



Software Requirements Specifications

For

BUDGET TRACKER

Prepared By

Name:

Niyati Joshi

Urmi Vyas

Dipti Joliya

Insiya Maldeviwala

EnrollmentNo.:

22SOECA21023

22SOECA21057

22SOECA21022

22SOECA21034

BUDGET TRACKER

Table of Contents

1. Introduction	3
1.1 Purpose	3
1.2 Document Conventions	3
1.3 Intended Audience and Reading Suggestions	4
1.4 Project Scope	4
2. Overall Description	5
2.1 Product Perspective	5
2.2 Product Features	5
2.3 User Characteristics	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	7
2.6 Assumptions and Dependencies	7
3. System Features	8
4. External Interface Requirements	9
4.1 User Interfaces	9
4.2 Hardware Interfaces	10
4.3 Software Interfaces	10
5. Other Nonfunctional Requirements	11
5.1 Performance Requirements	11
5.2 Safety Requirements	12
5.3 Security Requirements	12
5.4 Software Quality Attributes	13
6. Other Requirements	14
Appendix A: Glossary	14
Appendix B: Analysis Models	14
6.1 ER Diagram	15
6.2 Flow Chart	16
Appendix C: Future Enhancements	17
➤ Wireframe	18
➤ Layouts	19
7. Reference	27

1. Introduction

1.1 Purpose & Introduction

- This app is used for daily accounting and expense managing.
- The main purpose of this app is to provide overview of user's spending habits and financial situation, allowing them to make better financial decisions.
- In this money managing app user role is to utilize the app to track and manage your personal finances.
- It allow users to easily track their expenses, providing a clear picture of where their money is going.
- This app is easy to use and you can easily access this app anywhere.

1.2 Document Conventions

- The main heading for software requirements specification, application name, version number, prepared by, organization and date are written in right handed alignment.
- Apart from this, the entire document for software requirements

Convention for Heading:

- Font face: Times New Roman
- Font style: Bold
- Font size:28

Convention for Main title:

- Font face: Times New Roman
- Font style: Bold
- Font size:22

Convention for Sub title:

- Font face: Times New Roman
- Font style: Bold
- Font size:18

Convention for Body:

- Font face: Times New Roman
- Font style: Regular
- Font Size: 12

1.3 Intended Audience and Reading Suggestions

- The document is intended for all the customers and the developers – designers, coders, testers and maintainers. The reader is assumed to have basic knowledge of Mobile OS, databases and user accounting along with knowledge and understanding of a various diagrams such as ER diagrams . User must understand the flow of the applications and various redirections too.

1.4 Project Scope

- The objective of the budget tracker app is to help users manage their expenses by storing a our day to day use and to keep record.
- It provides a platform to track their income and expenses.
- It is useful for Housewives, Students, Hostel Students etc.
- It has an user-friendly interface that is easy to navigate and understand.

2. Overall Description

2.1 Product Perspective

- A product perspective for a budget tracker project would focus on designing a software solution that provides users with a comprehensive platform for managing their personal or day to day expenses. The budget tracker software should provide users with a range of features to manage their expenses effectively. This could include adding an expense or incomes, categorizing expenses, generating reports.
- The user interface should be easy to use, intuitive, and provide users with a clear view of their financial situation. It should also be visually appealing and easy to navigate.
- Overall, a product perspective for a budget tracker app should focus on providing users with a comprehensive, easy-to-use platform for managing their finances. By incorporating features such as user-friendly interfaces, robust functionality, secure integration with financial services, the budget tracker app can provide users with a reliable tool for managing their financial goals effectively.

2.2 Product Features

- **Budget creation:** The first step in budget tracker is creating a budget plan. This involves estimating income and expenses over a given period, usually monthly or yearly.
- **Expense tracking:** After creating a budget plan, it's important to track expenses to ensure that you are staying within your budget. This can be done manually or through the use of budgeting software or apps.
- **Goal setting:** Budget management can help you set financial goals, such as saving for a down payment on a home or paying off debt.
- **Forecasting:** By analyzing past spending patterns and current trends, you can manage your budget.
- **Flexibility:** A good budget management plan should be flexible and adaptable to changing circumstances, such as unexpected expenses or changes in income.
- **Regular review:** To ensure that your budget is effective, it's important to review it regularly and make adjustments as needed.
- **Expense categories:** Users should be able to categorize their expenses by type (e.g. food, transportation, rent) to gain insight into their spending habits.

2.3 User Characteristic:

- Users can access every feature of the application and even by performing login or registration activities. User would be responsible for entering your income and expenses into the app.
- User may need to categorize your transactions, so the app can track where your money is going. For example, you might categorize a purchase as "groceries," "entertainment," or "transportation".
- User would need to regularly review the reports generated by the app to see how you're doing. This could include date wise report.
- Based on the reports, you may need to adjust your budget and spending habits. This could involve cutting back on certain expenses, finding ways to increase your income.

2.4 Operating Environment

An operating environment or integrated applications environment is the environment in which users run their application. The environment consists of a user interface provided by an applications manager.

