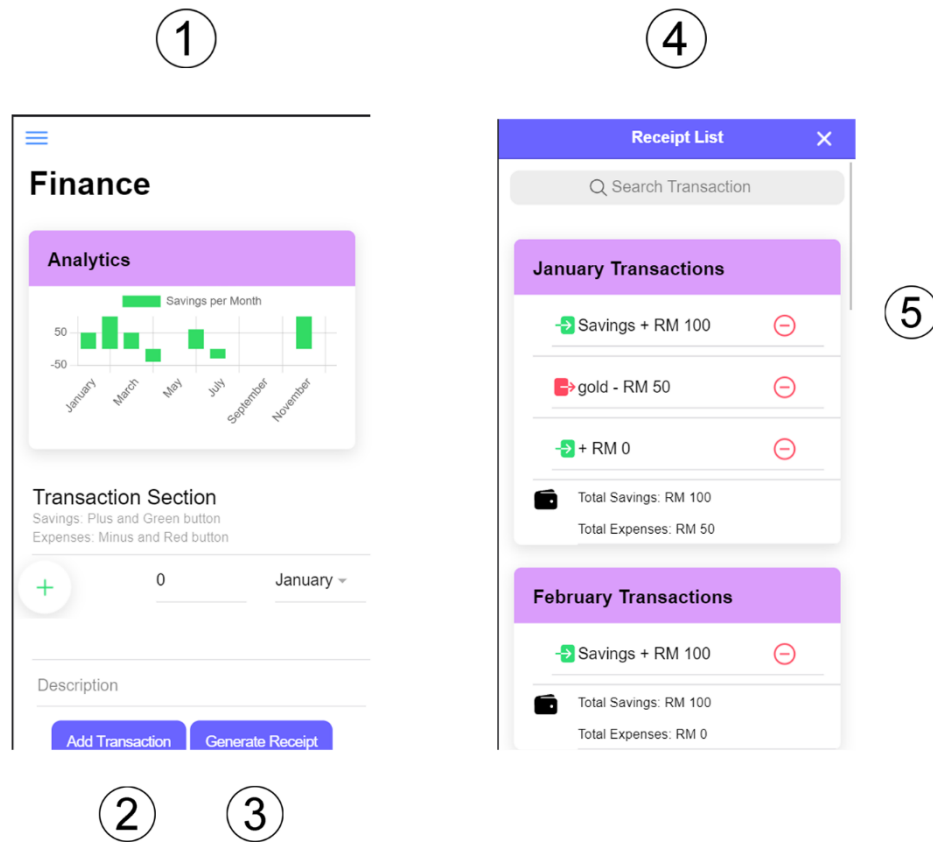


Figure 4.25 User Manual: Manage Finance Module

Table 4.5 User Manual: Manage Finance Module

User Manual: Manage Finance Module	
No	Description
1	Finance page
2	Add transaction button
3	Generate receipt button
4	Transaction list modal

5	Transaction detail
Manual Instruction	
<ol style="list-style-type: none"> <li>1. The user can access this page via side navigation on the money icon and the system will show Finance page (No.1).</li> <li>2. The user can add either an expense or saving transaction by changing the plus and minus button.</li> <li>3. After inputting other details, user can add the transaction by clicking add transaction button (No.2) the transaction will be updated in the chart.</li> <li>4. Users can view the transaction list modal (No.4) by clicking the generate receipt button (No.3).</li> <li>5. The user can view a specific transaction (No.5) and users also can delete that specific transaction.</li> </ol>	

#### **4.4 Testing and Result Discussion**

##### **4.4.1 User Acceptance Test**

The final stage of this application testing procedure is the user acceptance test (UAT). Actual users test the program to ensure that it can do the needed activities in a real setting and in accordance with requirements during UAT. UAT, in general, entails using the program by members of the target audience while documenting and fixing any flaws that are found. It allows users the chance to interact with the software and determine whether everything functions as it should or whether any features have been missed or improperly explained, for example.

This section serves to describe the system's User Acceptance Testing (UAT) procedure. The approval of this testing signifies that the reviewers are sure that the final system User will be regarded as completely tested and qualified for installation after the test plan has been carried out. The Appendix A file contains the whole UAT as it has been distributed through the google form.

#### **4.4.2 Result Discussion**

After User Acceptance Test (UAT) is done, the results are going to be analyzed and discussed to ensure the Life Planner Application has met the requirements and objectives. From the UAT, the non-registered tester is unable to log in because the tester does not have an account. The account cannot be login or created if the required field is left blank. The tester can sign up and log in to the account, fill in the profile, view the home page, add a note, view notes, edit notes, view calendar, view events, add event, edit event, add a transaction, deduct a transaction, view a transaction, and view charts and lists of transactions. Based on the result from UAT, the application was able to meet the requirement and objectives of this project. This can be seen through appendix B. In terms of design, the application however may need some improvements.

#### **4.5 Conclusion**

In this chapter, the findings of the research as well as the testing or implementation of the Life Planner mobile application on several users were shown. The implementation process using what tools and technology as well as the development of application were also discussed in order to have a better understanding of the application. User manual was also presented to let the users informed on ways to use the application. Based on the User Acceptance Test, the result showed that the application was able to run successfully although with a little bit of bug in the application.

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 Introduction**

The project and research that were completed for the Life Planner (LP) mobile app are summarised in this chapter. It will provide an overview of the research's key findings and the project's implications for the findings. Additionally, it will talk about the obstacles this project experienced when it was being developed. In using this application, users were able to improve their lifestyle as well as they have a better management in terms of their finance. The RAD technique was used to design this application. In conclusion, the Life Planner (LP) project is a good fit for the RAD process model because it enables speedy delivery based on user feedback, active client involvement, flexibility and adaptability to changing requirements, and efficient development. Finally, in this chapter also will provide recommendations for prospective topics for additional research as well as future upgrades and enhancements to the project.

#### **5.2 Research Constraint**

These are the few constraints that were faced in doing this project:

1. **Time constraints:** A mobile app requires a lengthy and difficult development procedure. The research and development of the app might only have a short window of time, which could influence the project's scope.
2. **Budget constraints:** A mobile app can be expensive to develop, and there might be financial restrictions that limit the resources the project can use.



3. Technical constraints: The development of the app might be subject to technical restrictions, such as restrictions on the technology or programming languages that can be employed.
4. Platform constraints: The Life Planner (LP) mobile app will be built for a variety of platforms, including Android, iOS, and web applications; this may entail more development time and resources.

### **5.3 Future Work**

These are the several enhancement or future updates for this project in the future:

1. Integration with other apps and services: To give consumers a more seamless experience, the app might be integrated with other well-known apps and services like calendars, to-do lists, and financial management tools.
2. Code Optimization: To give a better performance to the application. It also can provide a better quality to the user in terms of its deliverability to the user.
3. Offline functionality: The software might be improved so that users could access and use it even when they weren't online.
4. Multi-language support: To reach a larger audience, the software could be translated into additional languages.

### **5.4 Conclusion**

In this chapter, the conclusion of the project as well as the research were shown. This chapter summarizes the research constraint and the challenges that the developers were having throughout the building of this application. The future works were also identified as there were lots of ways that the application can be improved on.



## REFERENCES


1. AltexSoft. 2022. Non-functional Requirements: Examples, Types, How to Approach. [online] Available at: <<https://www.altexsoft.com/blog/non-functional-requirements/>>
2. Play.google.com. 2022. [online] Available at: <<https://play.google.com/store/apps/details?id=com.sgcdeveloper.moneymanager&hl=en&gl=US>>
3. Play.google.com. 2022. [online] Available at: <<https://play.google.com/store/apps/details?id=com.appxy.planner>>
4. Play.google.com. 2022. [online] Available at: <<https://play.google.com/store/apps/details?id=com.socialnmobile.dictapps.notepad.color.note>>
5. Guru99. 2022. *What is RAD Model? Phases, Advantages and Disadvantages.* [online] Available at: <<https://www.guru99.com/what-is-rad-rapid-software-development-model-advantages-disadvantages.html>> [Accessed 14 April 2022].
6. Techopedia.com. 2022. *What is a Mobile Application? - Definition from Techopedia.* [online] Available at: <<https://www.techopedia.com/definition/2953/mobile-application-mobile-app>>.
7. Modernanalyst.com. 2022. *What is a Context Diagram and what are the benefits of creating one?.* [online] Available at: <<https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/1433/What-is-a-Context-Diagram-and-what-are-the-benefits-of-creating-one.aspx>>

8. Visual-paradigm.com. 2022. *What is Use Case Diagram?*. [online] Available at: <<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>>
9. Smartdraw.com. 2022. *Activity Diagram - Activity Diagram Symbols, Examples, and More*. [online] Available at: <<https://www.smartdraw.com/activity-diagram/#:~:text=An%20activity%20diagram%20visually%20presents,can%20be%20sequential%20and%20concurrent.>>
10. Thakur, D., 2022. *Data Design in Software Engineering - Computer Notes*. [online] Computer Notes. Available at: <<https://ecomputernotes.com/software-engineering/data-design>>
11. Lucidchart. 2022. *What is an Entity Relationship Diagram (ERD)?*. [online] Available at: <<https://www.lucidchart.com/pages/er-diagrams>>
12. Library.ucmerced.edu. 2022. *What Is a Data Dictionary? | UC Merced Library*. [online] Available at: <<https://library.ucmerced.edu/data-dictionaries>>
13. Firestore &nbsp;|&nbsp; Firebase (no date) Google. Google. Available at: <https://firebase.google.com/docs/firestore>.
14. Firebase realtime database (no date) Google. Google. Available at: <https://firebase.google.com/docs/database>.

## APPENDIX A: USER ACCEPTANCE TEST

# LIFE PLANNER

  Firebase



### Life Planner (LP) Mobile Application using Ionic and Firebase

Dear all,



This is a **User Acceptance Test** for the **Life Planner Mobile Application**. This application is an application that is focus on improving the lifestyle of the user.

This UAT is to fulfil the requirement for my Final Year Project (FYP) for BACHELOR OF COMPUTER SCIENCE (SOFTWARE ENGINEERING) WITH HONOURS. Do take part in this testing stage.

Thank You!

**CB19140 AHMAD NAQUIB BIN ALI**  
**BCS - BACHELOR OF COMPUTER SCIENCE (SOFTWARE ENGINEERING) WITH HONOURS**

Supervised by:  
DR. MOHD FAIZAL BIN AB RAZAK  
FACULTY OF COMPUTING  
UNIVERSITI MALAYSIA PAHANG

 [a.naquib5@gmail.com](mailto:a.naquib5@gmail.com) (not shared) [Switch account](#) 

**\* Required**

**Email \***

Your answer \_\_\_\_\_

**Institution \***

IPTA/IPTS

POLYTECHNIC

Primary / Secondary School

Other: \_\_\_\_\_

**Gender \***

Male

Female

**Age \***

Your answer \_\_\_\_\_

[Next](#) [Clear form](#)

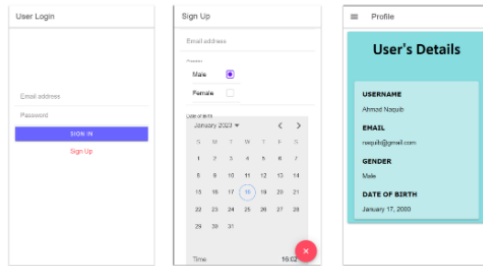
## User Application Testing

This section will show the preview of the application if the user decided on not wanting to download the application

Users can download the application (APK) through the link provided, using mobile phone is recommended:

<https://drive.google.com/file/d/1hzZqMEZBirwiqVxqDBpLSECNLNuAOuPd/view?usp=sharing>

### Interface for the Login & Signup Module



Manage to register into the application? \*

- Yes  
 No

Manage to login into the system with the correct credentials? \*

- Yes  
 No

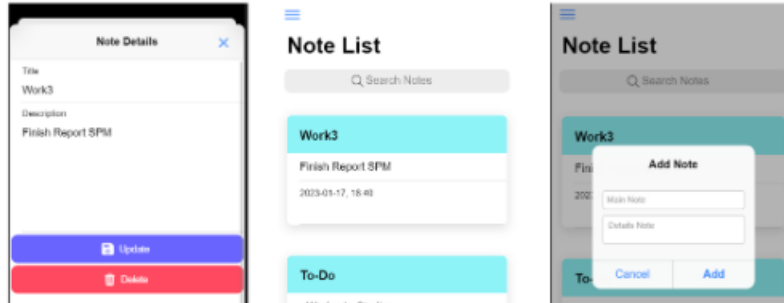
System was able to redirect to the the note page? \*

