

Department of Electrical and Communication Engineering

Session: 2021-22, Semester B

BSc (Honours) in IT Management for Business

MHW225671: Honours Project

Final Report

Project Title : Financial management mobile application for college students in Oman

Student Name : Mohsina Firdous .K

Student Number : 180096

Supervisor's Name : Dr. K.Vijayalakshmi

i. ABSTRACT

College students confront various problems and develop many skills that will help them in their college adventure and future, but one of the essential skills students need is to handle their funds properly. The project is proposed to develop a personal finance mobile application for college students in Oman that will help them manage their money and provide valuable insights based on their data. The application increases students' financial awareness and literacy while encouraging young people to make prudent financial decisions. Various periodicals were studied for the financial management tools and approaches necessary for students. Predictive machine learning algorithms proposed by earlier authors for financial planning are used to examine the many financial management techniques and tools needed to improve financial management in general and for college students. The agile development methodology was utilized to develop the application. Android Studio and Flutter were used to develop the application and Firebase as a database. The application is separated into sub-applications, each of which promotes financial awareness and better decision-making. The home page integrates the whole program and displays a summary of the user's financial progress. The program produces beneficial insights into their current and future financial success using data analysis and machine learning while also exhibiting their last progress for their reference. The app also helps students save money and take advantage of their status by displaying deals and student discounts in their immediate vicinity. Financial literacy is further fostered by including links to financial articles that promote improved financial planning suggestions and practices. Furthermore, all application features are successfully tested to ensure that the program meets its requirements. The application successfully used the user's data to produce beneficial insights and used machine learning to predict the user's upcoming month's income and expenses using linear regression from flutter libraries. A successful application that analyses and indicates the student data was developed to prepare college students to be better financial planners and decision-makers before entering the real world.

Key Words: Financial Management, data analysis, machine learning, Financial Literacy, Students

II. TABLE OF CONTENTS

ACKNOWLEDGMENT	Error! Bookmark not defined.
i. ABSTRACT	2
iii. LIST OF FIGURES.....	5
iv. LIST OF TABLES.....	7
1. CHAPTER 1 INTRODUCTION.....	8
1.1. Project background.....	8
1.2. Project Motivation.....	9
1.3. Aim.....	10
1.4. Objectives	10
1.5. Project Scope	10
1.6. Project Methodology.....	10
1.7. Project Feasibility.....	11
1.8. Project Challenges	12
1.9. USDG goals.....	12
1.10. Summary.....	13
2. CHAPTER 2 LITERATURE REVIEW	14
2.1. Financial management for college students	14
2.2. Mobile application development for personal finance	18
2.3. Machine learning for financial management	20
2.4. Literature review in comparison with the proposed project	23
2.5. Summary.....	24
3. CHAPTER 3 PROJECT DESIGN AND METHODOLOGY	25
3.1. Software Development Lifecycle	25
3.2. System Requirements Analysis.....	27
3.2.1. Software Requirements.....	27
3.2.2. Hardware Requirements	28
3.3. Design	29
3.3.1. Project Prototype.....	29
3.3.2. User Interface Design	30
3.4.1. Flow chart	41
3.4.2. Database Design	42
3.5. Data Collection	43
3.6. Summary.....	43

4. CHAPTER 4 IMPLEMENTATION AND TESTING	44
4.1. Database implementation	44
4.2. Application implementation.....	47
4.2.1. Login and registration.....	48
4.2.2. Home page.....	55
4.1.3. Budget app.....	61
4.1.4. Offers	75
4.1.5. Financial read.....	77
4.1.6. Profile page.....	78
4.1.7. Admin page.....	79
4.2. Testing	81
4.3.1. Test cases.....	81
4.3. Summary.....	85
5. CHAPTER 5 RESULTS AND DISCUSSIONS.....	86
5.1. User Data Analysis results	86
5.2. Machine Learning	89
5.3. Application comparison.....	94
5.4. Summary.....	94
CHAPTER 6 CONCLUSION	95
CHAPTER 7 FUTURE WORK.....	96
GANTT CHART	97
REFERENCES.....	98
APPENDIX 1: PROJECT PROPOSAL (APPROVED)	Error! Bookmark not defined.
ORIGINALITY REPORT.....	Error! Bookmark not defined.

III. LIST OF FIGURES

Figure 2.1Set of Data for House Bookkeeping and Personal Finance (Rukhiran and Netinant, 2020)	19
Figure 2.2Execution rule on three dimensions(Rukhiran and Netinant, 2020).....	19
Figure 2.3three layer User Interface Composition model(Rukhiran and Netinant, 2020).....	20
Figure 2.4equation used for the model(Kumar, 2021)	23
Figure 2.1Seven phases of SDLC (Jevtic, 2019)	Error! Bookmark not defined.
Figure 2.2 Agile development lifecycle (Khan, 2020).....	26
Figure 2.3 Prototype design flow chart (self,2022)	29
Figure 2.4 Home Overview (self, 2022).....	33
Figure 2.5 Home Stats (self, 2022)	33
Figure 2.6 Home Menu (self, 2022).....	31
Figure.2.7 Income transaction (self, 2022),Figure 2.8 Expense transaction (self, 2022) Figure 2.9 Liability transaction (self, 2022)....	32
Figure 2.10 Income Statistics (self, 2022),Figure 2.11 Expenses Statistics (self, 2022)	33
Figure 2.12 Budget Data (self, 2022),Figure 2.13 Loan Data (self, 2022)	34
Figure 2.14Reminder page (self,2022)	34
Figure 2.15 Add Expenses (self, 2022),Figure 2.16 Add Income (self, 2022) , Figure 2.17 Add Liability (self, 2022), Figure 2.18 Add Reminder (self,2020	35
Figure 2.19 Offers (self, 2022) , Figure 2.20 Student discounts (self, 2022)	35
Figure 2.21 Financial Reads (self, 2022).....	36
Figure 2.22 Login Page (self, 2022)	37
Figure 2.23 Create Account (self,2022), Figure 2.24 Student Information (self, 2022)	38
Figure 2.25 Income Page (self,2022), Figure 2.26 Expenses Page (self, 2022) , Figure 2.27 Liability page (self, 2022).....	39
Figure 2.28 Budget Page (self,2022) , Figure 2.29 Savings page (self, 2022)	39
Figure 2.30 Admin Home page(self, 2022)	40
Figure 2.31 Profile Page (self, 2022).....	40
Figure 2.32 Flowchart application (self,2022).....	41
Figure 2.33 Firebase Database Design (self,2022)	42
Figure 4.1 Firebase Console (self,2022)	44
Figure 4.2 Firebase libraries in flutter (self,2022).....	44
Figure 4.3 Authentication user database	45
Figure 4.4 Firestore Database (self,2022)	45
Figure 4.5 User data path (self,2022).....	46
Figure 4.6 Flutter Libraries (self,2022)	47
Figure 4.7 Main page code (self,2022).....	48
Figure 4.8 Login page code (self,2022)	49
Figure 4.9 Error messages using flutter toast.....	50
Figure 4.10 User credentials validation (self,2022), Figure 4.11 Validating User data (self,2022).....	51
Figure 4.12 Pages redirected from login(self,2022).....	52
Figure 4.13 Text field error in create account (self,2022).....	53
Figure 4.14 user Authentication code.....	53
Figure 4.15 Validation code	54
Figure 4.16 Login and registration files (self,2022)	54
Figure 4.17 User Registration Procedure	55
Figure 4.18 Adding data to the flutter database (self, 2022).....	55
Figure 4.19 Home page body (self,2022)	56
Figure 4.20 Home Page Body change code (self,2022).....	57
Figure 4.21 Home page design(self,2022)	58
Figure 4.22 User Data Access(self,2022).....	58
Figure 4.23 Navigation bar(self,2022)	59
Figure 4.24 Navigation Bar code snippet(self,2022).....	59

Figure 4.25 Overview elements code snippet (self,2022)	60
Figure 4.26 budget app bar (self,2022)	61
Figure 4.27 Budget app code(self,2022)	62
Figure 4.28 Transaction page(self,2022)	63
Figure 4.29 Transaction add(self,2022)	64
Figure 4.30 Transaction data access and classification (self,2022)	65
Figure 4.31 Edit and delete data (self,2022)	66
Figure 4.32 Statistics page(self,2022)	67
Figure 4.33 bar chart code (self,2022)	68
Figure 4.34 Pie chart code (self, 2022)	68
Figure 4.35 Monthly budget progress (self,2022)	70
Figure 4.36 Monthly budget page code snippet (self,2022)	71
Figure 4.37 Reminder page (self,2022)	72
Figure 4.38 Reminder page code snippet (self,2022)	73
Figure 4.39 Add reminder code	74
Figure 4.40 Offers page(self,2022)	75
Figure 4.41 Element code (self,2022)	76
Figure 4.42 Financial Reads (self,2022)	77
Figure 4.43 Profile page (self,2022)	78
Figure 4.44 Admin Page(self,2022)	79
Figure 4.45 Add information snippet(self,2022)	80
Figure 5.1 Income and expenses data analysis(self,2022)	86
Figure 5.2 Progress Bar Income and expenses (self,2022)	87
Figure 5.3 Savings progress (self,2022)	87
Figure 5.4 Loan progress (self,2022)	88
Figure 5.5 Machine learning prediction in home stats(self,2022)	89
Figure 5.6 Income prediction code (self,2022)	90
Figure 5.7 Machine learning algorithm (self,2022)	91
Figure 5.8 Predicted value container code (self,2022)	92

iv. LIST OF TABLES

<i>Table 2 . 1 Literature review comparison table</i>	28
<i>Table 5.1 Application Comparison Table(self,2022).....</i>	98
<i>Table 7.1 Gantt chart.....</i>	101

CHAPTER 1 INTRODUCTION

1.1. Project background

In present times, college-going young students have gained a reputation for being poor money managers due to their irrational and impulsive spending habits. Furthermore, their low financial literacy and lack of financial planning have led them to increased debts and inadequate savings amounts in their banks (Widener, 2017). These demotivate the students financially and keep them away from being financially independent or self-sufficient before completing their college or university degrees. Financial management refers to the behavior and perception of how well the finances and managing of money are done (Azer and Mohamad, 2018). Knowing financial management is very important as having money is not enough to guarantee financial independence. Still, the proper knowledge to manage or take care of the money will help the students be self-sufficient (Styles, 2018).

Though financial management is essential for all age groups demographically, college students have different challenges as they are at a time of drastic transformation in their lives, especially as they transform from a teenager to an adult between the ages of 18 to 25. These students usually take student loans from banks for their education and are expected to pay back their education loans. Due to the lack of proper financial planning, students tend to have increased debts through loans and other borrowings and later find it difficult to pay back the loans (Syakirah, n.d.) According to Student Loan Hero, over 60 percent of students fear paying back their student loans (Dickler, 2017), which could further increase due to the pandemic situation and economic problems. Students are further pressured to be independent and well-settled to establish an identity in society (Azer and Mohamad, 2018). But the thought of financial independence and timely repayment of debts further adds to their pressures as they are usually unguided and financially uneducated on managing their finances. This further leads them to do things on an impulse that may lead them to poor planning and heavy borrowing such that they have more to pay or give out than their annual or monthly income (Widener, 2017).

According to iotuition, less than 45% of their university students believe they understand basic financial concepts, while more than 50% believe they do not practice or understand financial management. (iotuition, 2016) This data is alarming as people with poor financial management or knowledge tend to make wrong financial decisions in their lives. The ability to manage their finances is essential for our everyday lives. A financially educated person can make better decisions that lead them to higher financial security and well-being (Falahati et al., 2011).

Though the problem seems enormous and dangerous, there are various ways to gain financial independence. There are many financial practices that can help students be responsible planners, which can further help towards a better future (Styles, 2018). A student needs to learn proper budgeting. It gives

the student conscious control over their spending and is a vital financial management skill required for financial freedom (planning et al., n.d.). Essential financial practices such as saving, investing, etc., can further help enhance the student's financial life and should be common knowledge. Even a basic understanding of such financial practices can help college students in numerous ways and will be the beginning of the right financial path (Styles, 2018). Though all these techniques seem simple, they are usually not easy to track down and manage all the time. Furthermore, without proper financial guidance, students frequently end up causing more financial harm, particularly through the use of credit cards and a lack of emergency funds. All of this can be managed and assisted with the help of better tracking and management systems, and, as technology has helped to change most lives in a positive way, it is important to use it to enhance the financial life of college students as technically they are the country's future and present assets that need to be guided properly so that they can contribute positively to society (Styles,

Thus, the financial management mobile application is a personal financial management app that focuses on financially helping college-going 'young' students to manage their finances with the help of something they know very well, technology and, in this case, a mobile app (Solution Analysts, 2020). The mobile application aims at assisting college-going young students to become more financially literate and become mindful spenders by helping them budget for their home, college, and other expenses and debts by giving them insights into their spending habits and with the help of technology and data. Thus, helping students to become self-sufficient in their lives. (Planning et al., n.d.)

1.2. Project Motivation

It's not always about how much money you make that makes you financially independent; it's about how effectively you finance the money that may make or break your financial independence (Ruth, 2015). As college-going young students start their journey to be financially independent, there are many ups and downs and also a lot of traps that lead them away from their financial dream. Common reasons for this issue could be due to ignorance and failure to plan financially, lack of financial literacy, and accumulation of unnecessary debt to cover their expenses and so (Williams, 2008).

The main motivation is to bridge the gap in financial literacy among college students with the help of data and technology so that they may be financially ready and prepared before graduating. In countries like Oman, where students are being encouraged to be skilled, which could also include pursuing higher education to promote and contribute to the country, it is important to prepare them financially as they do take loans for education and other expenses (TAS News Service, 2021). Thus, proper and disciplined financial planning at the beginning can keep them away from the depths of debt and also help them in their careers with the confidence that they have the knowledge and skill to handle their own money in such a way that they can avoid many of the mistakes and live financially independent.

1.3. Aim

To develop a mobile application for college-going young students to monitor their spending patterns and help them become more financially literate by providing insights into their spending habits and guiding them to save and learn.

1.4. Objectives

- To conduct research on the different financial planning techniques and software applications for financial management and promote financial literacy among students.
- To investigate and contrast different software development life cycles and mobile application development softwares.
- To design and develop a financial assistance mobile application powered by machine learning using Android Studio and Flutter and make it easy to use for students new to financial planning.
- To test the application with a small group of participants and test it with a small group of participants and compare its efficiency with existing applications.

1.5. Project Scope

In the era of finance and technology, it is important to provide access to easy finance to everyone, and the youth or the newly graduated youth who are considered the future of a country should be prepared at the beginning. Moreover, the technology of predicting the next move with the help of machine learning can help the student be ready beforehand. Though there are financial assistance or financial management application that target college students like mint etc., there are none targeted to students in Oman, which can hinder a student from using and gaining knowledge from such useful applications. Through the project and application, we aim toward the college students in Oman to help them manage their finances through budgeting and also use machine learning to give them future insights into their spending and saving habits.

1.6. Project Methodology

A hybrid development and testing approach will be used in the project since it reflects on the spending habits of the students and thereby allows the students to improve their financial literacy and become stronger and more conscientious spenders in the future

At the beginning of the project, a literature review is done to study the various financial management techniques and tools used to enhance financial literacy and independence. The literature review will help in gathering user requirements, which will allow one to get a better understanding of the problem domain of college students' financial needs and knowledge through research. The literature review also focuses more on previous authors' works or journals on the machine learning part in financial budgeting, especially in organizations, and furthermore will discuss how they can be used in the existing system to achieve the aim of the project.

The next stage is to design the application's front, and back ends. The literature review stage will serve as the foundation for the design, development, and testing phases. During the design process, the first focus is on the functionality and user interface design to get a better view of the application's features and functionality as a whole (BuildFire, 2016). The front end will serve as a reference for designing and planning the application's backend.

During the development stage, the framework is built using the selected mobile application development framework and database application while maintaining functionality and upgrading and fixing any flaws in the plan or design. The machine learning part of the project will be separately developed and then integrated into the mobile application.

In the software development lifecycle, the testing process is critical. At this point, a review will be done to see how the program still functions after integrating the front and back ends, as well as if any of the user specifications have been fulfilled. During this process, the code is tested for any bugs or errors and to make any necessary changes to the application.

1.7. Project Feasibility

The financial management system is a mobile application with machine learning that takes the financial data from the user on a monthly basis and displays their finances with the help of budgeting. It also predicts the amount they would save and spend for the next month and further promote them to budget and save the amount for further use. The application is built and personalized for students in Oman and shows them the possible offers and saving present in Oman.

The project requirements and functionality are feasible as we are using android studio with Java and SQLite for the database. The machine learning part of the application will be developed using python and will be implemented into the android studio by converting the model to tensor flow lite.

The project is aimed toward college students, especially in Oman, and also to make their lives easier with the help of tracking and budgeting their finances which almost every student, especially those with student loans, will find the application practical and helpful.

1.8. Project Challenges

There are many challenges that are faced in the project, and the following are the potential challenges:

- Inadequate data for the machine learning algorithm that identifies students' spending and saving patterns. Inadequate data is a very common problem in machine learning, and there are very few datasets that track the spending habits of working adults, and there is a high chance of not finding any spending data of college students (Stewart, 2019). Thus in such a situation, there will be an effort
- Made to acquire a sample dataset drawn from other research projects, such as those found on Kaggle. If no such datasets are found, then with the help of data acquired from the user, a linear regression model will be used to make the needed predictions.
- Failure to select the appropriate software for developing and deploying the application. Certain IDEs (Integrated Development Environment) have a good interface to work with, while some have a very complicated interface which can lead to confusion when developing an application. For the project, it's also crucial to have an IDE that can integrate machine learning algorithms in it; thus, a good and huge support community is needed while developing the application. Thus IDE and development language with the most active support community will be chosen (e.g., Android studio)
- One of the most common issues in application development is Time limitations, and poor code quality can add up to the time issue. In such cases, it is better to work within the time limit and use Gantt charts, Work breakdown structures, and Object-Oriented diagrams to ensure that the application is completed on schedule. It is also important to be realistic with time and application features and focus more on the specializing a feature in the application than feature overloading the application (Council, 2019).
- Receiving the right data from the user, such as receiving the correct amount of money as income and money spent, is very important to show and predict the amount for the present and next month. If any improper data and or no data is given by the user, it could affect the algorithm of the application and may display wrong data to the user affecting the functionality of the application.
- Complex algorithms and a lack of understanding of a certain subject and the code associated with it. To ease the coding process, the application will be developed into smaller modules and integrated.

1.9. USDG goals

The project focuses on enhancing the financial literacy of college students that will help them to be financially independent, which basically means to be well off financially while reducing issues like poverty and inequality in society. Thus the USDG goals met by this project are the eighth and tenth USDG goals.

The tenth goal aims at reducing inequality within and among countries which can be achieved by making citizens of the country financially aware and literate. The eighth goal is to promote sustained, inclusive, and sustainable economic growth, full and productive employment, and decent work for all as through the project, we aim to improve the citizens financially, which will further promote the economic growth of the country and economic stability (United Nations, 2015).

1.10. Summary

The introductory chapter is the first chapter, and it is intended to introduce the reader to the topic and goal of the project. The project background, in the beginning, throws light on the problem statement and also provides insight into how the project plans to tackle the problem. The aim and objective provide a concise introduction to the project topic and how the project goal will be achieved. The project motivation and scope provide a small insight into why this problem statement or topic was chosen and how beneficial it may be to youth development. The project feasibility statement describes how the project can be completed with the resources available, and the project challenges provide a comprehensive problem and a potential solution to the problems that may affect the project's completion. In the conclusion, the USDG goals are used to demonstrate how the project may benefit the world

CHAPTER 2 LITERATURE REVIEW

A literature review is an in-depth examination of prior work done by various researchers and writers on a certain topic (Royal Literary Fund, 2014). This research focuses on the project objectives, methods, and findings from the investigation. The chapter is divided into three parts. The first part is reviewing financial management for college students. The second part reviews mobile application development for personal finance, and the third part reviews machine learning.

2.1. Financial management for college students

The article produced by (Widener, 2017) outlines the issues highlighted about finances and why students struggle to keep up with their finances. The author examines prior publications and journals and provides a thorough literature review centered on the money management challenges that college-aged students face, as well as the important elements and consequences of their financial mismanagement. The author also highlights the need to know key relationships in order to train the future generation to be more financially literate or to be able to handle their finances properly. The author also mentions that there are several studies on personal finance. However, very few are aimed toward college students and their financial mishaps, emphasizing the importance of the presented problem.

According to the author, financial literacy plays a significant role in influencing money-making decisions, and financial situations also play a vital role in how well students understand money. Financial literacy is an essential aspect of everyone's life, but students fail to understand it. The lack of financial awareness also plays a crucial role in making decisions regarding money. Some students know their essential finances but lack the awareness of the present changes in the economic facts and numbers, such as interest rates, which leads them to miss out on opportunities that could have helped them lower their expenditures. In the paper, according to the study by Koposko et al. (2016), it is said that students were highly aware of the importance of planning their financial futures and their retirement. However, when the researchers questioned, they realized that students were not committed to putting their financial knowledge into practice. Another reason why students fail to be financially independent is due to their lack of seriousness about the consequences of bad financial planning or not planning their financial futures at all. However, countering the previous study, Goetz et al. (2011) discovered a demand for personal finance courses among the students, confirming that students have both the desire and the awareness to gain financial knowledge. Some students want to make an effort to be financially aware, while there are also those who, even after having the necessary knowledge, fail to use it for their futures, which again could be for reasons such as financial background and lack of awareness. The paper further highlights that students are not up to date with financial management and lack the discipline for proper financial planning, and are very prone to overspending (L. Archuleta, Dale, and M. Spann, 2013). At the same time, another common problem for overspending is the student's reliance on credit cards which further increases their expenses and debt. Their low income and tendencies to spend irrationally with credit cards lead them to a financial disaster

(Goetz et al., 2011). The author also states that financial terminologies such as saving, retirement, etc., are not common among students, which reduces their need for that knowledge. They do not understand the importance of starting financial planning at the earliest stage possible.

The author goes on to say that financial factors have an impact on students' attempts to comprehend their money. According to the author, demographics, family influence, and backgrounds all have a significant part in a student's desire for financial literacy and independence. Furthermore, the author discusses how students' financial decisions are influenced by their lack of parental support, their own financial choices, or both. Students' misuse of resources might be connected to a lack of parental support because they did not have to undertake any financial planning when living with their parents. Their transition from dependent to independent frequently leads to hasty decisions, and sound financial planning may help avoid excessive spending. The debt aspect also has a severe impact on their financial path, particularly in college. According to the author, students who stay in college for a longer period of time have greater debt than those who do not continue in college for a longer time. Rather, there is no solid evidence that certain degrees may aid students with their practical financial knowledge; however, finance students demonstrated a great interest in the importance of knowing and learning personal finances. (Widener, 2017).

In the end, all these negative financial influences have a consequence on their present and future finances and life. The societal need to be financially independent and the lack of financial knowledge often lead students to the inherent risk of being in debt even after completing their degree. They also have to bear the risk of not paying back the loan if they are unable to get a promising job after the degree. Moreover, according to the various researches mentioned in the paper, the consequences that the students face due to their lack of financial planning lead them to have various stress-related health issues affecting their mental and physical health, which can further disrupt their financial behaviors and which go on affecting other aspects of their life such as their social life and even their careers. Being in debt even after completing their degree can affect a person's financial well-being and security which end up having a negative psychological impact that continues to affect a person's confidence in managing their finances. The author then concludes the need to understand the student's approach to money and why it's important to set up the coming up generation for success (Widener, 2017).

The article underlined critical data and studies that underscored the need for financial planning and management for college students. The author emphasized the study's limitations, such as the association between a student's relationship status and faith with their financial practices. Furthermore, the author stresses that only a few papers talk about personal finances targeting college-going young students and the growing youth. Overall the paper contributes to the need for the topic of the project but lacks a solution. The study discusses what students encounter due to their financial decisions, but it does not discuss how to prevent these financial catastrophes in a student's life. The author also misses the technical side that could help in a person's finance and falls short of discussing the tools that students might require to live a financially secure lifestyle. The study (Styles, 2018) investigates young student's level of financial literacy and focuses on the growing debt levels among college students in the United States, as well as how a lack of knowledge, motivation, and implementation of financial

management applications has contributed to the debt increase among college students. The author discusses financial management and the necessity of financial management in the paper, as well as the many forms of debts and the most prevalent sorts of debts utilized by college students. In addition, the author provides a brief history of debt and how it continues to influence students today. The author also discusses financial literacy and what remedies exist to prevent such mounting debts and improve financial management among college students.

When the author understood that this is a common issue among students his age and that the great majority of Americans, mainly, are financially illiterate when it comes to financial management, his interest in financial management grew. According to the author, this might be due to a lack of parental direction from the outset and a failure to comprehend and apply basic financial ideas (Kim and Chatterjee, 2013). The author (Widener, 2017) also mentioned this as a common problem among students. According to the author, there is a link between effective financial management and lower debt levels. One cannot thrive in a society without money, emphasizing the significance of financial management among all people, not just college students. There are other financial management methods that one may employ; nevertheless, the author concentrates on saving in the research. Saving is considered one of the most important financial management practices. Many people wind up spending on large goods without saving any money in the end. The author discusses how difficult it is to survive only on one's income at any one moment and how savings might assist prepare for unforeseen futures or disasters. Though saving reduces the amount of money available now, forcing people to live within their means, the inconvenience will save them significantly. While saving is an excellent financial management technique, the author neglected to include inflation and the discounting factor of money, which may further diminish the desire to save among the youth. As a result, advising the youngster to invest in areas that will build their savings and keep them up with inflation will improve their financial well-being and preparation. Budgeting is another practice mentioned by the author. Budgeting is also a highly important technique since it helps plan how to distribute and manage one's money, as well as how a budget may assist monitor a person's spending habits. Budgeting helps prevent overspending since the user is more aware of where their money is going and receives notifications on how much they might have saved if they had budgeted. Budgeting is a critical part of boosting the student's knowledge of the project application. Investing is another management tool meant to increase one's financial assets. Though present with risks, various opportunities are available to support safely. The author goes on to discuss how paying bills in full and on time may help decrease future debts and how not paying bills might harm a person's credit score (Styles, 2018).

The author also mentions credit card usage and a lack of emergency money, which may be readily minimized by employing the financial management practices outlined by the author. Unfortunately, credit cards are nothing more than a deception that money is available for free, while in fact, it is borrowed and must be repaid at a higher interest rate. This creates a debt cycle that can be difficult to break and disastrous. According to the author, fewer people practice good and beneficial management while some don't know how to or choose not to live, having that financial mindset in order to have a specific lifestyle (Comai-Legrand, 2019). There are several types of debts, such as unsecured debt, which is a typical type of debt that does not need any assets or collateral, such as a credit card, and secured debt, which is secured by existing assets and decreases the lender's risk, such as

bank loans. According to the author, secured debt is more suggestible in terms of debt since, when handled appropriately, it may be a great help because it has fewer risks than unsecured debt. On the other hand, unsecured debt has a more significant possibility of costing more than the borrowed amount. While student loans are the most prevalent sort of secured loan taken out by students, there are several risks compared to the rewards of the borrowed money. College degrees, mainly undergraduate degrees, have less demand in the job market than in previous years, so students cannot guarantee well-paying jobs that can help them repay their loans quickly (Hollis, 2014). Furthermore, students do not have the opportunity to declare bankruptcy, which in the future can overburden the students and lead to other issues, as mentioned in the study before. While debt is not a new phenomenon, credit cards are a recent debt problem that arose as a consequence of the ease and convenience that is also a revolving element that has led to the severe growth in debts over the last several decades. The author attributes this to a lack of financial literacy, which has contributed to the increased usage of credit cards and debt; as the demand for credit cards increased, the understanding of how to use them efficiently and responsibly did not. Furthermore, the ease with which student loans could be obtained had a significant influence on America's debt issue. Finally, the issue is not debt but the creation of 'ease' of debt, which has had a long-lasting influence, particularly in America today (Styles, 2018).

Finally, the author proposes a possible answer to the problem is making financially illiterate individuals literate. This method has been repeatedly emphasized in the literature study. However, the question is whether it is as simple as it sounds to implement the solution. According to the author, the answer is complex since studies done by (Lusardi and Mitchell, 2014) have determined that with the little research available, there have been flaws found in the data and experiments to establish that financial literacy has a correlation with good financial behavior have been discovered. With the information available, it's difficult to infer that there is any good association between financial behavior and intelligence. The author believes that the lack of research and evidence is the reason why the students still struggle with financial literacy and good financial behaviors. The author concludes by stating that, because the right way has yet to be discovered, it is critical to teach students personal finance, including the financial management techniques suggested by the author, at the high school level, as this will help build awareness among the students, which can lead to a good financial culture.

In the paper, the author (Styles, 2018) focuses on the debt issue in America among college students and has statements supporting the article by (Widener 2017). The author further details why credit card usage is detrimental and how parental influence may play a key role in promoting financial literacy among youth. The solutions provided in the paper supplement the approaches required in the application to assist students in better managing their finances and discuss several additional studies on why there is a need for further research and experimentation on financial literacy among college students. Furthermore, the paper is another literature review that discusses different approaches to aid students with financial illiteracy but ends with poor backing due to a lack of research and data.

2.2. Mobile application development for personal finance

The article "A practical model from multidimensional layering: personal finance information framework using mobile software interface operations" by (Rukhiran and Netinant, 2020) proposes the development of a personal finance application using multidimensional layering that mainly focuses on the user experience. The application is illustrated in a three-layer user interface composition model. The author suggests that an integrated view of the software system would make the design and implementation consistent with supporting the framework. The application has a practical framework that has all four phases of analysis, design, implementation, and evaluation. The author's primary research is based on developing a framework that can result in simple, enthusiastic, and aesthetically operative information, function, aspects, and layers with higher and better separation of concerns.

The authors state the importance of User Interface (UI) and its importance in the software development cycle. However, as developers focus more on designing UI, it also leads to a complex user interface. According to (Hays, 2014), complex user interfaces can encounter delayed reaction time and unsatisfied users, affecting the functionality of the application. Thus developing a UI based on the needs of the users has a very beneficial role in software development (Rukhiran and Netinant, 2020).

In simple words, separation of concern is divided as a part of the software that represents a single functionality. An aspect-oriented approach such as the Aspect-Oriented Software Development (AOSD) is used (Netinant and Elrad, 2016). The AOSD approach reduces duplication and dependency codes; the evolution strategy of AOSD focuses on expressing the rules and definition of events, conditions, and actions for supporting changes in computation environments. This evolution is concerned with running time. Thus the separation of concern can result in a reusable, extensible, and adaptable system (Rubio

et al., 2005). According to the author, breaking software into smaller pieces is the solution to allow achievable data and reusable functions. The application is first divided into two segments. One segment is the multidimensional layering of the personal-finance information, and the other is the aspect element to operating functional methodology. The three-dimensional layering enabled flexible and adaptable designs. In the three-dimensional layering, the layers hold the coordinates X, Y, and Z that belongs to expenditure, income, and liabilities, where each dimension is further divided into sub-dimensions. The available data is correlated to the system information that cuts across the layering of three dimensions. Figure 2.1. shows how the application house bookkeeping is divided into three concerns.

Dimension	Sub-dimension	Functional data
Income (I)	Earned Income (EI)	working, owning a business, consulting, gambling
	Portfolio Income (PoI),	trading paper assets, selling real estate, investment
	Passive Income (PaI)	rental income, bonus, insurance, retirement, interest, bank interest, stocks
Expenditure (E)	Daily Expenses (DE)	food, transportation
	Personal Expenses (PE)	clothing, travel, sports, books, social & entertainment
	House Expenses (HE)	mobile phone bill, Internet, repairing equipment, parking fee
	Family Expenses (FE)	tuition fee, alimony, medical fee, donation
Liabilities (L)	Current Liabilities (CL)	credit card debt, home equity loan, interest, taxes, rental mortgage
	Long-term Liabilities (LL)	bonds payable, notes payable, bank loan, deferred revenue, mortgage

Figure 2.1 Set of Data for House Bookkeeping and Personal Finance (Rukhiran and Netinant, 2020)

The functionality data records from one field to n-fields, and the aspect elements are defined as a set of computational properties such as insert update, delete, etc., which also starts corporately from more than one aspect to m-aspects. An aspect-oriented technique's main function is to decompose concerns and methods into aspects.

The operational semantics used for the three layers were analyzed using components as shown in figure 2.2, which is the execution rule on the three dimensions as aspect elements in the case study done by the authors.