The operating environment for Budget Tracker is as follows:

- Client Side : Android OS
- Operating system: Windows
- Database: Firebase database
- Platform: Android studio
- Preferred language: Java

2.5 Design and Implementation Constraints

System or implementation constraints describe how the product operates inside various circumstances and limit the options designers have if building the product/application. This section specifies design constraints imposed by other standards, hardware limitations, communication interface limitations, etc. It also includes the constraints regarding the databases. Here in Budget Tracker we are using the firebase database to maintain and store the data of our users. So here we are facing constraints such as:

- The size of data downloaded from the database at a single location should be less than 256MB for each read operation.
- It should allow users to input and track their income and expenses quickly and easily, and generate visual reports to help them analyze their spending patterns.
- Due to the online community of the firebase, no offline entries can be done. Network here, can be considered as a constraints.

2.6 Assumptions and Dependencies

Assumptions:

- Users have access to a smartphone with an internet connection.
- Users have basic knowledge of how to use a smartphone and navigate apps.
- Users have a need or desire to track their personal finances.
- Users will input accurate and up-to-date information regarding their income and expenses.

Dependencies:

- The app relies on accurate and timely data entry by the user to function correctly.
- The app will need to have a user-friendly interface that is easy to navigate and understand.
- The app's effectiveness is influenced by the user's ability to interpret and act on the information provided by the app.

3. System Features

The entire system features consists of functional requirements as well as user-interfaces .By understanding these requirements, entire system features can be understood easily. Talking about Budget Tracker Application, the functional requirements consists of the following:

- The user shall be able to register to the system.
- Users shall have two options on the main screen i.e the dashboard and they are Add Expense and Add Income.

Talking about the user-interfaces which include the information about front-end, back-end as well as database, the Budget Tracker Application have the following in context of this:

- Front end: It is usually referred to as the application's "client side." The frontend consists of everything that the user sees when interacting with the application, such as text colors and styles, photos, graphs and tables, buttons, colors, the navigation menu, and much more.
- Back end: It is generally the logical part or say the coding criteria which is determined on the basis of the functionalities of the front-end. While talking about android studio, we can use java as well as kotlin language for coding purposes but widely java is used. In this application also, java language is preferred for the back-end.
- Database: While talking about the Budget Tracker Application, we have used firebase database connectivity for the purpose of storing the data of the user. The Firebase Realtime Database is a cloud-hosted NonSQL database that lets you store and sync data between your users in real time.

4. External Interface Requirements

4.1 User Interfaces

- A user interface for a budget tracker app is designed with simplicity, ease of use, and user engagement in mind, giving them convenience while they are using it. The application makes sure at every point that the user spends most of the time using the device rather than figuring out how to use it.
- A dashboard is the first screen users see upon logging in. This screen should provide an overview of their finances, including account balances ,total expense and income.
- The user interface has easy-to-use navigation that allows users to move between screens quickly and easily.
- The user can select any of the one tab from the dashboard, and is taken to the respective screen. Every screen displays the back button too if the user wants to get back to the previous page. The user can click on any one of the options and is taken to the screen of their choice.
- If the user does not know how to use any functionality or has any queries,then software requirements specification documentation can be referred .
- While talking about our application, in designing the xml file, we used various layouts to make it more attractive and convenient. Mostly constraint layout and linear layout of android studio are in use for the purpose of front-end that is designing.

4.2 Hardware Interfaces

- The hardware interface for a budget tracker Android app will include the following considerations:
- Screen size and resolution: The app is designed to work on different screen sizes and resolutions of Android devices.
- Touchscreen interface: The app is designed to work with touchscreens for easy navigation and interaction.
- Data storage: The app should store data securely on the device's internal storage or external memory card.
- Input Data :The app has the ability to input data from the users.
- Redirects : The app has the ability to redirect the users to other links.
- User Friendly : The app is user friendly. Any group of age can understand the functionality.

4.3 Software Interfaces

Software interfaces (programming interfaces) are the languages, codes and messages that programs use to communicate with each other and to the hardware. budget trackey Nest too have the software interfaces just like all other applications Following are the software interfaces used for the budget tracker application:

- Android SDK: The Android Software Development Kit (SDK) provides tools and APIs to build Android apps, including libraries for UI development, database management, and networking.
- Android Studio: This platform is chosen for the development of our application, it is an IDE for android app development that provides features such as code editor, debugging tools, and emulator.
- Database: Firebase database is used in the budget trackeNest application to save the record of the users. It is most commonly used as well as a convenient database to use. also it is the most preferable database compared to SQLite.
- Operating System: We have chosen Windows operating system for its best support and user-friendliness.
- Java: To implement the project we have chosen the Java language for its more interactive support.

5. Other Non-functional Requirements

5.1 Performance Requirements

Performance requirements define how well the application accomplishes certain functions under specific conditions. The performance requirements of our Budget Tracker include the following :-

- **Speed:** The app should load and process data quickly, with minimal waiting times for users. This can be measured by metrics such as network latency, data transfer speeds, and app load times.
- **Memory usage:** The app should use minimal device memory to ensure optimal performance and reduce the risk of crashes or errors. This can be achieved by optimizing data caching, reducing data duplication, and avoiding memory leaks.
- **Scalability:** The app should be designed to handle large amounts of data and users, without sacrificing performance or usability. This can be achieved by optimizing data storage and retrieval, and implementing efficient data structures and algorithms.
- **Usability:** The usability refers to the usage of the application. Usability of application depends on the ease of accessibility. Accessibility again, should be considered as an important aspect of the performance requirement. Application should be easily accessible as well as it must be easy to use also.
- **Flexibility:** Flexibility is the ease with which the application can respond to uncertainty in a manner to sustain or increase its value delivery. Uncertainty is a key element in the definition of flexibility. Uncertainty can create both risks and opportunities in a system, and it is with the existence of uncertainty that flexibility becomes valuable. The application should be flexible enough to deal with the uncertainties.
- **Portability:** Our application can be used on any android phones and tablets.