- Yes  
 No

System was able to show correct user information in the Profile age? \*

- Yes  
 No

## Interface for the Manage Note Module



Manage to add and view the the updated note into the application? \*

- Yes
- No

Able to edit a specific note in the application? \*

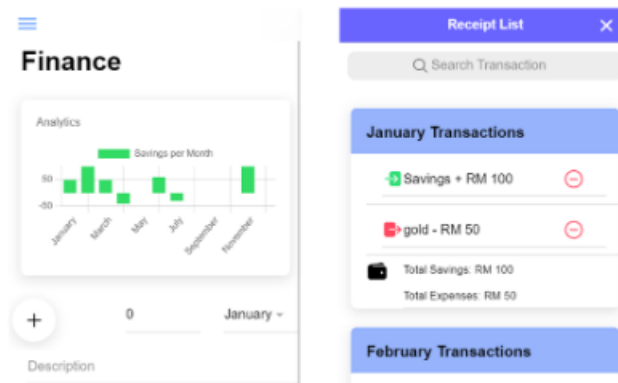
- Yes
- No

Able to search a specific note in the application? \*

Yes

No

Interface for the Manage Finance Module



Able to add a transaction in the specific month? \*

Yes

No

Able to view the updated chart in the application? \*

Yes

No

Generate Receipt button is able to show the list of user's transaction? \*

Yes

No

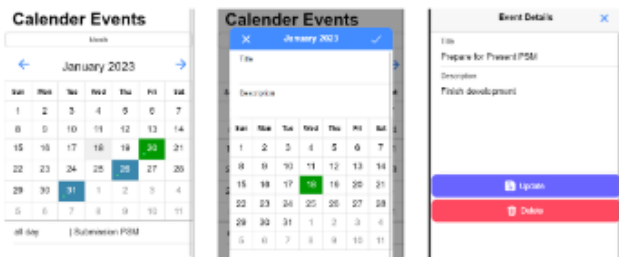
Able to delete a specific transaction and able to view the updated chart? \*

Yes

No



## Interface for the Manage Event Module



Able to add an event in a specific date? \*

- Yes  
 No

Able to view the updated calendar with the specific event? \*

- Yes  
 No

Able to view the details of the specific event? \*

- Yes  
 No

Able to update or delete the specific event? \*

- Yes  
 No

## Untitled Section

Rate the overall design for this application \*

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

Rate the overall experience for this application \*

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

Would you recommend other users to use this application? \*

	1	2	3	4	5	
No thanks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Yes definitely

Do you have any suggestion on improving the app?

Your answer

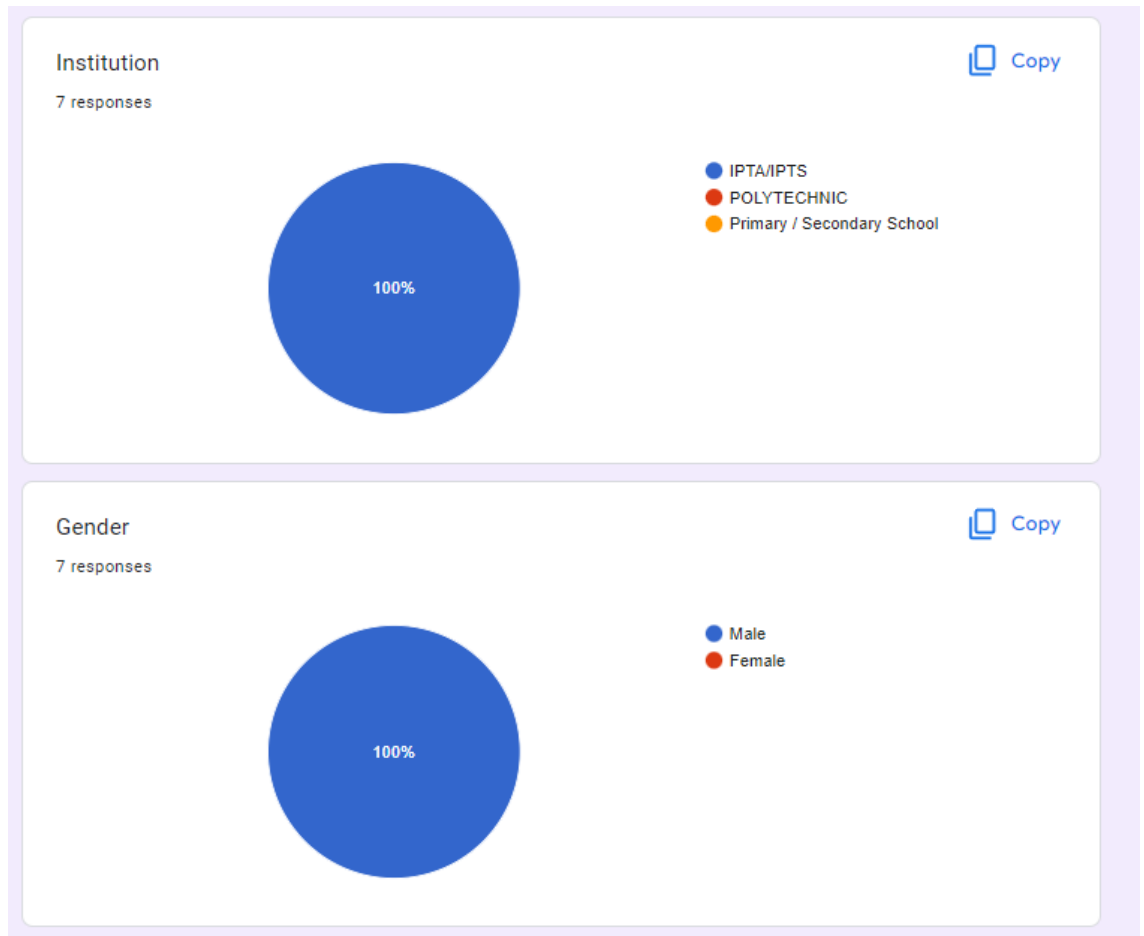
---

[Back](#)

[Submit](#)

[Clear form](#)

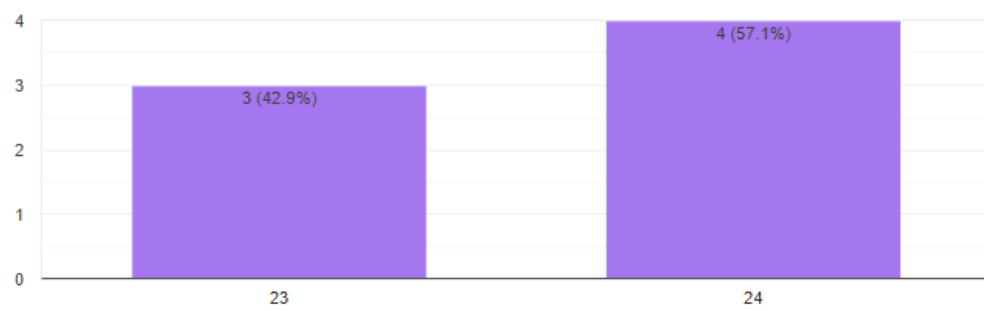
## APPENDIX B: UAT RESULT



## Age

 Copy

7 responses

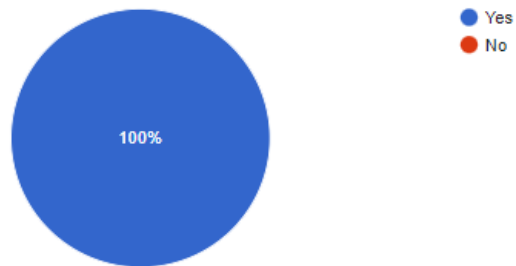


## User Application Testing

### Manage to register into the application?

 Copy

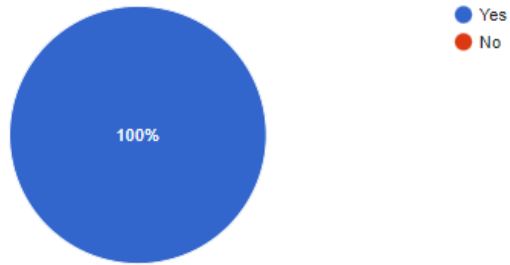
7 responses



Manage to login into the system with the correct credentials?

 Copy

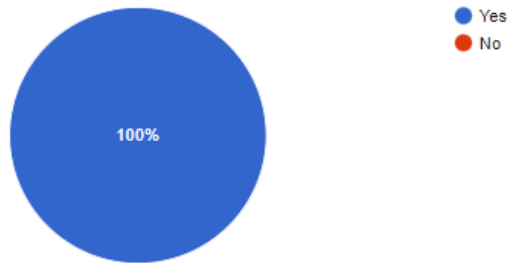
7 responses



System was able to redirect to the the note page?

 Copy

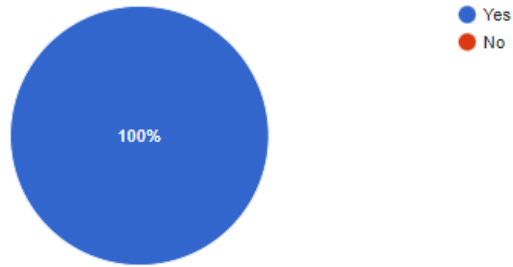
7 responses



System was able to show correct user information in the Profile age?

 Copy

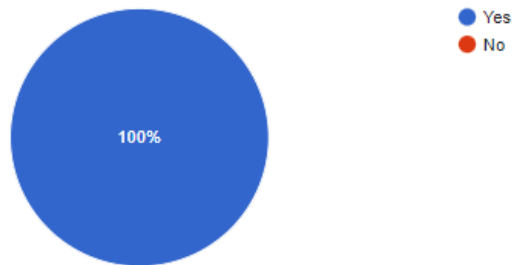
7 responses



Manage to add and view the the updated note into the application?

 Copy

7 responses



Able to edit a specific note in the application?

 Copy

7 responses



Yes  
 No

Able to search a specific note in the application?

 Copy

7 responses

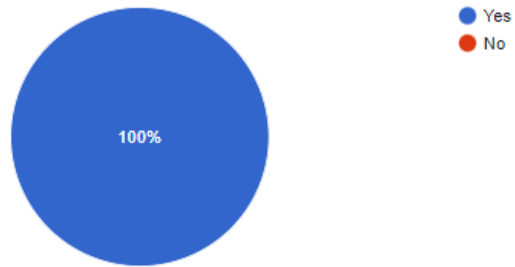


Yes  
 No

Able to add a transaction in the specific month?

 Copy

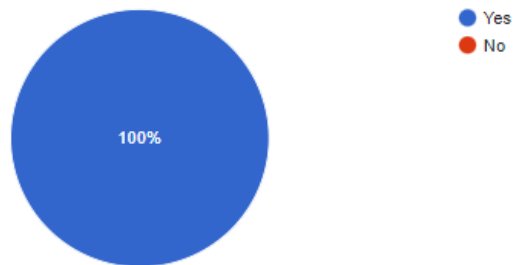
7 responses



Able to view the updated chart in the application?

 Copy

7 responses





Generate Receipt button is able to show the list of user's transaction?

 Copy

7 responses



Yes  
 No

Able to delete a specific transaction and able to view the updated chart?

 Copy

7 responses

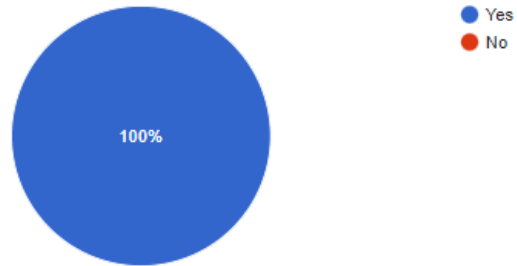


Yes  
 No

Able to add an event in a specific date?

 Copy

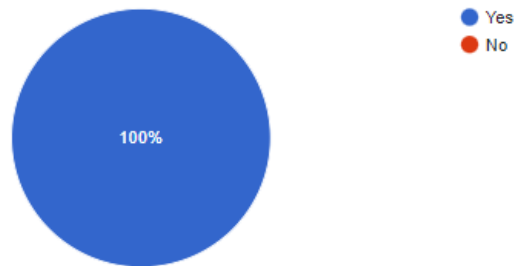
7 responses



Able to view the updated calendar with the specific event?