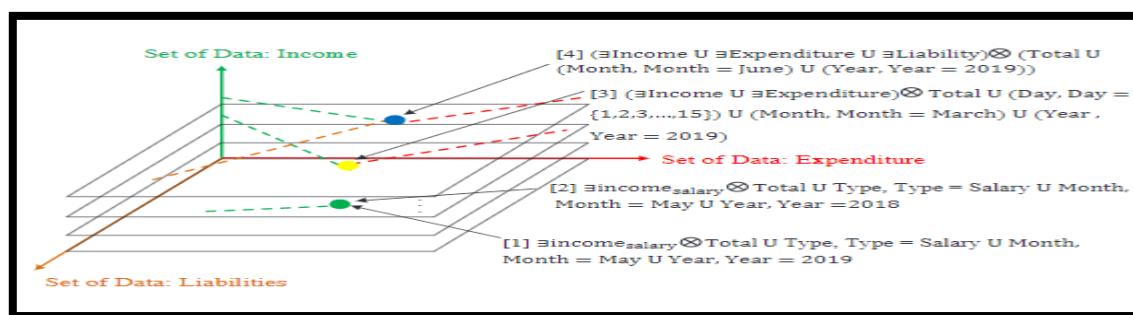


Figure 2.2 Execution rule on three dimensions(Rukhiran and Netinant, 2020)

To further support the design of the multidimensional information layering, figure 2.3. Represents the architecture of the three-layer User Interface Composition Model. Here the UI layer on the top facilitates the visual page that allows the end-users to interact with the devices. In contrast, the component layer provides components that

decompose the Functional Data(FD) and Aspect elements (AE). Finally, the data layer manipulates the data acquisition from the different calling stages through three-dimensional layering. This model responded to UI and gave a better performance that illustrates a cooperative UI design to separate different layers.

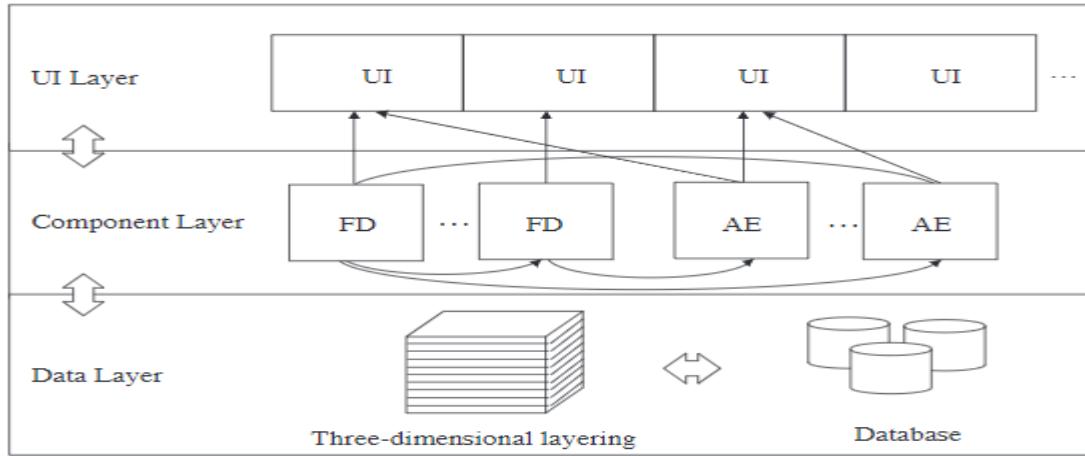


Figure 2.3 three layer User Interface Composition model(Rukhiran and Netinant, 2020)

The author, through the paper, introduces other types of mobile application development that focus on enhancing the functionality of the application while at the same time enhancing the User Interface through the separation of concerns method and the AOSD development cycle. The application promotes the development of the personal-finance mobile application and also gives a good idea of how such techniques can build an efficient and executable application. Although article related to personal finance in a mobile application is very scarce, and an article like this has only contributed to the project in a positive way, it lacks an explanation of the human-computer interaction part of the user interface and only on how to build an application, even though there were explanations about the need of a good user interface and why there is allot of time spent in that part of software development.

2.3. Machine learning for financial management

The journal "Predictive, regression, and model building for driver-based Budgeting" (Kumar, 2021) highlights the need for driver-based budgeting using the predictive model in machine learning businesses. It provides the steps to achieve predictive budgeting. According to the author, budgets are essential for administration control systems as it encourages communication and coordination within an organization. According to (Palchowdhuri and Roy, 2018), budgeting is the process of assigning financial values to the goals and targets that makes progress measurable and turn strategic concepts into simple, practical methods. Every corporation obtains goals and strategies through a long-term strategic plan, and budgeting helps corporations efficiently meet their goals. In the wake of technological advancements and how technology-enhanced the financial sector and planning, it becomes evident that traditional budgeting methods have become inefficient and expensive. According to the author, conventional budgeting methods don't focus on strategies and contradict each other, thus not adapting to economic and business changes. Therefore, the driver-based budgeting plan focuses on the critical business

drivers. These drivers allow organizations to make efficient decisions. According to the author, predictive modeling is used to make more precise and objective judgments than human experts (Kumar, 2021).

As Businesses need to adapt to changes constantly, it is evident that there should be alternatives to the traditional budgeting methods, one such being the use of predictive modeling and regression-based modeling. According to the author, simply putting all the data in the machine learning process is not the right approach. Thus, the following steps were highlighted by the author. The first step to developing a predictive budgeting model using machine learning is to get the quantifying data to budget and predict. This can help quantify the company's objectives to a single number. Next would be to include exploratory analysis obtained from the organization's IT assets and data. The third stage would be to construct a machine learning development sample where data analysts interpret the business data and its organization. Data is cleaned through the development sample by excluding invalid data, inconsistent data, and unexplainable data. The fourth stage is to prepare the data to identify any patterns. Once sufficient data is collected, the fifth stage is to pre-process and remember preliminary variable choices. Standardization and transformation are typically used for pre-processing the data into a format for the machine learning algorithm. The sixth step is model construction, which involves data presentation to machine learning software. The seventh step is the algorithm's execution and the prediction model's generation. As the goals have been identified, the last step is to identify the primary drivers to build a financial model to reach those goals (Kumar, 2021).

The journal highlights the steps to build a prediction model using machine learning which again helps in the development part of the project with the machine learning part. The journal also states how machine learning is helpful to businesses when it comes to making predictions and better decisions and how to implement them. However, the author does speak about replacing the human experts with the help of machine learning, which in itself is incomplete as human experts are the ones that prepare and analyze the data and also help present the data to the necessary stakeholders. The need for human experts with technology could have been explored as the amalgamation of both advanced minds can further enhance the organizations to make the right decisions. The paper also failed to mention what type of machine learning is good for budgeting and what type of output should be expected to interpret.

In the journal "Machine learning for financial forecasting, planning and analysis: recent developments and pitfalls," (Wasserbacher and Spindler, 2021) explains that as the availability of big data increases with new analysis tech available gives an opportunity for accurate financial planning and analysis, especially at a faster rate that generates value for a company. In modern companies, financial forecasts and planning for effective and efficient resource allocation are core in modern companies. This is also very crucial in a volatile and fast-moving business environment, and high-quality financial forecasting is a defining characteristic (Roos et al., 2020). The need for accurate financial forecasting in businesses to stay ahead has brought in machine learning that successfully applies to a variety of predictive tasks. The aim of the paper is to show how machine learning can be used in financial planning and analysis while also highlighting the advantages and disadvantages of using it.

The main goal of financial planning analysis is to inform and support the decisions of management, and there are different routes to do so (Oesterreich et al., 2019). Analyzing business environment and dynamics is an important part of financial planning, and analysis and insights generated with the help of analysis can help in the assessment of how these plans are likely to succeed. Financial planning and analysis rely largely upon quantitative analysis to generate forecasts and plans, whose inputs are sales, expenses, and balance sheet positions. The recent development and comprehensive spread in the IT systems over decades have increased the amount and variety of data available readily for financial planning and analysis. Big data and increasing digitalization will further promote the development of financial planning and analysis. The key output of financial planning and analysis is the financial forecasts and plans. According to the author, with all the tools and resources available, it becomes important to choose the right ones for financial planning, and machine learning is one such tool that can provide those. Machine learning methods are especially suitable for financial planning and analysis for forecasting as they focus on predictive performance. The methods manage to identify and generalize patterns that work well with the given data and identify any complex structures that have not been specified. Machine learning supports a high degree of automation in the generation of forecasts. Moreover, a large section of the easy to use and are readily available (Mullainathan and Spiess, 2017). Financial planning and analysis also provide recommendations for the design of financial plans and suggest corrective actions when deviations occur from the plans. According to the author, this requires causal inference techniques, which are quite different from forecasting. Even though machine learning does ease and give accurate predictions based on the data, it does not produce reliable estimates for the parameters that govern the relationship between input and output variables, and machine learning is typically not built for this. According to (Taddy, 2019), the machine learning community has pursued the aim of maximizing the predictive performance when compared to understanding the model parameter. Moreover, believing that a tool built for forecasting possesses the properties required for causal inference in economic applications can be misleading (Mullainathan and Spiess, 2017). It may also be necessary to sacrifice predictive accuracy for a correct understanding of the relationships that are relevant for making decisions about interventions (Athey, 2018). Some authors also highlight that the lack of understating the causal relationship is the fundamental obstacle for machine learning (Pearl, 2019).

In the paper, the author provides results based on a small simulation study to further understand the types of data and questions that companies with Financial planning and analysis could face. This section mainly focuses on how accurate the data is in the case of sales forecasting and also understands how Financial planning and analysis assess the effectiveness of promotional activities in generating sales. In the first simulation, the author evaluates the accuracy of the sales forecast with the help of a sample evaluation approach. For this, the author uses the first four years of data to build and train the forecast models; financial planning and analysis are then used to compare the forecast that has been generated by the models to the real values that were available in the last years' dataset. This approach has been described as a sufficient way to compare the results even though there are many other alternatives to it. The out-of-sample forecasting approach is evaluated for each simulation separately. The author used the following equation as shown in figure

(5.1)

$$y_n = \alpha d_n + x_n' \beta_p + \varepsilon_n$$

Figure 2.4 equation used for the model(Kumar, 2021)

The author then uses two forecasting approaches, one with the traditional linear regression based on the ordinary least squares (OLS) method and the second with the post lasso classic machine learning technique. From the simulation, the author concludes that OLS increases significantly between in-sample and out-of-sample data. In comparison, the post lasso RMSE is stable between in-sample and out-of-sample data. In the second simulation, the author distinguishes between forecasting and planning, where forecasting involves prediction while planning involves causal inference. The author also highlights that the confusion between forecasting and planning can be a major pitfall that should be avoided. Furthermore, specific approaches to causal machine learning have begun to gain traction as naïve machine learning applications can fail if gone beyond prediction.

The paper highlights how machine learning is used in financial planning and analysis for companies and how its accuracy in predictions can help enhance an organization's efficiency and proper management of resources. The author did their best to prove this through simulations and proved which techniques of machine learning as useful. However, the author did not explore the other types of machine learning, such as deep learning, which might give better results when compared to the techniques mentioned in the simulations. The author themselves highlights that they used a simple data generation process that does not complement the real-life trends and changes, which is a very common thing in business. Even with these limitations, the paper did give some very valuable insights into financial planning and analysis with machine learning.

2.4. Literature review in comparison with the proposed project

The following table compares the reviewed previous papers and the proposed system.

Features	(Widener, 2017)	(Styles, 2018)	(Rukhiran and Netinant, 2020)	(Kumar 2021)	(Wasserbacher and Spindler, 2021)	Proposed Project
Financial management for college students	✓	✓				✓
Financial management techniques		✓				✓

Mobile application for financial management			✓			✓
Machine learning for financial management				✓	✓	✓

Figure 2.1. Literature review comparison table

2.5. Summary

The chapter reviews the prior literature done by authors that are linked to the intended project scope and identify the debate related to the themes as well as the development process and the authors' findings. The best financial management strategies were studied in the literature review and are intended to be incorporated into the proposed application, and features of the application were introduced as a result of the review. There was also a discussion on developing mobile applications for personal finance and improving the application's functionality and user experience. There were literature reviews that went into detail regarding predictive machine learning. In addition, the chapter includes a comparison of the examined publications and the suggested proposal. Better methodologies and functionality were grasped and presented as a result of the literature study, which might be of tremendous use to the proposed project.

CHAPTER 3 PROJECT DESIGN AND METHODOLOGY

Selecting a software development approach becomes crucial given the time and resources available. In software development, the project design and methodology stages are essential because they determine how the project is prioritized and developed in a systematic manner. Software requirements, the initial step of software development, were evaluated in Chapter 2 through a series of literature review; this information aids in the emergence of software design and requirements. The software design and development model utilized are extensively outlined in this chapter.

3.1. Software Development Lifecycle

The Software Development Life Cycle (SDLC) is a collection of procedures for developing software applications. These phases break down the development process into manageable activities that can be assigned, done, and monitored. The SDLC is a method for evaluating and improving the development process. It enables a fine-grained study of each process phase. The Software Development Life Cycle essentially lists all of the steps involved in creating a software application. This helps to eliminate waste and improve the development process' efficiency. SDLC ensures that the project stays on schedule and is a viable investment for the business (Jevtic, 2019). SDLC is broken down into generally 6-8 stages which can be combined or omitted based on the project scope (Khan, 2020). However, without a thorough grasp of the software development life cycle's phases, it's difficult to execute it intelligently and effectively (Sajna, 2022). Figure 3.1 illustrates the seven stages or phases of SDLC.

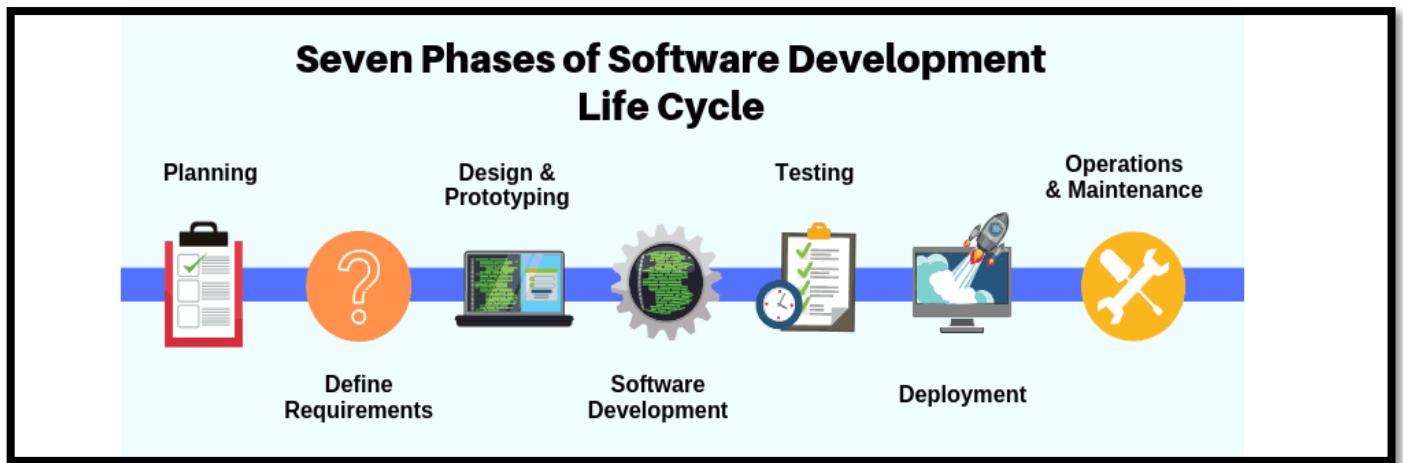


Figure 3.1 Seven phases of SDLC (Jevtic, 2019)

SDLC offers various unique methodologies, and finding the proper one is highly dependent on not just the intended output but also the project's criteria (Half, 2019). Based on the project size, type, and time the 'Agile' SDLC model was chosen. The agile model is adaptable, detailed, accurate, consistent, and

simple. This includes a pre-built framework that will aid in the development process. The agile process entails breaking down a large project into multiple iterations (Trunkett, 2020). In essence, it is planned and tested in stages rather than after the full product is completed. Furthermore, agile's strict planning, testing, and integration reduce the chances of a project failing. Agile development makes it easy to keep track of progress since much of the work has been completed rapidly. Agile methodologies provide rapid delivery; thus, they can be considered the best option for the tight deadline (Intergy, 2019). Figure 3.2 illustrates the stages involved in the agile development model. Figure 3.2 further highlights how adaptable the model is and also has room for any requirements changes and error correction.

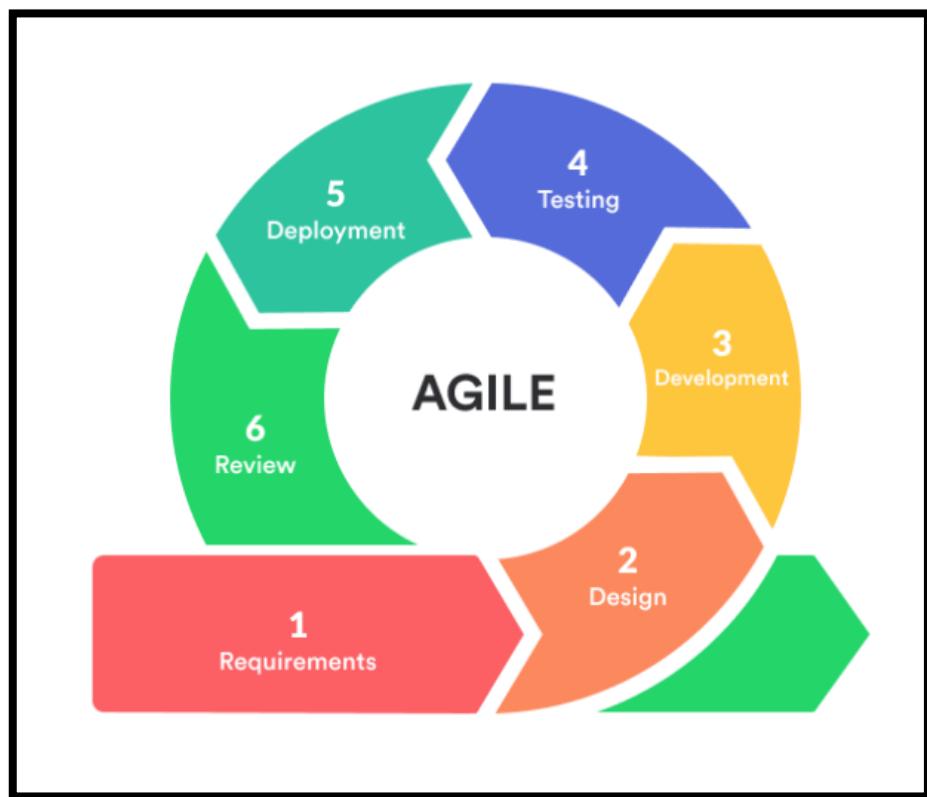


Figure 3.2 Agile development lifecycle (Khan, 2020)

The first step, as indicated in figure 3.2, is the requirements or brainstorming session, during which the project expenses, time, resources, and risks are assessed in order to determine whether or not to proceed with the project. All of these details were provided in Chapter 1 in order to get the project started. Requirements analysis is also done during the planning stage, as illustrated by the study of numerous literature reviews in Chapter 2. The prototype is designed using Uizard in the design stage, with all of the criteria given in the planning step. The chapter goes through this period in great depth. The software is produced in the following stage, which is the development stage. This stage, which takes the most time, is broken down into smaller projects such as a budget app, an offer app, a financial read, an admin page, and a home page. Flutter, Android Studio, and Firebase are used for coding and implementation. During the testing step, several tests will be performed on the completed application to see if there are any errors and to confirm that the program meets all of the

criteria. If any flaws or defects are discovered, the program will be updated. After that, the application is deployed and sent out for evaluation.

The agile methodology is adopted above other traditional techniques such as the waterfall model because of agile's flexible and collaborative approach across the SDLC, as opposed to the waterfall approach's rigid structure. Agile initiatives are more successful and less risky than waterfall projects, according to (The Standish Group, 2020) research, since agile programs have fewer problems and failures. Agile is also desirable because it allows for more continuing research and iteration, which helps to improve the program anytime a new concept arises. At the outset, the waterfall method rigorously follows study and planning (Ta, 2020). Though the waterfall model is simple and straightforward, it is fraught with risk and unpredictability. However, the rigorous structure of waterfalls aids in the classification and prioritization of tasks, which may not be the case with agile since new requirements and modifications made during the project may surpass the project's planned time limit (ExisTek, 2018). When implementing agile, a little rigidity is recommended while simultaneously making use of agile's flexibility.

3.2. System Requirements Analysis

In this section, the software and hardware requirements for developing the application will be discussed

3.2.1. Software Requirements

A software requirement is a condition or capacity that a program or system component must have in order to address a real-world problem. Problems might include automating a portion of a system, correcting flaws in an existing system, controlling a device, and so on (Thakur, 2013). The following are the software requirements to develop the application.

3.2.1.1. Uizard

Uizard is a rapid prototyping tool for automatically converting wireframes into prototypes, creating custom style guides, exporting as Sketch files, downloading frontend code, and iterating as quickly as feasible. It removes the issue and allows time to develop the low-fidelity wireframe into a high-fidelity wireframe (Ogunsola, 2021).

Uizard is used to design the User Interface of the application system. Through the app, the prototypes and wireframes are designed, which will be used to check the requirements of the application and used as a reference to develop the application in a more efficient way.

3.2.1.2. Android Studio

Android Studio is the Integrated Development Environment for Google's android operating system to develop Android Applications that are built on IntelliJ IDEA, a Java-based Integrated Development Environment for software. It includes code editing and developer tools. It employs a Gradle-based build system, emulator, code templates, and Github integration to

allow application development within the Android operating system (TechTarget Contributor, 2019).

The main mobile application will be developed using the android studio as the integrated development environment. The User interface and application will be developed using Flutter and dart as the programming language for development.

3.2.1.3. Flutter

Flutter is a free and open-source mobile user interface framework developed by Google in May 2017. In a nutshell, Flutter enables you to develop a native mobile app using only one codebase. This implies that one can design two separate apps using the same programming language and codebase, which can be used on iOS and Android (Thomas, 2019).

Flutter is written in the Dart programming language. Dart is an object-oriented programming language for creating the front end of mobile and web apps (Thomas, 2019).

3.2.1.4. Firebase

Firebase is a Backend-as-a-Service (BaaS) business that expanded into a next-generation app-development platform on the Google Cloud Platform. Through Firebase, it is not required to manage servers. You do not need to create APIs. Firebase is your server, API, and data store, all designed in a way that allows you to customize it to meet your specific requirements. Yes, you will need to leverage other parts of the Google Cloud for complex applications on occasion. Firebase cannot be all things to all people. However, it comes quite near (Esplin, 2016).

3.2.2. Hardware Requirements

For developing the mobile application, a desktop or laptop is required to develop the application with the following requirements (Intuz.com, 2021):

- Microsoft Windows 8/10/11 (64-bit).
- GNOME or KDE or Unity desktop on Ubuntu or Fedora or GNU/Linux
- 16 GB RAM recommended
- 8GB Disk space
- 1GB for Android SDK
- Flutter SDK
- 1280x800 screen resolution

3.3. Design

This section explains how the application was divided into sub-application and how the user interface was designed for each of the sub-apps and what color schemes and designs were used to design the User interface of the application.

3.3.1. Project Prototype

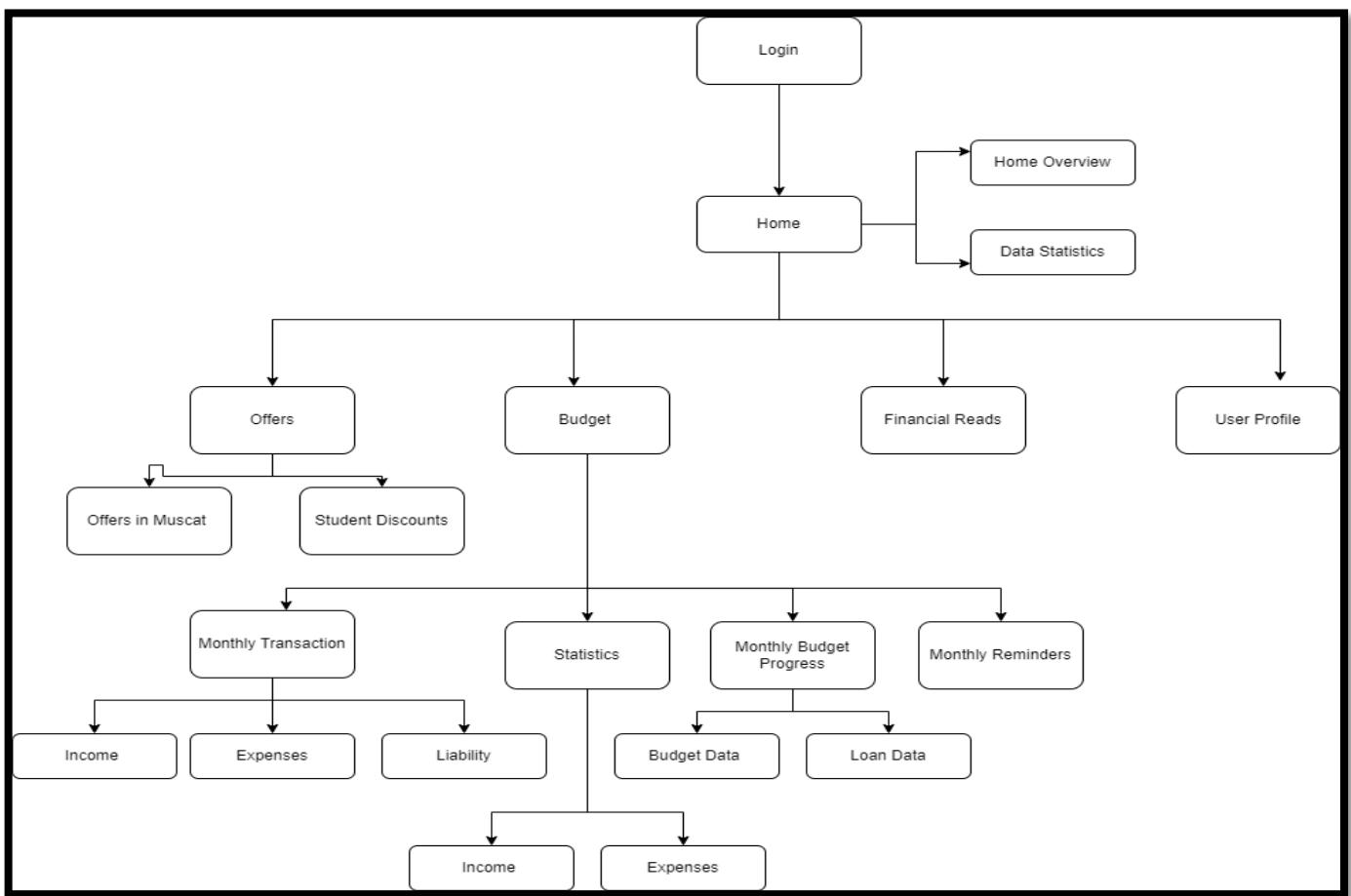


Figure 3.3 Prototype design flow chart (self,2022)

As illustrated in figure 3.3, the application is designed by splitting the application into sub-apps. Budget, Offers, Financial Reads, Profile, Home, and Login are all sub-apps of the financial management system. Each sub app has a similar but distinct User Interface and color scheme, making it easy for the user to distinguish between them. The login app will be created initially, as shown in figure3.3. Logging in and registering new users will both be done using the login app. The home app is the most significant app since it connects all of the other sub-applications. The budget app is one of the most prominent and has the most complicated user interface. The offers app will show users' offers, and student discounts, whilst the financial reads app will show all articles that would assist the user financially. The user may check their details and log out of the application from the profile page. In section 3.3.2, User Interface Design, further in-depth design aspects for the sub-app will be covered.

3.3.2. User Interface Design

User Interface (UI) Design is concerned with predicting what users may need to accomplish and ensuring that the interface has features that are simple to access, understand, and utilize in order to assist those activities. Interaction design, graphic design, and information architecture are all combined in UI (Usability.gov, 2014). The emphasis is on maximizing usability and delivering easy and enjoyable experiences (Malvik, 2020). The user interface is critical for meeting user expectations and ensuring that the site functions properly. Through contrasting graphics, clean design, and responsiveness, a well-executed user interface promotes successful interaction between the user and the program, app, or machine. When creating a user interface for your website, keep in mind the user's expectations in terms of accessibility, aesthetics, and simplicity of use. Because it anticipates the demands of the user and then meets those needs, an appropriate blend of engaging aesthetics and efficient responsiveness help keep the user engaged with the app (Indeed Editorial Team, 2021).

This section covers the actual user interface design of the mobile application, with a detailed explanation of each page.

As the application has many sub-apps, as mentioned and illustrated in section 3.3.1 and Figure3.3, the UI has been designed in such a way that each sub-app has a different color so that the user can clearly differentiate each app . Each color is selected in such a way that resonates with the sub apps functionality and also look good at the same time. The following are the detailed UI explanation of each color of the app:

- **The Home app:** The home app is a very integral part of the application as it integrates all the different sub apps. It is also the first page the user views after successfully logging in. Purple was chosen as the home page and the dominating color because of its sweetness, tranquillity, trust, confidence, or contemplation (Slava Vaniukov, 2020). It also evokes notions of wealth, with the application aiming to improve the users wealth through personal financial management. The colors grey and white represent wholesomeness, clarity, purity, and innocence, which instils trust and ease in consumers (Slava Vaniukov, 2020). The design aims to be simple and uncluttered for the user in order to eliminate misunderstanding and put the user at rest. This section also has the summary or overview of the whole application and it's meant to simplify and summarise the user's progress of the month as shown in figure 3.6 ,3.7 and 3.8.

The base of the application was kept white to simplify the application and not overcomplicate the colors for the user.

The Buttons in this section is grey with purple font for contrast. The buttons navigate between two different pages that is 'Overview' and 'Stats' .The heading is all in white with all the icons in black color which was done so for contrast again.

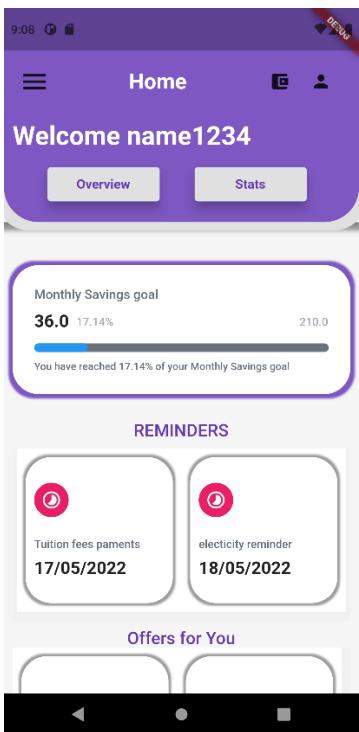


Figure 3.4 Home Overview (self, 2022)

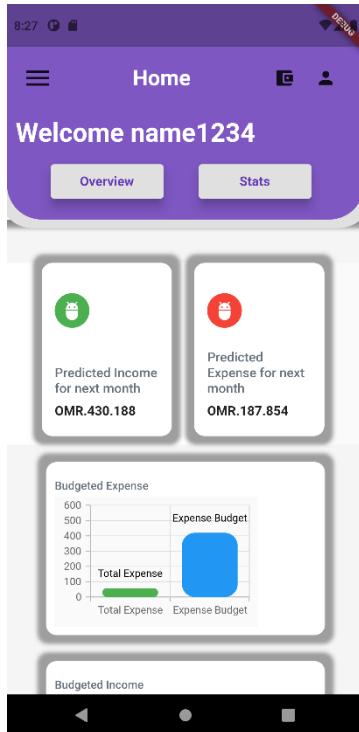


Figure 3.5 Home Stats (self, 2022)

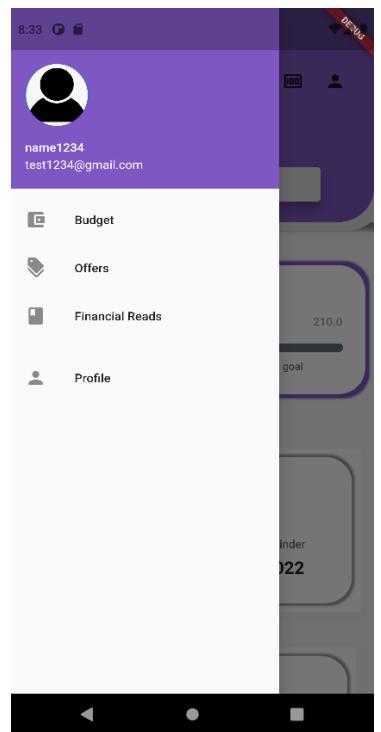


Figure 3.6 Home Menu (self, 2022)

- **The Budget App:** The Budget app helps the user manage their budget by viewing their transactions, progress, reminders and visualizing their data. The user can choose to add, edit and delete any necessary data. The Budget app has a deep pink color, which was chosen for its distinct and eye catchy look. The color pink is associated with calmness and keeps the viewer's happy (Thorpe, 2015). As this Budget app is the biggest part of the main application and the section where the user would spend most time, it seems to be obvious why the pink color was selected. The budget app has a footer with the icons of each of the section in the budget page and also a dominating icon of the home page which will direct the user to the home page. The footer is also majorly white in color with pink highlighting the color of the section the user is presently viewing or actively viewing and grey color of the icon being the inactive section.