5.2 Safety Requirements

- Safety requirements specifications are specifications that describe every required safety function that must be performed by a safety instrumented system.
- Talking about safety aspects, the user should be well aware about their password's which they set during the registration process. This is the way they can go for safety measures or safety requirements.
- In addition to it, safety requirements also include that no other person than you can use this application as it is registered by your name and password is only know to you.
- So these safety requirements are assumed to be handled by the user. And users are also advised to take care that no one by knowingly or unknowingly access the features and misuse it which then creates a problem to the user.

5.3 Security Requirement

- A security requirement is a goal set out for an application at its inception. Every application fits a need or a requirement. Security vulnerabilities allow software to be abused in ways that the developers never intended.
- Requirements related to access control, data integrity, authentication, and wrong password lockouts fall under security requirements.
- If the user is logged out due to some reasons and now wishes to log in or sign in again, then they must have to do the proper authentication. Passwords which they set during registration for the very first time, should match now, as this authentication information of the user is recorded in admin's database which is firebase database.
- In other words, again using a firebase database we are looking forward to the best security aspect as it is the most preferable and capable one to store the data of the user.
- Security principals also include Confidentiality, integrity and availability. Users should preserve the access control and disclosure restrictions on information . Moreover users should avoid the improper (unauthorized) information modification or destruction.

5.4 Software Quality Attributes

Software Quality Attributes are features that facilitate the measurement of performance of a software product by Software Testing professionals, and include attributes such as availability, interoperability, correctness, reliability, learnability, maintainability, readability, extensibility, testability, efficiency, and portability. High scores in Software Quality Attributes enable software architects to guarantee that a software application will perform as the specifications provided by the client. Software quality attributes for this application are as follows:

- **Performance:** The app should be fast and responsive, with minimal waiting times or delays. This can be achieved through optimization of app resources, such as network and memory usage, and the use of efficient algorithms and data structures.
- **Portability:** Our application can be used on any android phones and tablets.
- **Maintainability:** Maintainability is defined as the probability that a failed component of application will be restored or repaired to a specified condition within a specified period or time when maintenance is performed in accordance with prescribed procedures. So the application remains maintained.
- **Reliability:** The app should be reliable and trustworthy, with a low rate of errors or crashes. This can be achieved through testing and quality assurance practices, such as automated testing, error logging and reporting, and monitoring and analysis of app usage and performance.
- **Reusability:** An application that can be adapted for different customers without changing the source code of the system. Application systems have generic features and so can be used/reused in different environments.
- **Readability:** Readability is a measure of how easy a piece of text is to read. The level of complexity of the text, its familiarity, legibility and typography all feed into how readable your text is .The readability of our application is simple and sobar. Users can easily understand without any difficulty .
- **Extensibility:** Extensibility is a measure of the ability to extend a system. Our application shows high extensibility as many new features are yet to be introduced and much modification might take place soon in the application.

6. Other Requirements

Other requirements of Budget Tracker include the legal requirements which means the permissions from the users to access some or the other thing in their device. These are the requirements which are taken into consideration at the very first time when the application is installed and ready to use. In this application permissions such as “Allow application to access your location? ”, “Allow application to make calls? ”, “Allow application to do messages? ” is seen

In addition to this, with regards to the Firebase database, There is no need for an application server to access the Firebase Real-time database. We can access it directly from a mobile device or web browser. Data validation and security are available through the Firebase Real-time Database Security Rules, expression-based rules executed when data is read or written.