 Copy

7 responses

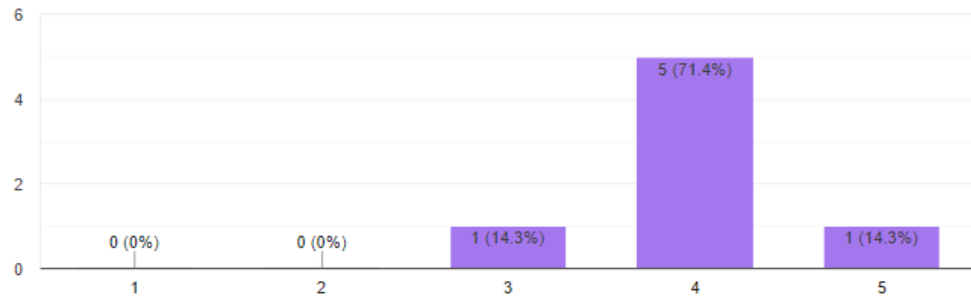


## Untitled Section

Rate the overall design for this application

 Copy

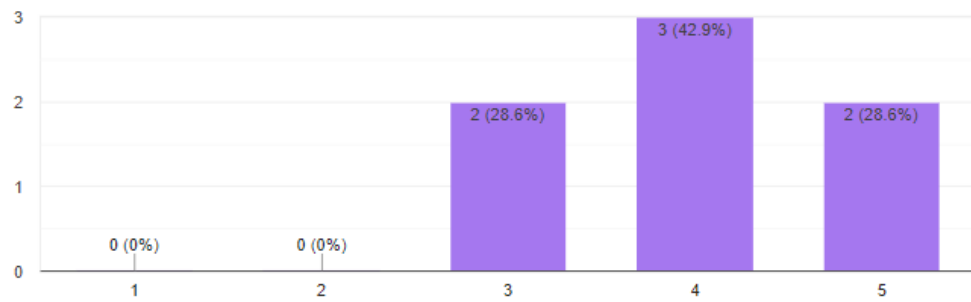
7 responses



Rate the overall experience for this application

 Copy

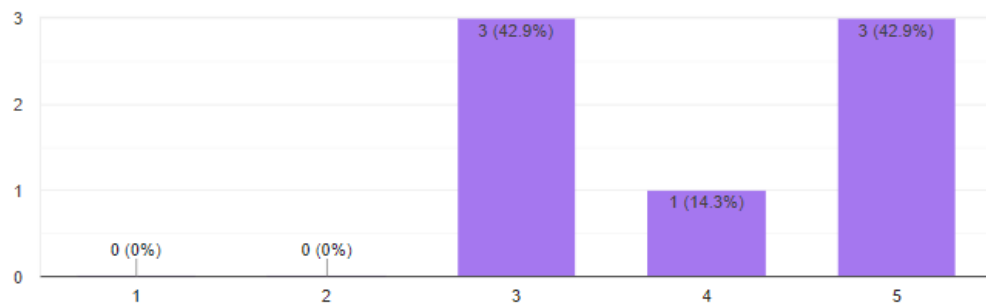
7 responses



### Would you recommend other users to use this application?

 Copy

7 responses



### Was there any bug or problem in using the app?

3 responses

Yes , other user can update and delete the note from other user

Yes can see other user' notes

I can edit my details in profile without edit/update button, and there is no confirmation that the profile is updated.

### Do you have any suggestion on improving the app?

4 responses

baik ah nakeb

Remembet me function so we do not need to sign in everytime we open the app

Color the interface to make it more interesting

In my opinion, this application can be improve by

- Design/color: Its a bit empty (maybe more icon/image/animation), blue and cyan didn't really match IMO 😬
- Calendar: maybe display all the events in list for each month below the calendar
- Finance: maybe adjust the "add expense/saving" layout by putting the description first, then add expense/saving and month. More gap between the "analytic graph" and "transaction section", or just put the transaction section at the bottom.
- Finance: different color for expenses bar (maybe red/orange) to differentiate it with savings bar.
- Profile: add edit button & confirmation, so user can edit their profile instead of tap at the text and type with no confirmation.

\*Overall it's a really good application, most of my suggestion is on the design, the functionality is more important and everything is working fine 👍 Good Job Naquib! 😊

2020

# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

[LIFE PLANNER APP]



## DOCUMENT APPROVAL

	Name	Date
<p>Authenticated by:</p> <hr/> <p>Name</p>		
<p>Approved by:</p> <hr/> <p>Client</p>		

--	--	--

Software :

Archiving Place :

## TABLE OF CONTENT

CONTENT	PAGE
<b>DOCUMENT APPROVAL</b>	<b>2</b>
<b>TABLE OF CONTENT</b>	<b>4</b>
<b>LIST OF FIGURES</b>	<b>6</b>
<b>LIST OF TABLES</b>	<b>7</b>
<b>LIST OF APPENDIXES</b>	<b>8</b>
<b>1.1 PROJECT DESCRIPTION</b>	<b>1</b>
<b>1.2 SYSTEM IDENTIFICATION</b>	<b>1</b>
<b>1.3 CONTEXT DIAGRAM</b>	<b>2</b>
<b>1.4 DATA FLOW DIAGRAM</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>



<b>2.1</b>	<b>USE CASE DIAGRAM AND DESCRIPTION</b>	<b>1</b>
<b>2.2</b>	<b>SEQUENCE DIAGRAM</b>	<b>11</b>
<b>3.1</b>	<b>INTERFACE DESIGN</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>3.2</b>	<b>HARDWARE AND SOFTWARE SPECIFICATION</b>	<b>18</b>

## LIST OF FIGURES

Figure 1.1	Context Diagram	2
Figure 2.1	Use Case Diagram	1
Figure 2.2	Use Case for Login Module	2
Figure 2.3	Use Case for Manage Notes Module	4
Figure 2.4	Use Case for Manage Finance Module	7
Figure 2.5	Use Case for Manage User Module	9
Figure 2.6	Sequence Diagram- Manage Login Module	11
Figure 2.7	Sequence Diagram - Manage Notes Module	12
Figure 2.8	Sequence Diagram - Manage Finance Module	12
Figure 2.9	Sequence Diagram - Manage User Module	13
Figure 3.1	Login Page	14
Figure 3.2	Sign Up Page	15
Figure 3.3	Home Page	16
Figure 3.4	Add Note Page	17
Figure 3.5	Finance Page	18

## LIST OF TABLES

Table 2.1	Use Case Description for Login Module	2
Table 2.2	Use Case Description for Manage Notes Module	4
Table 2.3	Use Case Description for Manage Finance Module	7
Table 2.4	Use Case Description for Manage User Module	9
Table 3.1	Hardware Specification	19
Table 3.2	Software Specification	19

## LIST OF APPENDICES

## CHAPTER 1

### 1.1 PROJECT DESCRIPTION

Life Planner application has four modules in total. The first one is manage note module, second is manage finance module, third one is the manage user module and the last one is login module. First module, manage notes is the application can take note of what the user wishes to store in the application. Users can fill in the details on the note with items such as date, title, and details. They are also able to toggle whether they want to notify them or not based on what time they put in.

Other than that, manage finance users are also able to manage their finance in the application. Users can input their initial balance into the wallet. They can then input either the expenses or income in the application within the month. The app also can generate a report on the list of expenses and income based on the month. The third module is the manage user's module where only the admin can view the list of users and can edit them or delete their profiles from the system. The last module is the login module where the user can log in and signup if they have no account setup yet.

### 1.2 SYSTEM IDENTIFICATION

System Title: *Life Planner Application*

System Abbreviation: LPA

System Identification Number: #012345

### 1.3 CONTEXT DIAGRAM

A context diagram or a level 0 Data Flow Diagram (DFD) is a chart that shows the input as well as the output to show how the data is processed. The overall system data flow is only represented by one process to show how the data is interacted. Based on figure 1.1 below, there will be a data interaction between the user and the system.

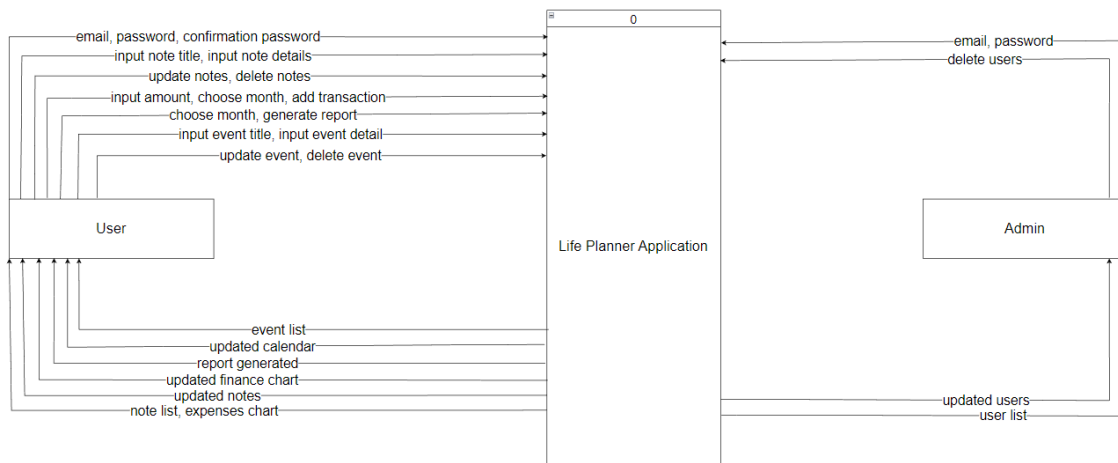


Figure 1.1 Context Diagram

For the context diagram there will be two external entities that will be using the system, which are user and admin. From the user, there will be five flows from user to the system process, however only four data flow will be flowing back to the user. For Admin entities there are email, password, and update users, and delete users that will be inputted into the system process while same goes with vice versa.