The following are divided to explain the UI of each section. All sections have the same colors as mentioned in the budget app.

- **Transaction Page:** The transaction page showcases all the monthly transactions of the user. It showcases the income, expenses and liabilities of the month. The main background of the application is in white color while the texts and icons are in black color. The header connects the three sections income, expenses and liability which the user can navigate through the buttons present. The buttons are in pink color associating to the budget app while the fonts are in white for contrast .The header is same for all the three sections while the main body is almost similar as illustrated figure 3.7, 3.8 and 3.9. All three sections have the same calendar with the selected month being highlighted in pink while the other are in simple white and black. In the body there it can be noticed that money in income is shown in green, while it is red for expenses and liability. This done so because green is associated with money and moving

forward, while red is associated with debt and danger (Today's Eggspert, 2017). Thus as income being good for the user it's in green color and expenses and liability being bad and to control it's in red. There is edit and delete icons in each transaction in grey color so that the user can easily select and make any changes to the transactions.

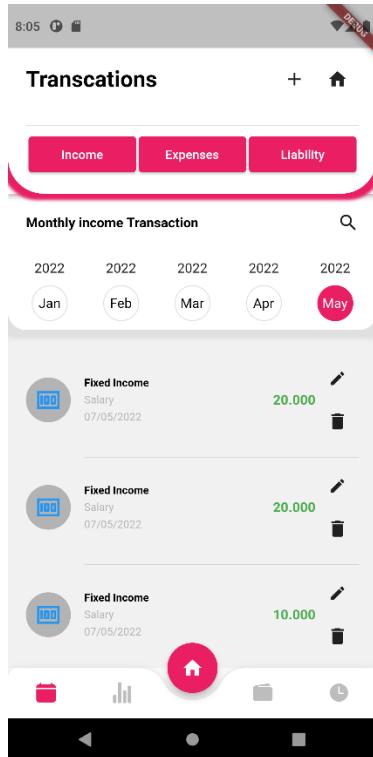


Figure 3.7 Income transaction (self, 2022)

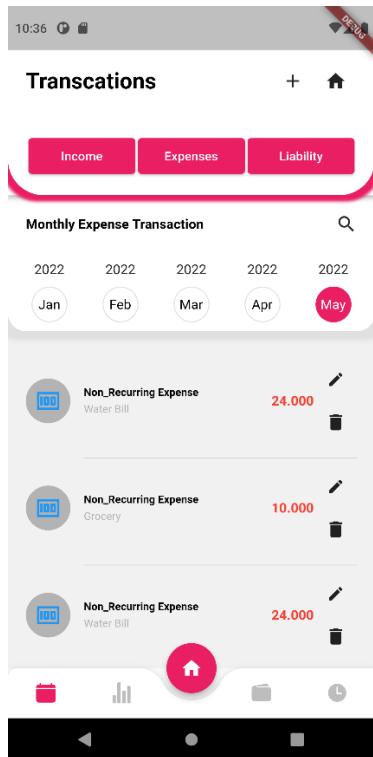


Figure 3.8 Expense transaction (self, 2022)

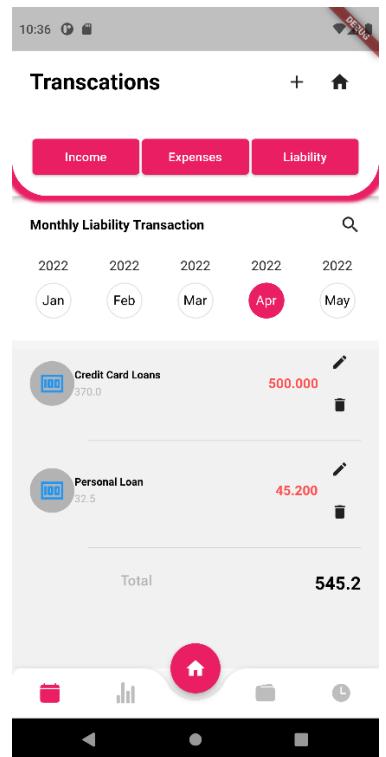


Figure 3.9 Liability transaction (self, 2022)

- Statistics Page: The statistics page which is again similar to the previous page is for visualizing and summarizing the income and expenses of the user in a glance for the present month. The heading is similar to the transaction page except there are only two buttons for the user to navigate between income and expenses. Here various types of charts are with different colors to beautify and at the same time declutter the section. The design is the same for both income and expenses.

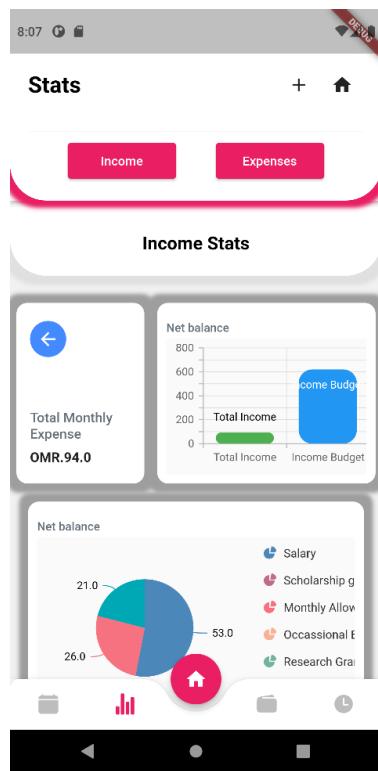


Figure 3.10 Income Statistics (self, 2022)

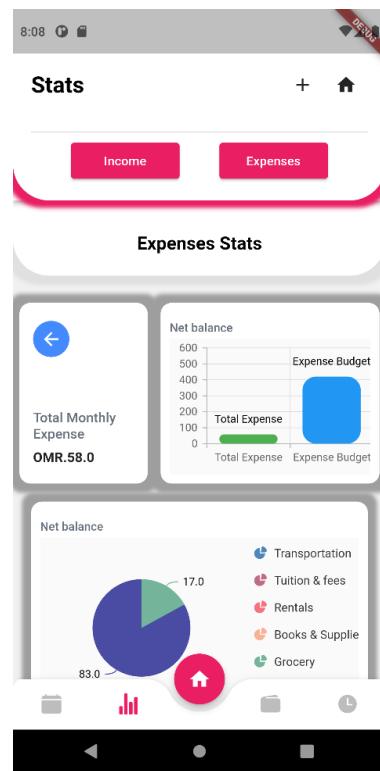


Figure 3.11 Expenses Statistics (self, 2022)

- o Monthly Budget Progress Page: Similar to the statistics page heading and footer the Monthly budget page aims to summarize and visualize the user budget progress, savings progress and loan progress. The Budget data page as illustrated in figure 3.12 is a very colorful and youthful page with Total spent summary associated with the home page color purple , green associated with income budget progress, red associated with expense budget progress and Budget page pink associated with the savings progress. This was done to classify each progress. Each progress section has a small summary about the user progress so that the user can easily understand and plan accordingly. There is an edit sign present in all progress section except the first one, so that the user can easily find and edit the budget or savings for that month. The loan data as illustrated in figure 3.13 is similar to budget data but has red as the dominating color as loans are debt and are considered bad. Thus, warning the user while also showing their loan progress. In each progress section the progress bar has green for loan paid and red for loan to pay . Like the Budget data there is a small summary in each progress section.

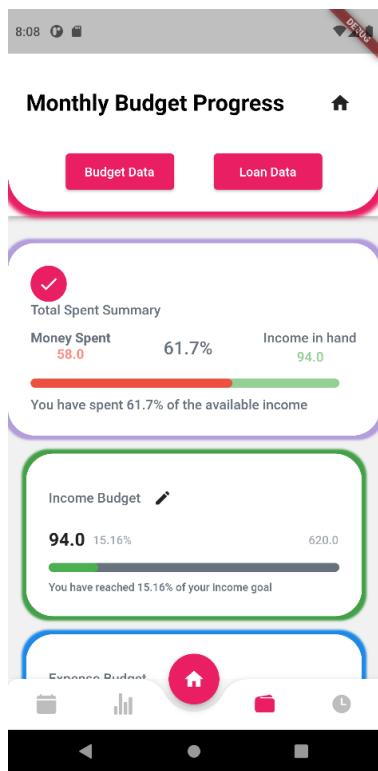


Figure 3.12 Budget Data (self, 2022)

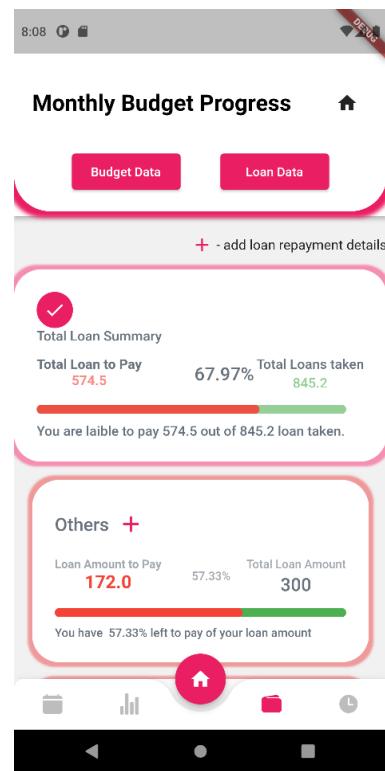


Figure 3.13 Loan Data (self, 2022)

- **Reminder Page:** The reminder page is where all of the user reminders are can be viewed ,edited And deleted. It is similar to the previous pages in the budget app as illustrated in figure.3.14.

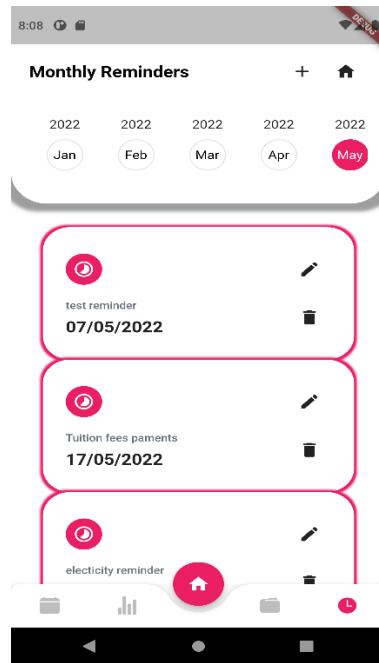


Figure 3.14 Reminder page (self, 2022)

- **Add new :** This section represents the UI for adding new information for income, expenses, liability and Reminder as illustrated below. The design and color are same in all sections and

the textboxes are designed with icons and textbox name so that the user can easily identify each textbox to add data in .

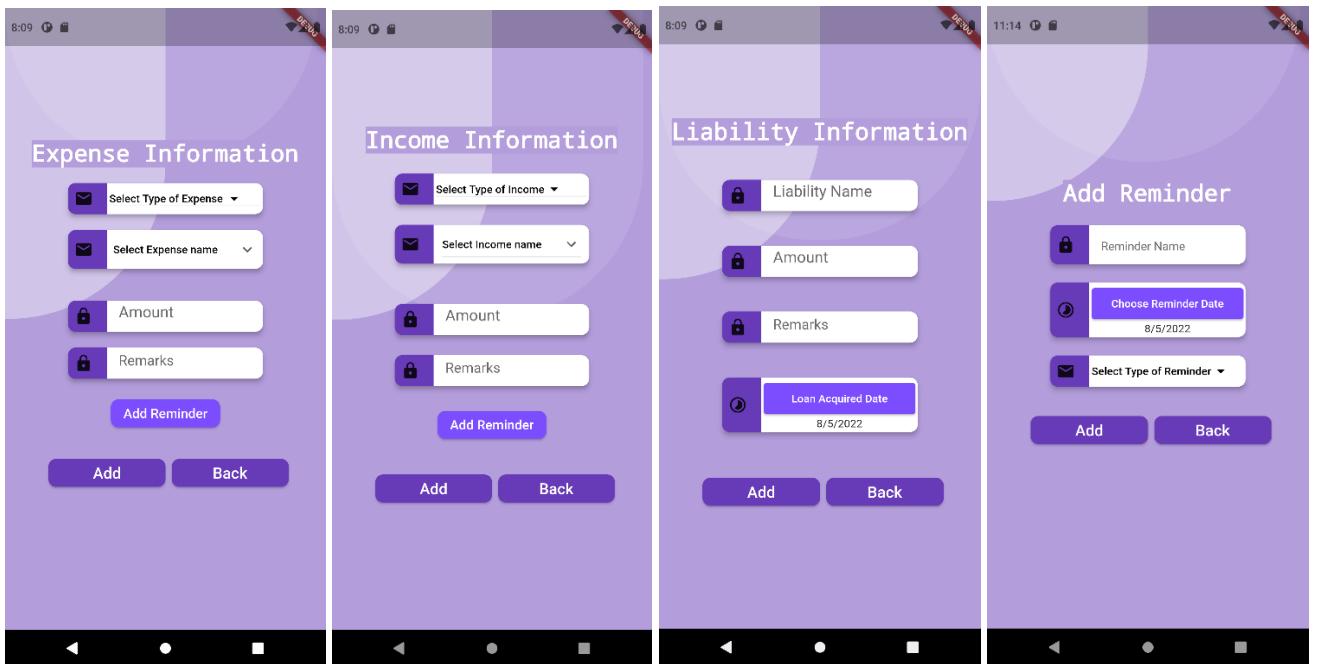


Figure 3.15 Add Expenses (self, 2022)

Figure 3.16 Add Income (self, 2022)

Figure 3.17 Add Liability (self, 2022)

Figure 3.18 Add Reminder (self, 2020)

- **Offers app:** The offers app which the user will be redirected from the home page nav bar is presented in green color . Green is usually associated with money and savings is considered good for the users money, thus this section of the application is green in color while the background is white for simplicity (Today's Eggspert, 2017). This section is also similar to the Budget app except that the header is in green color and the buttons to navigate between Offers and student discounts are grey with green fonts. The body is also same for both the sections with 2 offers and the student discounts having images and offer details in each row .

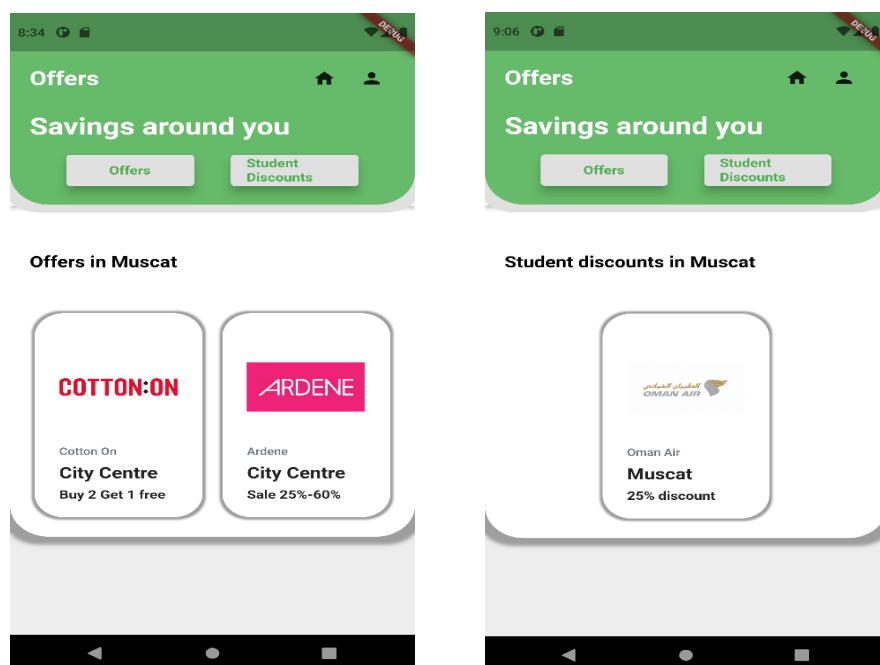


Figure 3.19 Offers (self, 2022)

Figure 3.20 Student discounts (self, 2022)

- **Financial Reads:** The financial reads app is presented in Blue color. This section aims at displaying selected financial articles for the user to read and learn to be more financially literate. The color blue is associated with knowledge, seriousness, power and integrity (Super Color Digital, 2020). Thus, this section is associated with seeking financial knowledge. In this section the user can view and scroll through the various article topics and select the learn more button option to be redirected to the article website.



Figure 3.21 Financial Reads (self, 2022)

- **Login section:** The login section has all the pages related to logging in, new user registration and admin login.

As the main color of the application is purple (same as home page) , thus this section major background color is also in purple.

- **Login:** This is the first page the user will view if the user is new or if the registered user has logged out of the application. As illustrated in figure 3.22. The user can login using their registered email address and password. If the given credentials are incorrect then an error message will be displayed in the top of the screen .The text fields were designed in a way so that the user can choose to read the information required to be given from the text field or through the icons present next to the text field. The aim of the design is to be simple yet trendy. If the user is new to the application, then the user can select the sign up option. The same design has been followed for Registration of new user. More details is in the new user registration section after the login.

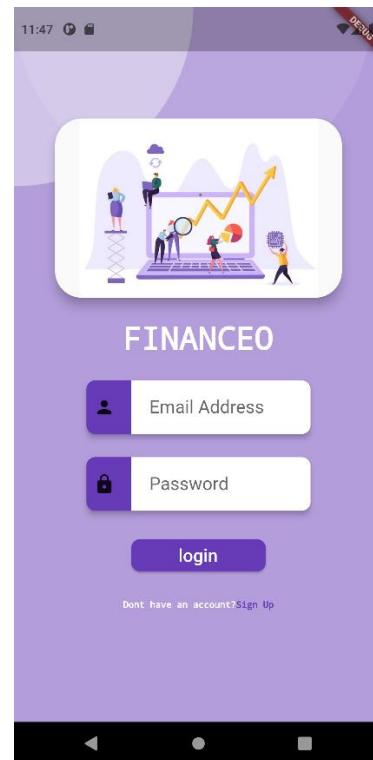


Figure 3.22 Login Page (self, 2022)

- **New User Registration:** As mentioned in the login section the design is very similar to the login page in terms of style and color. The user can create an account through the create account page as illustrated in Figure 3.23 . If the user is already registered then an error message will be shown in the top. If the password is not more than 6 characters and if the confirm password is not the same as password, then an error message in red will be shown in the password text field and confirm password text field. After the user successfully creates an account in the database then the user will be directed to the student information page as illustrated in Figure 3.24.

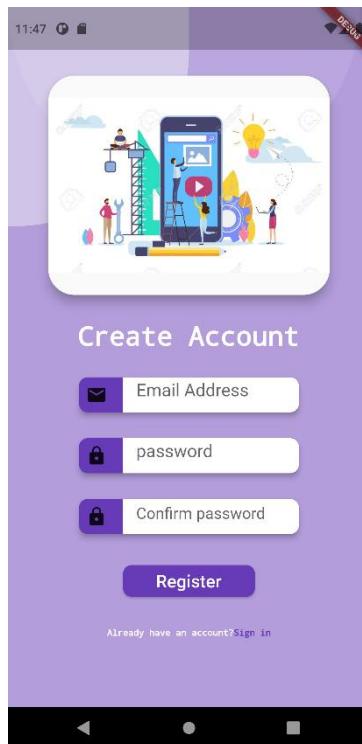


Figure 3.23 Create Account (self, 2022)



Figure 3.24 Student Information (self, 2022)

The user will then have to give their income, expenses and liabilities details to start using the application and the UI design is almost same for the three pages. The user can add necessary information for that month by clicking the add button and then click next after the necessary information is given. The first page is the Income page as illustrated in Figure. 3.25 , the next page is expenses page as illustrated in figure 3.26 and then the liability page is shown as illustrated in figure 3.27.

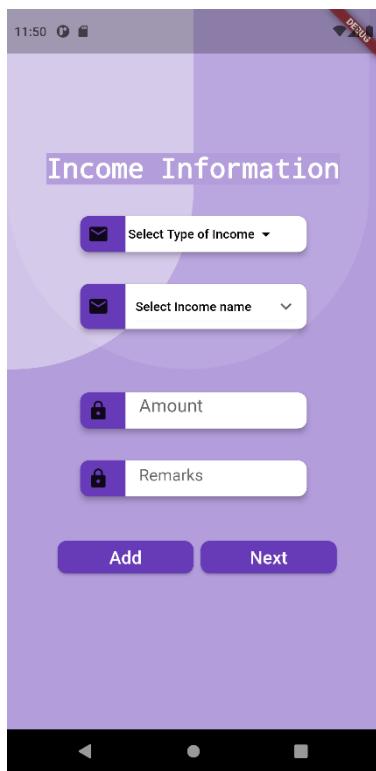


Figure 3.25 Income Page (self, 2022)

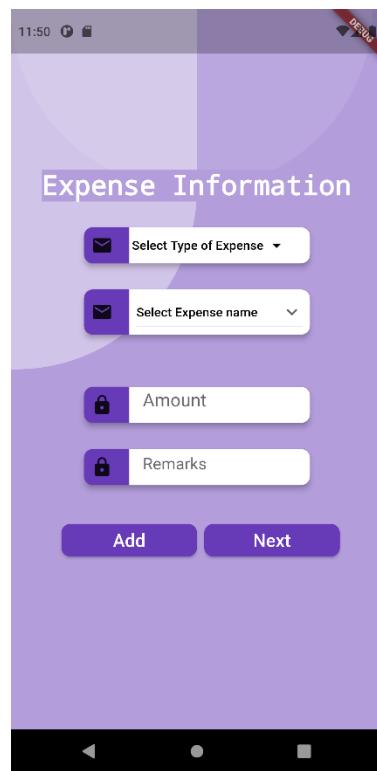


Figure 3.26 Expenses Page (self, 2022)

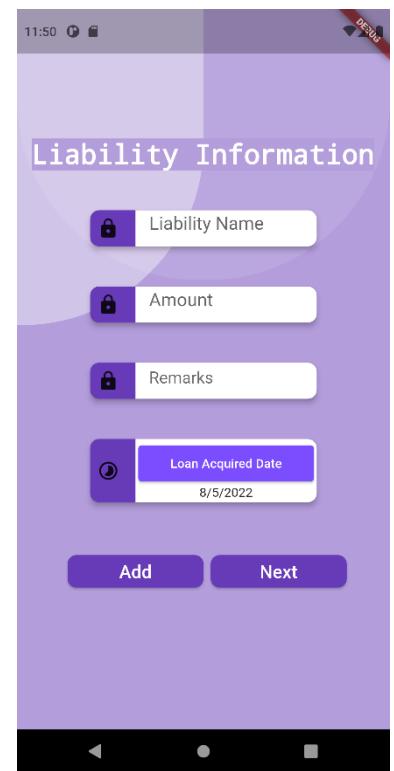


Figure 3.27 Liability page (self, 2022)

After giving necessary income, expenses and liability details, the user will then have to give the planned budget and savings details as illustrated in Figure 3.28 and Figure.3.29

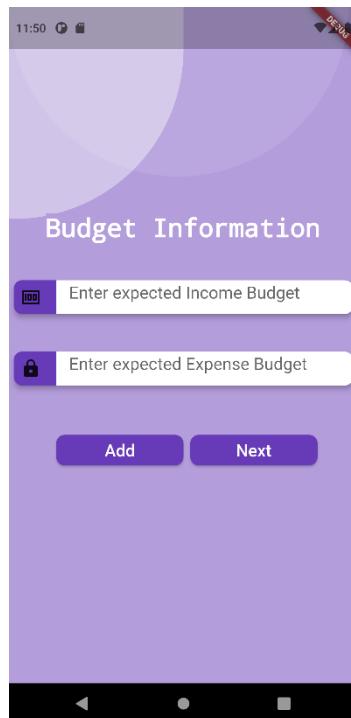


Figure 3.28 Budget Page (self, 2022)

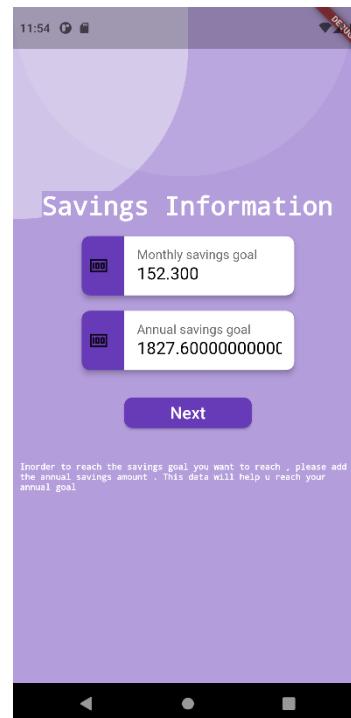


Figure 3.29 Savings page (self, 2022)

- **Admin Page:** The admin page is designed for the admin to add any new offers , student discount and financial reads for the apps registered user. The design is similar to the main apps home page as shown in figure 3.30. yet kept simple and compact as this is only for the admin and not the users

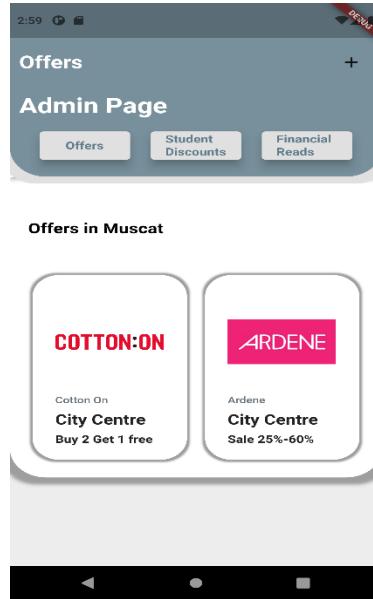


Figure 3.30 Admin Home page(self, 2022)

- **Profile Page :** Every registered user once logged in can access their profile page . This page was designed for the user to view their registered information and also sign out through the sign out button as shown in figure 3.31.which is located at the end so that the user can click it only if they want to . The page is kept simple for the user to understand.

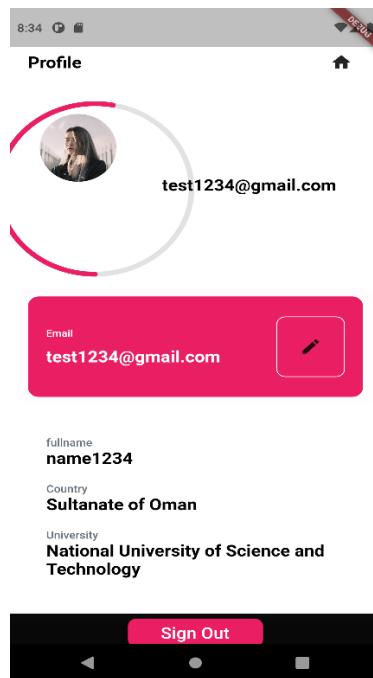


Figure 3.31 Profile Page (self, 2022)

3.4.1. Flow chart

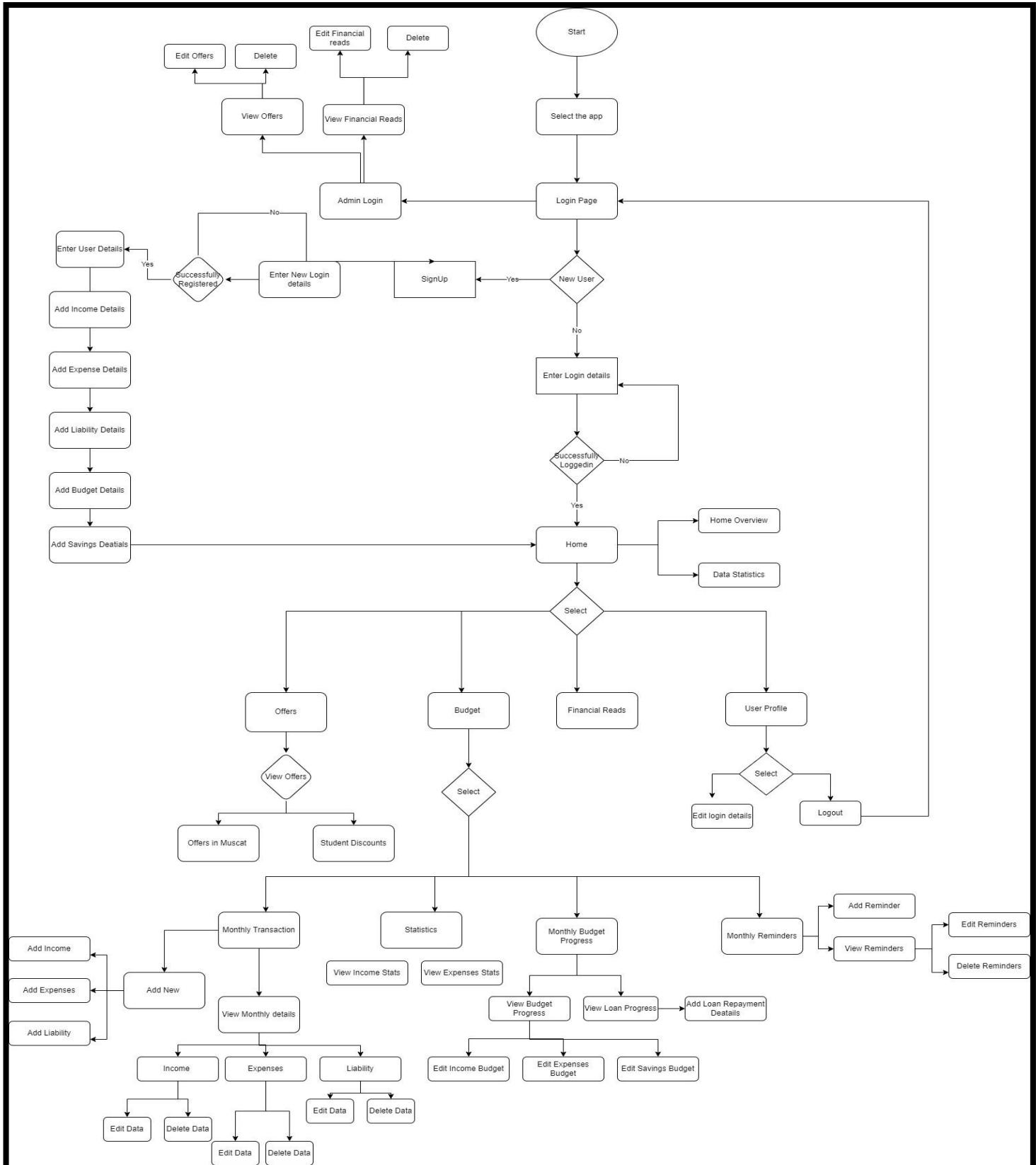


Figure 3.32 Flowchart application (self,2022)

A flow chart visual depiction of a process that depicts each step in sequential order. It is represented by a separate symbol and includes a brief explanation of the stage. The flow chart icons are connected by arrows that indicate the process flow direction. It is a frequent process analysis tool and one of the seven fundamental quality tools (Hebb, 2019). In figure 3.1, the flowchart of the proposed project is

presented. The flowchart is designed for a personal financial management system to manage the expenses and income of the user. When the user visits the app, the user will have to give his/her user details if registered in the database to log in. After successful login, the user will be directed to the application home page. Suppose the user is not registered in the database, which is a new user. In that case, the user will have to give the necessary details such as personal details, expense details, and income details to register. Once the user is successfully registered, the user will be directed to the application home page. Once the user is on the homepage, the user can view the financial overview, such as savings goal progress, reminders, and links to add or make changes to the expense and income data. The user can also visualize the income, expenses, and liability through the data option. The user will also be redirected to the offers page in the menu, displaying all available offers in Oman for the student to take advantage of. The user can add, edit, and delete the expense, income, savings goals, and reminders through other features, which are calculated monthly. The user can log out of the application just after viewing the overview or making any necessary changes to the data given by the user.