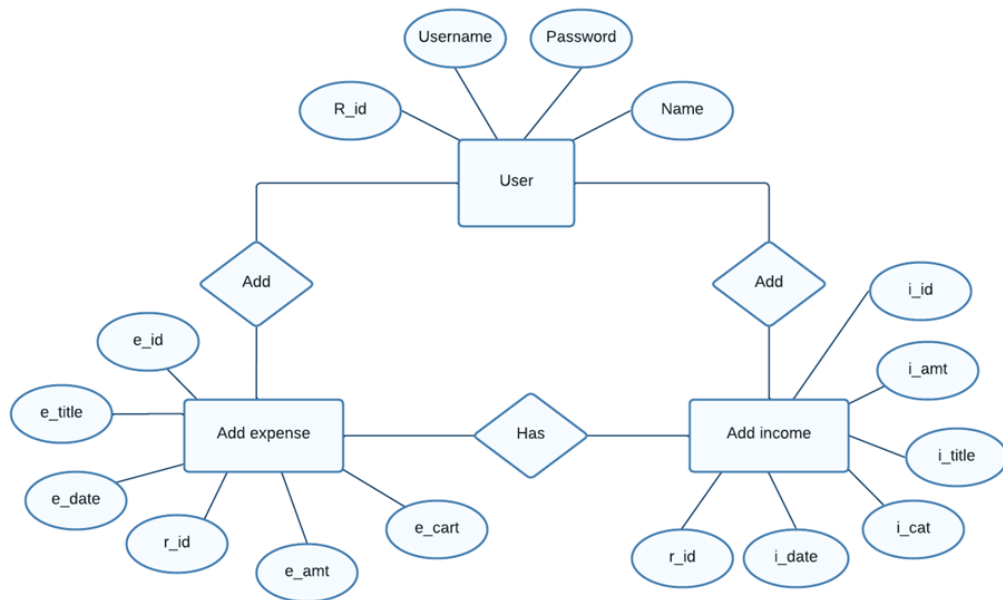
Appendix A: Glossary

- UI - User interface
- DFD- Data Flow Diagram
- ER- Entity Relationship
- SWOT – Strength , Weakness, Opportunity and Threat
- OS- Operating System

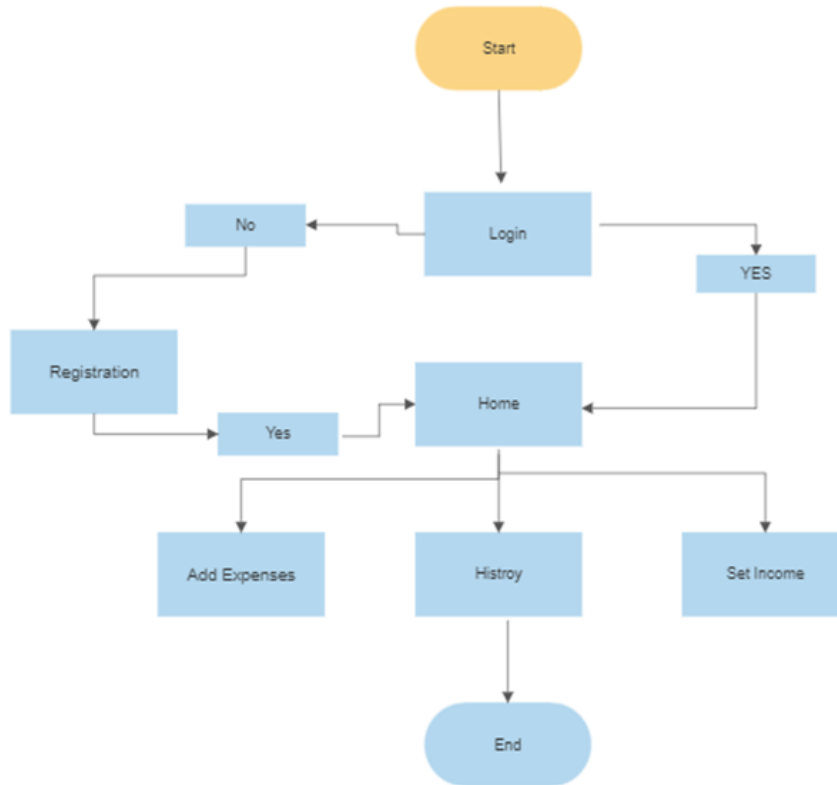
Appendix B: Analysis Models

This section includes the analysis models for our application. This analysis model includes UML diagrams such as Flow Chart and ER Diagram.