CHAPTER 2

2.1 USE CASE DIAGRAM AND DESCRIPTION

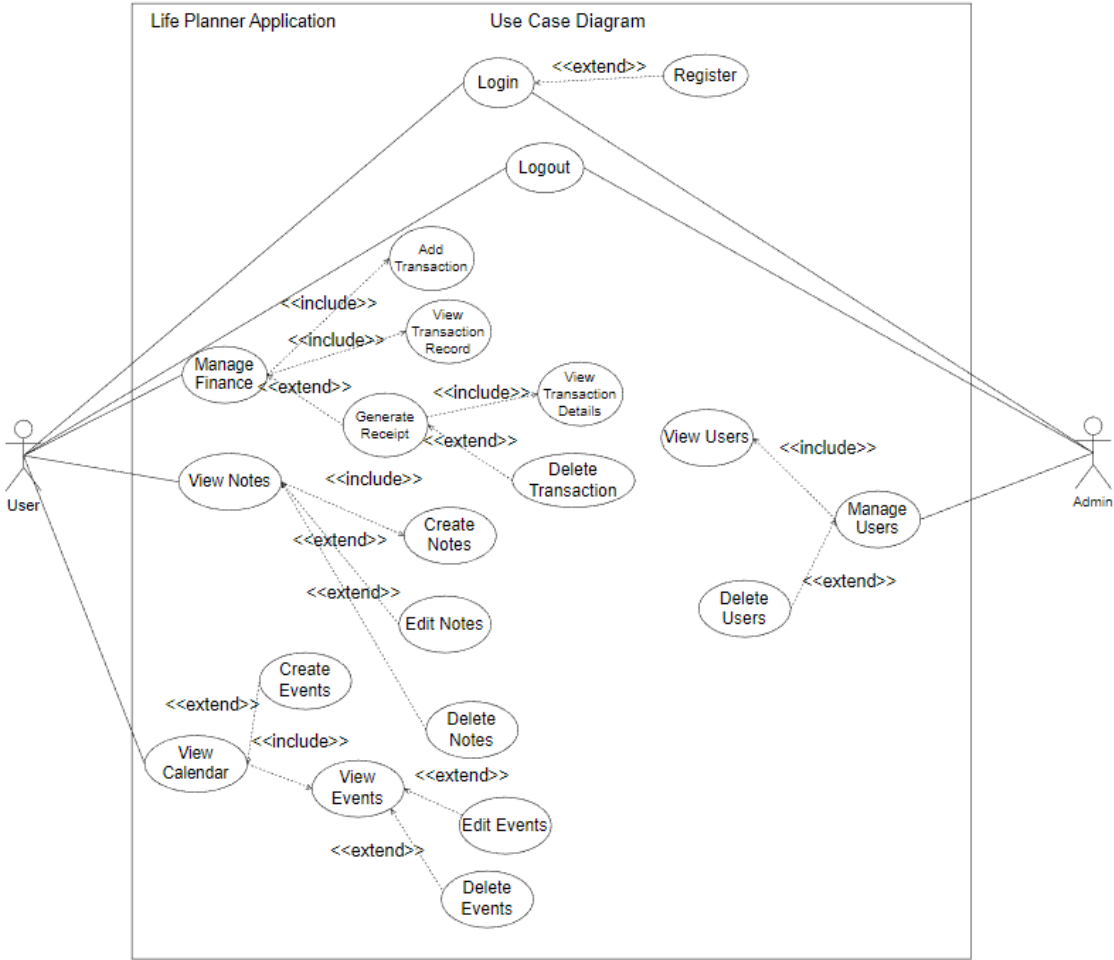


Figure 2.1 Use Case Diagram

## 2.1.1 Manage Login Module

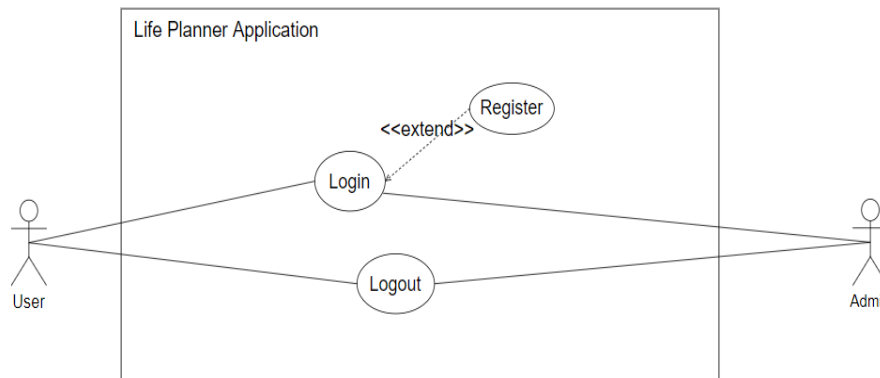


Figure 2.2 Use Case for Login Module

Table 2.1 Use Case Description for Login Module

Use Case ID	UC001
Brief Description	This use case describes on the process of how the users will login or register into the system
Actor	User, Admin
Pre-Conditions	<ol style="list-style-type: none"> <li>1. Users have downloaded the application</li> <li>2. Users can view the login page</li> </ol>
Basic Flow	<ol style="list-style-type: none"> <li>1. Use case start when the users view the login page. [A1- Users does not have an account]</li> <li>2. Users input their email and password</li> <li>3. Users click on the login button</li> </ol>



	<p>[E1-Wrong login credentials]</p> <ol style="list-style-type: none"> <li>4. System will redirect users to the home page of the system</li> <li>5. Use case end</li> </ol>
Alternative Flow	<p>[A1- Users does not have an account]</p> <ol style="list-style-type: none"> <li>1. Users click on the register button</li> <li>2. System will display the register page</li> <li>3. Users input all the credentials that is needed</li> <li>4. Users click the sign-up button</li> </ol> <p>[E2 – Wrong sign-up credentials]</p> <ol style="list-style-type: none"> <li>5. System will redirect users to login page</li> <li>6. Use case will resume on step 2 of basic flow</li> </ol>
Exception Flow	<p>[E1-Wrong login credentials]</p> <ol style="list-style-type: none"> <li>1. System will show a dialogue box on the errors</li> <li>2. Use case resume on step 2 of basic flow</li> </ol> <p>[E2- Wrong sign-up credentials]</p> <ol style="list-style-type: none"> <li>1. System will show a dialogue box on the errors</li> </ol>

	<ol style="list-style-type: none"> <li>2. Use case will resume on step 3 of alternative flows 1</li> </ol>
Post-Conditions	<ol style="list-style-type: none"> <li>1. Users will be able to login into the system</li> <li>2. Users will be able to view the home page</li> </ol>
Rules	<ol style="list-style-type: none"> <li>1. Needs to have a smartphone device</li> <li>2. Needs to have a connection with an internet</li> </ol>
Constraints	-

### 2.1.2 Manage Notes Module

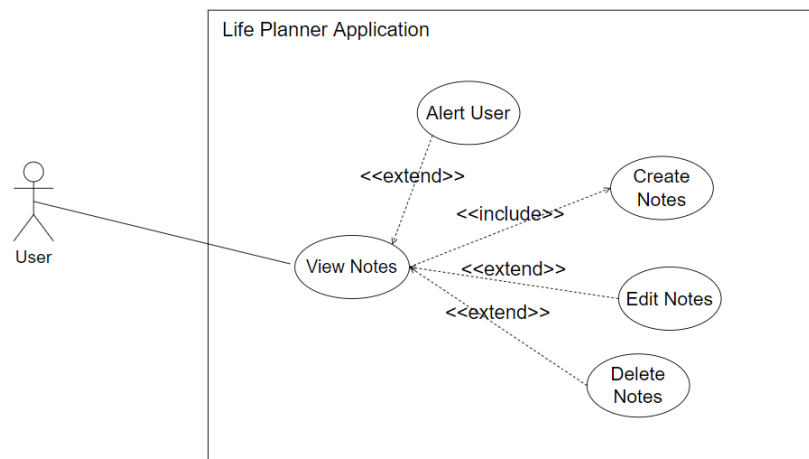


Figure 2.3 Use Case for Manage Notes Module

Table 2.2 Use Case Description for Manage Notes Module

Use Case ID	UC002
-------------	-------

Brief Description	This use case will show the flows on how the user will be able to manage their notes on the application.
Actor	Users
Pre-Conditions	<ol style="list-style-type: none"> <li>1. Have downloaded the application.</li> <li>2. Have an account in the system.</li> <li>3. Mobile device is connected to any internet connection</li> </ol>
Basic Flow	<ol style="list-style-type: none"> <li>1. Use case start when the users view the home page of the application.</li> <li>2. System will display a list of notes that has been inserted by the users from previous session.  [A1 – Edit Notes]</li> <li>3. Users click on the add logo icon on the bottom of the application.</li> <li>4. System will display a page to create a new note.</li> <li>5. Users will insert the needed information on the page given.</li> <li>6. Users click on the submit button.</li> <li>7. System will direct users to the home page.</li> <li>8. Use case end</li> </ol>

Alternative Flow	<p>[A1 – Edit Notes]</p> <ol style="list-style-type: none"> <li>1. Users click on a specific note on the home screen.</li> <li>2. System will display a page to edit the specific note.</li> <li>3. Users will update any information that the user want.</li> <li>4. Users click on the close button .</li> <li>5. Use case will resume on the step 2 of the basic flow</li> </ol>
Exception Flow	-
Post-Conditions	<ol style="list-style-type: none"> <li>1. Users will be able to view all the notes from previous session.</li> <li>2. Users will be able to create a new note.</li> <li>3. Users will be able to update the selected note</li> </ol>
Rules	<ol style="list-style-type: none"> <li>1. Needs to have a smartphone device.</li> <li>2. Needs to have a connection with an internet.</li> <li>3. Have an account on the application</li> </ol>
Constraints	-

### 2.1.3 Manage Finance Module

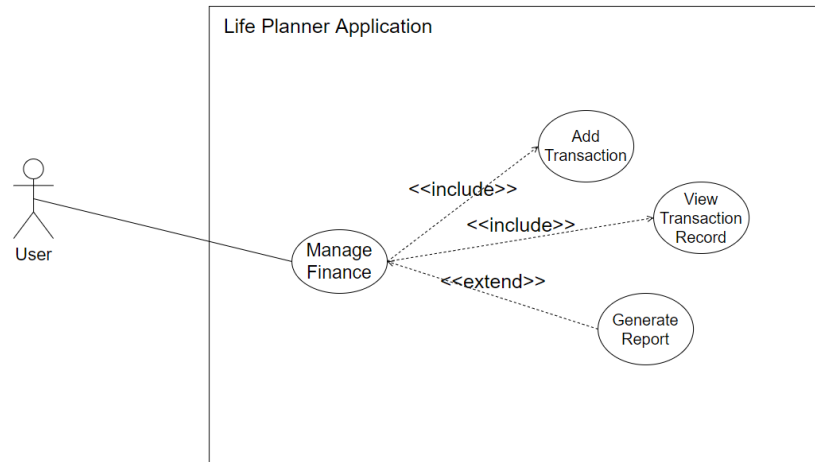


Figure 2.4 Use Case for Manage Finance Module

Table 2.3 Use Case Description for Manage Finance Module

Use Case ID	UC003
Brief Description	This use case will show the flows on how the user will be able to manage their finance on the application.
Actor	Users
Pre-Conditions	<ol style="list-style-type: none"> <li>1. Have downloaded the application</li> <li>2. Have an account in the system</li> <li>3. Mobile device is connected to any internet connection</li> </ol>
Basic Flow	<ol style="list-style-type: none"> <li>1. Use case start when the users view the finance page</li> </ol>

	<ol style="list-style-type: none"> <li>2. System will display a chart that contains the finance information throughout the whole year</li> <li>3. Users choose on a particular month that the user wants to update  [A1 – Generate Report]</li> <li>4. Users will select either to input a new finance information as an expense or saving with their information details</li> <li>5. System will update the chart and generate it on the finance page</li> <li>6. Use case end</li> </ol>
Alternative Flow	<p>[A1 – Generate Report]</p> <ol style="list-style-type: none"> <li>1. Users select on the generate report button</li> <li>2. System will display a list of information on their finances on the selected month</li> <li>3. Use case will resume on step 3 on the basic flow</li> </ol>
Exception Flow	-
Post-Conditions	<ol style="list-style-type: none"> <li>1. Users will be able to see the updated charts of their finance information</li> </ol>
Rules	-
Constraints	-

## 2.1.4 Manage User Module

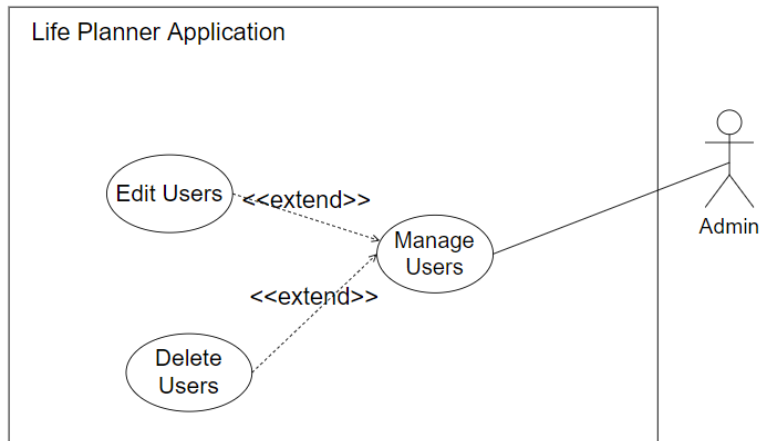


Figure 2.5 Use Case for Manage User Module

Table 2.4 Use Case Description for Manage User Module

Use Case ID	UC004
Brief Description	This use case will show the flows on how the user will be able to manage their users on the application.
Actor	Admin
Pre-Conditions	<ol style="list-style-type: none"> <li>1. Have downloaded the application</li> <li>2. Have an account in the system</li> <li>3. Mobile device is connected to any internet connection</li> </ol>
Basic Flow	<ol style="list-style-type: none"> <li>1. Use case start when the system shows the login page</li> </ol>

	<ol style="list-style-type: none"> <li>2. Admin will need to input their special credential as an admin</li> <li>3. System will display list of users that is registered in the system</li> <li>4. Admin will choose on a particular user to update</li> <li>5. System will display a specific page on the user's info</li> </ol> <p>[A1 – Delete User]</p> <ol style="list-style-type: none"> <li>6. Admin will update any of the info from the user that is needed</li> <li>7. Admin clicks on the close button</li> <li>8. System will redirect users to user list page</li> <li>9. Use case end</li> </ol>
Alternative Flow	<p>[A1 – Delete User]</p> <ol style="list-style-type: none"> <li>1. Admin will click on the delete button</li> <li>2. System will display a confirmation dialogue</li> <li>3. User clicks on OK</li> <li>4. Use case resume on step 3 of the basic flow</li> </ol>
Exception Flow	-



Post-Conditions	<ol style="list-style-type: none"> <li>Admin will be able to view the list of updated users</li> <li>Admin will be able to update user's information or delete any users if necessary</li> </ol>
Rules	-
Constraints	-

## 2.2 SEQUENCE DIAGRAM

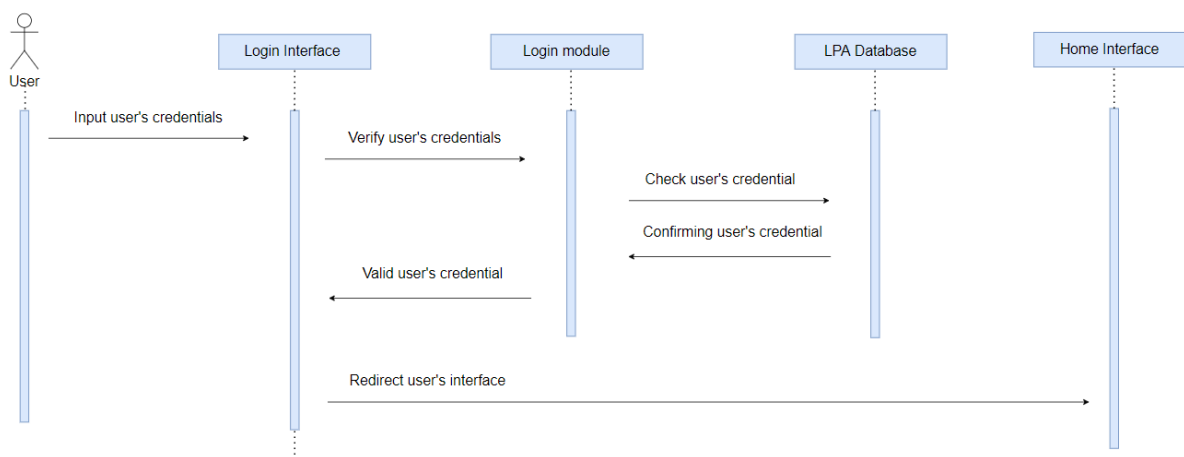


Figure 2.6 Sequence Diagram- Manage Login Module

Figure 2.6 above shows the basic flows on the process of manage login module in relate to UC001 (Manage Login Module).