3.4.2. Database Design

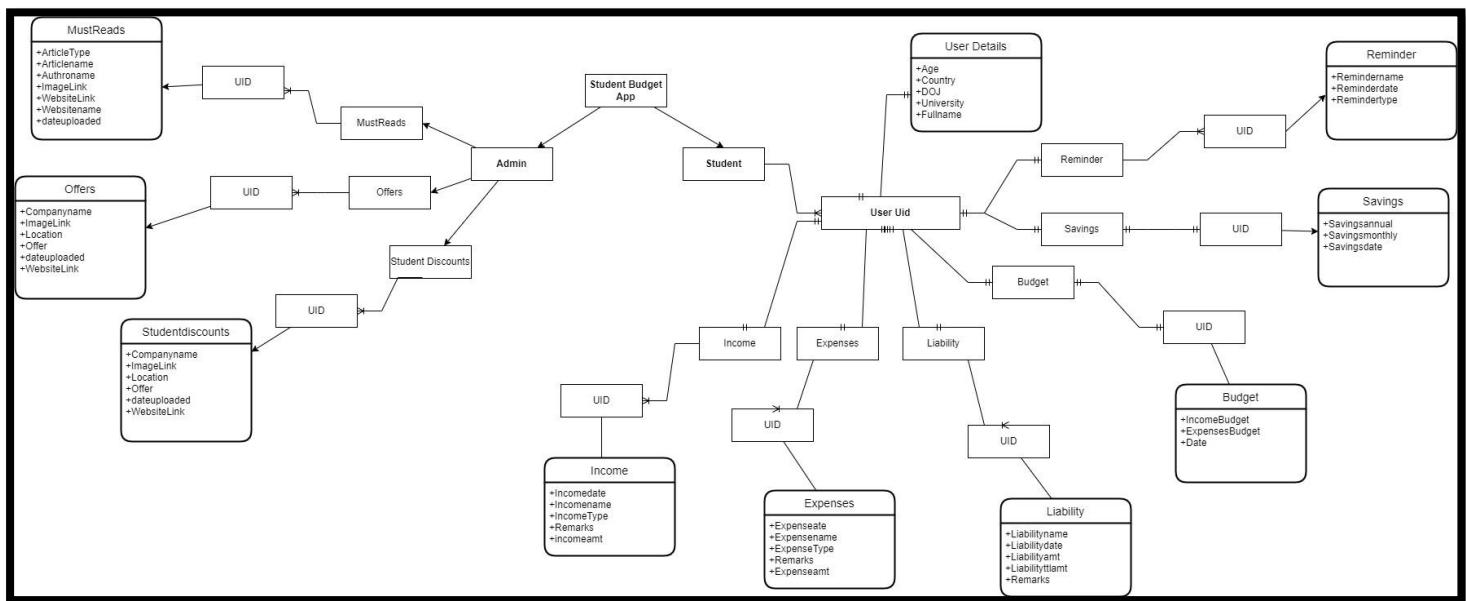


Figure 3.33 Firebase Database Design (self,2022)

Database design is a set of actions or procedures that help plan, develop, deploy, and maintain enterprise data management systems. Maintaining a database is reduced, which improves data consistency, and the cost-effective methods are highly impacted in terms of disk storage space. As a result, a great database design idea is required (javatpoint, n.d.). Figure 3.6. Illustrates the database design for firebase that has been used as the database for the application. The Google Firebase Real-time Database is a web-based database maintained by Google. Data is saved in JSON and synced in real time to all linked mobile and other clients. It allows developers to create complex collaborative apps with data stored locally for a responsive user experience (hackolade, n.d.). The most common type of NoSQL database is a JSON database. Traditional relational databases have trouble storing data outside of columns and rows; hence NoSQL database

management and JSON are distinct (Morris, 2021). Instead of rows, columns, or even tables, the database uses collections and documents. For example, there are two collections in the student finance management app: admin and student. Must-Reads, offer, and student discount papers are all found in the admin collection, and each admin collection only has one of these documents. Each document has numerous sub-documents, each with its unique identifier and all of the fields listed in Figure 3.6.

Similarly, each student or user in the student collection has a unique user id produced when the person registers. The user id contains information about the user and other records such as income and expenses. As shown in figure3.6, liability, budget, savings, and reminders each have their unique id and details.

3.5. Data Collection

During the development stage of the application, no data was collected. All of the information provided throughout the development stage is fictitious. Although no data is available during the development stage, once the prototype is deployed, the user data will provide all the information needed to display the user's progress, details, and statistics. Machine learning will be used to anticipate savings based on user data. Because no initial data was taken when the machine learning component of the project was developed, the user data will be used to remodel every time the user updates or adds new data, allowing the machine to forecast the user data with more accuracy as the user's data grows.

3.6. Summary

In conclusion, the chapter discusses the approach used to design the application and how it compares to other methodologies. The system requirements are examined and the software and hardware needs for development. The application's design is described in detail, including the prototype and user interface designs. Details of any data gathered are also explained.

CHAPTER 4 IMPLEMENTATION AND TESTING

This chapter walks you through creating a database using Firebase and designing and implementing a mobile app with flutter using the Android Studio IDE. It demonstrates how each layout component is used in the interfaces.

4.1. Database implementation

The first step to use the firebase services from the Google Cloud is to set up a project on the Firebase console. After creating the project and following the necessary steps mentioned by firebase, the user will be directed to firebase homepage. At this stage a connection between the firebase and the mobile application has been established.

For the project a new project named ‘Financial Management app’ was created as shown in figure 4.1. Two apps were created were the login app is a test app and the budget app is where the data is saved in.

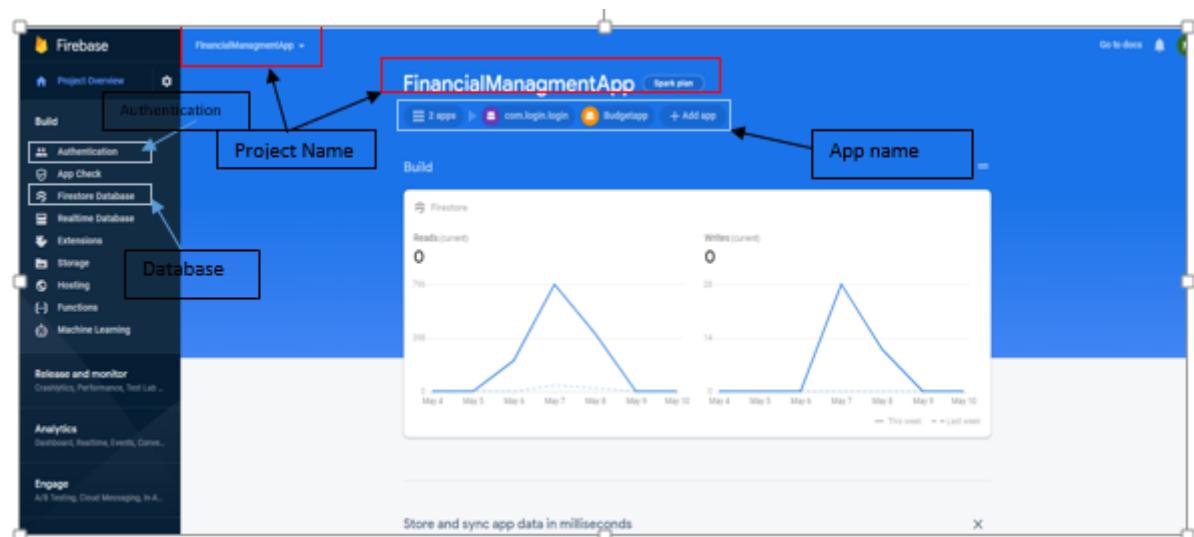


Figure 4.1 Firebase Console (self,2022)

The following libraries were utilized in flutter to access the required functions after connecting Firebase to the mobile application, as illustrated in figure 4.2.

```
dependencies:
  flutter:
    sdk: flutter
  firebase_auth: ^3.3.11
  firebase_core: ^1.13.1
  cloud_firestore: ^3.1.11
  firebase_database: ^9.0.11
```

Figure 4.2 Firebase libraries in flutter (self,2022)

Firebase has the ability to create new accounts and utilize firebase authentication to determine whether the user's email address and password are correct before allowing access to their home page. All the users account are stored in the Authentication section as shown in figure 4.3. Here the admin can also disable or delete the user account.

The screenshot shows the Firebase Authentication console for a project named "FinancialManagementApp". The left sidebar lists various services: Build, Authentication (selected), App Check, Firestore Database, Realtime Database, Extensions, Storage, Hosting, Functions, and Machine Learning. The main area is titled "Authentication" and shows a table of users. The table has columns for Identifier, Providers, Created, Signed in, and User UID. Two rows are visible:

Identifier	Providers	Created	Signed in	User UID
test1234@gmail.com	✉	2 Apr 2022	8 May 2022	o8m68RQanCVYjezOQWGQrNHR...
test@gmail.com	✉	19 Mar 2022	29 Mar 2022	VvvuoAYT44Zv7Ssm6Dw5R386tc2

At the bottom right of the table, there are buttons for "Rows per page" (set to 50), "1 - 2 of 2", and navigation arrows.

Figure 4.3 Authentication user database

A collection for the users' data as student and the admin has been created in the firestore database as shown in figure 4.4. The student collection stores all of the user's data, including income, expenses, liabilities, savings, budget, reminders, and student information. While the data for all offers, student discounts, and financial reads is maintained in the admin database. The admin data will be the same for all users, but the data for different students will be unique.

The screenshot shows the Cloud Firestore console for the same project "FinancialManagementApp". The left sidebar includes sections for Release and monitor, Analytics, and Engage, along with a Spark section. The main area is titled "Cloud Firestore" and shows a "Data" view. A "Student" collection is selected, showing a document with the ID "o8m68RQanCVYjezOQWGQrNHRcv2". The document contains the following fields:

- Budget
- Expenses
- Income
- Liability
- Age: 45
- Country: "Sultanate of Oman"
- DOJ: 7 January 2022 at 00:00:00 UTC+4
- University: "National University of Science and Technology"
- fullname: "name1234"

Figure 4.4 Firestore Database (self,2022)

The Firestore database can store the data given from the application and read, write, update, and remove data. To read, write, remove, or update data from Firebase, a path must be specified each time the database is used in flutter. An example is illustrated in figure 4.5.

```
Future <User>Data() async{
  List studentList=[];
  DocumentSnapshot snapshot;
  final CollectionReference collectionref=FirebaseFirestore.instance.collection("Student");
  try{
    await collectionref.where(
      FieldPath.documentId,
      isEqualTo: user.uid
    ).get().then((querySnapshot){
      for (var result in querySnapshot.docs){
        studentList.add(result.data());
      }
    });
  }

  return studentList;
}
catch(e){
  debugPrint('Error - $e');
  return null;
}
}
```

Figure 4.5 User data path (self,2022)

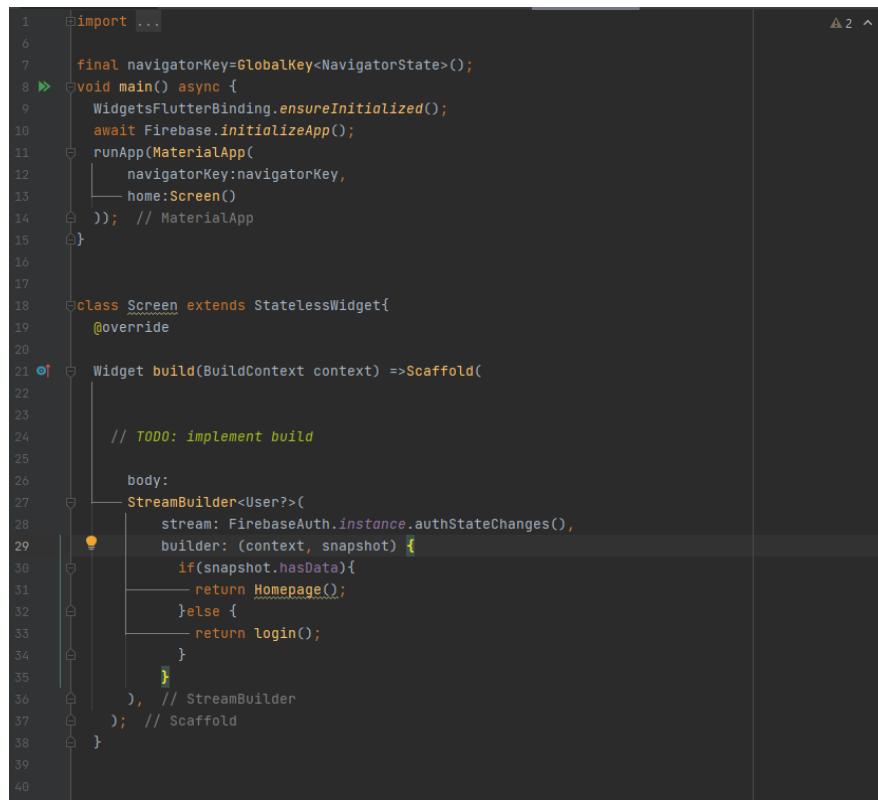
4.2. Application implementation

Flutter was used to create the app, with Dart as the programming language and Android Studio as the IDE. Various libraries were obtained throughout the implementation phase, as indicated in figure. 4.6, in order to improve the UI and various functionalities of the application.

```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.2
animated_bottom_navigation_bar: ^0.3.2
percent_indicator: ^4.0.0
flutter_staggered_grid_view: ^0.3.2
email_validator: ^2.0.1
flutertoast: ^8.0.8
firebase_auth: ^3.3.11
firebase_core: ^1.13.1
cloud_firestore: ^3.1.11
firebase_database: ^9.0.11
provider: ^6.0.2
intl: ^0.17.0
time_machine: ^0.9.16
excel: ^1.1.5
concentric_transition: ^1.0.1+1
lottie: ^0.7.0+1
syncfusion_flutter_charts: ^20.1.48
chart_sparkline: ^1.0.10
flutter_circular_chart: ^0.1.0
font_awesome_flutter: ^10.1.0
flutter_sparkline: ^0.1.0
charts_flutter: ^0.11.0
flutter_local_notifications: ^8.2.0
table_calendar: ^3.0.5
ml_algo: ^16.5.2
df: ^0.4.0-nullsafety.0
url_launcher: ^6.1.0
jiffy: ^4.1.0
```

Figure 4.6 Flutter Libraries (self,2022)

The main function is a very essential and crucial part for an application and the dart programming language. The main program execution starts from the main function and it can only be used once. All types of execution, including user-defined statements, functions, and libraries, are handled by the main() function (Elijah, n.d.). In the project the main function checks if the user is active or logged out from the application with the help of firebase. If the user did not log out of the application then the firebase library directs the registered user to the home page. If the user is new or has logged out of the application then the user will be redirected to the login page. This is illustrated in figure 4.7 as the code snippet for the main page.



```

1 import ...
2
3 final navigatorKey= GlobalKey<NavigatorState>();
4
5 void main() async {
6   WidgetsFlutterBinding.ensureInitialized();
7   await Firebase.initializeApp();
8   runApp(MaterialApp(
9     navigatorKey:navigatorKey,
10    home:Screen()
11  )); // MaterialApp
12 }
13
14
15
16
17 class Screen extends StatelessWidget{
18   @override
19
20   Widget build(BuildContext context) => Scaffold(
21
22
23
24   // TODO: implement build
25
26   body:
27   StreamBuilder<User?>(
28     stream: FirebaseAuth.instance.authStateChanges(),
29     builder: (context, snapshot) {
30       if(snapshot.hasData){
31         return Homepage();
32       }else {
33         return login();
34       }
35     },
36   ), // StreamBuilder
37 ); // Scaffold
38 }
39
40

```

Figure 4.7 Main page code (self,2022)

The following are the interface and implementation details of each page:

4.2.1. Login and registration

If a user has logged out or is new to the application, the first page that will appear is the Login page. Figure 4.8 shows the code for the login page. This page, in conjunction with the flutter authentication library, determines whether the user exists and whether the password provided is correct. The code authenticating the user is illustrated in figure 4.10. The error message as shown in fig 4.9.will be presented at the top of the page using flutter toast if the login information do not match the current user database. Other error messages as illustrated in figure 4.9., such as an incorrect email address or password, will be shown at the top of the page using flutter. Figure 4.11 shows the code for validating the data entered by the user.

```
1 import ...
8
9
10 class Login extends StatefulWidget {
11
12
13     @override
14     _loginState createState() => _loginState();
15 }
16
17 class _loginState extends State<Login> {
18
19     final TextEditingController emailController=TextEditingController();
20     final TextEditingController passwordController=TextEditingController();
21     bool _isemailValid = false;
22     bool _ispasswordValid = false;
23
24
25     @override
26     void dispose(){
27         emailController.dispose();
28         passwordController.dispose();
29
30         super.dispose();
31     }
32     Widget build(BuildContext context) {
33
34         return
35             Form(
36                 child: Container(
37                     width:MediaQuery.of(context).size.width,
38                     height:MediaQuery.of(context).size.height,
39                     color: Colors.deepPurple[200],
40                     child: Stack(
41                         children:<Widget>[
42                             Align(

```

Figure 4.8 Login page code (self,2022)

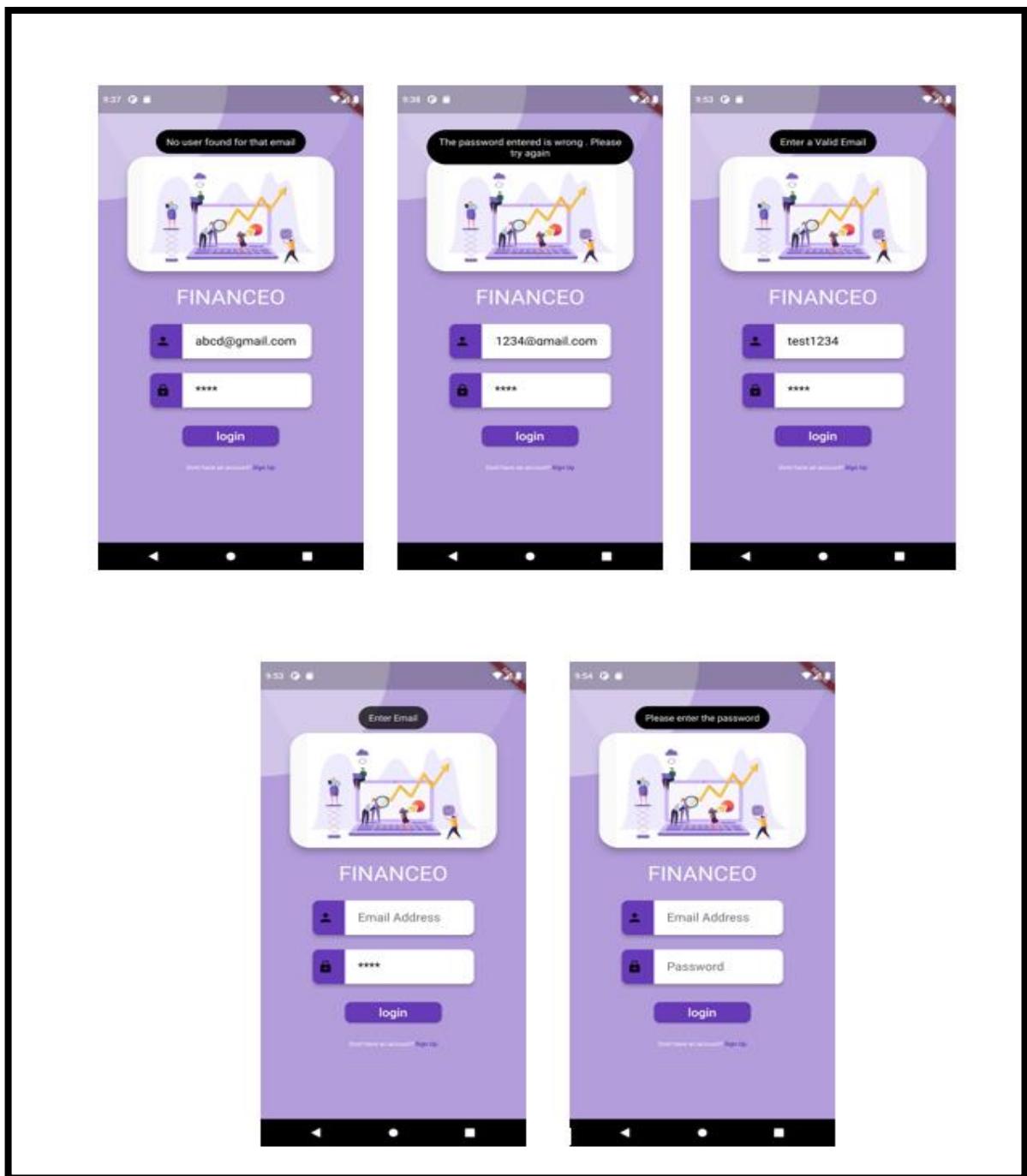


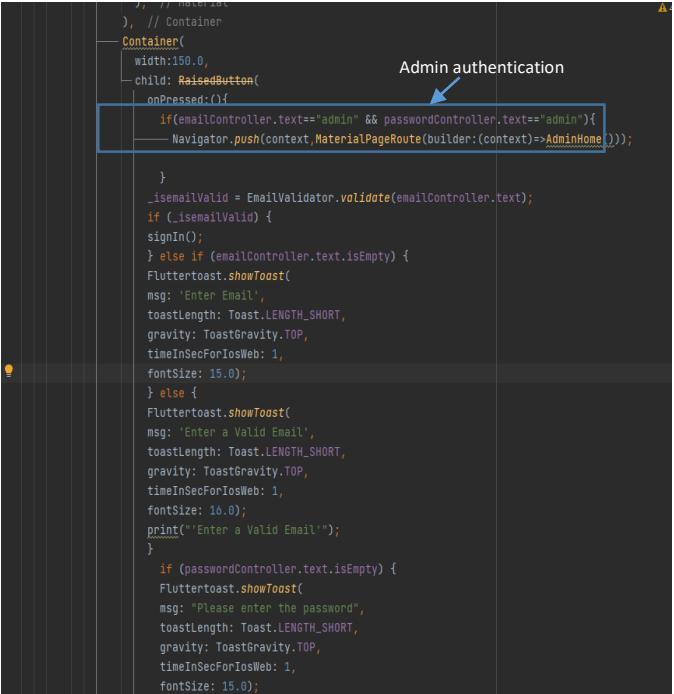
Figure 4.9 Error messages using flutter toast

```

Future signIn() async{
  try {
    await FirebaseAuth.instance.signInWithEmailAndPassword(
      email: emailController.text.trim(), password: passwordController.text.trim());
  } on FirebaseAuthException catch (e) {
    if (e.code == 'user-not-found') {
      Fluttertoast.showToast(
        msg: 'No user found for that email',
        toastLength: Toast.LENGTH_SHORT,
        gravity: ToastGravity.TOP,
        timeInSecForIosWeb: 1,
        fontSize: 16.0);
      print('No user found for that email.');
    } else if (e.code == 'wrong-password') {
      Fluttertoast.showToast(
        msg: 'The password entered is wrong . Please try again',
        toastLength: Toast.LENGTH_SHORT,
        gravity: ToastGravity.TOP,
        timeInSecForIosWeb: 1,
        fontSize: 16.0);
      print('Wrong password provided .');
    }
  }
  navigatorKey.currentState!.popUntil((route)=>route.isFirst);
}

```

Figure 4.10 User credentials validation (self,2022)



```

    }, // Container
    width:150.0,
    child: RaisedButton(
      onPressed:(){
        if(emailController.text=="admin" && passwordController.text=="admin"){
          Navigator.push(context, MaterialPageRoute(builder:(context)=>AdminHome()));
        }
      }
    );
  }
  _isemailValid = EmailValidator.validate(emailController.text);
  if (_isemailValid) {
    signIn();
  } else if (emailController.text.isEmpty) {
    Fluttertoast.showToast(
      msg: 'Enter Email',
      toastLength: Toast.LENGTH_SHORT,
      gravity: ToastGravity.TOP,
      timeInSecForIosWeb: 1,
      fontSize: 15.0);
  } else {
    Fluttertoast.showToast(
      msg: 'Enter a Valid Email',
      toastLength: Toast.LENGTH_SHORT,
      gravity: ToastGravity.TOP,
      timeInSecForIosWeb: 1,
      fontSize: 10.0);
    print("Enter a Valid Email");
  }
  if (passwordController.text.isEmpty) {
    Fluttertoast.showToast(
      msg: "Please enter the password",
      toastLength: Toast.LENGTH_SHORT,
      gravity: ToastGravity.TOP,
      timeInSecForIosWeb: 1,
      fontSize: 15.0);
  }
}

```

Figure 4.11 Validating User data (self,2022)

The user will be led to the other pages via the login page, as seen in figure 4.12. The user will be routed to the Home page if they are already registered and their credentials are correct. If the user is new and wishes to create an account, they can click sign up and be directed to the Create Account page. If the user is an admin, the admin will be directed to the admin page using the admin credentials. Figure 4.11 shows the admin authentication code.

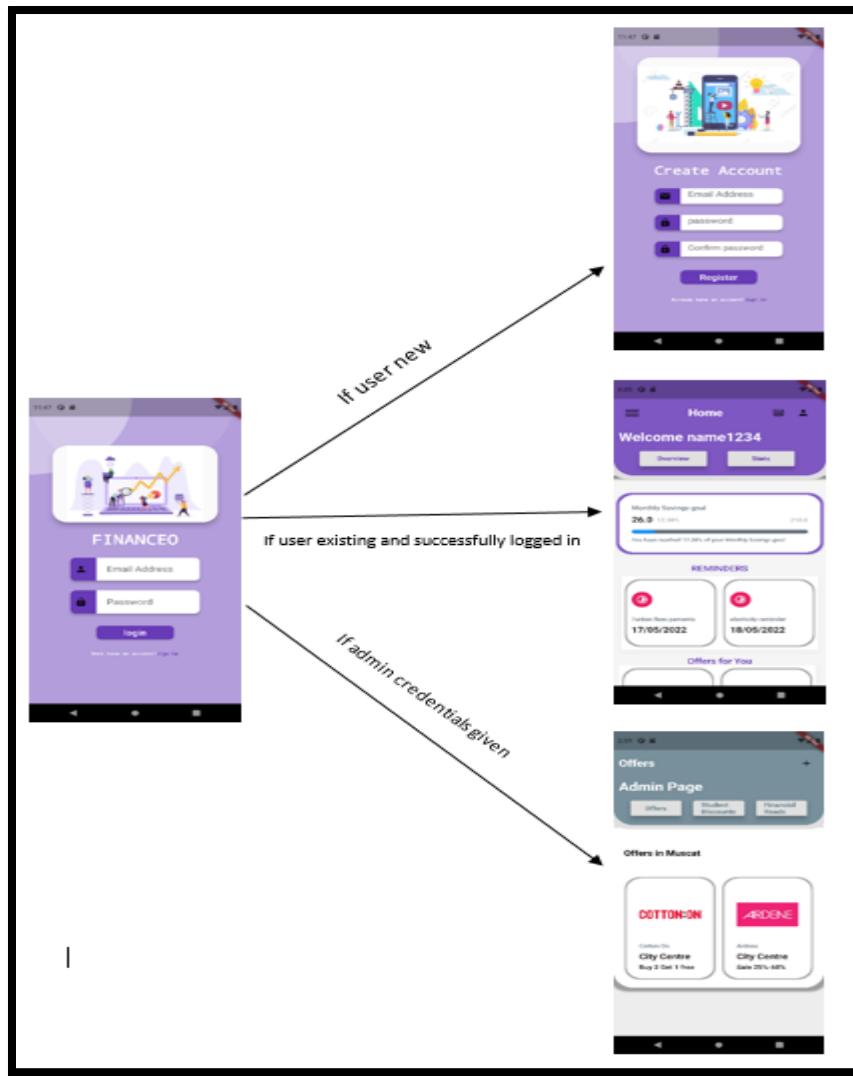


Figure 4.12 Pages redirected from login(self,2022)

The new user is led to the create account page, where they must provide a unique user email address and password in order to register using the flutter library. If the user email address exists, an error message will be displayed using flutter toast, and if the password and confirm password are not the same, an error message will be displayed in the text field, as illustrated in figure 4.13. Figure 4.14 shows the user login code, whereas Figure 4.15 shows the text field validation code. Only an Astrid sign, as illustrated in figure 4.15, may reveal the password and confirm password.

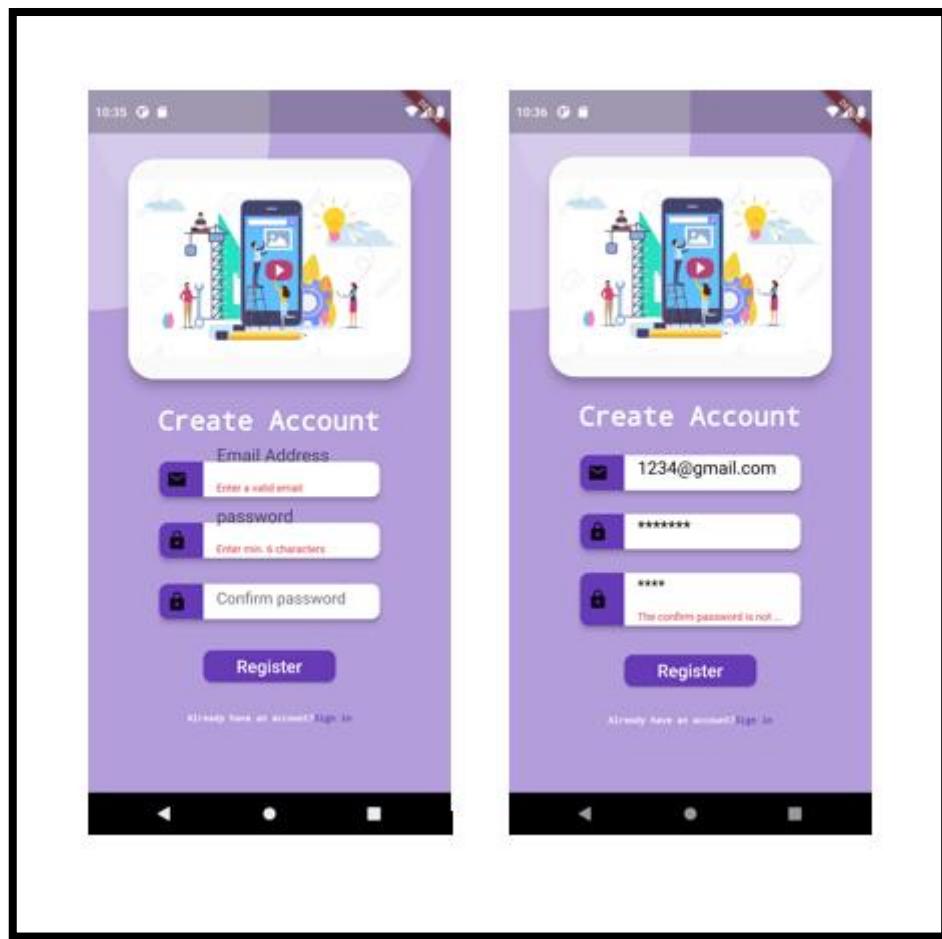


Figure 4.13 Text field error in create account (self,2022)

```

Future signUp() async{
  final isValid=formKey.currentState!.validate();
  if(!isValid) return;
  try {
    await FirebaseAuth.instance.createUserWithEmailAndPassword(
      email: emailController.text.trim(), password: passwordController.text.trim());
  } on FirebaseAuthException catch (e) {
    print(e);
    Fluttertoast.showToast(
      msg: 'The Useremail and password already exists .Please try logging in or give another user email',
      toastLength: Toast.LENGTH_SHORT,
      gravity: ToastGravity.TOP,
      timeInSecForIosWeb: 1,
      fontSize: 12.0);
    print('The Useremail and password already exists .Please try logging in or give another user email');
  }
}

```

Figure 4.14 user Authentication code

```

    child: TextFormField(
      autovalidateMode: AutovalidateMode.onUserInteraction,
      validator:(value)=>
      value!= null && value!=passwordController.text
        ? "The confirm password is not the same as the password entered"
        :null,
      obscureText: true,
      obscuringCharacter: "*",
    ),
    validation_confirm password

    child: TextFormField(
      controller: passwordController,
      autovalidateMode: AutovalidateMode.onUserInteraction,
      validator:(value)=>
      value!= null && value.length<6
        ? "Enter min. 6 characters"
        :null,
      obscureText: true,
      obscuringCharacter: "*",
    ),
    Validation password

    child: TextFormField(
      controller: emailController,
      autovalidateMode: AutovalidateMode.onUserInteraction,
      validator:(email)=>
      email!= null && !EmailValidator.validate(email)
        ? "Enter a valid email"
        :null,
    ),
    validation_email address
  
```

Figure 4.15 Validation code

As illustrated in Figure 4.16, all login and registration codes are kept in a single directory called login. After successfully registering, the user must enter their personal information, followed by their income, expenses, liabilities, budget, and savings information in the order illustrated in figure. 4.17. All the design and code after create account are similar. In the data add page, all of the data is uploaded to the database using the flutter library, as seen in figure 4.18. After successfully confirming the savings data, the user will be taken to the Home page.