6.1 ER Diagram:



6.2 Flow chart:

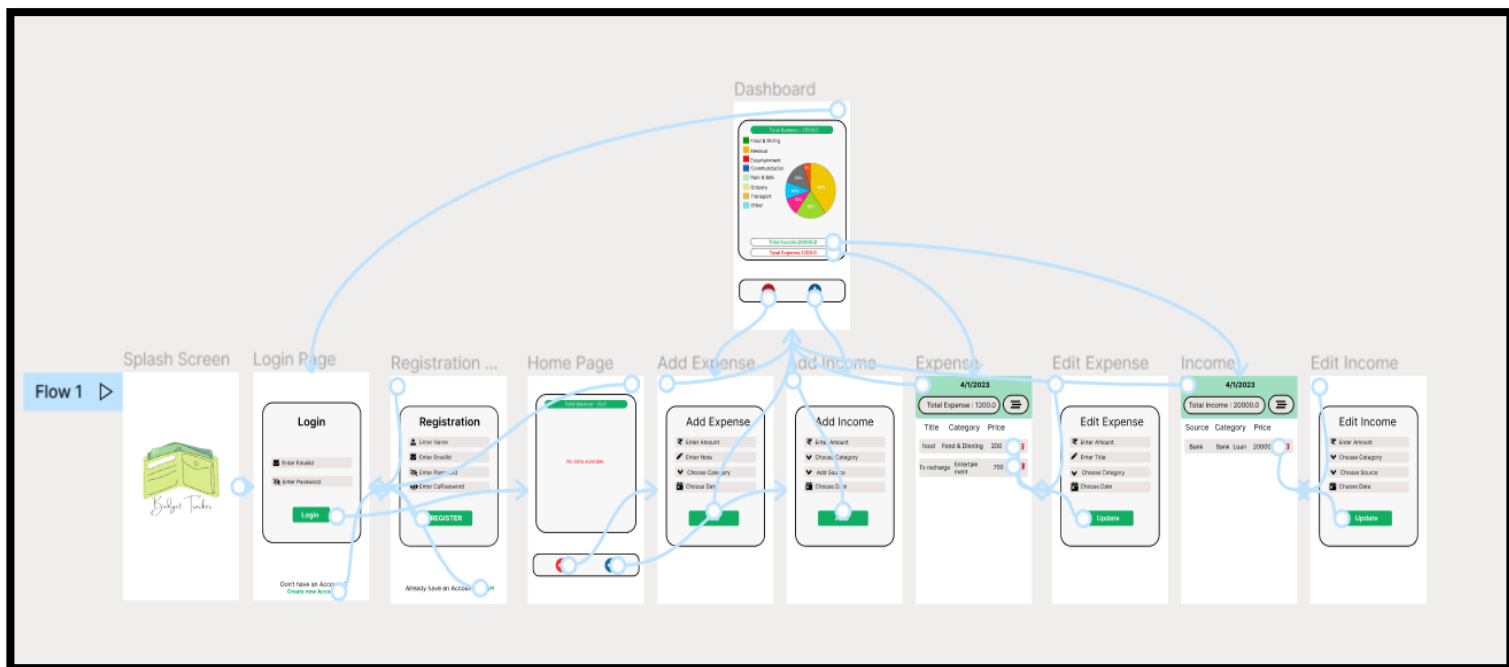


Appendix C: Future Enhancements

There are several potential future enhancements for a budget tracker app that can improve the user experience and provide additional value. Here are a few points:

- **Real-time Financial Insights:** Providing real-time financial insights and alerts to users, such as spending habits, income streams, and investment opportunities.
- **Customization:** Allowing users to customize the app's interface, layout, and features based on their unique financial needs and preferences.
- **Photo Uploading :** User can upload photo of their bill and can be usable for future.

Wireframe



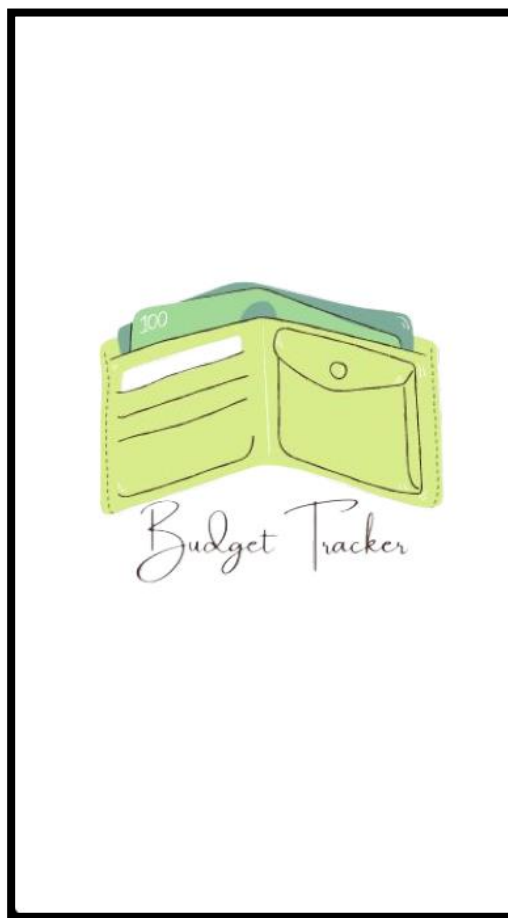
For clear view
click here :

<https://www.figma.com/file/rspEZNCdxBQ19GPL91nlFW/Untitled?type=design&node-id=0-1&t=XnsR1rXdY9B6BDkV-0>

Layouts :

Splash Screen

- Splash screen is the first graphical page which is seen when we visit app.
- It also signifies that you have to wait for a few seconds before landing on the actual screen of the application.
- Image seen in Splash Screen is the logo of our application.
- After a few seconds users are redirected to the main screen which is the login page.



Login Page

- This is login page , where the registered user can login and access the application.
- You have to use your email id and password which was set at the time of registration.
- If you don't have an account and want to register than you can click on below link of "Create new Account " and it will redirect you to a registration page.