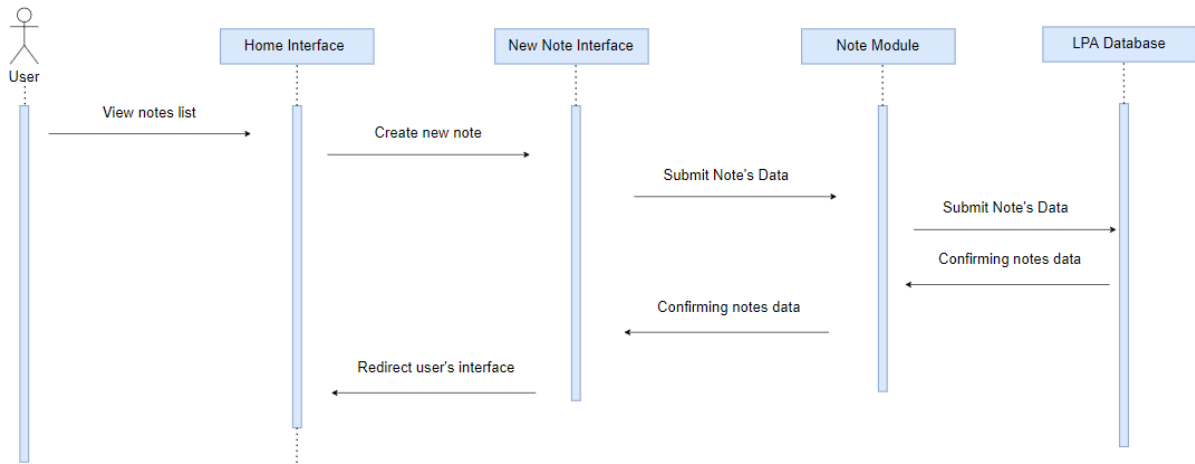


Figure 2.7 Sequence Diagram - Manage Notes Module

Figure 2.7 above shows the basic flows on the process of manage notes module in relate to UC002 (Manage Notes Module).

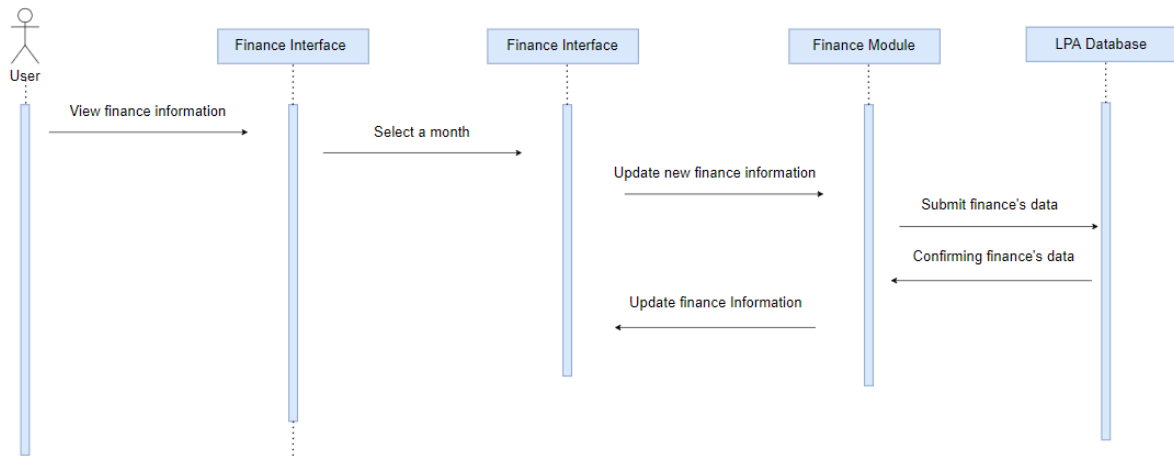


Figure 2.8 Sequence Diagram - Manage Finance Module

Figure 2.8 above shows the basic flows on the process of manage finance module in relate to UC003 (Manage Finance Module).

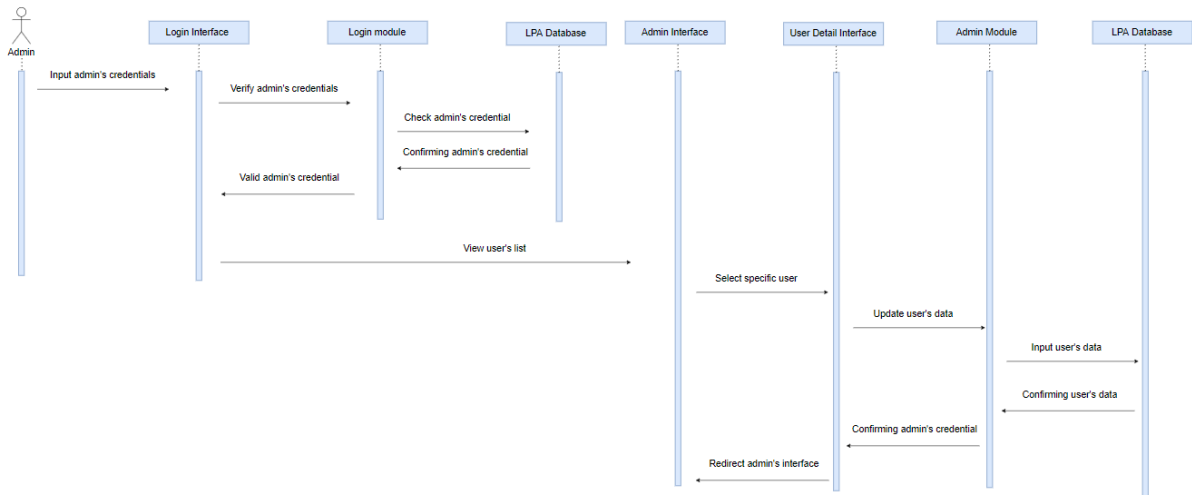
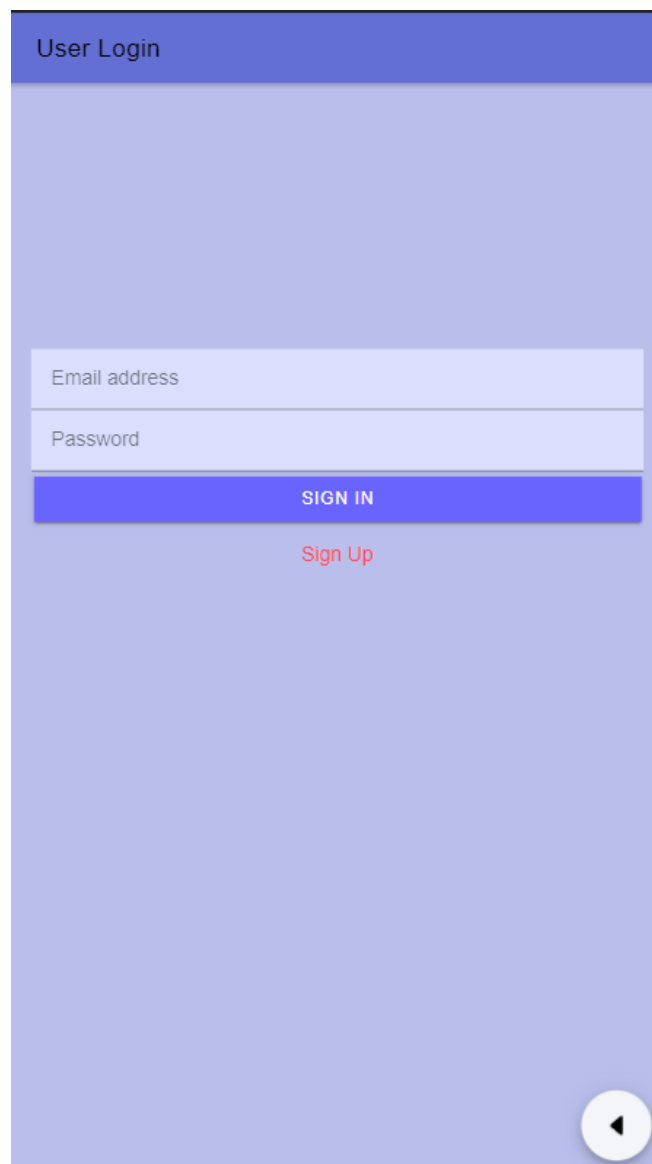


Figure 2.9 Sequence Diagram - Manage User Module

Figure 2.9 above shows the basic flows on the process of manage user module in relate to UC004 (Manage User Module).

## CHAPTER 3

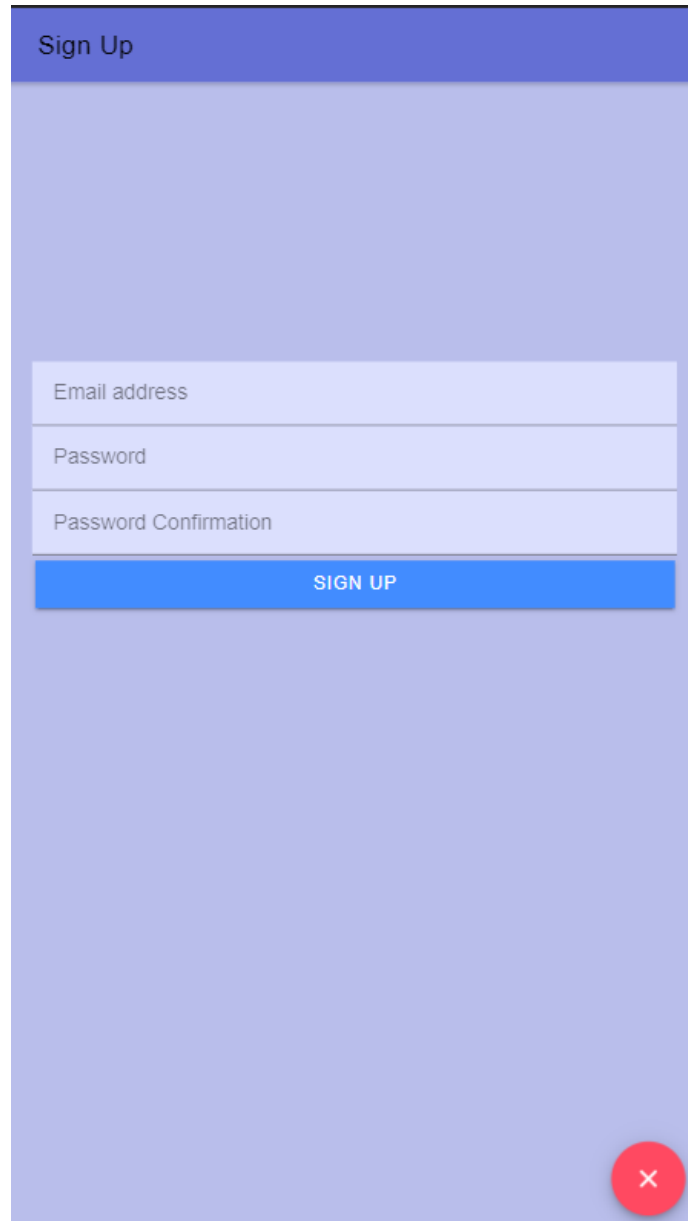
### 3.1 INTERFACE DESIGN



The image shows a mobile application interface for a user login page. At the top, there is a dark blue header bar with the text "User Login" in white. Below the header, the background is a light blue gradient. The login form consists of two input fields: "Email address" and "Password", both with light blue backgrounds and thin borders. Below the password field is a dark blue button with the text "SIGN IN" in white. Underneath the button is a red link that says "Sign Up". In the bottom right corner, there is a white circular button with a black left-pointing arrow, serving as a back navigation control.