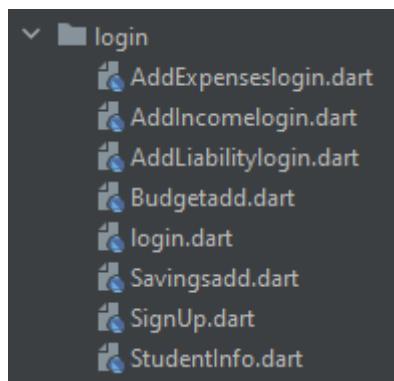


Figure 4.16 Login and registration files (self,2022)

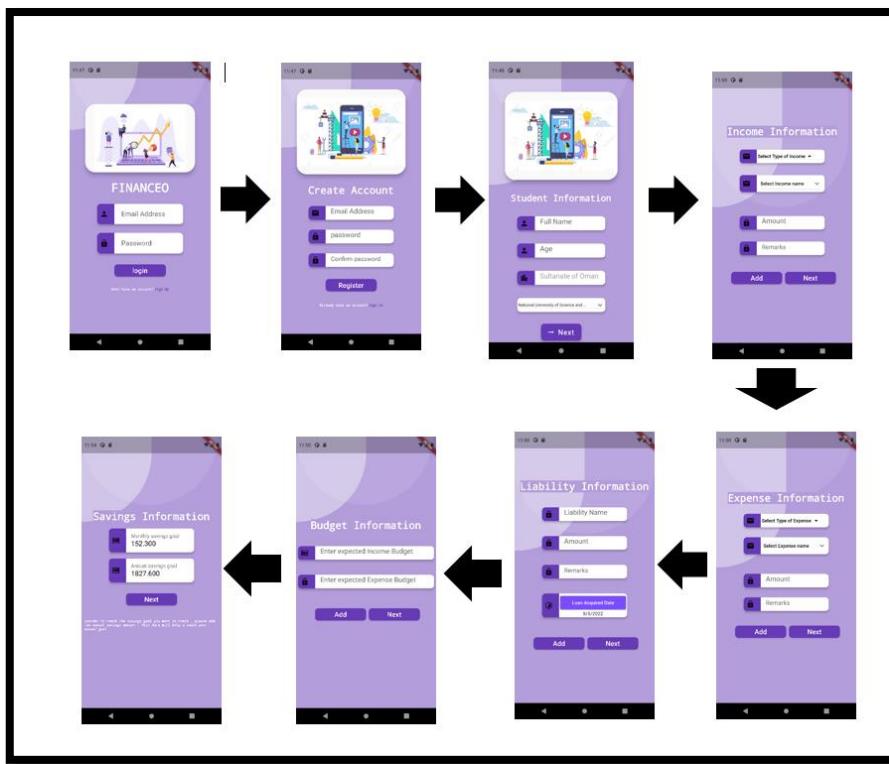


Figure 4.17 User Registration Procedure

```

1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:firebase_auth/firebase_auth.dart';
3 import 'package:fluttertoast/fluttertoast.dart';
4 import 'package:intl/intl.dart';
5
6 CollectionReference users = FirebaseFirestore.instance.collection('Student');
7 DateFormat dateFormat = DateFormat("yyyy-MM-dd HH:mm:ss");
8 String update = dateFormat.format(DateTime.now());
9 final user = FirebaseAuth.instance.currentUser!;
10 var now = DateTime.now();
11
12 Future<void> addIncome( String incometype, String incomename, double incomeamt, String Remarks, DateTime incomedeate) async {
13   // Call the user's CollectionReference to add a new user
14   if(incometype=="Fixed Income"){
15     }
16   return await users.doc(user.uid).collection("Income").add({
17     'Incometype': incometype,
18     "Incomename": incomename,
19     'incomeamt': incomeamt,
20     'Remarks': Remarks,
21     "Incomedeate":incomedeate
22   })
23   .then((value) { Fluttertoast.showToast(
24     msg: 'Income added successfully',
25     toastLength: Toast.LENGTH_SHORT,
26     gravity: ToastGravity.TOP,
27     timeInSecForIosWeb: 1,
28     fontSize: 16.0);
29 })
30   .catchError((error) { Fluttertoast.showToast(
31     msg: 'Failed to add details: $error.',
32     toastLength: Toast.LENGTH_SHORT,
33     gravity: ToastGravity.TOP,
34     timeInSecForIosWeb: 1,
35   );
36 })
37 }
```

Figure 4.18 Adding data to the flutter database (self, 2022)

4.2.2. Home page

The Home page is an important aspect of the application since it ties everything together. Each app/page has its own color scheme for easier classification and identification, as described in the User Interface design. Purple is the primary color on the home page. The user will first see the homepage with overview data as the home page body after successfully registering or signing in. As shown in figure. 4.19, the app bar is the sole design on the Home page, and it is set up such that the home page body changes to overview or stats depending on which button the user selects. Figure 4.20 shows the code for the home page application.

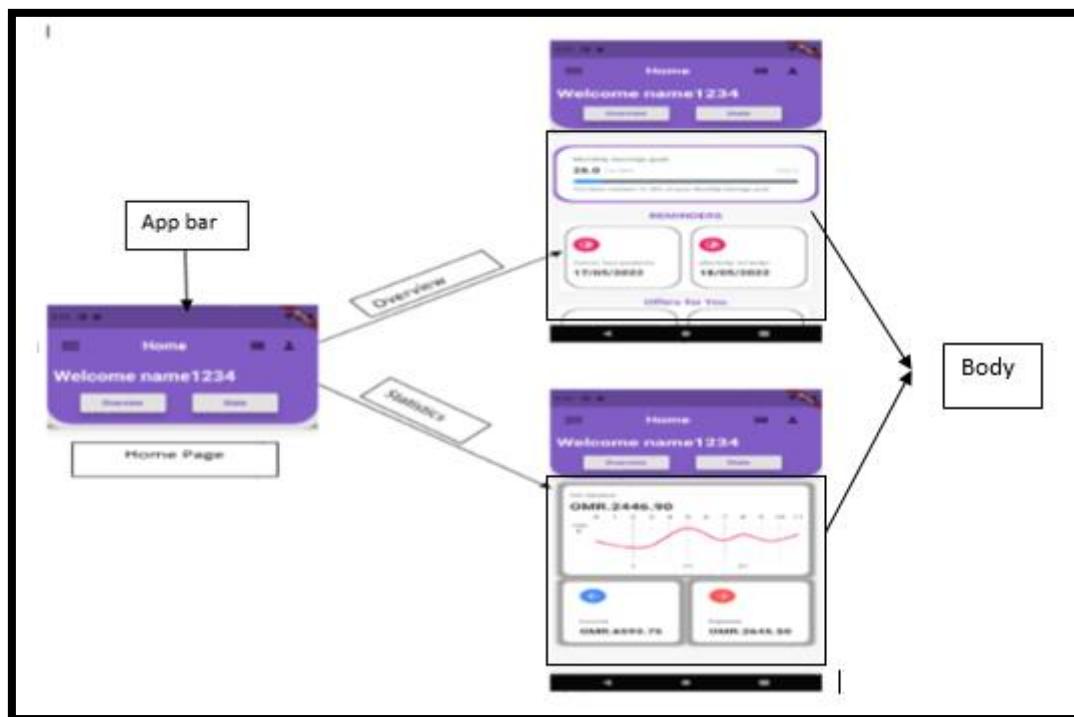


Figure 4.19 Home page body (self,2022)

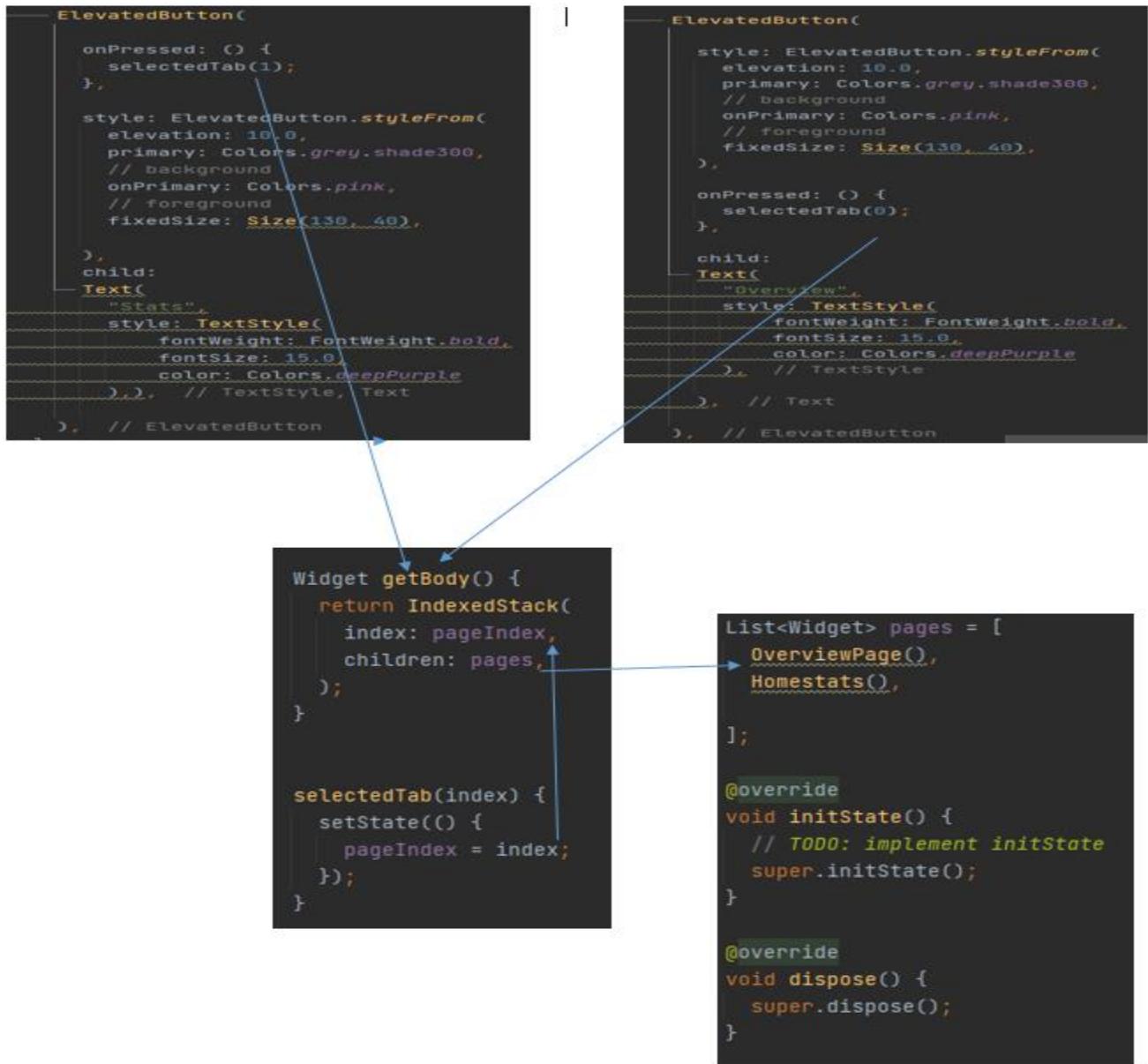


Figure 4.20 Home Page Body change code (self,2022)

The app bar, as seen in figure.4.21, contains the majority of the functionality of the home page. The user will be directed to the Budget app, user profile page, navigation bar, and the body of the home page will be altered to Overview or Stats based on button clicked . As the code illustrates in Figure 4.22. The welcome message with the user name is displayed on the home page by calling the user data from another file while specifying the data path that connects to and accesses the data from the firebase console.

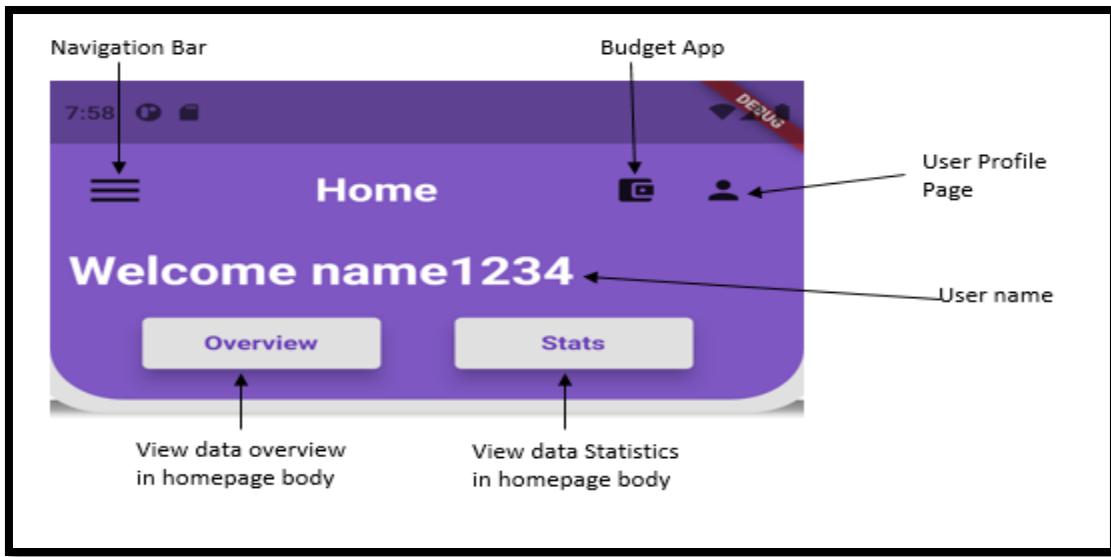


Figure 4.21 Home page design(self,2022)

```

Widget build(BuildContext context) {
  return FutureBuilder<UserData>(
    future: FireStoreDataBase().UserData(),
    builder: (context, snapshot) {
      if (snapshot.hasError) {
        return const Text(
          "Something went wrong",
        );
      }
      if (snapshot.connectionState == ConnectionState.done) {
        data = [];
        data = snapshot.data as List;
        var index = data.length;
      }
      return Scaffold(
        body: getBody(),
        key: _scaffoldKey,
      );
    },
  );
}

Future <UserData>() async{
  List studentList=[];
  DocumentSnapshot snapshot;
  final CollectionReference collectionRef=FirebaseFirestore.instance.collection("Student");
  try{
    await collectionRef.where(
      FieldPath.documentId,
      isEqualTo: user.uid
    ).get().then((querySnapshot){
      For (var result in querySnapshot.docs){
        studentList.add(result.data());
      }
    });
  };
  return studentList;
}
catch(e){
  debugPrint('Error - $e');
  return null;
}
}

Future <IncomeData>() async{
  try{
    await collectionRef.doc(user.uid).collection("Income").get().then((querySnapshot){
      For (var result in querySnapshot.docs){
        studentList.add(result.data());
        reminderdata.add({'id':result.id,result.id:result.data()});
      }
    });
  };
  return studentList;
}
catch(e){
  debugPrint('Error - $e');
  return null;
}
}

```

Figure 4.22 User Data Access(self,2022)

Due to space and design constraints, not all app functions are visible in the Home page app bar. As seen in Figure.4.23, the navigation bar comes into play since the user may simply navigate to the budget app, Offers, Financial Reads, and profile page. Figure 4.24 shows the code that is included in the home page app bar.

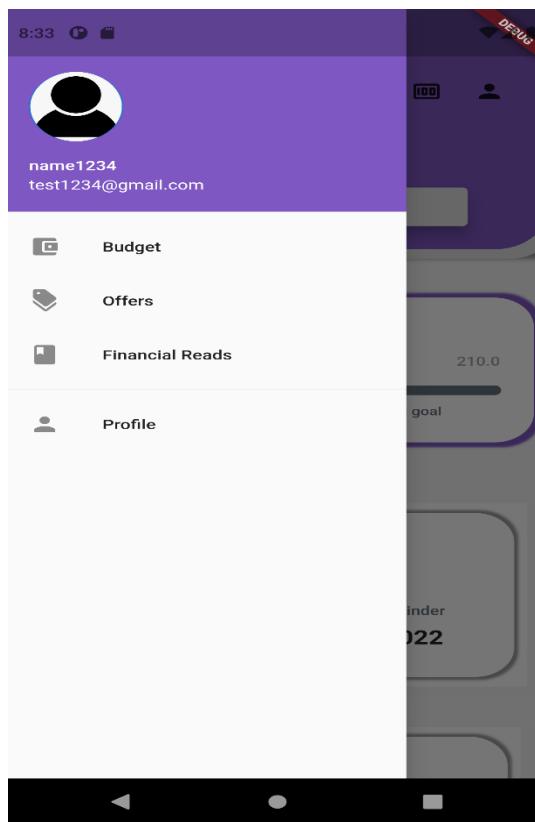


Figure 4.23 Navigation bar(self,2022)

```

    drawer: Drawer(
      child: ListView(
        padding: EdgeInsets.zero,
        children: [
          UserAccountsDrawerHeader(
            accountName: Text(data[0]["fullname"]),
            accountEmail: Text(user.email.toString()),
            currentAccountPicture: CircleAvatar(
              child: ClipOval(
                child: Image.network(
                  "https://www.kindpng.com/picc/m/173-1731325_person-icon-png-transparent-png.png",
                  fit: BoxFit.cover,
                  width: 70,
                  height: 90,
                ),
              ),
            ),
            decoration: BoxDecoration(
              color: Colors.deepPurple.shade400,
            ),
          ),
          ListTile(
            leading: Icon(Icons.account_balance_wallet),
            title: Text('Budget'),
            onTap: () => Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => RootApp(),
              ),
            ),
          ),
          ListTile(
            leading: Icon(Icons.discount),
            title: Text('Offers'),
            onTap: () => Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) =>
                  Studentsavingspage(),
              ),
            ),
          ),
          ListTile(
            leading: Icon(Icons.book),
            title: Text('Financial Reads'),
            onTap: () => Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) =>
                  MustReads(),
              ),
            ),
          ),
        ],
      ),
    ),
  ),
)

```

Figure 4.24 Navigation Bar code snippet(self,2022)

As previously stated, the user views the Home page first, with the Data Overview serving as the body of the page. The data overview is structured in such a way that each element in the body of the overview links to a different page in the financial management application. The user may see their savings progress, which with

a 'double click' will take to the budget page on the budget app. The user may also check their most recent and upcoming reminders, which will take them to the budget app's reminder page, where they can see all of their current, prior, and upcoming reminders. The user may also look at nearby Offers and be directed to the Offers page. From the overview page, the user may also enter new income or expenses. Instead of reworking the entire code and increasing the memory, the code was constructed in such a manner that just the page element is called. Figure 4.24 illustrates this.

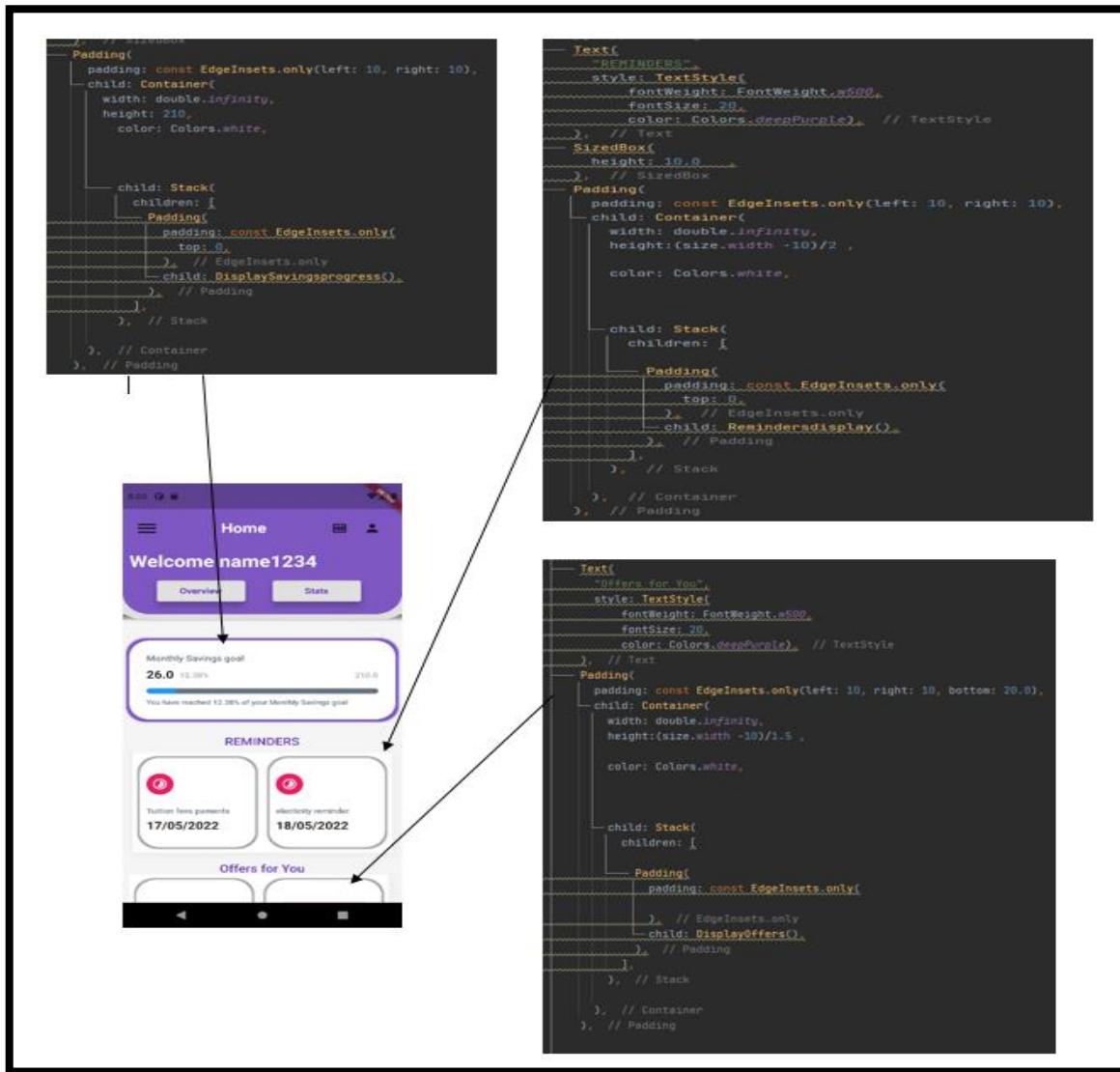


Figure 4.25 Overview elements code snippet (self,2022)

4.1.3. Budget app

The budget app is the biggest app with the most functionalities in the financial management application. The color palette for the budget app, like the Home page, was decided to be pink. The budget app, like the home page, is structured in such a manner that the footer brings the user to the many features of the app as well as direct the user to the home page. Other features as illustrated in figure 4.26 of the budget app include monthly transaction data, a monthly statistics page, a budget page, and a reminders page. The footer on all pages of the budget app serves as the navigation bar. The code explaining how the user are directed to other pages through the navigation bar are illustrated in figure 4.27. The body and app bar will alter according on what the user selects from the footer navigation bar. The home icon will exit the budget app and direct the user to the home page of the financial management application.

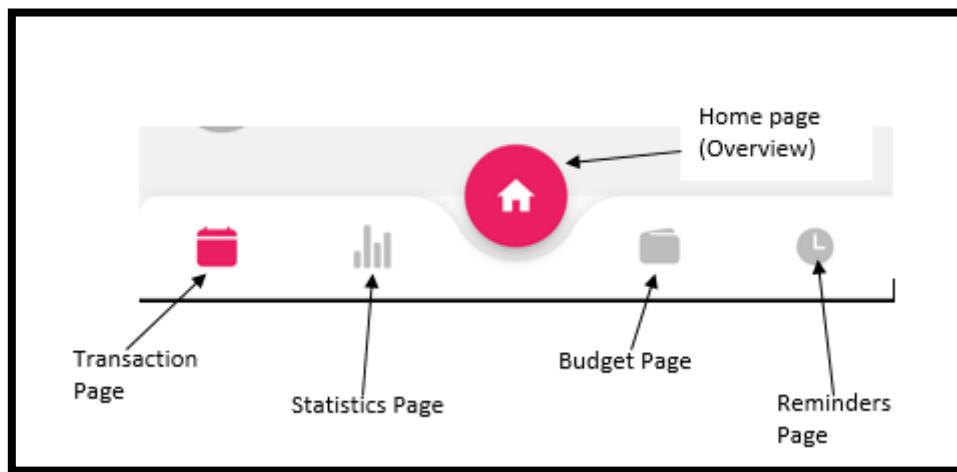


Figure 4.26 budget app bar (self,2022)

```

Widget build(BuildContext context) {
  return Scaffold(
    body: getBody(),
    bottomNavigationBar: getFooter(),
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => const Homepage()),
        );
      },
      child: Icon(Icons.home.size: 25),
      backgroundColor: Colors.pink
    ),
    floatingActionButtonLocation:
    FloatingActionButtonLocation.centerDocked,
  ); // Scaffold
}

Widget getBody(){
  return IndexedStack(
    index:pageIndex,
    children: pages,
  );
}

Widget getFooter(){
  List<IconData> iconItems = [
    Ionicons.calendar,
    Ionicons.stats_chart,
    Ionicons.wallet,
    Ionicons.time,
  ];
  return AnimatedBottomNavigationBar(
    icons: iconItems,
    activeColor: Colors.pink,
    splashColor:Colors.white ,
    inactiveColor: Colors.black26,
    activeIndex: pageIndex,
    gapLocation: GapLocation.center,
    notchSmoothness: NotchSmoothness.softEdge,
    leftCornerRadius: 10,
    iconSize: 25,
    onTap: (index){
      selectedTab(index);
    });
}

selectedTab(index) {
  setState(() {
    pageIndex = index;
  });
}

```

```

List<Widget> pages = [
  DetailsPage(),
  StatsPage(),
  BudgetPage(),
  ReminderPage()
];

```

The figure shows three code snippets from a budget app. The top snippet is the main scaffold structure, which includes a floating action button (FAB) with a home icon and a pink background, and a bottom navigation bar. The middle snippet defines the body of the scaffold as an indexed stack of pages, and the footer as an animated bottom navigation bar with four tabs. The bottom snippet is a list of four pages: DetailsPage, StatsPage, BudgetPage, and ReminderPage.

Figure 4.27 Budget app code(self,2022)

More details of the various functionalities offered by the budget app such as the transaction page , statistics page , budget page and reminders page are explained in this section.

4.1.3.1. Transaction page:

The transaction page displays the user monthly transaction income, expenses and liabilities and their total. The transaction page with the income transactions as the body is the first page the user will view once the user clicks the budget option in the home page. Based on the button the user picks as illustrated in figure 4.28., the transaction page will display monthly income, spending, and liabilities transactions, where the user can view, edit or delete the data. The app bar, like the main page, is the same, but the body varies depending on which button the user clicks. The income, expense, and liability bodies have the same design and code. The user may examine their prior monthly transaction by selecting the month from the user body.

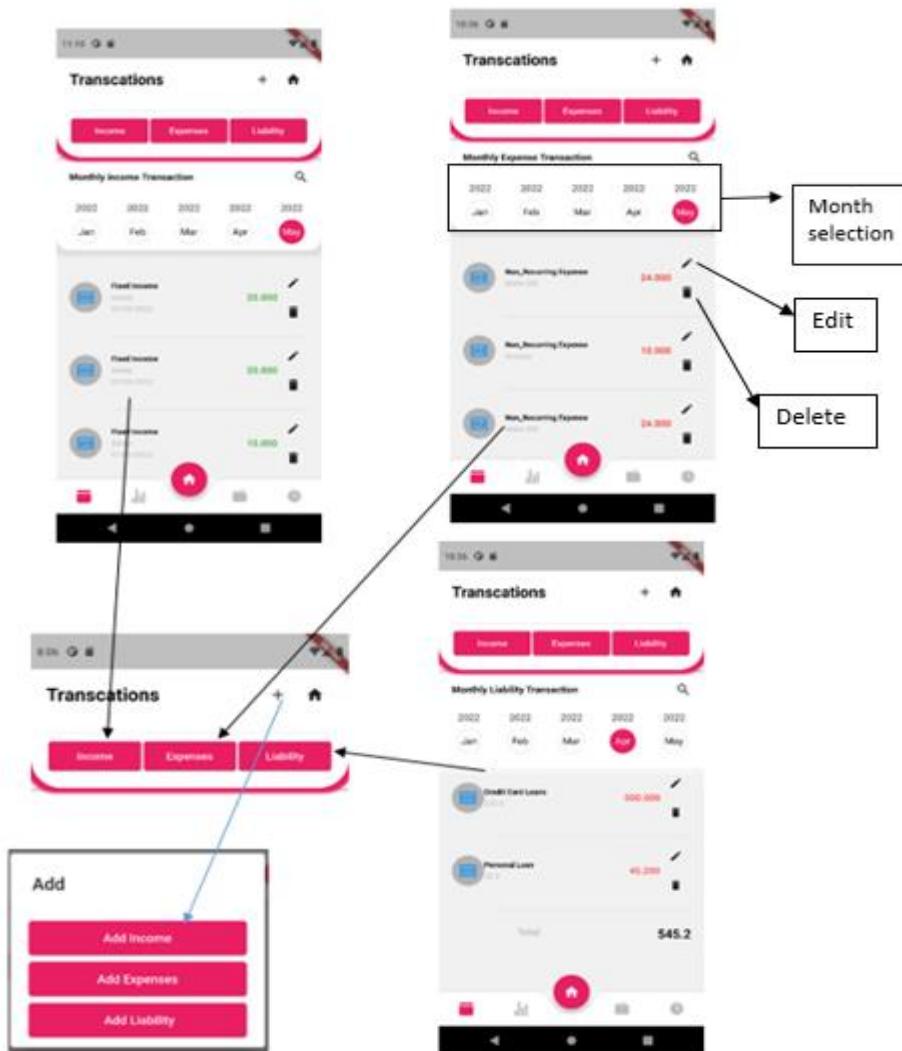


Figure 4.28 Transaction page(self,2022)

The user may also add any income, expense, or liability information by clicking the add button, which will open a dialog box where the user can pick the option they wish to add and be sent to the appropriate pages. Figure 4.29 explains this further.