Login

✉ abc12@gmail.com

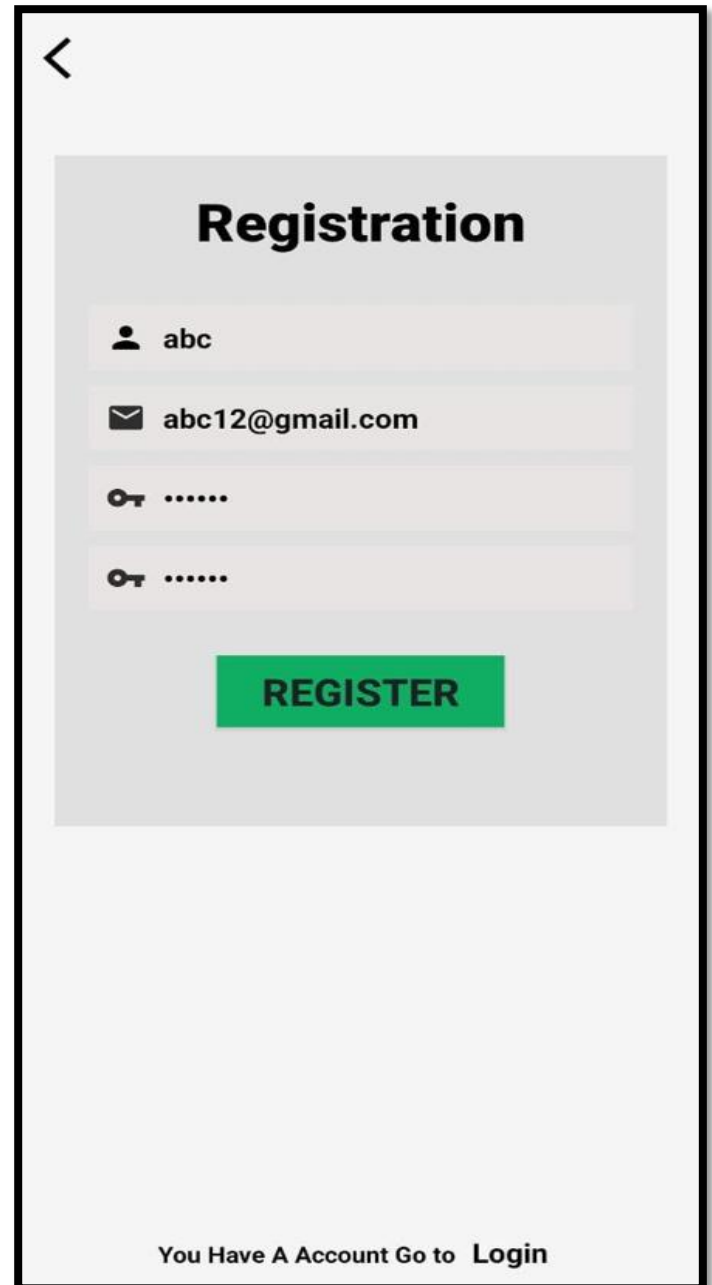
🔑

LOGIN

You Don't Have A Account
Create New Account

Register Page

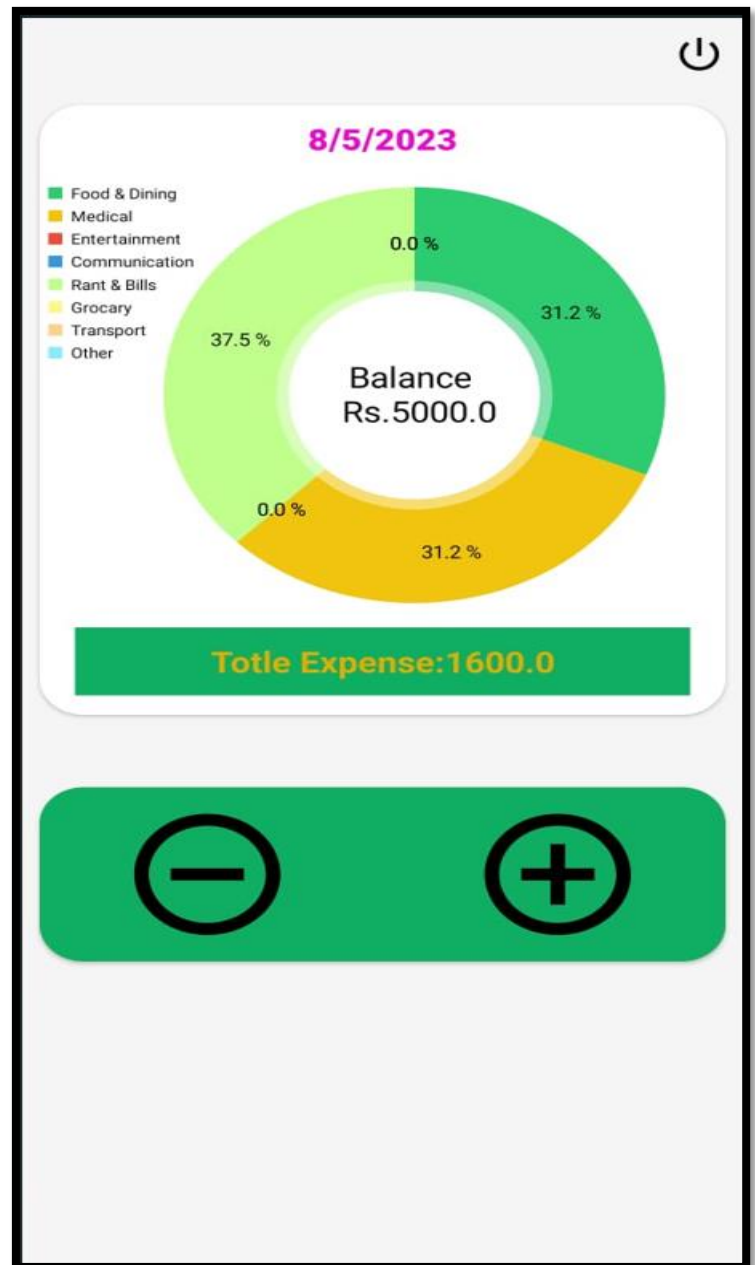
- This is Registration page , where the user can register themselves and create their account.
- Only registered users can access the application after logging in .
- You have to enter a valid username , email id , password and confirm password .
- If you have already have an account and want to login , then you can click on below link of “Login” and it will redirect you to a login page.