Figure 3.1 Login Page

The first page for the application when the system start is the login page. The user will need to input their email as well as their password. If the credential is correct, then the user will be directed to the home page. If not, then the user will receive an error message regarding their errors. On the bottom is the widget if the user wants to login using any other social media such as Google or Facebook.



The image shows a mobile application sign-up screen. At the top, there is a dark blue header with the text "Sign Up" in white. Below the header is a light blue gradient background. In the center, there are three stacked text input fields with the labels "Email address", "Password", and "Password Confirmation". Below these fields is a bright blue button with the text "SIGN UP" in white. In the bottom right corner, there is a red circular button with a white "X" inside, likely a close or cancel button.

Figure 3.2 Sign Up Page

Figure 3.2 is the sign-up page where any user who does not have an account will be able to register. The credential that is needed is a valid email address as well as password which 6 is the minimum character that is needed to be input. If the user already

has an account, then user can go back to the login page via the cancel button below. If the user successfully entered the correct credential, then user will be directed to the login page where the user will be needed to login again.

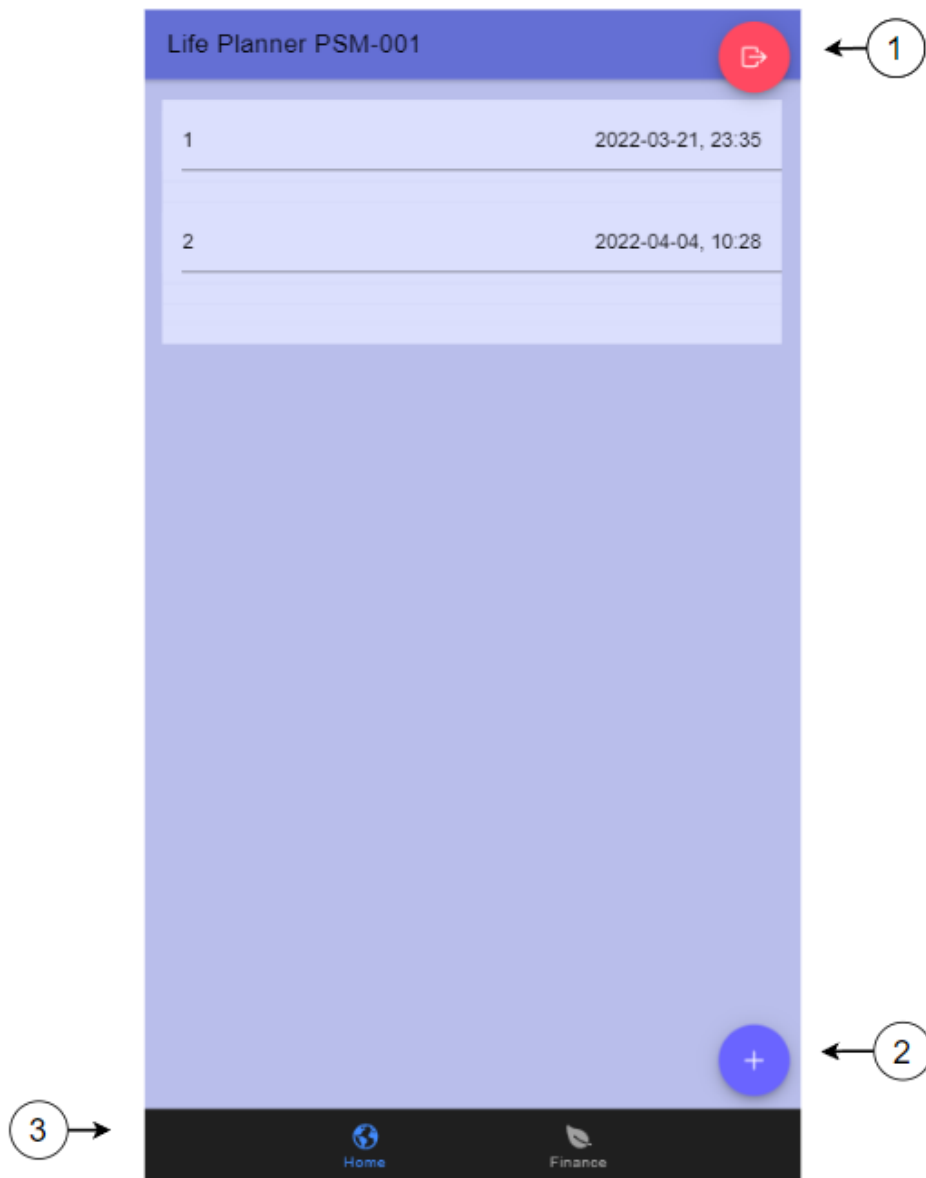


Figure 3.3 Home Page

Figure 3.3 is the home page where the user will be directed when the user successfully entered the correct credential in the login page. Home page is where all the notes will be listed in according to their time created. According to the figure above, 1 is the logout button where the user can logout and back to the login page. No 2 is for when

the user wants to create a new note by filling in the details of the note. No 3 is where all the tabs of the application, on the left is the home page and on the right is the Finance page.

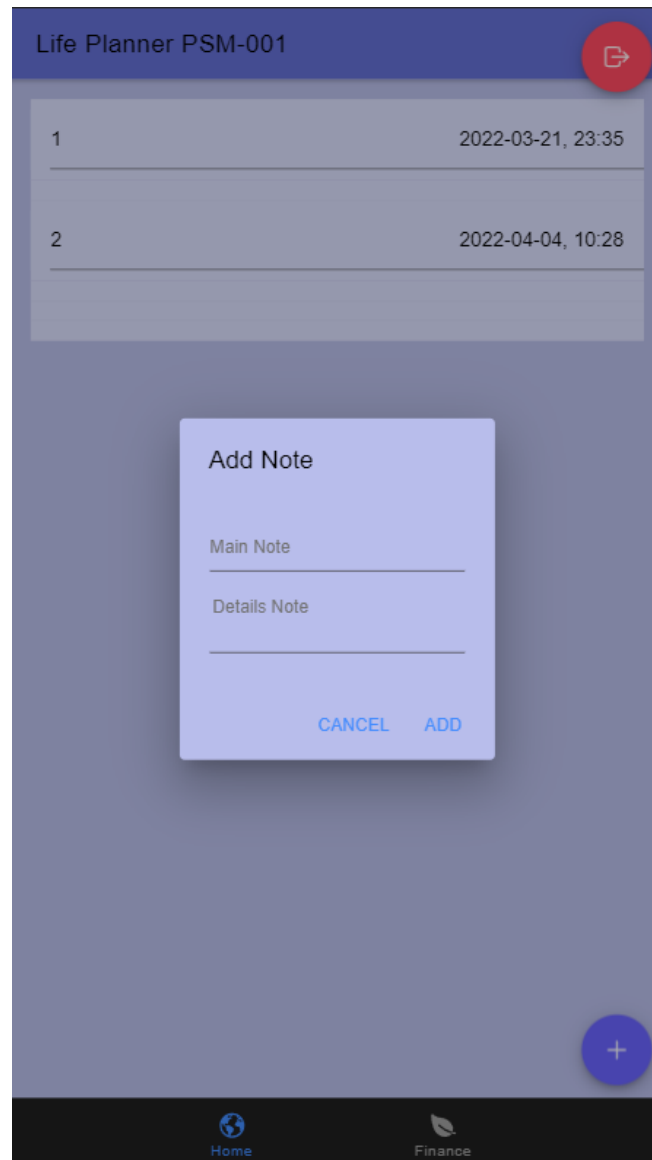


Figure 3.4 Add Note Page

Figure 3.4 is the Add Note page where the user able input the new notes with their main note as well as the details of the note.

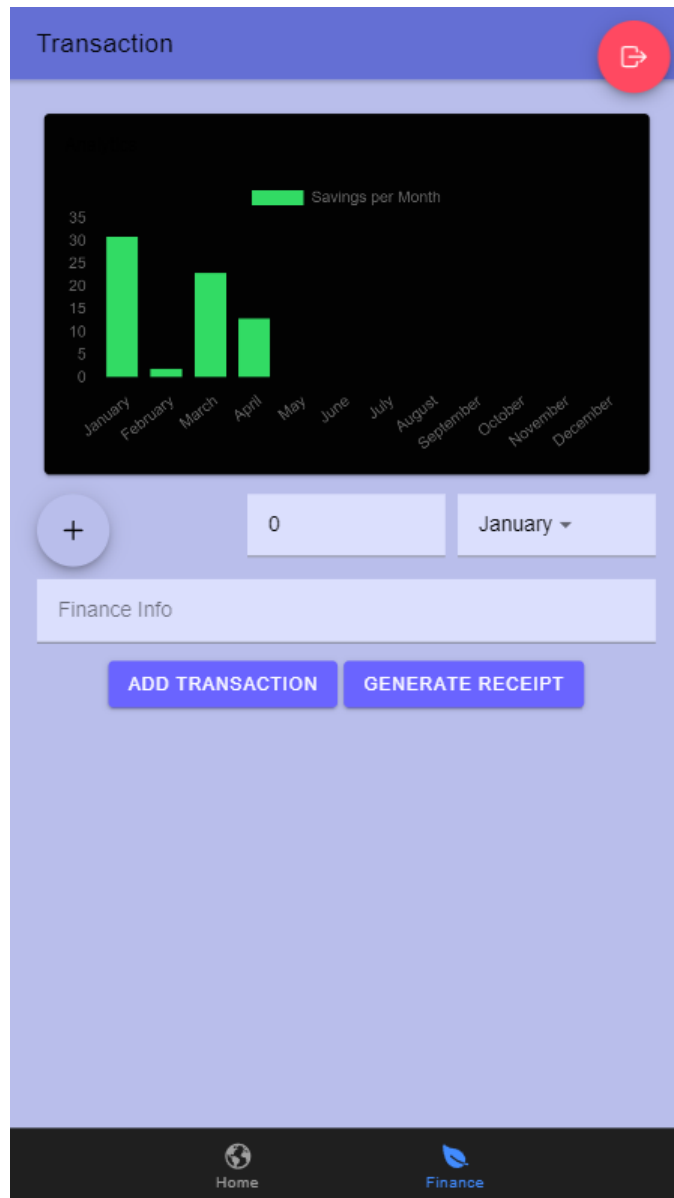


Figure 3.5 Finance Page

Figure 3.5 is the Finance Page where the user will be able to view records of the transaction throughout the whole year. Users can also input a new transaction either it is a saving or a withdraw money. Receipt can also be generated with their financial info.

### 3.2 HARDWARE AND SOFTWARE SPECIFICATION



**Table 3.1 Hardware Specification**

Hardware	Specification	Description
Asus ROG Gaming (Personal Computer)	<p>Operating System: Windows 10</p> <p>Processor: AMD Ryzen 7 4800H with Radeon Graphics@2.90 GHz</p> <p>RAM: 8GB</p> <p>GPU GEFORCE GTX</p>	For development phase, this laptop has been used as it has the necessary tools for the development of the system.
Redmi Note 9 Pro (Smart Mobile Device)	<p>Operating System: MIUI 12.5.5 (Android 11)</p> <p>Processor: Octa-core Max 2.32GHz</p> <p>RAM: 8GB</p> <p>Storage: 128 GB</p>	Mobile device that has been used for testing and deploying the application to test its usability.

**Table 3.2 Software Specification**

Software	Version	Description
Visual Studio Code	1.67.2	Code editor software that is being used due to the efficiency that is offered by the software

Git	2.36.1	Software that is used to keep track on any changes that is being applied to the system.
Draw.io	18.1.3	Drawing software tools that are being used to draw any kind of diagrams to give a better understanding of the system.
Google Chrome	-	Browser that has been used to test the application.