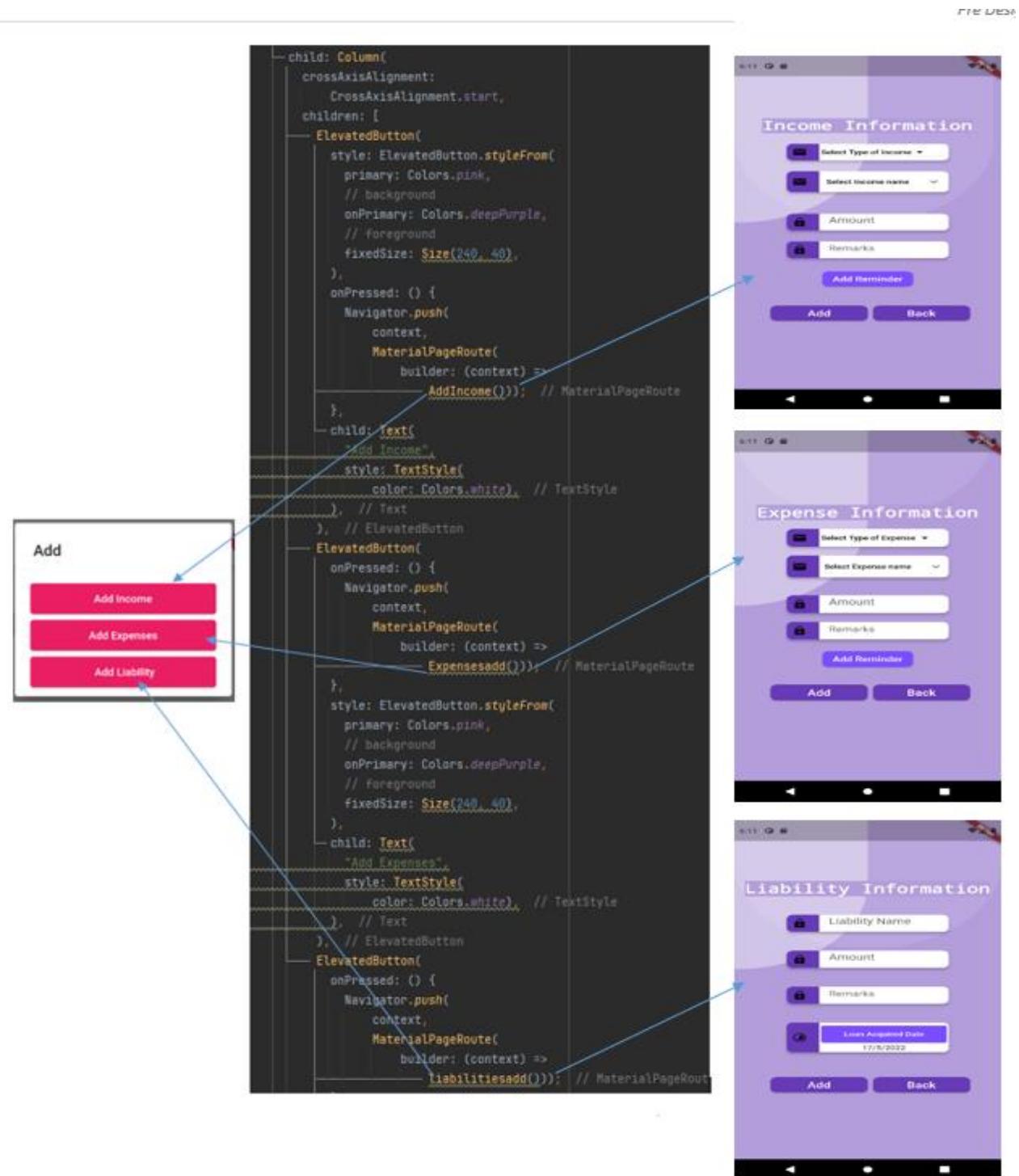


Figure 4.29 Transaction add(self,2022)

As previously stated, the user may examine the transactions based on the month they have chosen. The code is designed in such a manner that the data is first received from the Firestore database by giving the

path and then placed in a list that is then classified by month. Therefore, the user may readily access previously classified data based on the month of choice. This is illustrated in figure 4.30.

```

Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey.withOpacity(0.1),
    body: FutureBuilder(
      future: FireStoreDataBase().getIncomeData(),
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return const Text(
            "Something went wrong",
          );
        } // Text
        if (snapshot.connectionState == ConnectionState.done) {
          data = snapshot.data as List;
          data.sort((a, b) => a["incomedate"].compareTo(b["incomedate"]));
        }
        for (var i = 0; i < data.length; i++) {
          DateTime date = (data[i]["incomedate"] as Timestamp).toDate();
          data[i]["Income"] = date;
        }
        sum = 0.0;
        dataList = [];
        for (var i = 0; i < data.length; i++) {
          if (data[i]["Income"].month == activeDay+1) {
            dataList.add(data[i]);
            sum += data[i]["Incomeamt"];
          }
        }
        return getBody();
      }
    );
    return const Center(child: CircularProgressIndicator());
  );
}

```

```

Future getIncomeData() async{
  try{
    await collectionRef.doc(user.uid).collection("Income").get().then((querySnapshot){
      for (var result in querySnapshot.docs){
        studentlist.add(result.data());
      }
    });
    return studentlist;
  }
  catch(e){
    debugPrint('Error - $e');
    return null;
  }
}

```

Figure 4.30 Transaction data access and classification (self,2022)

The user can also delete and edit data from the Firestore database by specifying the path in the code as illustrated in figure 4.31. The data in the Firestore database will be modified or removed after the user acknowledges that they wish to edit or delete it. The modifications may be displayed in the transaction page once they have been successfully performed in the Firestore database.

```

Future deleteData(Map<String, dynamic> deletedata) async {
  List reminderdata = [];
  Map dataadd = Map<String, dynamic>();

  String datato = deletedata.toString();

  final CollectionReference collectionref =
  FirebaseFirestore.instance.collection("Student");

  final user = FirebaseAuth.instance.currentUser;
  try {
    await collectionref
      .doc(user.uid)
      .collection("Income")
      .get()
      .then((querySnapshot) {
    for (var result in querySnapshot.docs) {
      dataadd[result.id] = result.data().toString();
      reminderdata.add({result.id: result.data()});
    }
  });
  var Key = dataadd.keys
    .firstWhere((k) => dataadd[k] == datato, orElse: () => null);
  await collectionref
    .doc(user.uid)
    .collection("Income")
    .doc(Key)
    .delete();
  Fluttertoast.showToast(
    msg: 'Data deleted successfully successfully',
    toastLength: Toast.LENGTH_SHORT,
    gravity: ToastGravity.TOP,
    timeInSecForIosWeb: 1,
    fontSize: 16.0);
  return Key;
} catch (e) {
  debugPrint('Error - $e');
  Fluttertoast.showToast(
    msg: 'Failed to delete details: $e',
    toastLength: Toast.LENGTH_SHORT,
    gravity: ToastGravity.TOP,
    timeInSecForIosWeb: 1,
    fontSize: 16.0);
  return null;
}
}

Future updatedata(Map<String, dynamic> deletedata, double Incomeamt) async {
  var dataoo = deletedata.toString();
  Map dataadd = Map<String, dynamic>();

  final CollectionReference collectionref =
  FirebaseFirestore.instance.collection("Student");

  final user = FirebaseAuth.instance.currentUser;
  try {
    await collectionref
      .doc(user.uid)
      .collection("Income")
      .get()
      .then((querySnapshot) {
    for (var result in querySnapshot.docs) {
      dataadd[result.id] = result.data().toString();
    }
  });
  var Key = dataadd.keys
    .firstWhere((k) => dataadd[k] == datato, orElse: () => null);

  await collectionref.doc(user.uid).collection("Income").doc(Key).update({
    'incomeamt':Incomeamt,
  });
  Fluttertoast.showToast(
    msg: 'Data updated successfully successfully',
    toastLength: Toast.LENGTH_SHORT,
    gravity: ToastGravity.TOP,
    timeInSecForIosWeb: 1,
    fontSize: 16.0);
  return Key;
} catch (e) {
  debugPrint('Error - $e');
  Fluttertoast.showToast(
    msg: 'Failed to update details: $e',
    toastLength: Toast.LENGTH_SHORT,
    gravity: ToastGravity.TOP,
    timeInSecForIosWeb: 1,
    fontSize: 16.0);
  return null;
}
}

```

*Delete Data**Update data*

Figure 4.31Edit and delete data (self,2022)

4.1.3.2. Statistics page

The statistics page summarizes the user's revenue and spending information so that the user may quickly comprehend their monthly budgeting performance. The budget page, like the previous pages, is the app bar and will modify the content of the budget page depending on whether the user selected income or expenses. This is illustrated in figure 4.32. The income and expense body designs are identical. The user's income or spending total is presented on this page, along with a graph indicating their actual income and budgeted income, so the user can see how much of their intended budget or goal they have met. The user's income and expenses are then displayed in a pie chart that is categorized depending on the name of the income or expense so that the user may understand which sort of income or expense is greater and budget appropriately. Another pie chart is shown dependent on the type of income or expense. The sync fusion flutter library was used to design and develop the graphs. The graphs were designed in other pages and then positioned the budget income and expenses page. The bar chart and pie chart code is illustrated in figure 4.33 and figure 4.34.

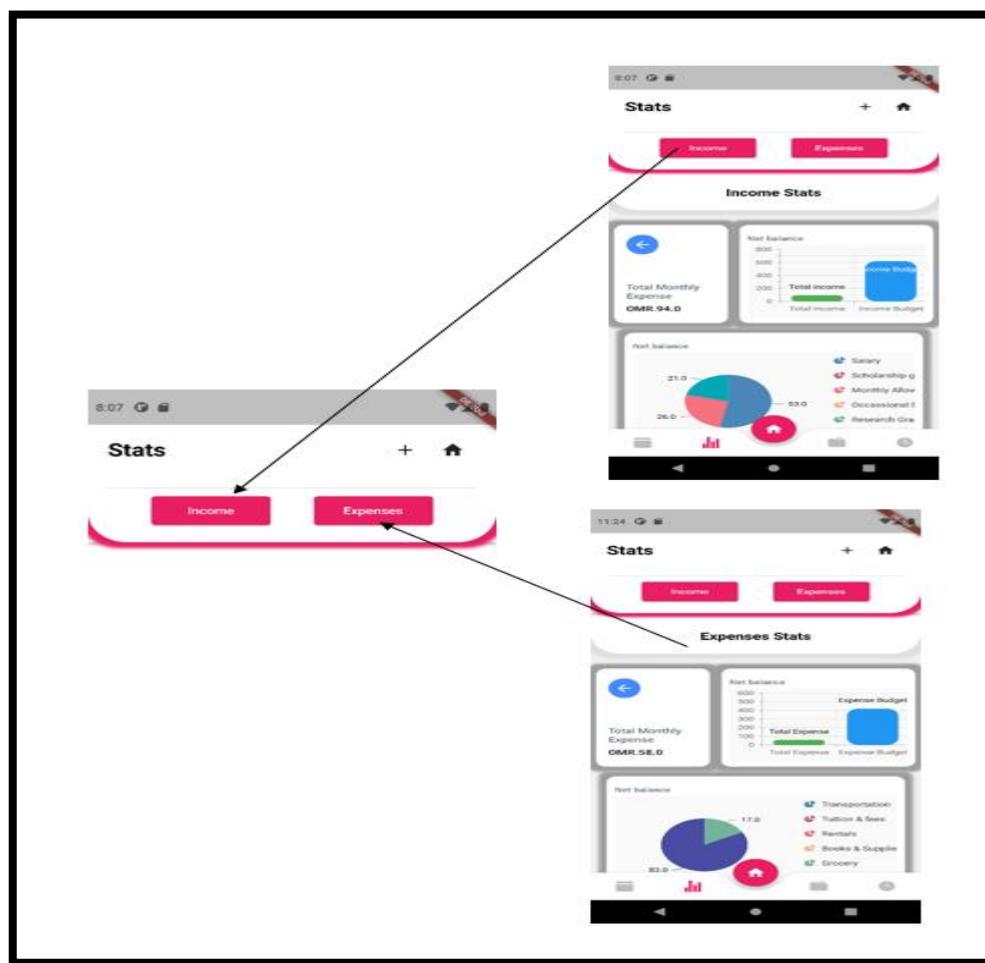


Figure 4.32 Statistics page(self,2022)

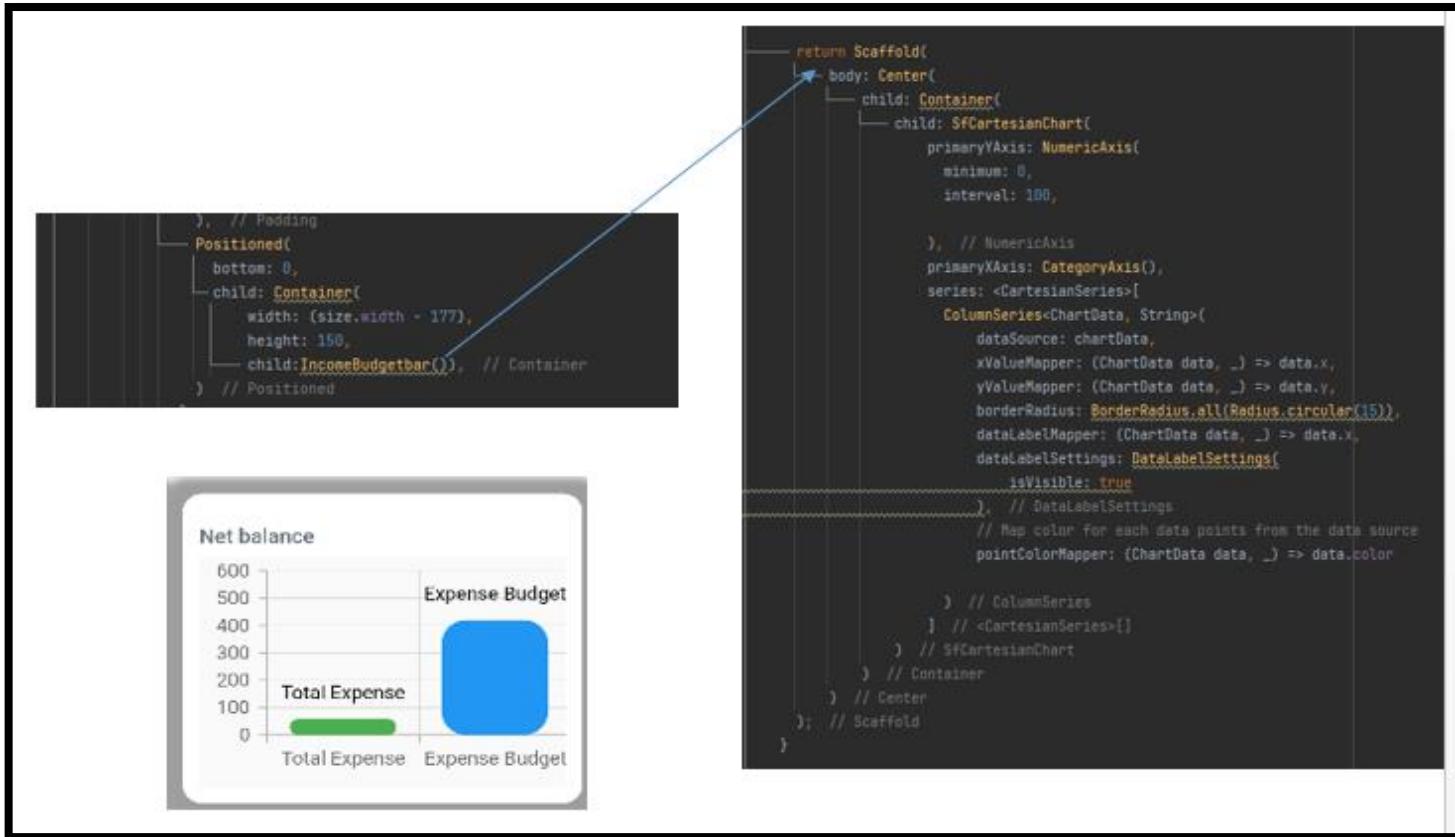


Figure 4.33 bar chart code (self,2022)

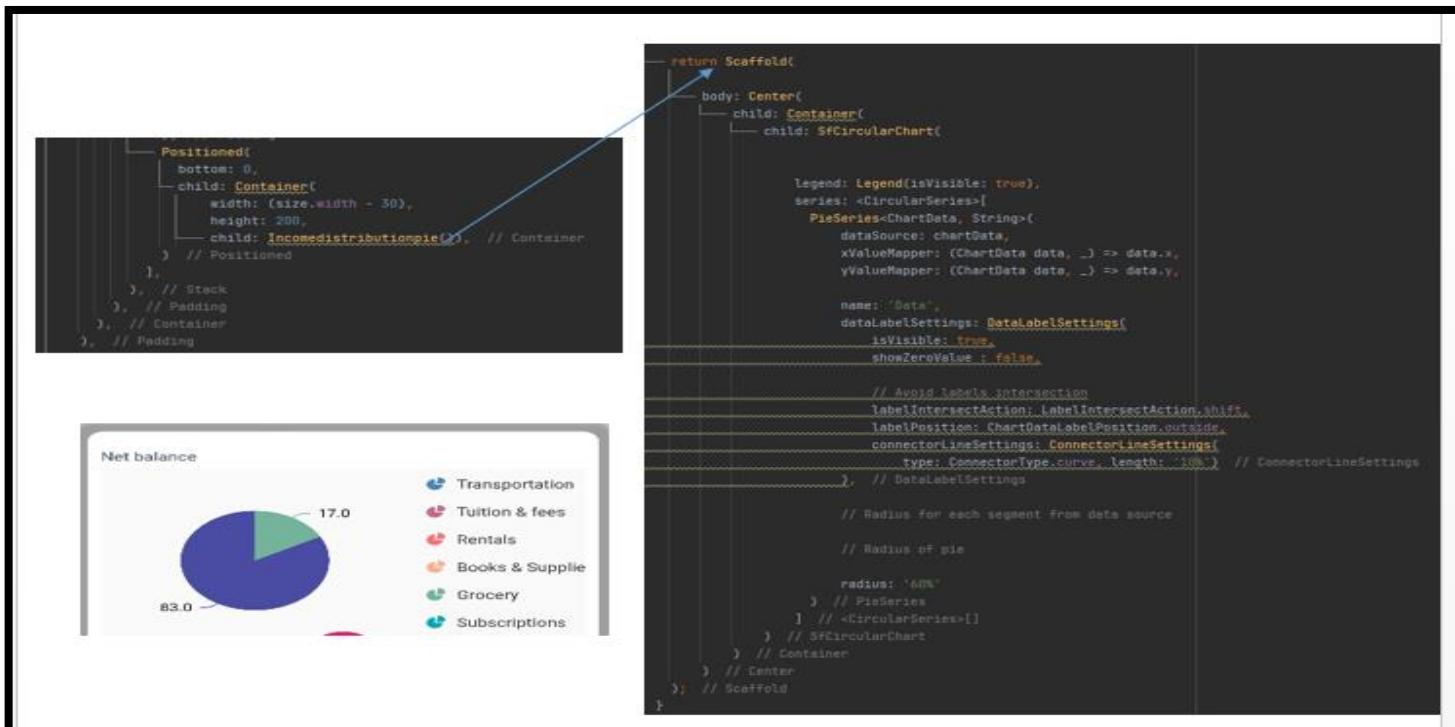


Figure 4.34 Pie chart code (self, 2022)

4.1.3.3. Monthly budget page

The goal of the budget app is to summarize the user's budget and loan progress. This page, like the statistics page, was developed so that users may quickly analyze their budget, savings, and loan progress. The budget page is an app bar, and the body of the budget page varies depending on whether the user chooses Budget data or Loan data as illustrated in figure 4.35. The code snippet is shown in figure 4.36. The budget data on this page presents the total spent summary, which shows how much of their earned income they have spent. The income and expense budget is then presented, indicating how much of their intended budget has been met. The user may also see how much of their savings target they have met. Furthermore as illustrated in figure 4.35, using the edit icon option presented in each of their containers, the users may alter their monthly income or expenses budget, and savings. Each element on the budget data page includes a progress bar that shows their performance in relation to their target at the end and their actuals at the beginning. Each part also includes a percentage in the center and a summary at the conclusion to help the user understand how far they have progressed with their budget and how far they still have to go.

The loan data, like the Budget data, attempts to summarize all of the user's liabilities and how much the user is obligated to pay back. The design is also consistent across the budget and loan data. The user should be able to see how much of their liabilities they have taken and how much they have paid. This page will provide the overall liability summary as well as any loans taken out by the user. The first feature that the viewer may examine is the total loans taken summary, which displays the sum of all loans taken at the end and the amount paid at the beginning. The progress bar is intended to In a sense, red represents 'they have to pay' and green represents 'they paid this amount towards their liabilities.' The loan percentage and summary are also highlighted. Following the summary, the remaining elements utilize the same design as the loan summary to display each loan the user has taken. The user as shown in figure 4.35 can also give their loan repayment details by clicking the plus sign in each element and a dialog box asking the loan repayment details which will be showcased. The loan repayment will alter liability progress and added to the database as an expense. This section's goal is to make the user aware of their liabilities and make decisions accordingly.

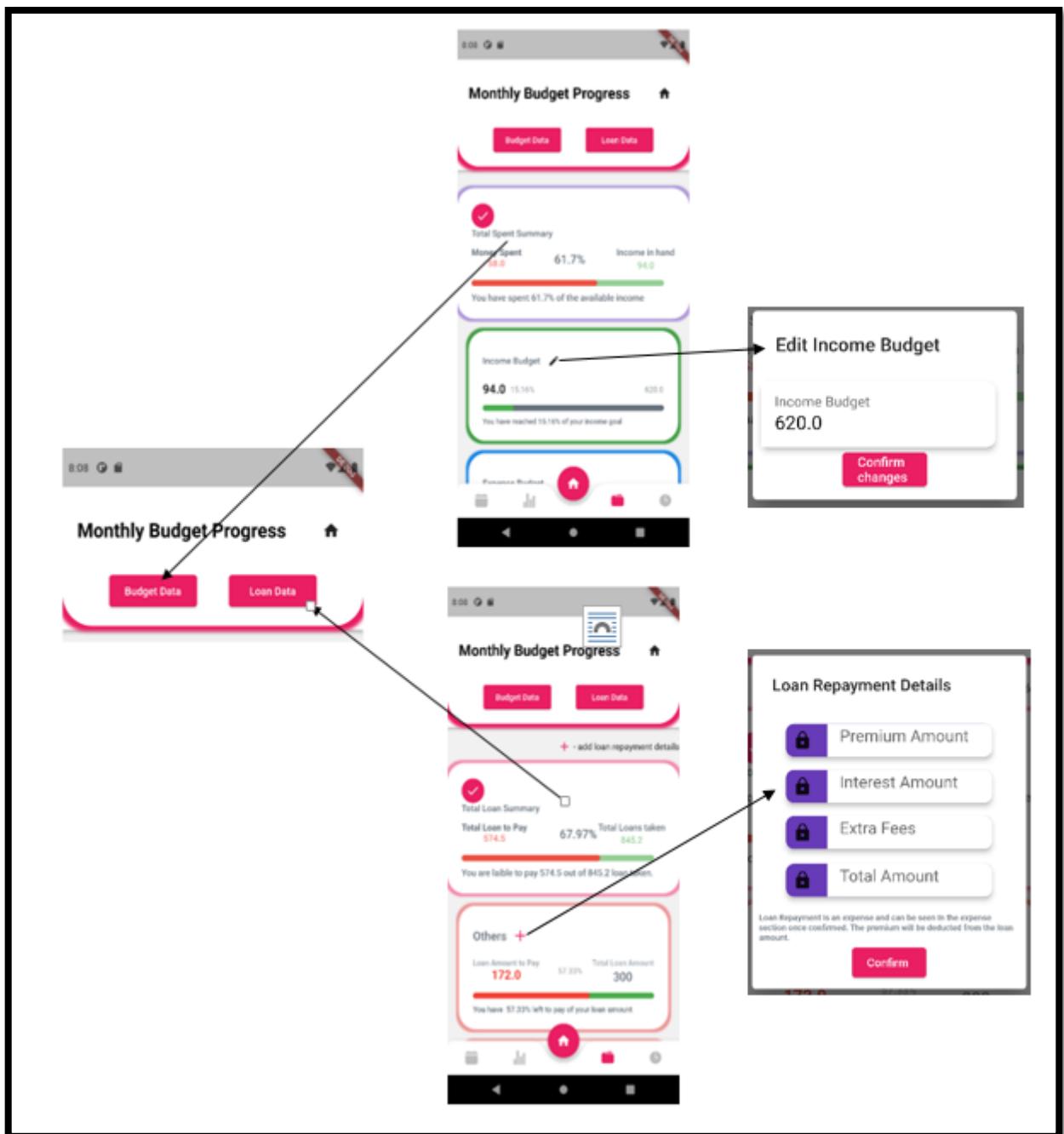


Figure 4.35 Monthly budget progress (self,2022)

```
- Container(
  width: 40,
  height: 40,
  decoration: BoxDecoration(
    shape: BoxShape.circle, color: Colors.pink), // BoxDecoration
- child: Center(
  child: Icon(
    Icons.done,
    color: Colors.white),
), // Icon, Center
), // Container
- Text(
  "Total Spent Summary",
  style: TextStyle(
    fontWeight: FontWeight.w500,
    fontSize: 15,
    color: Color(0xff67727d)),
), // Text
- SizedBox(
  height: 10,
), // SizedBox
- Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Row(
      children: [
        Column(
          children: [
            Text(
              "Money Spent",
              style: TextStyle(
                fontWeight: FontWeight.bold,
                fontSize: 15,
                color:
                Color(0xff67727d).withOpacity(1.0)),
            ), // Text
            Text(
              expenselist[0]['monthlyExpense'].toString(),
              style: TextStyle(
                fontWeight: FontWeight.bold,
                fontSize: 15,
                color: Colors.red.withOpacity(0.6),
            ), // TextStyle
            ), // Text
          ],
        ), // Column
      ],
    ), // Row
  ],
), // Row
```

Figure 4.36 Monthly budget page code snippet (self,2022)

4.1.3.4. Reminder page:

The reminder page, as the name implies, displays all of the reminders. As illustrated in Figure 4.37, the user can view, add, edit, and delete their reminders from this page. Figure 4.38 shows a code snippet for the reminder page. The user may access all of their reminders based on the month they have chosen. The data was retrieved from the Firestore database, categorized by month, and then displayed to the user. The reminder's name and date are shown for the user to view. When the user clicks the + symbol, they will be led to the add reminder page, where they may input the reminder name, pick whether they want recurring or non-recurring reminders, and also select the reminder date. As seen in figure. 4.39, the data will then be uploaded to the Firestore database. Changes to the Firestore database and the reminders page will result from editing or deleting reminders from the page.

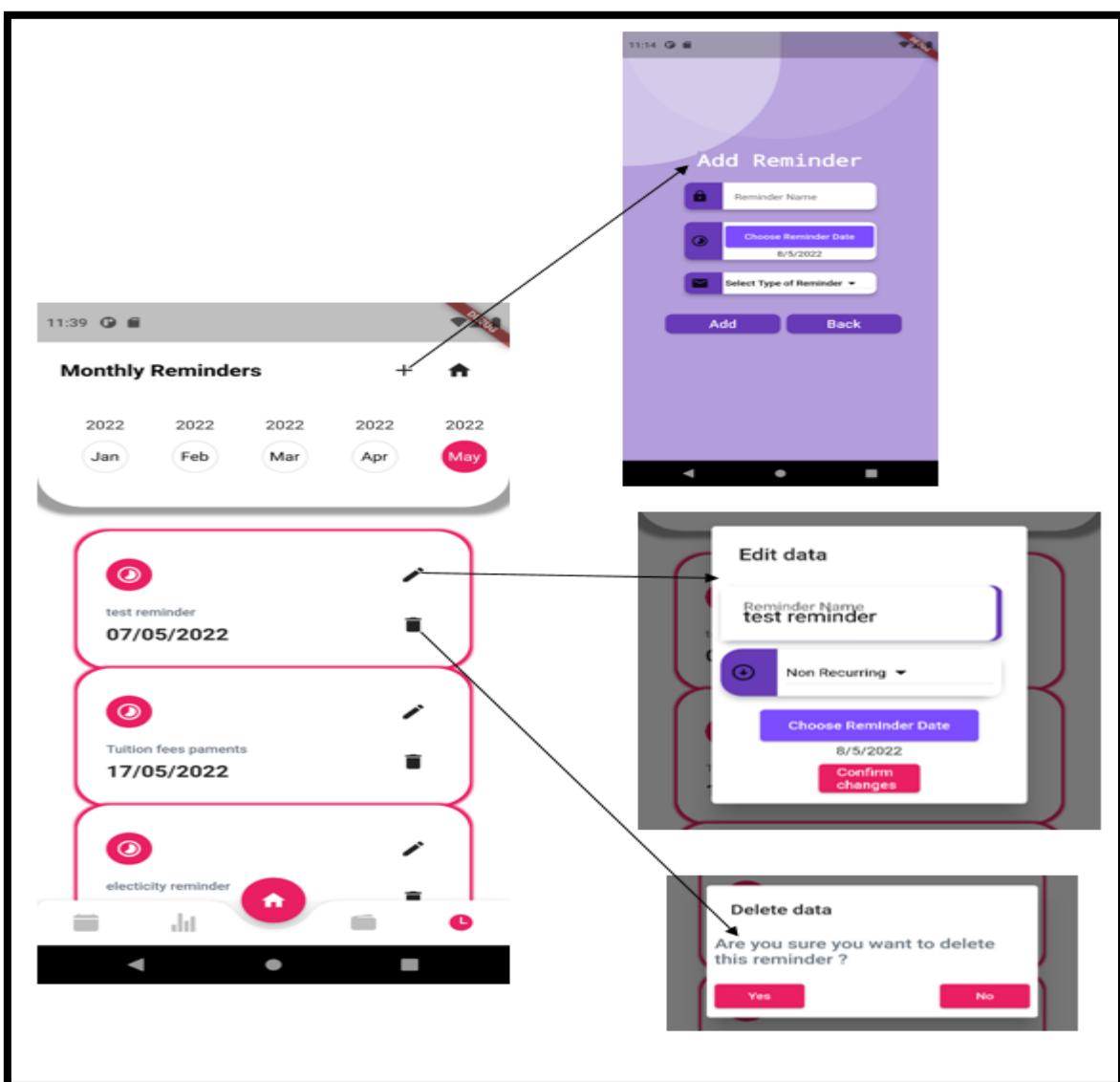


Figure 4.37 Reminder page (self,2022)

```

import ...

class Reminderpage extends StatefulWidget {
  const Reminderpage({Key? key}) : super(key: key);

  @override
  _ReminderpageState createState() => _ReminderpageState();
}

class _ReminderpageState extends State<Reminderpage> {
  @override
  int activeDay = DateTime.now().month - 1;
  bool isSwitched = false;
  DateTime selectedDate = DateTime.now();

  List<dynamic> dataList = [];
  List<dynamic> data = [];
  List<dynamic> users = [];
  String? dropdownvalue;
  List student = [];
  final TextEditingController Reminderdate = TextEditingController();

  final user = FirebaseAuth.instance.currentUser!;

  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey.withOpacity(0.15),
      body: FutureBuilder(
        future: FireStoreDataBase().getReminderData(),
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            return const Text(
              "Something went wrong",
            ); // Text
          }
          if (snapshot.connectionState == ConnectionState.done) {
            data = snapshot.data as List;
            data.sort((a, b) => a["Reminderdate"].compareTo(b["Reminderdate"]));

            for (var i = 0; i < data.length; i++) {
              DateTime date = (data[i]["Reminderdate"] as Timestamp).toDate();
              data[i]["Reminder"] = date;
            }

            dataList = [];
            for (var i = 0; i < data.length; i++) {
              if (data[i]["Reminder"].month == activeDay + 1) {
                dataList.add(data[i]);
              }
            }
          }
        },
      ),
    );
  }
}

```

Figure 4.38 Reminder page code snippet (self,2022)

```
1 import ...
10
11 class Reminderadd extends StatefulWidget {
12   const Reminderadd({Key? key}) : super(key: key);
13
14   @override
15   _ReminderaddState createState() => _ReminderaddState();
16 }
17
18 class _ReminderaddState extends State<Reminderadd> {
19   @override
20   String? dropdownvalue;
21   bool isSwitched = false;
22   DateTime selectedDate = DateTime.now();
23   final TextEditingController Remindername=TextEditingController();
24   final TextEditingController Reminderdate=TextEditingController();
25   @override
26   Widget build(BuildContext context) {
27     return Form(
28       child: Container(
29         width:MediaQuery.of(context).size.width,
30         height:MediaQuery.of(context).size.height,
31         color: Colors.deepPurple[200],
32         child: Stack(
33           children:<Widget>[
34             Align(
35               alignment: Alignment.bottomRight,
36               widthFactor: 0.5,
37               heightFactor: 0.5,
38               child: Material(
39                 color: Color.fromRGBO(255, 255, 255, 0.4),
40                 borderRadius: BorderRadius.all(Radius.circular(200.0)),
41                 child: Container(
42                   width:500.0,
43                   height: 500.0,
44                 ), // Container
45               ), // Material
46             Align(
47               alignment: Alignment.bottomLeft,
48               widthFactor: 0.5,
49               heightFactor: 0.5,
50               child: Material(
51                 color: Color.fromRGBO(255, 255, 255, 0.1),
52                 borderRadius: BorderRadius.all(Radius.circular(200.0)),
53                 child: Container(
54                   width:600.0,
```

Figure 4.39 Add reminder code

4.1.4. Offers

The offers page is intended to assist students in taking advantage of local deals and student discounts. This page's color scheme is green, which represents money and serenity. This page is designed to be simple so that the user can see their most recent offers and locations at a glance. The app bar and the content of the offers page, like the other pages, alter depending on the user's decision between offers and student discounts. Figure 4.40 provides an illustration of this. The user can exit the offers app by clicking the home symbol, which will take them to the home page, or by clicking the profile icon, which will take them to their profile page. Only two offers are visible in a row, but numerous offers can be viewed by scrolling down depending on what offers are available in the database . The database for student discounts and offers is managed by the administrator. Figure 4.41 shows the detailed code for the element displaying the offer.