The image shows a mobile application registration screen. At the top left is a back arrow icon. The title "Registration" is centered in bold. Below the title are four input fields: a username field with the text "abc", an email field with "abc12@gmail.com", a password field with ".....", and a confirm password field with ".....". Each field has a small icon on the left (person, envelope, and two keys respectively). Below the input fields is a large green button with the text "REGISTER" in white. At the bottom of the screen, there is a link that says "You Have A Account Go to Login".

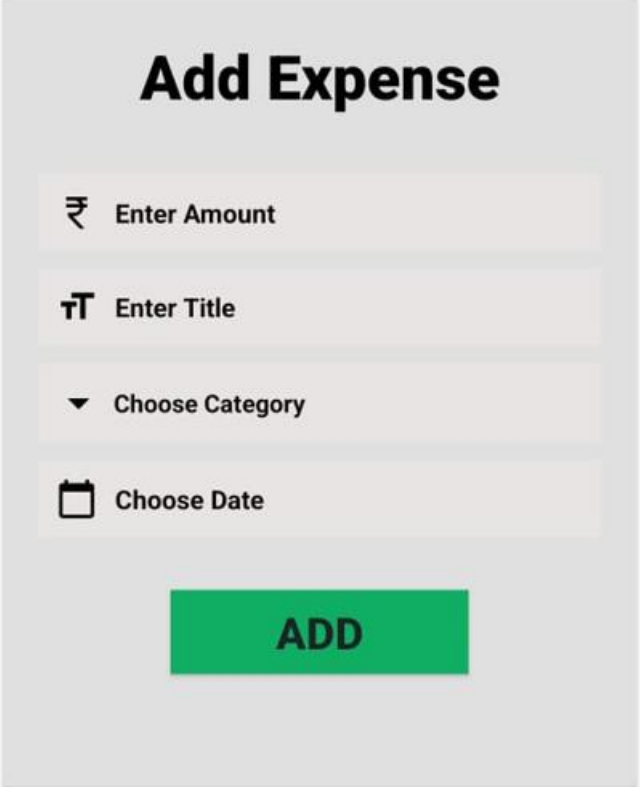
Home Page

- This is Home page , It will show the details of total expense and balance category wised.
- Plus (+) means to add income. When you click on this symbol you will be redirect to add income page to add your incomes .
- Minus (-) means to add expenses. When you click on this symbol you will be redirect to add expense page to add your expenses
- On the Upper Right Side , there is a Logout icon were by clicking you be logged out from the app.



Add Expense

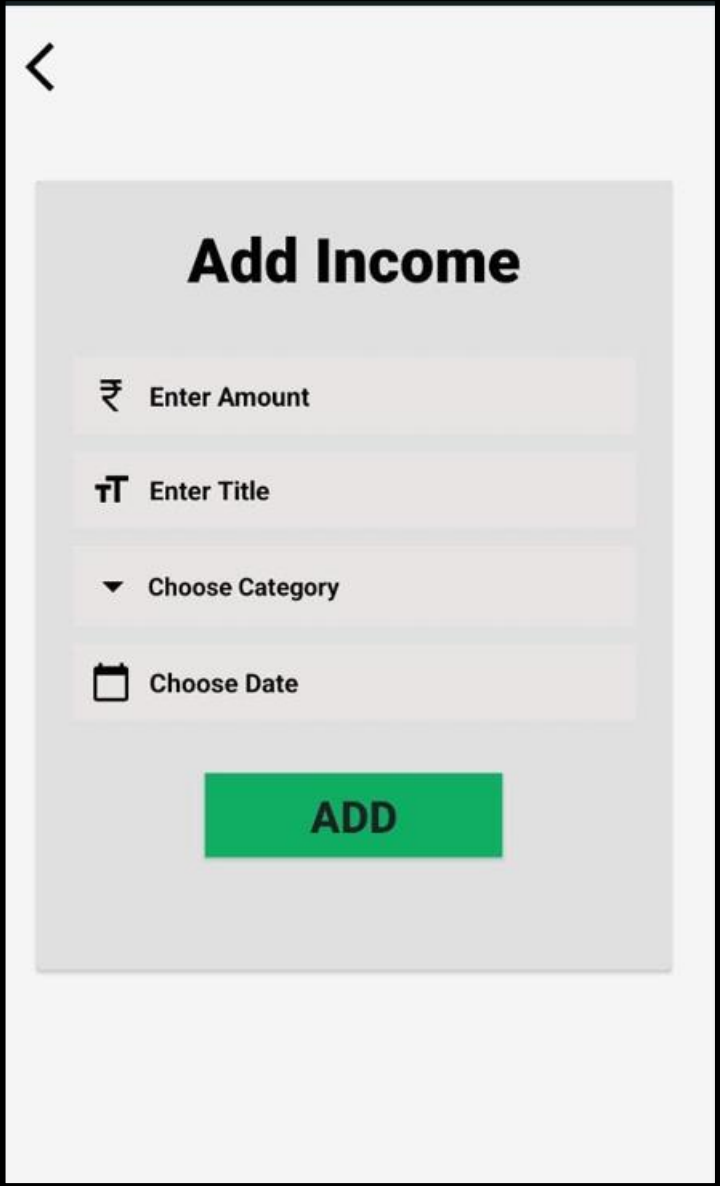
- In this , user can add their daily expenses .
- There are 4 fields , in add amount – enter the amount of expense you have to add
- In title field you can enter name of your expense.
- In choose category , user can choose category of your expense like transport , grocery, education etc.
- At last , choose date of the expense.