2020

# SOFTWARE DESIGN DESCRIPTION (SDD)

[LIFE PLANNER APP]



## DOCUMENT APPROVAL

	Name	Date
<p>Authenticated by:</p> <hr/> <p>Name</p>		
<p>Approved by:</p> <hr/> <p>Client</p>		

--	--	--

Software : LIFE PLANNER APP

Archiving Place :

## TABLE OF CONTENT

CONTENT	PAGE
<b>DOCUMENT APPROVAL</b>	<b>2</b>
<b>TABLE OF CONTENT</b>	<b>4</b>
<b>LIST OF FIGURES</b>	<b>6</b>
<b>LIST OF TABLES</b>	<b>7</b>
<b>LIST OF APPENDICES</b>	<b>8</b>
<b>1.1</b>	<b>PROJECT DESCRIPTION</b>
	<b>1</b>
<b>1.2</b>	<b>SYSTEM IDENTIFICATION</b>
	<b>1</b>
<b>1.2</b>	<b>ARCHITECTURE / BLUEPRINT</b>
	<b>2</b>

1.3.1 Application Layer	2
1.3.2 Business Service Layer	3
1.3.3 MVC Package Relationship	3

<b>1.4</b>	<b>ARCHITECTURE / BLUEPRINT DESCRIPTION</b>	<b>4</b>
------------	---	----------

1.4.1 Manage User Interface	4
1.4.2 Manage Notes Interface	4
1.4.3 Manage Finance Interface	5
1.4.4 Login Interface	6

<b>CHAPTER 2</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
------------------	-------------------------------------

<b>2.1</b>	<b>DATA DICTIONARY</b>	<b>17</b>
------------	------------------------	-----------

2.2.1 Users Table	17
2.2.2 Notes Table	18
2.2.3 Finance Table	18
2.2.4 DetailsFinance Table	19

## LIST OF FIGURES

Figure 1 Application Layer	2
Figure 2 Business Service Layer	3
Figure 3 MVC Package	3
Figure 4 Manage User Interface	4
Figure 5 Manage Notes Interface	5
Figure 6 Manage Finance Interface	5
Figure 7 Login Interface	6



## LIST OF TABLES

Table 1 Users Table	17
Table 2 Notes Table	18
Table 3 Finance Table	18
Table 4 DetailsFinance Table	19

## LIST OF APPENDICES

## CHAPTER 1

### PROJECT DESCRIPTION

Life Planner application has four modules in total. The first one is manage note module, second is manage finance module, third one is the manage user module and the last one is login module. First module, manage notes is the application can take note of what the user wishes to store in the application. Users can fill in the details on the note with items such as date, title, and details. They are also able to toggle whether they want to notify them or not based on what time they put in.

Other than that, manage finance users are also able to manage their finance in the application. Users can input their initial balance into the wallet. They can then input either the expenses or income in the application within the month. The app also can generate a report on the list of expenses and income based on the month. The third module is the manage user's module where only the admin can view the list of users and can edit them or delete their profiles from the system. The last module is the login module where the user can log in and signup if they have no account set up yet.

### SYSTEM IDENTIFICATION

System Title: *Life Planner Application*  
System Abbreviation: LPA  
System Identification Number: #012345