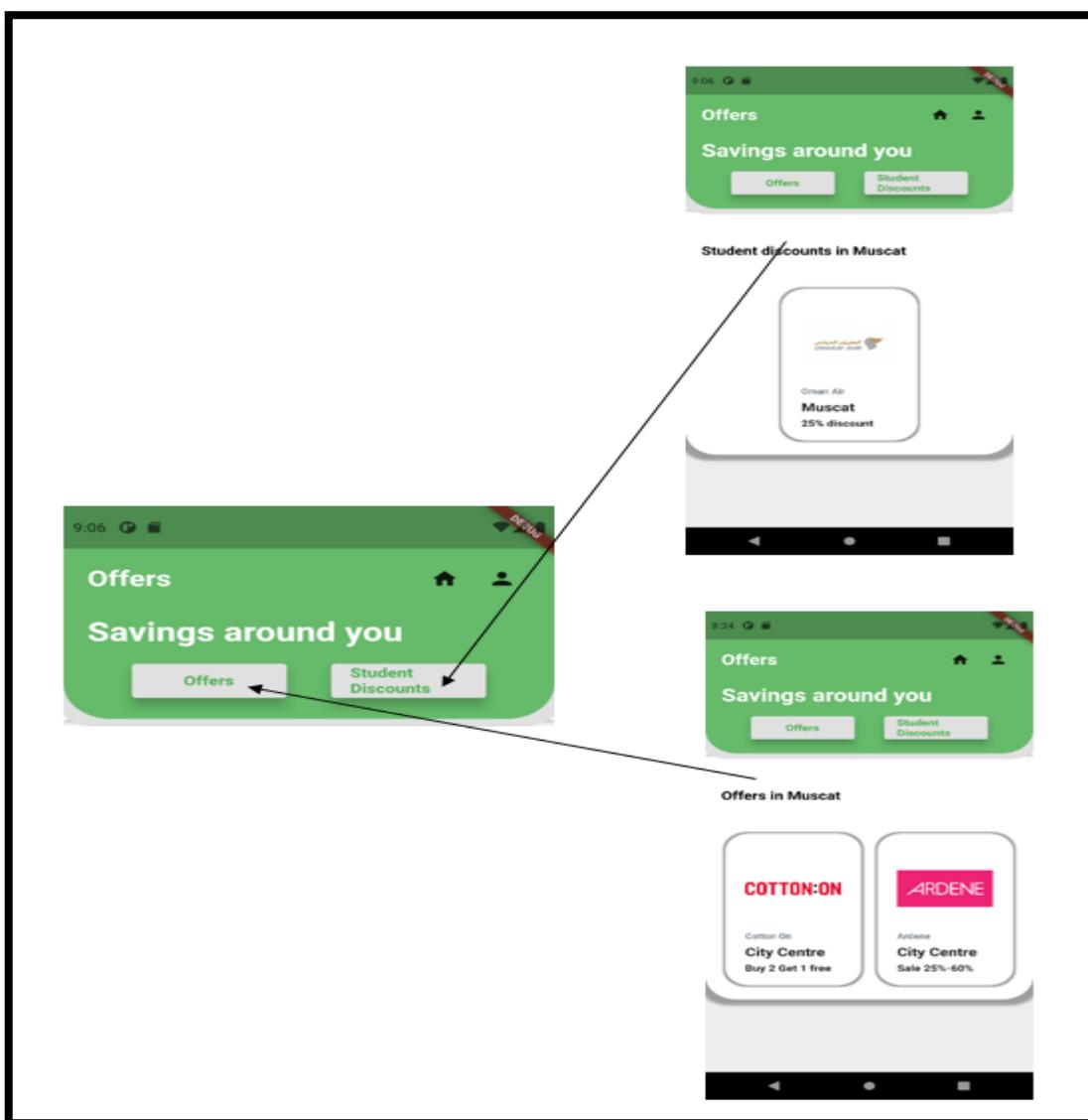


Figure 4.40 Offers page(self,2022)

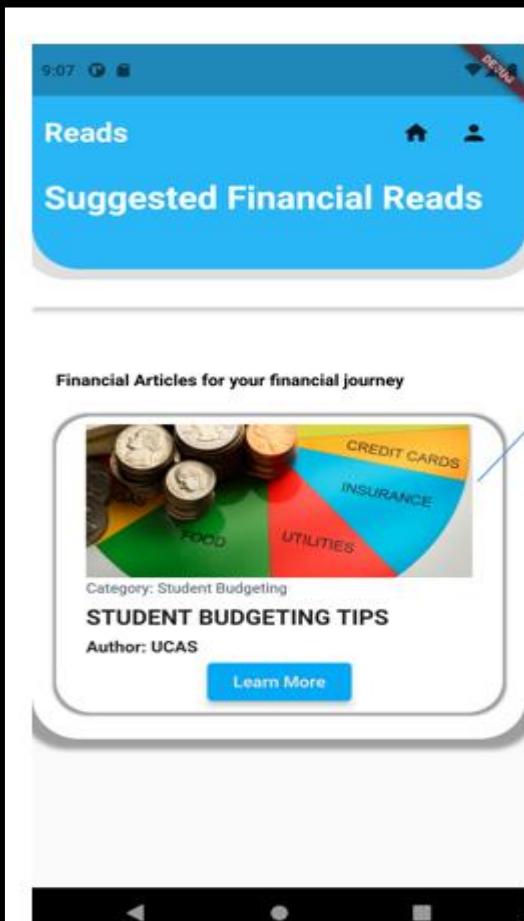


```
child: Padding(
    padding: const EdgeInsets.only(
        left: 25, right: 25, top: 20, bottom: 20), // EdgeInsets.only
    child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
            Container(
                child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                        Column(
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: [
                                Container(
                                    width: 120,
                                    height: 150,
                                    child: Image.network(data[index]["ImageLink"]),
                                ), // Image.network, Container
                                Text(
                                    data[index]["Companyname"],
                                    style: TextStyle(
                                        fontWeight: FontWeight.w500,
                                        fontSize: 13,
                                        color: Color(0xff67727d)), // TextStyle
                                ), // Text
                                SizedBox(
                                    height: 8,
                                ), // SizedBox
                                Text(
                                    data[index]["Location"],
                                    style: TextStyle(
                                        fontWeight: FontWeight.bold,
                                        fontSize: 20),
                                ), // TextStyle
                                SizedBox(
                                    height: 8,
                                ), // SizedBox
                                Text(
                                    data[index]["Offer"],
                                    style: TextStyle(
                                        fontWeight: FontWeight.bold,
                                        fontSize: 15),
                                ), // TextStyle
                            ],
                        ), // Column
                    ],
                ), // Row
            ),
        ],
    ), // Column
); // Column
```

Figure 4.41 Element code (self,2022)

4.1.5. Financial read

This page seeks to assist students in improving their financial knowledge by directing them to chosen articles on the internet. The financial readings app has a similar design and code to the Offers app. The viewer may examine the article title, author name, and be directed to the article website from this page. Figure 4.42 depicts the code.



```

Column(
  mainAxisAlignment: MainAxisAlignment
    .start,
  children: [
    Container(
      width: (size.width - 90),
      height: 150,
      child: Image.network(data[index]["ImageLink"],
        ), // Image.network, Container

    Text(
      "Category: "+data[index]["ArticleType"],
      style: TextStyle(
        fontWeight: FontWeight
          .w500,
        fontSize: 13,
        color: Color(0xff67727d)), // TextStyle
    ), // Text
    SizedBox(
      height: 8,
    ), // SizedBox
    Text(
      data[index]["Articlename"],
      style: TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 20,
      ), // TextStyle
    ), // Text
    SizedBox(
      height: 8,
    ), // SizedBox
    Text(
      "Author: "+ data[index]["Authorname"],
      style: TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 15,
      ), // TextStyle
    ), // Text
  ],
)
  
```

Figure 4.42 Financial Reads (self,2022)

4.1.6. Profile page

The profile page is also an essential component of the application. The user may access their personal information, username, and password. The user may also sign out of the application and Firebase by clicking the signout button in the profile page. The user can exit the profile page by clicking the home symbol, which takes them to the Home page. Figure 4.43 shows a code excerpt for the program.

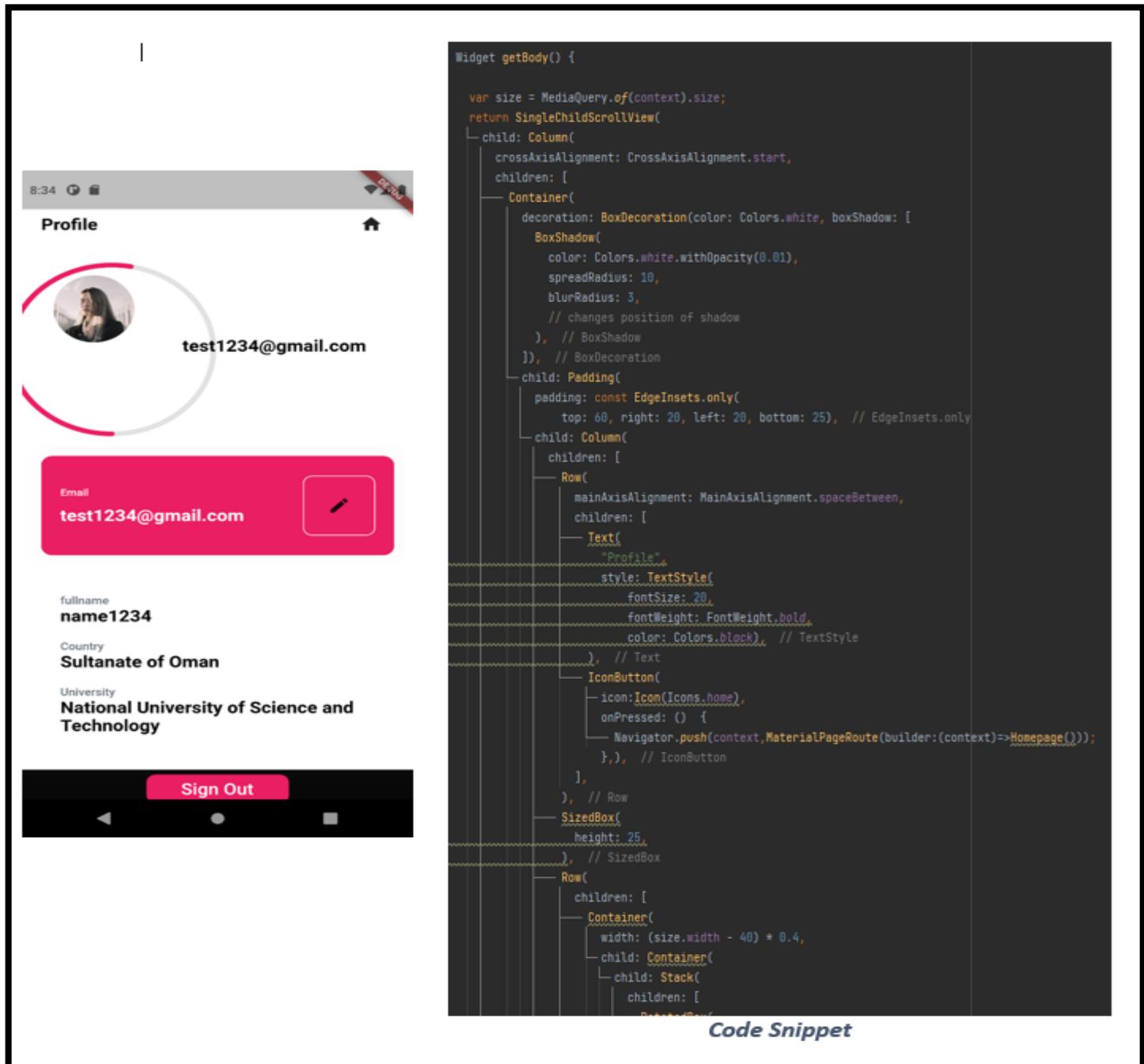


Figure 4.43 Profile page (self,2022)

4.1.7. Admin page

The admin page lets the admin to upload any new offers, student discounts, and financial reads so that the user has access to the most up-to-date information. The administrator may examine the offers, student discounts, and financial reads as the user views them and then make adjustments as needed. To add information, the administrator can click the plus sign, which will display a dialog box as shown in figure 4.44 and the code snippet as shown in figure 4.45. The administrator can then pick an option to be directed to the appropriate pages and enter the relevant data. There is no validation in this case since the administrator is part of the application and understands what is important or not. Furthermore, any updates to the existing data must be made using the Firebase console. All data is added to the Firestore database, which can subsequently be viewed on the user offers and must read page, as well as by the admin.

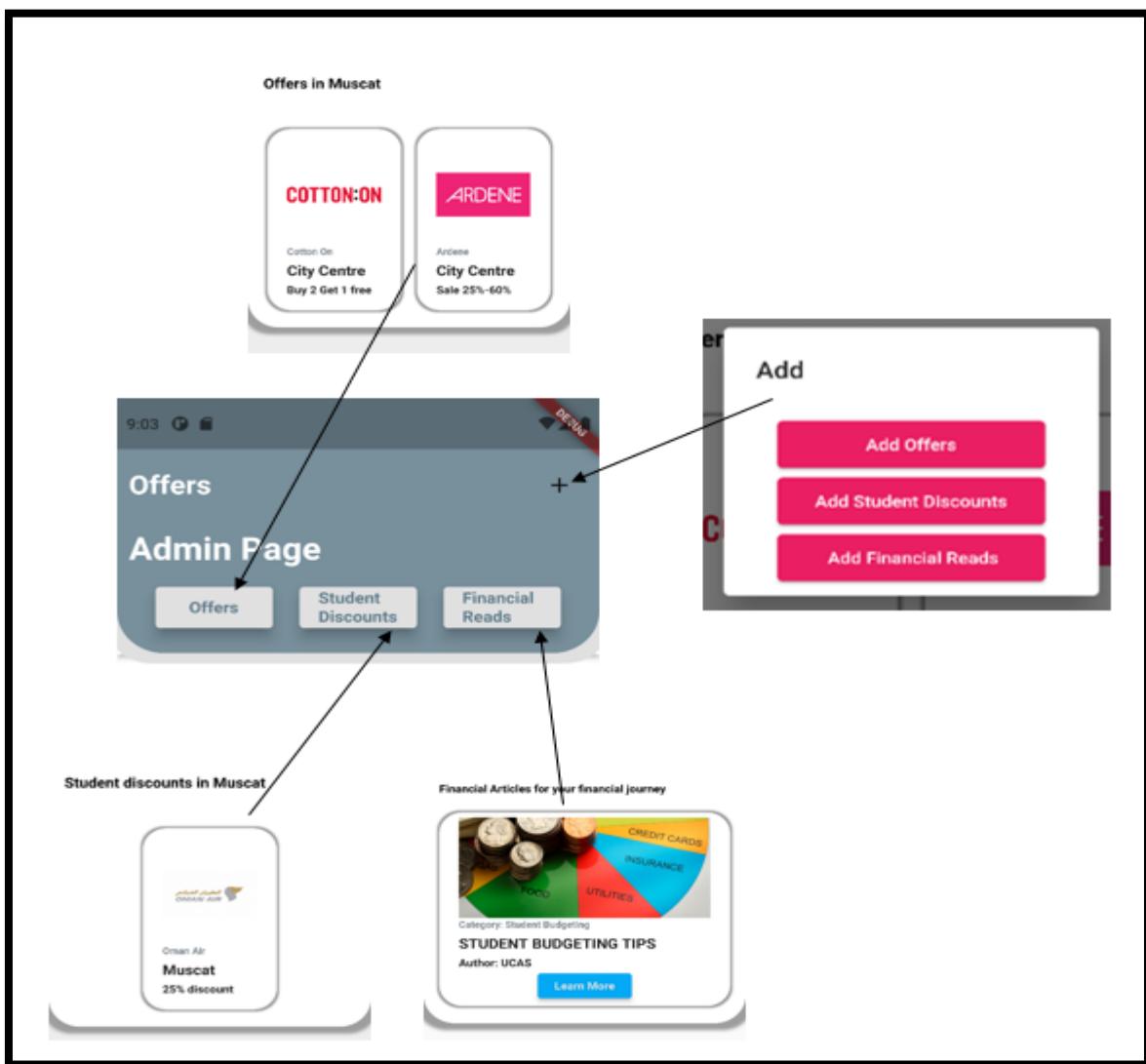
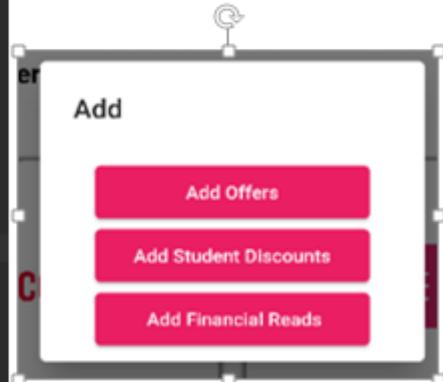


Figure 4.44 Admin Page(self,2022)



```
-children: [IconButton(  
  icon: Icon(Icons.add),  
  onPressed: () {  
    showDialog<String>(  
      context: context,  
      builder: (BuildContext context) => AlertDialog(  
        title: const Text('Add'),  
        actions: <Widget>[  
          Center(  
            child: Column(  
              mainAxisAlignment: MainAxisAlignment.start,  
              children: [  
  
                ElevatedButton(  
                  style: ElevatedButton.styleFrom(  
                    primary: Colors.pink, // background  
                    onPrimary: Colors.deepPurple, // foreground  
                    fixedSize: Size(200, 40),  
                  ),  
  
                  onPressed: () {  
                    Navigator.push(context, MaterialPageRoute(builder:(context)=>addOffers()));  
                  },  
  
                  child:  
                    Text(  
                      "Add Offers",  
                      style: TextStyle(  
                        color: Colors.white  
                      ), // TextStyle  
  
                    ), // Text  
  
                ), // ElevatedButton  
  
                ElevatedButton(  
                  onPressed: () {  
                    Navigator.push(context, MaterialPageRoute(builder:(context)=>addStudentdiscounts()));  
                  },  
  
                  style: ElevatedButton.styleFrom(  
                    primary: Colors.pink, // background  
                    onPrimary: Colors.deepPurple, // foreground  
                    fixedSize: Size(200, 40),  
                  ),  
                  child:  
                    Text(  
                      "Add Student Discounts",  
                      style: TextStyle(  
                        color: Colors.white  
                      ), // TextStyle  
                    ), // Text  
                ), // ElevatedButton  
              ],  
            ), // Column  
          ), // Center  
        ], // actions  
      ), // AlertDialog  
    );  
  },  
); // IconButton
```

Figure 4.45 Add information snippet(self,2022)

4.2. Testing

Software testing determines if the actual software application meets the standard requirements and ensures that the software application is free of defects. The goal of software testing is to find mistakes, gaps, or missing needs compared to the actual conditions (Hamilton, 2019) .In this section, various tests ensure that the application's features are operational and free of faults.

4.3.1. Test cases

Unit and integration testing is performed to determine whether or not the test cases perform well and without errors. Unit testing is performed to ensure that each software unit operates well, while integration testing is performed to ensure that the software functions work together. Unit testing is performed in the test cases to determine whether a component of the application is running correctly, and integration testing is performed to determine the links between various pages and the firebase.

	Test Performed	Testing method	Expected Output	Testing Result
1.	Login field texts validation	Unit	A message would be shown at the very top of the page if the user tried logging in without their registered email address and password	Pass
2.	Validate email and password by firebase authentication on the login page	Integration	Direct the user to Home Page or show an error message at the top of the page.	Pass
3.	Sign Up for page access from Login Page	Unit	The signup option on the login page should redirect the user to the Registration Page	Pass

4.	Registering user email address and password into firebase	Integration	The new user data should be seen on the firebase authentication page	Pass
5.	Checking if the user exists or not before registering	Unit	No user with the same email address can register twice. Thus, a message will be displayed at the top of the page stating that the user's email address exists	Pass
6.	Validating if the password and confirm password are the same and have more than six characters	Integration	If the characters are not more than six and if they confirm the password is not the same, then an error message will show in the text field	Pass
7.	All the data(income, expense, liability, budget, savings) is being added to the firebase when registering	Unit	The data is displayed in the Firestore database under the user UID in student collection.	Pass
8.	Fetch data from firebase and display them on the home page	Integration	Data is displayed on both the home page.	Pass
9.	Fetching data from the firebase, classifying them based on month, and	Integration.	All the transaction data can be seen concerning the month selected.	Pass

	displaying them on the transaction page.			
10.	Being able to view Income, expenses, and Liability data on the Transaction Page	Unit	The user can click the button and view their choice.	Pass
11.	Being able to edit the transactions	Unit	To confirm the transaction is successfully edited, a message will be displayed at the top of the page.	Pass
12.	Being able to delete the transaction data	Unit	To confirm the transaction is deleted, a message will be displayed at the top of the page.	Pass
13.	Adding income, expense, and liability data from the transaction page and the firebase.	Integration	The user will be directed to the respective add data pages, and a message that the user successfully added the data to the firebase, a message will be displayed at the top of the page.	Pass
14.	View income and expenses statistics	Integration	Should be able to view various summarized bar diagrams and two pie charts with a total amount summary on the statistics page	Pass

15.	View each of the graphs on the statistics page	Unit	The user should be able to view all the charts.	Pass
16.	Fetch data from firebase and summarize it to view in the budget page with the progress bar	Integration	The user can view their progress and small summary on the budget page. This is the same for the loan data page.	Pass
17.	Matching of the progress bar and the progress percentage	Integration	If the data is more than 100%, the progress bar should be complete, and the percentage should remain at 100.	Pass
18.	Fetch reminder data from firebase and display them on the reminder page	Integration	All the reminders of the chosen month will be displayed on the reminder page	Pass
19.	Edit and delete the data from the reminder page and Firebase	Integration	The reminder details are edited or deleted, and a message can be seen at the top with the reminder modified on the reminder page.	Pass
20.	View the latest offers and student discounts on the offers page	Integration	All the offers and student discounts can be viewed in the offers app	Pass
21.	Constructing emergency email for relative	Unit	Open an email filled with relative email as a receiver and text	Pass

			inside subject and text	
22.	Fetch user data from firebase and display them on the profile page	Integration	The user information should be displayed on the profile page	Pass
23.	Signout of the application from the profile page	Unit	The user should be redirected to the application login page.	Pass

4.3. Summary

The application was implemented with the Firebase database, various libraries, and methods that assured the application was completed and performed effectively. The Firebase database supported the application in authenticating, adding, editing, and deleting data from the database and retrieving data to create meaningful pages for the user. The application included several sub-applications created concurrently, and the home page was critical in unifying the entire application into one. Testing is essential to ensure that the program is effectively deployed and running correctly. The unit and integration tests confirmed that the program functioned adequately together.

CHAPTER 5 RESULTS AND DISCUSSIONS

The application effectively employed data analytics and machine learning to evaluate user data and provide detailed and summary insights on the user's income, spending patterns, budgeting progress, and credit or loan status. In addition, the machine learning successfully predicted the user's upcoming income and expenses. In this section, detailed information about analysis and machine learning is provided.

5.1. User Data Analysis results

Data analysis is the summarization of acquired data. It entails interpreting data acquired using analytical and logical reasoning in order to find patterns, connections, or trends (Coursera, 2022). The data analysis was utilized in the application to summarize the user data and provide insights that the user may examine at a glance via graphs, summary, and progress bar. This seeks to make the user financially aware and assist them in making financial decisions without requiring too many financial details that could potentially confuse them. This saves the user time and motivates the user to make wise financial decisions.

The application uses graphs to summarize the user income and expenses for that month. The data analysis for income and expenses using graphs is shown in Figure 5.1. This aims at letting the user know their income and expenses performance at a glance.

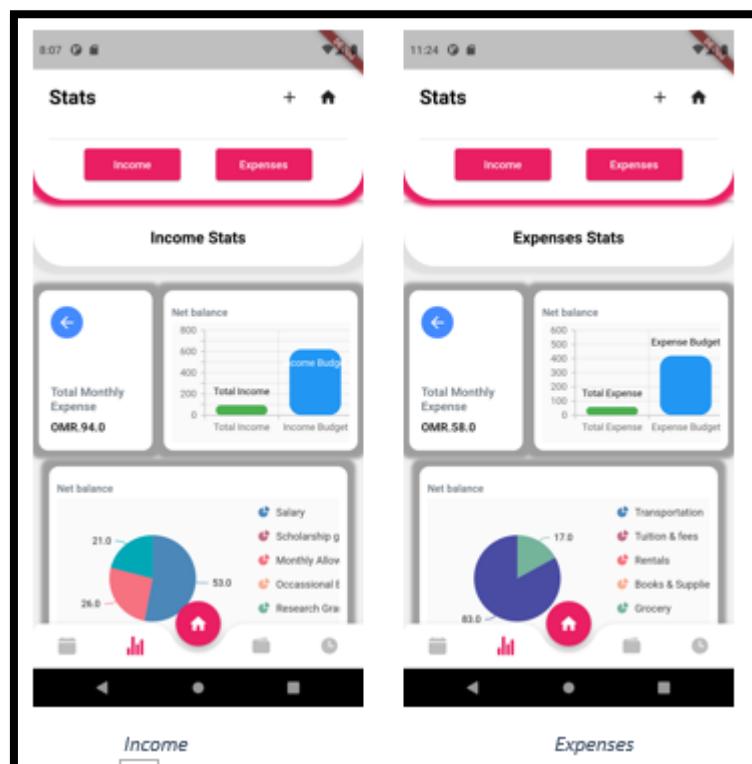


Figure 5.1 Income and expenses data analysis(self,2022)

Figure 5.2 shows how the program employs a progress bar to visualize the user's progress toward their spending and budget goals. This is also utilized since graphs are never sufficient to summarize facts for everyone. Each progress bar also provides a percentage and a word summary for further understanding of their progress.

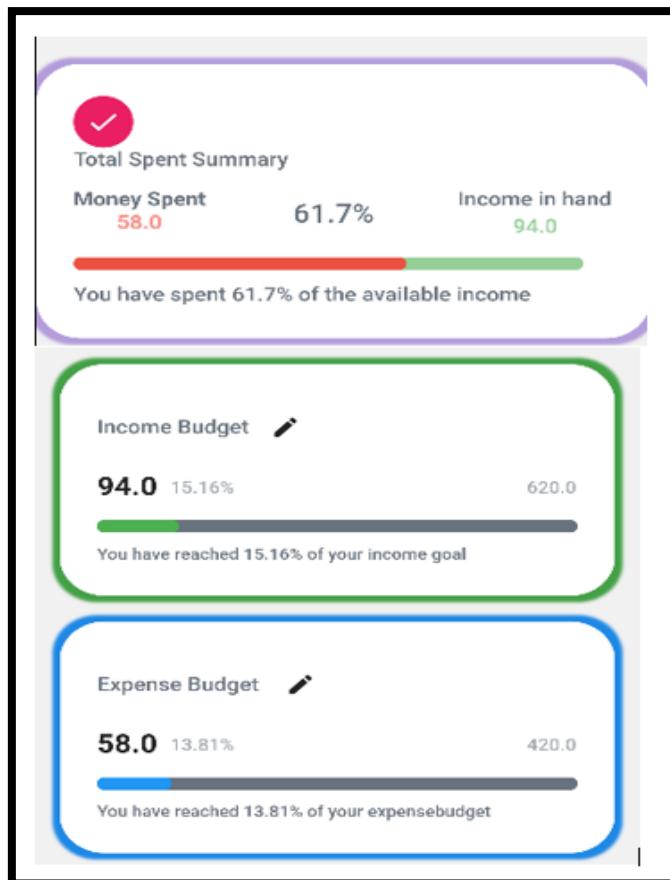


Figure 5.2 Progress Bar Income and expenses (self,2022)

Progress bars are also used to analyze the user's savings progress, as shown on the home overview page and budget page, as shown in Figure 5.3.

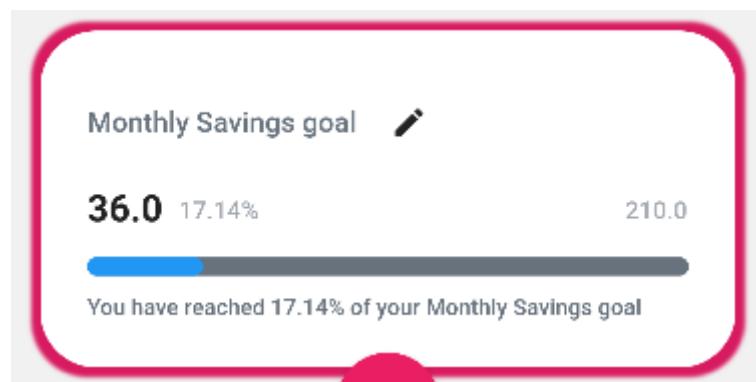


Figure 5.3 Savings progress (self,2022)

Loans are also a significant aspect of a person's life, and paying on time and keeping track of the loan progress is critical. This will promote and assist the user in making timely loan repayments while also keeping them aware of their commitments or obligations. Figure 5.4 depicts how a progress bar and summaries were utilized to display user loan progress for simple examination. The green color highlights good progress, whereas the red color highlights poor progress.

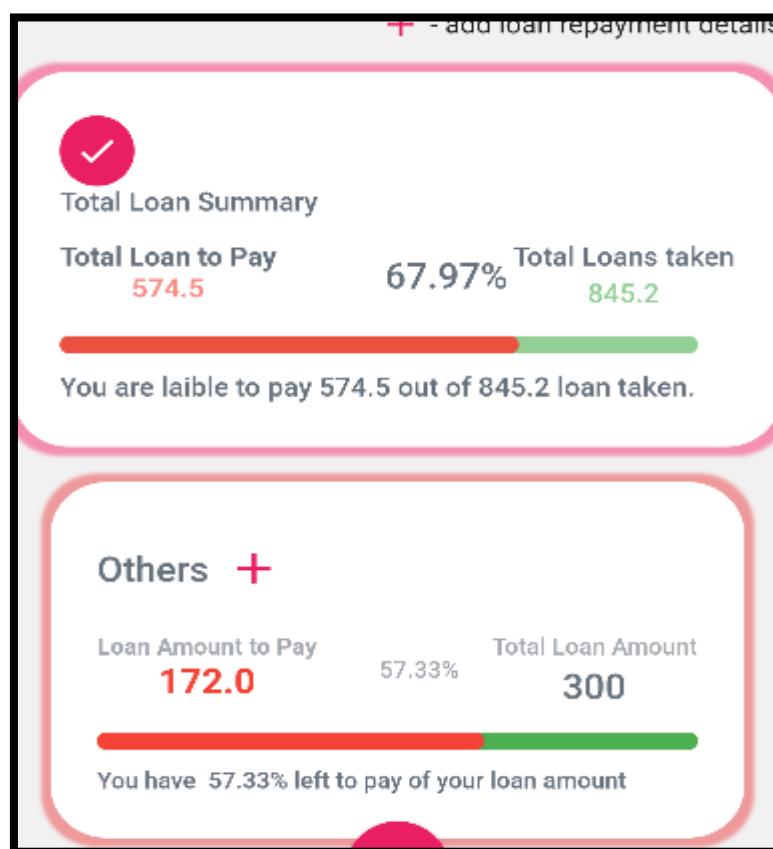


Figure 5.4 Loan progress (self,2022)

5.2. Machine Learning

The application intends to use predictive machine learning to forecast the students' earnings and expenditures for the upcoming month. This is done so that the student is aware of their prospective performance for the following month and may make changes to their spending patterns and become more financially aware and wise.

To anticipate the user's next month's data, machine learning is performed within the application using the flutter's ml algo and ml data frame libraries. To predict user data, linear regression or linear regressor (as it is known in the ml algo library) was chosen as the best machine learning algorithm for the prediction. Linear Regression is a supervised Machine Learning model that determines the linear relationship between dependent and independent variables (Deepanshi, 2021). As previously stated, no current datasets have been collected to train and forecast. Thus, the data provided by the user is categorized depending on the month and then run into the program to anticipate the user data using machine learning libraries. This is done every time a user enters new information into the database.

The application collects the user data, predicts the user income and expenses, and displays the data on the home stats page, as illustrated in Figure 5.5. The Home stats aims at summarizing the user statistics progress, and the predicted value is displayed at the top so that the user can view their predicted next month's performance at a glance.