The image shows a mobile application interface for adding an expense. It features a light gray background with a white card containing the form. At the top left of the card is a back arrow icon. The title 'Add Expense' is centered at the top of the card in a bold, black font. Below the title are four input fields, each with a small icon on the left and placeholder text: a rupee symbol for 'Enter Amount', a document icon for 'Enter Title', a downward arrow for 'Choose Category', and a calendar icon for 'Choose Date'. At the bottom of the card is a prominent green button with the word 'ADD' in white capital letters.

Add Income

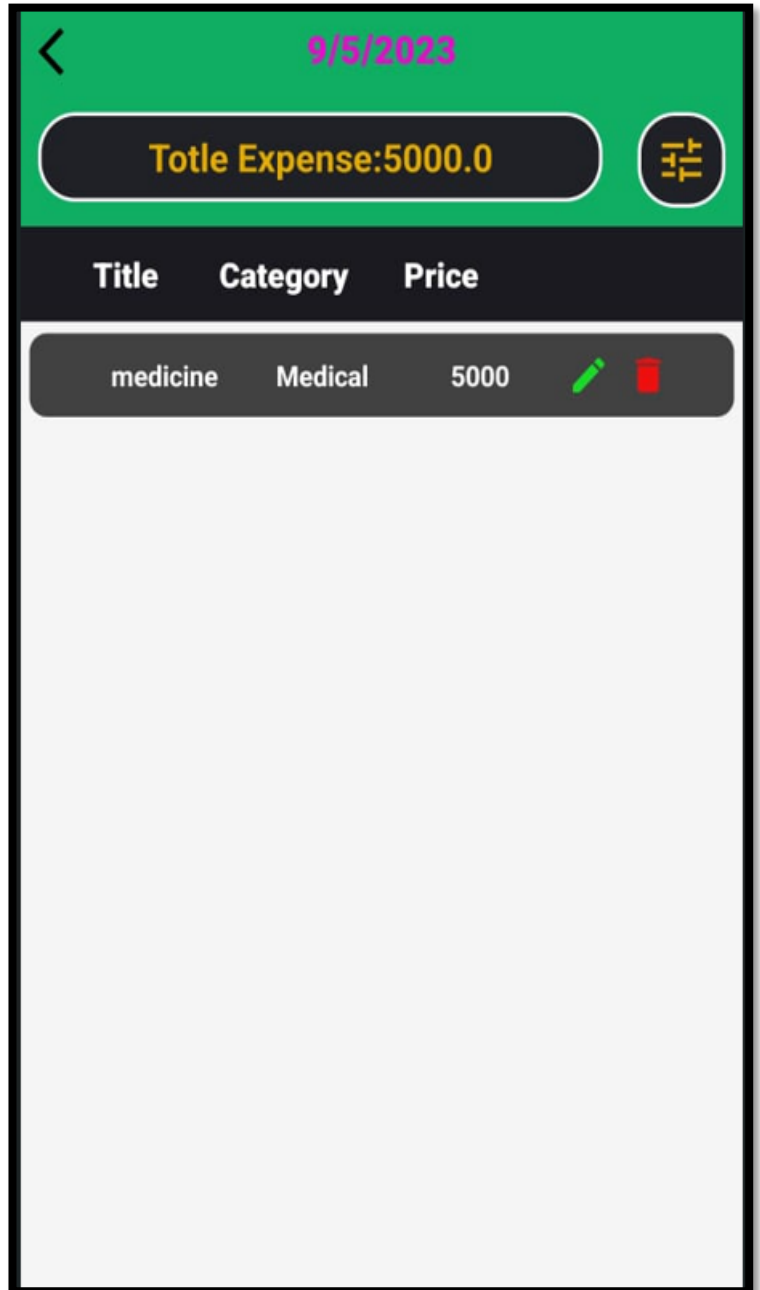
- In this , user can add their incomes.
- Here all the fields are same as add expense page.
- In this choose category field, you will get category like , loan or pocket money
- After that , click on Add button and it will add that details.



The image shows a mobile app interface for adding income. At the top left is a back arrow. The title 'Add Income' is centered at the top. Below it are four input fields: 'Enter Amount' with a rupee symbol, 'Enter Title' with a title symbol, 'Choose Category' with a dropdown arrow, and 'Choose Date' with a calendar icon. At the bottom is a green 'ADD' button.

Expense History

- In this , user can see their expense history.
- Here user can edit or delete their data.
- Green pen icon is of editing and red bin is for deleting.
- Total Expense amount is also seen at uppertime.



Edit Expense

- This is edit expense page.
- Here user can edit their data.
- After editing the fields , click on Update button and your data will be updated.

Edit Expense

₹ 5000

TT medicine

▼ Medical

📅 9/5/2023

UPDATE

7. References

For our application development as well as for the preparation of the software requirements specifications, we took various references from various websites and even a reference of an existing application is taken into consideration. The websites which we refer are as follows:

<https://www.youtube.com>

<https://stackoverflow.com>

<https://developer.android.com>

Other references are taken from such similar applications of budget tracker. The one which are taken into consideration are as follows:-

- Wallet
- MyBudget
- PocketGuard