# ARCHITECTURE / BLUEPRINT

## 1.1.1 Application Layer

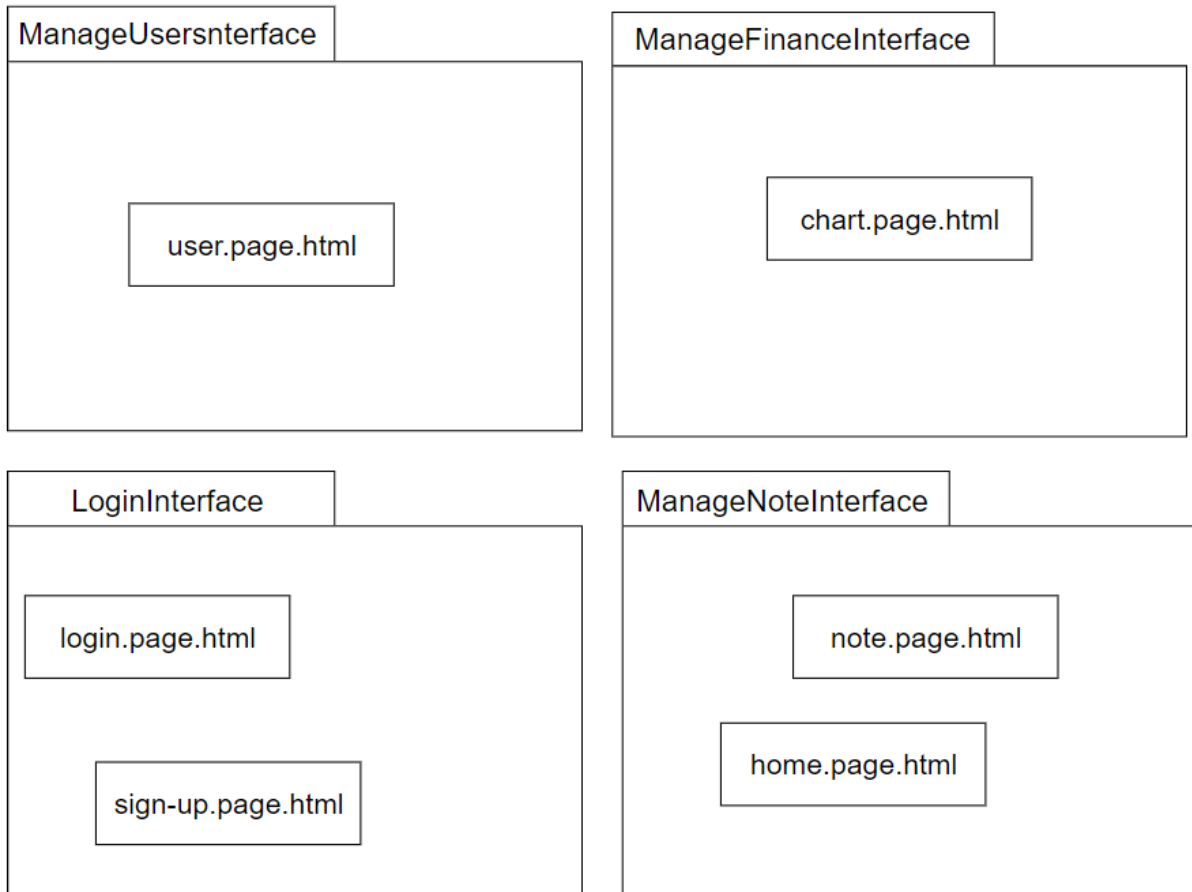


Figure 1 Application Layer

### 1.1.2 Business Service Layer

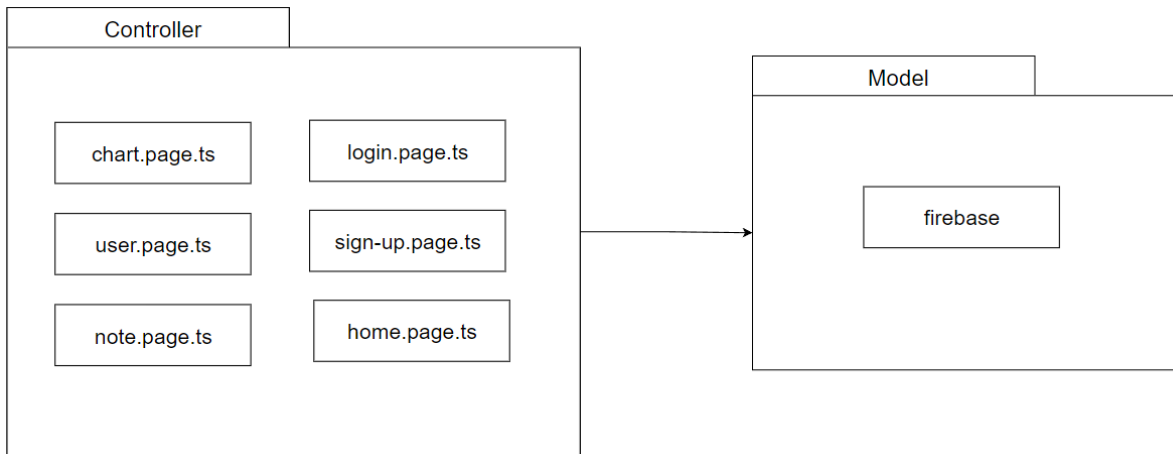


Figure 2 Business Service Layer

### 1.1.3 MVC Package Relationship

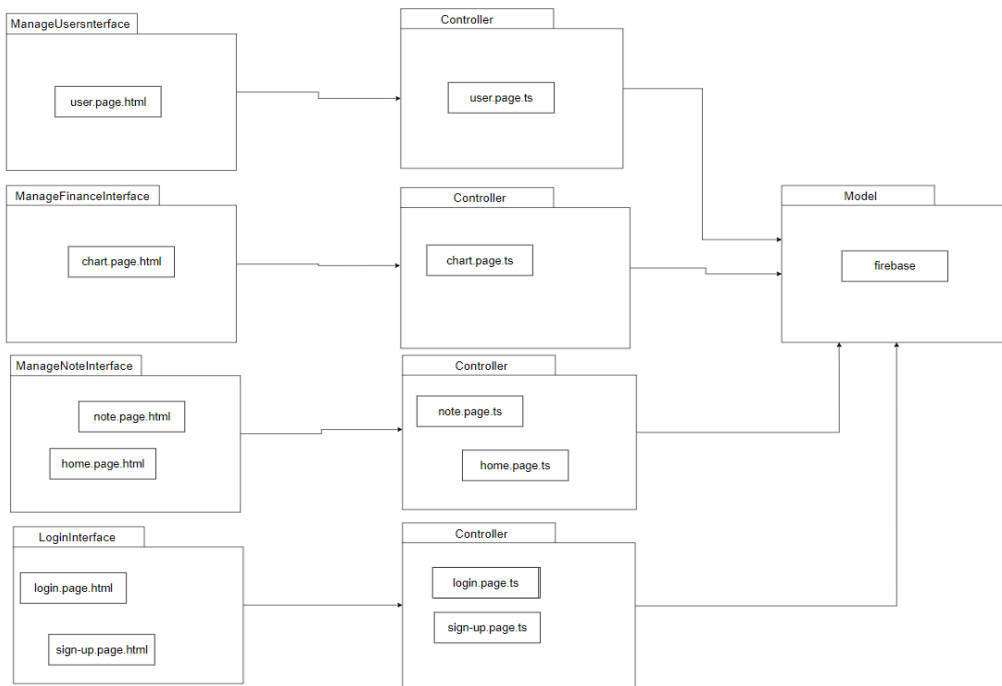


Figure 3 MVC Package

## ARCHITECTURE / BLUEPRINT DESCRIPTION

### 1.1.4 Manage User Interface

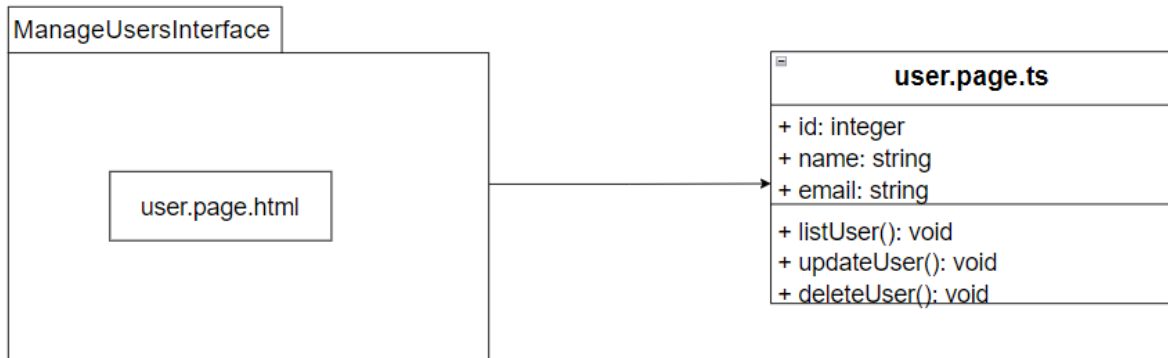


Figure 4 Manage User Interface

### 1.1.5 Manage Notes Interface

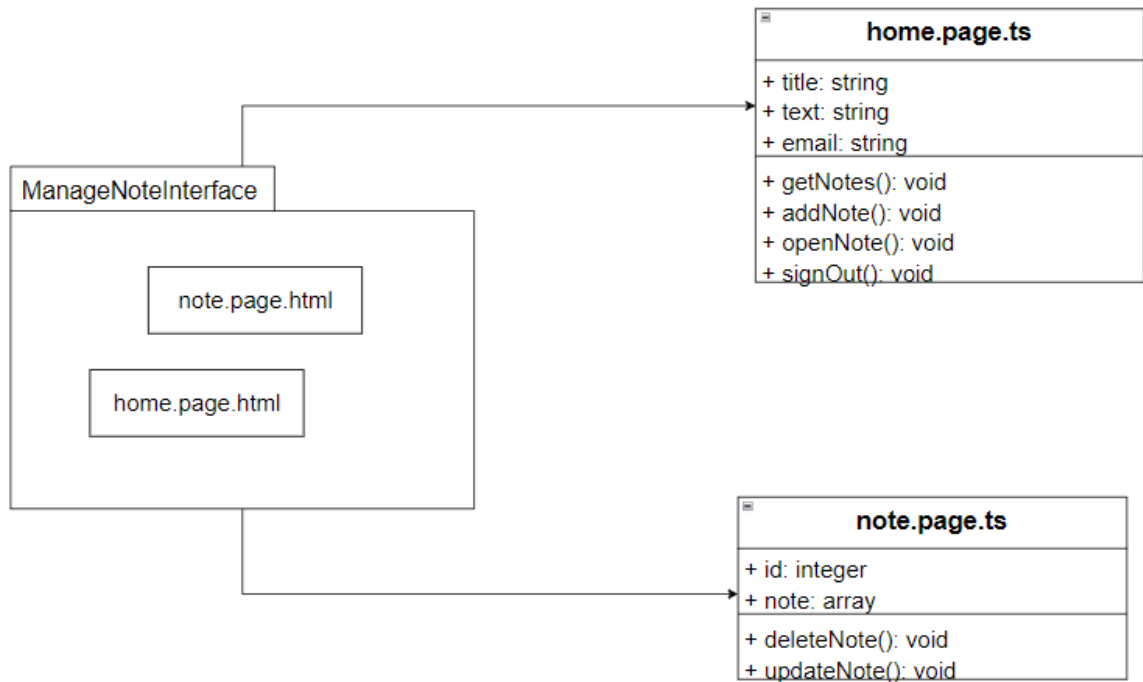


Figure 5 Manage Notes Interface

### 1.1.6 Manage Finance Interface

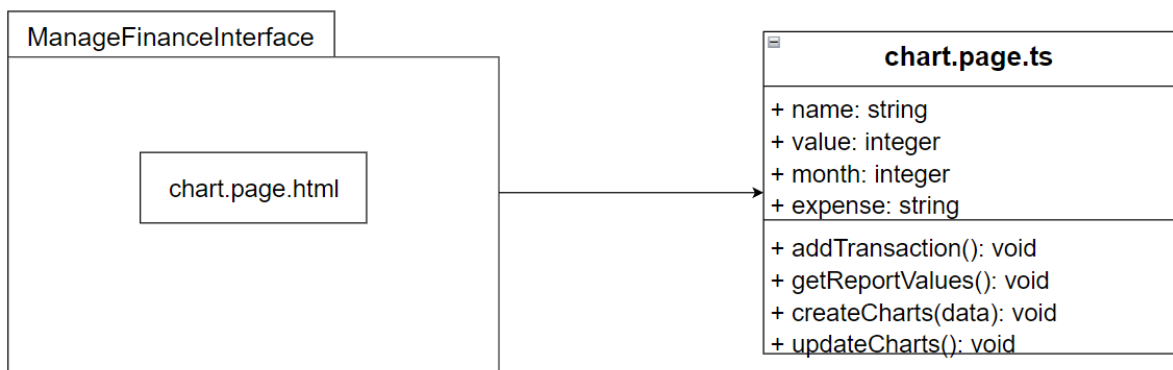


Figure 6 Manage Finance Interface

### 1.1.7 Login Interface

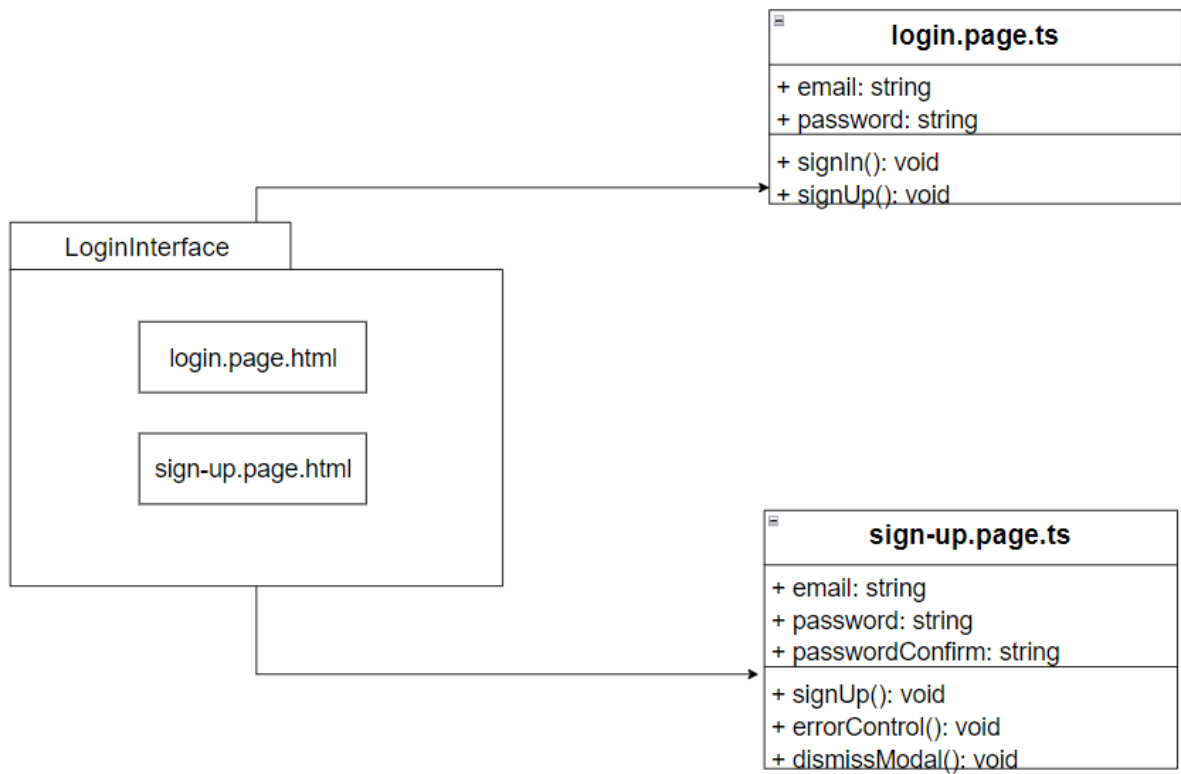


Figure 7 Login Interface



## CHAPTER 2

### 2.1 User.page.ts

Class Type	Boundary Class	
Responsibility	Controller for the manage user interface. Listing all the registered users that is available on the system, update user's information, and delete any users that is necessary in the system.	
Attributes	Attributes Name	Attributes Type
	Id	integer
	Name	string
	email	String
Methods	Method Name	Description
	listUser()	List all the registered users in the database
	updateUser()	Update the user's information if necessary
	deleteUser()	Delete any registered users if necessary
Algorithm	<b>listUser()</b>	

	<pre> START  STORE users table AS user  END  <b>updateUser()</b>  START  STORE user BY id AS userDocRef  UPDATE DOC BY userDocRef WITH User  END  <b>deleteUser()</b>  START  STORE user BY id AS userDocRef  DELETE DOC BY userDocRef  Modal DISMISS  END </pre>
--	---

## 2.2 Home.page.ts

Class Type	Boundary Class
------------	----------------

Responsibility	Home page of the system. List all the notes for the user, let user to view or add a specific note, and let users log out from the application.	
Attributes	Attributes Name	Attributes Type
	Title	string
	test	String
	email	string
Methods	Method Name	Description
	getNotes()	Let the system shows all the available notes for the user.
	addNote()	Open a modal that lets the user to fill in all the details of a new note and input all the data into the database.
	openNote()	Open a specific note to view its details
	signOut()	Let users to logout from the application
Algorithm	<b>getNotes()</b>  START  STORE data FROM notes table AS notes	

END

**addNote()**

START

STORE current time AS createdAt

CREATE modal FOR users to input data

SUBMIT data to NOTES TABLE

END

**openNote()**

START

CREATE modal for specific note

PULL notes BY note id

WAIT modal present

END

**signOut()**

START

CALL auth service FOR signout

REDIRECT to login page

	END
--	-----

### 2.3 Note.page.ts

Class Type	Boundary Class	
Responsibility	View a specific note and let the user to either update or delete it.	
Attributes	Attributes Name	Attributes Type
	Id	Integer
	Note	Array
Methods	Method Name	Description
	deleteNote()	Allow users to delete a specific note
	updateNote()	Allow users to update a specific note
Algorithm	<p><b>deleteNote()</b></p> <p>START</p> <p>STORE notes BY id AS noteDocRef</p> <p>DELETE DOC BY noteDocRef</p> <p>Modal DISMISS</p>	

	<p>END</p> <p><b>updateNote()</b></p> <p>START</p> <p>STORE notes BY id AS noteDocRef</p> <p>UPDATE DOC BY noteDocRef WITH Note</p> <p>END</p>
--	--

## 2.4 chart.page.ts

Class Type	Boundary Class	
Responsibility	Let users to view a chart that shows the value of the transaction for the whole year.	
Attributes	Attributes Name	Attributes Type
	Name	integer
	Value	Integer
	Month	integer
	expense	String
Methods	Method Name	Description

	addTransaction()	Let the users add a transaction for a specific month
	getReportValues()	Get a specific data for the specific month
	createCharts(data)	Let the system shows a chart for the transaction for the whole year
	updateCharts(data)	Update the charts with the newly updated data
Algorithm	<p><b>addTransaction()</b></p> <p>START</p> <p>STORE new data AS transaction</p> <p>PUSH transaction TO DATABASE</p> <p>END</p> <p><b>getReportValues()</b></p> <p>START</p> <p>LET reportByMonth AS monthly data</p> <p>STORE new data TO reportByMonth</p> <p>END</p> <p><b>createCharts(data)</b></p>	

	<p>START</p> <p>STORE data AS chartData</p> <p>LET chartData EQUAL TO getReportValues()</p> <p>LET valuesBarChart EQUAL TO chart element</p> <p>STORE chartData INTO valuesBarChart</p> <p>END</p> <p><b>updateCharts(data)</b></p> <p>START</p> <p>LET data AS chartData</p> <p>LET chartData EQUAL TO getReportValues</p> <p>UPDATE valuesBarChart FOR EACH chartData</p> <p>END</p>
--	--

## 2.5 login.page.ts

Class Type	Boundary Class	
Responsibility	Allow users to login into the system, either it is a user or an admin	
Attributes	Attributes Name	Attributes Type



	email	string
	Password	string
Methods	Method Name	Description
	signIn()	Allow users to sign in into the system with the correct credentials
	signUp()	Let the system shows the page for the users to fill to sign-up to the system
Algorithm	<p><b>signIn()</b></p> <p>START</p> <p>VALIDATE user's credential</p> <p>IF VALID</p> <p>REDIRECT TO home page</p> <p>IF NOT VALID</p> <p>SHOW message error</p> <p>END</p> <p><b>signUp()</b></p> <p>START</p>	

	<p>CREATE modal view for users to fill in for sign up</p> <p>AWAIT modal PRESENT</p> <p>IF user DISMISS</p> <p>FORM reset</p> <p>END</p>
--	--

## 2.6 sign-up.page.ts

Class Type	Boundary Class	
Responsibility	Controller to let the user to sign up into the system	
Attributes	Attributes Name	Attributes Type
	Email	string
	Password	String
	passwordConfirm	String
Methods	Method Name	Description
	signUp()	Let the users sign up into the system
Algorithm	<p><b>signUp()</b></p> <p>START</p>	

	<p>STORE user's credential AS credentialForm</p> <p>CHECK user's credential VALIDITY</p> <p>IF VALID</p> <p>Modal DISMISS</p> <p>SHOW confirmation message</p> <p>IF NOT VALID</p> <p>Modal DISMISS</p> <p>SHOW error message</p> <p>END</p>
--	--

## CHAPTER 3

### 3.1 DATA DICTIONARY

#### 3.1.1 Users Table

Table 1 Users Table

Field Name	Description	Data Type	Constraint
UserID	User ID	String	PK

email	User's email for login or signup	String	-
-------	-------------------------------------	--------	---

### 3.1.2 Notes Table

Table 2 Notes Table

Field Name	Description	Data Type	Constraint
NoteID	Note's ID	String	PK
userID	User's ID	String	FK
title	Main title for a specific note	String	
text	Description for the specific note	String	
date	Date when the note is created	Date	
alert	Conditional boolean if the user wants to be reminded or not	String	

### 3.1.3 Finance Table

Table 3 Finance Table

Field Name	Description	Data Type	Constraint
financeID	Finance's Id	String	PK
userId	User's Id	String	FK

expense	Condition if the income is expenses or not	String	
month	Month of the income that has been selected	Integer	
value	Value of the month that will be inputted	Integer	

### 3.1.4 DetailsFinance Table

Table 4 DetailsFinance Table

Field Name	Description	Data Type	Constraint
dfID	Detail Finance's ID	String	PK
financeID	Finance Table's ID	String	FK
title	Title for the specific finance information	String	
notes	Description or notes for the selected finance	String	
date	Date on the created finance	Date	