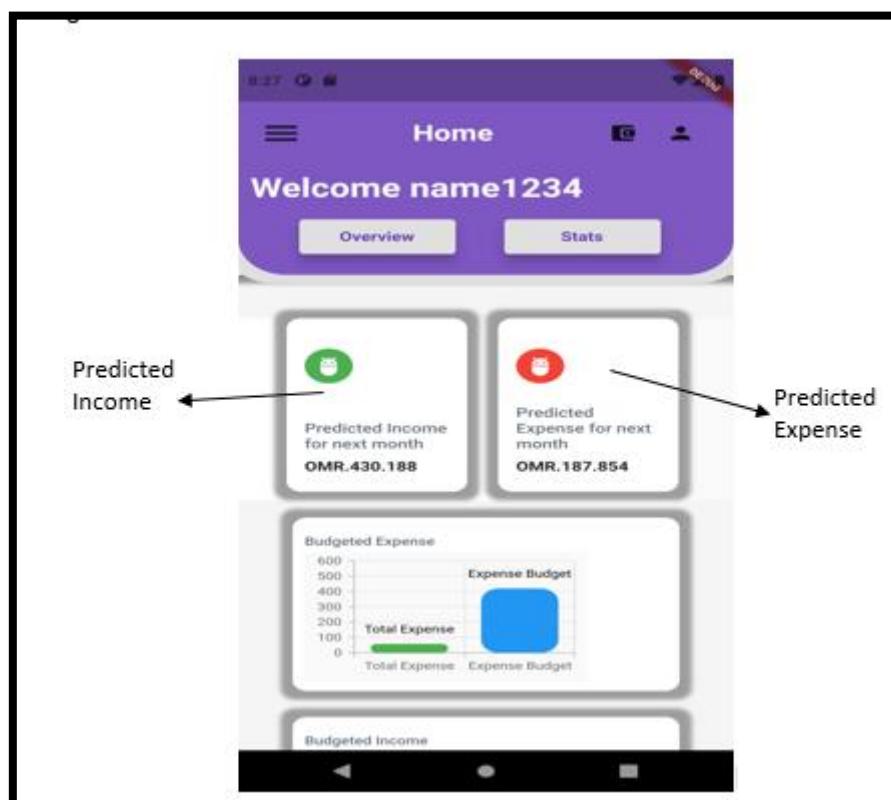


Figure 5.5 Machine learning prediction in home stats(self,2022)

Figure 5.6 shows how the income data was retrieved from the firestore database in firebase and then used to calculate the total sum of income for each month and their sum by executing the method get sum and adding it to an object list that would be passed to the ML function to estimate the user's forthcoming revenue.

```

Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey.withOpacity(0.10),
    body: FutureBuilder<QuerySnapshot>(
      future: FireStoreDataBase().getIncomeData(),
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return const Text(
            "Something went wrong",
          );
        } // Text
        if (snapshot.connectionState == ConnectionState.done) {
          data = snapshot.data as List;
          data.sort((a, b) => a["Incomedate"].compareTo(b["Incomedate"]));
          for (var i = 0; i < data.length; i++) {
            DateTime date = (data[i]["Incomedate"] as Timestamp).toDate();
            data[i]["Income"] = date;
          }
          List<List<Object>> incomedata = [];
          var ttl;
          for (var i = 1; i <= activeDay; i++) {
            ttl = getsum(i);
            incomedata.add([i, ttl]);
          }
          predictedvalue = ML(incomedata);
        }
        return const Center(child: CircularProgressIndicator());
      },
    ), // FutureBuilder
  ); // Scaffold
}

Widget getBody() {
  return Container(
    child: Column(
      children: [
        Text("Total Income:"),
        Text("Predicted Revenue:"),
      ],
    ),
  );
}

Future<List<Object>> getsum(int Income) {
  double sumvalue = 0.0;
  dataList = [];
  for (var i = 0; i < data.length; i++) {
    if (data[i]["Income"].month == Income) {
      dataList.add(data[i]);
      sumvalue += data[i]["incomeamt"];
    }
  }
  return Future.value(sumvalue);
}

ML(List<List<Object>> incomedata) {
  int row = incomedata.length + 1;
  var _data = incomedata..map((it) => [it, it])..removeLast();
  final data = [[x, y], ..._data];
  final samples = DataFrame(data, headerExists: true);
  final regressor = LinearRegressor(samples, 'y');
  var prediction = regressor.predict(DataFrame([[x, y], [row]]));
  var predictedvalue = prediction.rows.toString();
  return predictedvalue;
}

```

Figure 5.6 Income prediction code (self,2022)

Figure 5.7 shows how the income data was categorized and passed to the ML function to predict. The List is an object type list, and it is only this form of a list that can be used to translate data into vectors. The income data consists of two data points: the month in number, which is termed the 'x' or independent variable, and the 'y,' which is the dependent or variable to predict. The data is then turned into a data frame, and the Linear Regressor algorithm is trained and saved in the variable Regressor. The variable is then utilized to anticipate the following month's variable by providing a numerical value for the next month. The predicted value is returned to the widget built and kept in the variable predicted value, which will be used to display the data to the user.

```

ML(List<List<Object>> incomedata) {
    int row=incomedata.length+1;

    var _data = incomedata..map((it) => [it, it]) ;

    final data = [['x', 'y'], ..._data];

    final samples = DataFrame(data, headerExists: true);
    final regressor = LinearRegressor(samples, 'y');

    var prediction = regressor.predict(DataFrame([['x', 'y'], [row]]));
    var predictedvalue=prediction.rows.toString();

    return predictedvalue;
}

```

Figure 5.7 Machine learning algorithm (self,2022)

As illustrated in Figure 5.8, the predicted value is of the string type and is shown in a container in the body. This container is positioned on the Home stats page for the user to view from. This is the same method used to predict the user's upcoming expenses.

```

Widget getBody() {
  return Scaffold(
    body: Container(
      width: 150,
      height: 200,
      decoration: BoxDecoration(
        color: Colors.white,
        borderRadius: BorderRadius.circular(12),
        boxShadow: [
          BoxShadow(
            color: Colors.grey,
            spreadRadius: 10,
            blurRadius: 3,
            // changes position of shadow
          ), // BoxShadow
        ], // BoxDecoration
      ),
      child: Padding(
        padding: const EdgeInsets.only(
          left: 15, right: 15, top: 20, bottom: 20), // EdgeInsets
      ),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Container(
            width: 40,
            height: 70,
            decoration: BoxDecoration(
              shape: BoxShape.circle, color: Colors.green),
            child: Center(
              child: Icon(
                Icons. adb_outlined,
                color: Colors.white,
              ), // Icon, Center
            ), // Container
          ),
          Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
              Text(
                "Predicted Income for next month",
                style: TextStyle(
                  fontWeight: FontWeight.w500,
                  fontSize: 15,
                  color: Color(0xff67727d)), // TextStyle
              ), // Text
              SizedBox(
                height: 8,
              ), // SizedBox
              Text(
                "OMR."+predictedvalue.substring(2,9),
                style: TextStyle(
              ),
            ),
          ],
        ),
      ),
    ),
  );
}

```

Figure 5.8 Predicted value container code (self,2022)

Test Results :

As previously stated, no prior dataset was gathered to estimate user income. As a result, the machine learning will be trained and updated using user data classified by months. The prediction, or in this instance, the R-squared, will improve as the user's data grows. To put the machine learning linear regression and its findings to the test, a random dataset was produced based on the fact that a student's salary varies from month to month. Test data 1 in figure 5.9 has just 7 months of data and a low R-squared value, as shown in figure 5.11, but test data 2 as shown in figure 5.10 has 12 months of data and a higher R-squared value, as shown in figure 5.12. As a result, as the user data grows, so does the forecast of the user's impending monthly income and expenses.

```
'Months': [1, 2, 3, 4, 5, 6, 7],
'Income': [236, 528, 250, 526, 94, 230, 356]
```

Figure 5.9 Test data 1 (self,2022)

```
'Months': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
'Income': [236, 528, 250, 526, 94, 230, 356, 320, 258, 354, 190, 250]
```

Figure 5.10 Test Data 2 (self,2022)

```
PREDICTION: [[-215.34883721]]
Slope [[-32.81395349]]
Intercept [440.93023256]
MAE 118.6279069767442
r2 -4.253063580498519
```

Figure 5.11 Test 1 results (self,2022)

```
PREDICTION: [[151.]]
Slope [[-12.16666667]]
Intercept [394.33333333]
MAE 96.0
r2 -0.7144120544510111
```

Figure 5.12 Test 2 Results

5.3. Application comparison

The application's goal is to promote financial management among Oman's college students. Because there are no current financial management or budgeting applications in Oman, the application will be compared against financial management or budgeting applications from other countries. Table 5.1 compares the program features of the mint and Wally budgeting apps.

Features	Financial Management mobile application for college student in Oman	Mint (mint, n.d.)	Wally(Wally, n.d.)
Mobile application	✓	✓	✓
Applicable for students In Oman	✓		
Link application with the bank		✓	✓
Provide nearest student offers	✓		
Provide student discounts available	✓		
Provide financial article suggestions	✓		
Credit tracking	✓	✓	
Data analysis of the user financial data	✓	✓	✓
Predicting user data	✓		
Display annual insights		✓	✓

Table 5.1. Application Comparison Table (self,2022)

5.4. Summary

In summary, this section describes how data analysis and machine learning were used to assist the user in becoming more aware of their finances and making better financial decisions based on their own data. Numerous forms of data visualizations were created using data analysis and various other libraries available in flutter, resulting in a simple yet stunning summary of the user's progress. This allows the user to see their overall performance at a glance. In this part, machine learning was effectively utilized to estimate the users' future revenue and expense performance, which helps prepare the student ahead of time to control or manage their spending and income. This section demonstrates how past, present, and future data are leveraged to provide user with meaningful and easy insights . Additionally a comparison of the application with other budgeting applications are also made.

CHAPTER 6 CONCLUSION

College students are viewed as the youth who will represent and contribute to the country in the future, making it critical to prepare them in advance. Being financially literate and aware allows everyone to live a self-sufficient and stress-free life. As a result, it is crucial to prepare the youth financially. The application employs technology to assist students in becoming more financially aware and preparing them to make better financial decisions based on the information they have provided. The project aims to create a mobile application for college-aged young students in Oman that will track their spending habits and assist them in becoming more financially literate by offering insights into their purchasing habits and helping them save. Furthermore, the program encourages student savings by showing deals and student discounts in Oman and promoting financial literacy by suggesting financial articles for students to read and refresh their financial knowledge. Numerous study papers discovered and emphasized the necessity for financial planning among the youth and how the use of a mobile application may be used to reach the project aim. In addition, publications detailing machine learning in financial management to make predictions were reviewed. The pros and limitations of several software development lifecycles were investigated, and the agile development technique was chosen due to its flexibility and collaborative approach. In the Android Studio integrated development environment, the application is developed and implemented using the flutter framework based on the Dart programming language. Data analysis and machine learning show how past, present, and future data provide users with relevant and straightforward insights to make better financial decisions. Furthermore, machine learning tends to predict data given by the user , thus as the data grows the prediction accuracy also grows. Additionally, a comparison of the prototype and other existing budgeting applications are also made.

CHAPTER 7 FUTURE WORK

Based on the application's results, future work will concentrate on improving the application's performance, such as providing annual statistics and a summary with the monthly user summary. In addition, an API will be created to convert user bank statements into data for the Firebase database. Attempts will be made to immediately link the user bank with the program so that the user does not need to input data into the database on a regular basis. The machine learning method will be evaluated and improved further for improved performance and predictability. The application must also be tested among college students for at least a month in order to obtain proper feedback. The application in the future aims to make the youth more financially literate and aware in order to live financially independent early on.

GANTT CHART

PROJECT PLANNING GANTT CHART - (RM PHASE)														
KEY ACTIVITIES (TASKS)	DURATION 1 DIV = 1 WEEK													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1. Identifying and researching various topics for the project and communicating	✓	✓												
2. Completion and submission of project proposal			✓											
3. Coursework 1 Preparation				✓	✓	✓								
4. Coursework 1 Confirmation and Submission							✓							
5. Literature review and Predesign								✓	✓	✓	✓	✓		
6. Coursework 2 Confirmation and Submission												✓	✓	
PROJECT PLANNING GANTT CHART - (TP PHASE)														
KEY ACTIVITIES (TASKS)	DURATION 1 DIV = 1 WEEK													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14

REFERENCES

- Athey, S. (2018). *The Impact of Machine Learning on Economics*. [online] www.nber.org. Available at: <https://www.nber.org/books-and-chapters/economics-artificial-intelligence-agenda/impact-machine-learning-economics> [Accessed 16 Dec. 2021].
- Azer, I. and Mohamad, S.A. (2018). (PDF) *Exploring Financial Management Practices and Problems among Students*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/334496046_Exploring_Financial_Management_Practices_and_Problems_among_Students [Accessed 1 December 2021].
- B. Rubio, S. Romero, J. Troya, M. Diaz and E. Soler, "An Aspect Oriented Framework for Scientific Component Development," in *16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008)*, Lugano, Switzerland, 2005 pp. 290-296.doi: 10.1109/EMPDP.2005.11 Available at: <https://doi.ieeecomputersociety.org/10.1109/EMPDP.2005.11>[Accessed 7 Dec. 2021].
- BuildFire. (2016). *8 Steps to Understanding The Mobile App Development Lifecycle*. [online] Available at: <https://buildfire.com/understanding-mobile-app-development-lifecycle/> [Accessed 3 Dec. 2021].
- Comai-Legrand, L. (2019). *Should Teens Carry a Credit Card?* [online] firstalliancecu. Available at: <https://www.firstalliancecu.com/blog/should-teens-carry-a-credit-card> [Accessed 6 Dec. 2021].
- Council, E., Forbes Technology (2019). *Council Post: 12 Common Software Development Obstacles And How To Tackle Them.* [online] Forbes. Available at: <https://www.forbes.com/sites/forbestechcouncil/2019/06/05/12-common-software-development-obstacles-and-how-to-tackle-them/?sh=37808e195e01> [Accessed 3 Dec. 2021].
- Coursera (2022). *What Is Data Analysis? (with Examples)*. [online] Coursera. Available at: <https://www.coursera.org/articles/what-is-data-analysis-with-examples> [Accessed 17 May 2022].
- Deepanshi (2021). *Linear Regression | Introduction to Linear Regression for Data Science*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/#:~:text=In%20the%20most%20simple%20words> [Accessed 16 May 2022].
- Dickler, J. (2017). *Student loans take a mental toll on young people*. [online] CNBC. Available at: <https://www.cnbc.com/2017/10/17/student-loans-take-a-mental-toll-on-young-people.html> [Accessed 1 December 2021].

Elijah, M. (n.d.). *What is the main() function in Dart?* [online] Educative: Interactive Courses for Software Developers. Available at: <https://www.educative.io/edpresso/what-is-the-main-function-in-dart> [Accessed 5 May 2022].

Esplin, C. (2016). *What is Firebase?* [online] Medium. Available at: <https://howtofirebase.com/what-is-firebase-fcb8614ba442> [Accessed 18 Dec. 2021].

ExisTek (2018). *SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral | Existek Blog.* [online] Existek. Available at: <https://existek.com/blog/sdlc-models/> [Accessed 1 May 2022].

Falahati, L., Paim, L., Ismail, M., Haron, S. and Masud, J. (2011). Assessment of university students' financial management skills and educational needs. *African Journal of Business Management*, [online] 5(15), pp.6085–6091. Available at: <https://academicjournals.org/journal/AJBM/article-full-text-pdf/C1DC0DA14456.pdf> [Accessed 1 December 2021].

Goetz, J., Cude, B. J., Nielsen, R. B., Chatterjee, S., & Mimura, Y. (2011). *College-based personal finance education: Student interest in three delivery methods.* *Journal of Financial Counseling and Planning*, Available at: https://www.researchgate.net/publication/256019479_College-Based_Personal_Finance_Education_Student_Interest_in_Three_Delivery_Methods. [Accessed 5 Dec. 2021].

hackolade (n.d.). *Firebase.* [online] hackolade. Available at: <https://hackolade.com/help/Firebase.html#:~:text=The%20Google%20Firebase%20Realtime%20Database> [Accessed 8 May 2022].

Half, R. (2019). *6 basic SDLC methodologies: Which one is best?* | Robert Half. [online] Roberthalf. Available at: <https://www.roberthalf.com.au/blog/employers/6-basic-sdlc-methodologies-which-one-best> [Accessed 1 May 2022].

Hamilton, T. (2019). *What is Software Testing? Introduction, Definition, Basics & Types.* [online] Guru99.com. Available at: <https://www.guru99.com/software-testing-introduction-importance.html> [Accessed 9 May 2022].

Hayes, J. (2014). *User Interface Design for Online Social Media.* [online] California,USA: California Polytechnic State University. Available at: <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1110&context=grcsp> [Accessed 7 Dec. 2021].

Hebb, N. (2019). *What is a Flow Chart?* | BreezeTree. [online] Breezetreecom. Available at: <https://www.breezetreecom/articles/what-is-a-flow-chart> [Accessed 18 Dec. 2021].

Hollis, P. (2014). *A Degree in Debt* - ProQuest. [online] www.proquest.com. Available at: <https://www.proquest.com/openview/2dbab41d836d8af74dec37e3eed3d9b/1.pdf?pq-origsite=gscholar&cbl=41153> [Accessed 6 Dec. 2021].

Indeed Editorial Team (2021). *What is a User Interface? Definition, Types and User Interface Example.* [online] Indeed Career Guide. Available at: <https://www.indeed.com/career-advice/career-development/user-interface> [Accessed 4 May 2022].

Intergy (2019). *What's the difference between agile development and prototyping?* [online] Intergy. Available at: <https://www.intergy.com.au/agile-development-and-prototyping/> [Accessed 1 May 2022].

Iontuition (2016). *The state of college financial literacy programs [infographic]* | Financial literacy, Literacy programs, Finance infographic. [online] Pinterest. Available at: <https://www.pinterest.com/pin/288511919859939587/> [Accessed 1 December 2021].

javatpoint (n.d.). *Database Design - Javatpoint.* [online] javatpoint. Available at: <https://www.javatpoint.com/database-design> [Accessed 2 May 2022].

Jevtic, G. (2019). *What is SDLC? How the Software Development Life Cycle Works.* [online] PhoenixNAP Global IT Services. Available at: <https://phoenixnap.com/blog/software-development-life-cycle> [Accessed 1 May 2022].

Khan, M. (2020). *Which SDLC Model Is The Best For Your Business?* [online] RNF Technologies. Available at: <https://www.rnftechnologies.com/blog/best-sdlc-methodology> [Accessed 1 May 2022].

Kim, J. and Chatterjee, S. (2013). *Childhood Financial Socialization and Young Adults' Financial Management.* [online] Association for Financial Counseling and Planning Education. Available at: https://www.afcpe.org/wp-content/uploads/2018/10/v24_1_61-79.pdf [Accessed 5 Dec. 2021].

Koposko, J. L., Hershey, D. A., Bojórquez, M. I., & Pérez, A. E. (2016). *College student attitudes toward retirement planning: The case of Mexico and the United States.* *Journal of Personal Finance*, 15(2), 52–67. Available at: <https://web.p.ebscohost.com/abstract?direct=true&profile=ehost&scope=site&authtype=crawler&jrnln=15406717&AN=118018151&h=9kVzrvrg25LxA7HYY6Sv%2bCI5D3LGEBcR%2bI%2fwTggRkwZE3bxI9CMrswsXKGKC%2f%2f7DNcfj6gwmUyNNNgkqy%2f7wF2Q%3d%3d&crl=c&resultNs=AdminWebAuth&resultLocal=ErCrINotAuth&crlhashurl=login.aspx%3fdirect%3dtrue%26profile%3dehost%26scope%3dsite%26authtype%3dcrawler%26jrnln%3d15406717%26AN%3d118018151>. [Accessed 5 Dec. 2021].

Kumar, S. (2021). *Predictive, regression, and model building for driver-based Budgeting.* [online] Researchgate.net. Available at: https://www.researchgate.net/profile/Santhosh-Kumar-155/publication/354683675_Predictive_regression_and_model_building_for_driver-

based_Budgeting/links/61464224a595d06017d788ab/Predictive-regression-and-model-building-for-driver-based-Budgeting.pdf.[Accessed 5 Dec. 2021].

L. Archuleta, K., Dale, A. and M. Spann, S. (2013). *College Students and Financial Distress: Exploring Debt, Financial Satisfaction, and Financial Anxiety.* [online] EJ1043230. Available at: <https://files.eric.ed.gov/fulltext/EJ1043230.pdf> [Accessed 5 Dec. 2021].

Lusardi, A. and Mitchell, O.S. (2014). The Economic Importance of Financial Literacy: Theory and Evidence. *Journal of Economic Literature*, 52(1). [Accessed 6 Dec. 2021].

Malvik, C. (2020). *What Is User Interface Design? An Introduction for Aspiring Designers.* [online] Rasmussen University. Available at: <https://www.rasmussen.edu/degrees/design/blog/what-is-ui-design/> [Accessed 3 May 2022].

mint (n.d.). *Budget Tracker & Planner | Free Online Money Management | Mint.* [online] mint.intuit.com. Available at: <https://mint.intuit.com/> [Accessed 21 May 2022].

Morris, J. (2021). *Why NoSQL JSON Databases Are So Useful.* [online] The Couchbase Blog. Available at: <https://blog.couchbase.com/json-database/#:~:text=A%20JSON%20database%20is%20arguably> [Accessed 8 May 2022].

Mullainathan, S. and Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives*, [online] 31(2), pp.87–106. Available at: <https://www.aeaweb.org/articles?id=10.1257/jep.31.2.87> [Accessed 16 Dec. 2021].

Netinant, P. and Elrad, T. (2016). Separation of concerns in designing mobile software. *Rangsit Journal of Arts and Sciences*, [online] 6(1), pp.89–96. Available at: https://jcst.rsu.ac.th/files/issues/V6N1/2016_6_1_full_160601_20160824_1329.pdf [Accessed 7 Dec. 2021].

Oesterreich, T.D., Teuteberg, F., Bensberg, F. and Buscher, G. (2019). The controlling profession in the digital age: Understanding the impact of digitisation on the controller's job roles, skills and competences. *International Journal of Accounting Information Systems*, 35, p.100432. [Accessed 16 Dec. 2021].

Ogunsola, S. (2021). *Introduction to Uizard as a Rapid Prototyping Tool.* [online] Medium. Available at: <https://blog.prototypr.io/introduction-to-uizard-as-a-rapid-prototyping-tool-ab3b6bb8729e> [Accessed 18 Dec. 2021].

Palchowdhuri, Y. and Roy, P.S. (2018). Driver based statistical model for simulating land use land cover change in Indus river basin India. *Remote Sensing of Land*, [online] 2(1), pp.15–30. Available at: https://www.researchgate.net/publication/327275436_Driver_based_statistical_model_for_simulating_land_use_and_cover_change_in_Indus_river_basin_India [Accessed 4 Jan. 2020].

Palmer, L., Bliss, D., Goetz, J.W. and Moorman, D. (2010). *Improving Financial Awareness Among College Students: Assessment of a Financial Management Project*. [online] papers.ssrn.com. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2695748 [Accessed 1 December 2021].

Pearl, J. (2019). The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, [online] 62(3), pp.54–60. Available at: https://ftp.cs.ucla.edu/pub/stat_ser/r481.pdf [Accessed 16 Dec. 2021].

planning, F.B.F.L.S.E. is an expert in financial, Software, P.F., taxes, Researching, W.E., Wisconsin, teaching savings strategies for over 20 years S. earned her bachelor's in business administration from the U. of, Coursework, H.S.C.A., Administration, C. in P., Networking, C., Accounting, S.B. and Elmlad, small business management R.T.B. editorial policies S. (n.d.). *Attention College Students: Manage Your Money With These Tips*. [online] The Balance. Available at: <https://www.thebalance.com/how-to-manage-money-in-college-with-a-budget-1294240> [Accessed 1 December 2021].

PyCharm Help. (n.d.). Get started - Help | PyCharm. [online] Available at: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html> [Accessed 18 Dec. 2021].

Roos, A., Tucker, J., Rodt, M., Stange, S., Ego, P., Boudadi Hardik Sheth, A. and Sheth, H. (2020). *Lessons from Best-in-Class CFOs*. [online] BCG Global. Available at: <https://www.bcg.com/publications/2020/lessons-best-in-class-cfos> [Accessed 16 Dec. 2021].

Royal Literary Fund (2014). *What Is a Literature review?* [online] The Royal Literary Fund. Available at: <https://www.rlf.org.uk/resources/what-is-a-literature-review/> [Accessed 18 Dec. 2021].

Royal Literary Fund (2014). *What Is a Literature review?* [online] The Royal Literary Fund. Available at: <https://www.rlf.org.uk/resources/what-is-a-literature-review/> [Accessed 18 Dec. 2021].

Rukhiran, M. and Netinant, P. (2020). A PRACTICAL MODEL FROM MULTIDIMENSIONAL LAYERING: PERSONAL FINANCE INFORMATION FRAMEWORK USING MOBILE SOFTWARE INTERFACE OPERATIONS. *Journal of Information and Communication Technology*, [online] 19(3), pp.321–349. Available at: <http://www.e-journal.uum.edu.my/index.php/jict/article/view/jict2020.19.3.2/2271> [Accessed 6 Dec. 2021].

Ruth, A. (2015). *Robert Kiyosaki - It Isn't About How Much You Make*. [online] Due. Available at: <https://due.com/blog/robert-kiyosaki-it-isnt-about-how-much-you-make/> [Accessed 2 Dec. 2021].

Sajna, K. (2022). *The 7 Stages of the Software Development Life Cycle*. [online] Codilime. Available at: <https://codilime.com/blog/the-stages-of-the-sdlc/> [Accessed 1 May 2022].

Slava Vaniukov (2020). *Colors in UI Design: A Guide for Creating the Perfect UI - Usability Geek.* [online] Usability Geek. Available at: <https://usabilitygeek.com/colors-in-ui-design-a-guide-for-creating-the-perfect-ui/>. [Accessed 17 Dec 2021].

Solution Analysts. (2020). *10 Finance App Ideas For Startups to Consider in 2021.* [online] Available at: <https://www.solutionanalysts.com/blog/finance-app-ideas/>. [Accessed 10 Dec 2021].

Stewart, M. (2019). *The Limitations of Machine Learning.* [online] Medium. Available at: <https://towardsdatascience.com/the-limitations-of-machine-learning-a00e0c3040c6> [Accessed 3 Dec. 2021].

Styles, M. (2018). *A Financial Epidemic: How Financial Literacy Affects College Students' Financial Management Practices and the Debt Crisis in America* Mikala Styles. [online] East Tennessee State University. East Tennessee State University. Available at: <https://dc.etsu.edu/cgi/viewcontent.cgi?article=1509&context=honors#:~:text=The%20Problem&text=Some%20common%2C%20yet%20effective%2C%20financial>. [Accessed 1 December 2021].

Super Color Digital (2020). *The Meaning of the Color Blue.* [online] Super Color Digital. Available at: <https://www.supercolor.com/blog/the-meaning-of-the-color-blue/#:~:text=Dark%20blue%20represents%20knowledge%2C%20power%2C%20integrity%2C%20and%20seriousness>. [Accessed 4 May 2022].

Syakirah, N. (n.d.). The importance of financial management for students. www.academia.edu. [online] Available at: https://www.academia.edu/33991046/The_importance_of_financial_management_for_students [Accessed 1 December 2021].

Ta, K. (2020). *Traditional vs Agile SDLC: How To Skyrocket Your Project With Agile Model.* [online] AgileTech Vietnam. Available at: <https://agiletech.vn/traditional-sdlc-vs-agile-sdlc/> [Accessed 1 May 2022].

Taddy, M. (2019). *Business data science : combining machine learning and economics to optimize, automate, and accelerate business decisions.* [online] New York: McGraw-Hill Education. Available at: <https://www.oreilly.com/library/view/business-data-science/9781260452785/> [Accessed 16 Dec. 2021].

TAS News Service (2021). *Bank Muscat's Shahadati finance supports Omani youth in fulfilling their career dreams.* [online] The Arabian Stories News. Available at: <https://www.thearabianstories.com/2021/09/04/bank-muscats-shahadati-finance-supports-omani-youth-in-fulfilling-their-career-dreams/> [Accessed 2 Dec. 2021].

TechTarget Contributor (2019). *What is Android Studio? - Definition from WhatIs.com.* [online] SearchMobileComputing. Available at: <https://searchmobilecomputing.techtarget.com/definition/Android-Studio> [Accessed 18 Dec. 2021].

Thakur, D. (2013). *What is Software Requirement? Types of Requirements*. [online] Computer Notes. Available at: <https://ecomputernotes.com/software-engineering/softwarerequirement> [Accessed 18 Dec. 2021].

The Standish Group (2020). *CHAOS Report 2020*. [online] The Standish Group. Available at: <https://www.standishgroup.com/news/45> [Accessed 1 May 2022].

Thomas, G. (2019). *What is Flutter and Why You Should Learn it in 2020*. [online] freeCodeCamp.org. Available at: <https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/> [Accessed 1 May 2022].

Thorpe, J. (2015). *What 4 Common Colors Do To Your Brain*. [online] Bustle. Available at: <https://www.bustle.com/articles/123327-what-4-common-colors-do-to-your-brain> [Accessed 3 May 2022].

Today's Eggspert (2017). *6 Colors That Are Proven to Boost Sales*. [online] The Daily Egg. Available at: <https://www.crazyegg.com/blog/colors-proven-to-boost-sales/> [Accessed 4 May 2022].

Trunkett, O. (2020). *SDLC Methodologies: From Waterfall to Agile | Virtasant*. [online] virtasant. Available at: <https://www.virtasant.com/blog/sdlc-methodologies> [Accessed 1 May 2022].

United Nations (2015). *The 17 Goals*. [online] United Nations. Available at: <https://sdgs.un.org/goals> [Accessed 2 Dec. 2021].

Usability.gov (2014). *User Interface Design Basics*. [online] Usability.gov. Available at: [https://www.usability.gov/what-and-why/user-interface-design.html#:~:text=User%20Interface%20\(UI\)%20Design%20focuses](https://www.usability.gov/what-and-why/user-interface-design.html#:~:text=User%20Interface%20(UI)%20Design%20focuses) [Accessed 3 May 2022].

Vorobiova, A. (2021). *Software Development Life Cycle Methodologies – NIX Approach*. [online] NIX United. Available at: <https://nix-united.com/blog/software-development-life-cycle-nix-approach-to-sdlc/> [Accessed 1 May 2022].

Wally (n.d.). *Home*. [online] Wally 3.0. Available at: <https://www.wally.me/> [Accessed 21 May 2022].

Wasserbacher, H. and Spindler, M. (2021). Machine learning for financial forecasting, planning and analysis: recent developments and pitfalls. *Digital Finance*. [online] Available at: <https://arxiv.org/pdf/2107.04851.pdf> [Accessed 16 Dec. 2021].

Widener, K. (2017). *Financial Management Issues of College-Aged Students: Influences and Consequences*. [online] FireScholars. Southeastern University - Lakeland: Southeastern University FireScholars. Available at: <https://firescholars.seu.edu/cgi/viewcontent.cgi?article=1063&context=honors> [Accessed 1 December 2021].

Williams, A. (2008). *Essentials of Financial Planning 1* Running head: ESSENTIALS OF FINANCIAL PLANNING *The Essentials of Financial Planning for College Graduates*. [online] *The Essentials of Financial Planning for College Graduates*. Virginia, USA: Liberty University Digital Commons. Available at: <https://core.ac.uk/download/pdf/58820809.pdf> [Accessed 2 Dec. 2021].

Picture References:

Loginimage.png: IconScout (n.d.). *Product Development 3D Illustrations Designs, Images, Vectors, HD Graphics*. [online] IconScout. Available at: <https://iconscout.com/3ds/product-development> [Accessed 21 May 2022].

Loginimage6.png: PPM Works (n.d.). *Project and Planner Sync*. [online] PPM Works, Inc. Available at: <https://ppmworks.com/project-and-planner-sync/> [Accessed 21 May 2022].

Signupimages.jpg: cutewallpaper (n.d.). *Create Clipart posted by Sarah Sellers*. [online] cutewallpaper.org. Available at: <https://cutewallpaper.org/24/create-clipart/view-page-24.html> [Accessed 21 May 2022].

