**Efficient Android Expense Tracker: A Comprehensive Solution for Personal Finance Management**

**A**

**PROJECT REPORT**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD**

**DEGREE OF**

**BACHELOR OF COMPUTER APLLICATION**

Submitted by

SHAIK SHAMEER VALI

(280121861044)

Under the guidance of

Prof. Mohammad Ilyas

Department of BCA

St. Joseph’s Degree College

Submitted to



OSMANIA UNIVERSITY : HYDERABAD – 500 007.

(Re-Accredited by NAAC with A Grade) Approved by Distance Education

Council of India.

**ST. JOSEPH’S DEGREE & P.G COLLEGE**

***(Affiliated to Osmania University)***

**Boston Towers, Pillar No. 187, Upper Pally, Rajendranagar, R.R. Dist.**

**Ph: 04024018619 & 9619, 8125301483 & 86**

**CERTIFICATE**

This is to certify that the project title **"EXPENSES TRACKING ANDROID APPLICATION ON TRACKING DAILY EXPENSES"** is a record of the confined work done by **SHAIK. SHAMEER VALI** bearing hall ticket number **280121861044**. **BCA** (Bachelor’s of Computer Applications) student of **ST.JOSEPH'S DEGREE COLLEGE**.

The project is being submitted in partial fulfillment. Towards the requirement for the award of the degree of bachelor in computer applications from **OSMANIA UNIVERSITY** during the period od **(2021-2024).**

**Mrs. Jyothi Lakshmi**

**Principal**

**St. Joseph’s Degree College, 2-2-44, Boston Towers, Pillar # 187, Upperpally.**

**E-Mail ID:** [**stjjrcattapur@yahoo.com**](mailto:stjjrcattapur@yahoo.com) **Phone # 0091-8125301483 & 86** [**http://www.st-josephs.in/**](http://www.st-josephs.in/)

**ST. JOSEPH’S DEGREE & P.G COLLEGE**

***(Affiliated to Osmania University)***

**Boston Towers, Pillar No. 187, Upper Pally, Rajendranagar, R.R. Dist.**

**Ph: 04024018619 & 9619, 8125301483 & 86**

**CERTIFICATE**

This is to certify that **SHAIK SHAMEER VALI** is a bonafide student of **BCA III YEAR** of this institution with Hall **Ticket No: 280121861044** for this Academic year 2021-2024. He has submitted a project on **ANDROID BASED APPLICATION ON TRACKING DAILY EXPENSES.**

**Project Guide**

**St. Joseph’s Degree College, 2-2-44, Boston Towers, Pillar # 187, Upperpally**

**E-Mail ID:** [**stjjrcattapur@yahoo.com**](mailto:stjjrcattapur@yahoo.com) **Phone # 0091-8125301483 & 86 http://www.st-josephs.in**

**ST. JOSEPH’S DEGREE & P.G COLLEGE**

***(Affiliated to Osmania University)***

**Boston Towers, Pillar No. 187, Upper Pally, Rajendra nagar, R.R. Dist.**

**Ph: 04024018619 & 9619, 8125301483 & 86**

**CERTIFICATE**

This is to certify that  **SHAIK SHAMEER VALI** is a bonafide student of **BCA III YEAR** of this institution with Hall **Ticket No: 280121861044** for this Academic year 2021-2024. He has submitted a project on **ANDROID BASED APPLICATION ON TRACKING DAILY EXPENSES.**

**External Examiner Internal Examiner**

**St. Joseph’s Degree College, 2-2-44, Boston Towers, Pillar # 187, Upperpally**

**E-Mail ID:** [**stjjrcattapur@yahoo.com**](mailto:stjjrcattapur@yahoo.com) **Phone # 0091-8125301483 & 86** [**http://www.st-josephs.in/**](http://www.st-josephs.in/)

**ST. JOSEPH’S DEGREE & P.G**

***)***

**Boston Towers, Pillar No. 187, Upper Pally, Rajendranagar, R.R. Dist.**

**Ph: 04024018619 & 9619, 8125301483 & 86**

**CERTIFICATE**

This is to certify that **SHAIK. SHAMEER VALI** is a bonafied student of **BCA III YEAR** of this institution with Hall **Ticket No: 280121861044** for this Academic year 2021-2024. He has submitted a project on **ANDROID BASED APPLICATION ON TRACKING DAILY EXPENSES.**

**HOD**

**Signature**

**St. Joseph’s Degree College, 2-2-44, Boston Towers, Pillar # 187, Upperpally**

**E-Mail ID:** [**stjjrcattapur@yahoo.com**](mailto:stjjrcattapur@yahoo.com) **Phone # 0091-8125301483 & 86** [**http://www.st-josephs.in/**](http://www.st-josephs.in/)

**DECLARATION**

I am the student of **St. Joseph degree college**,upparpally, Hyderabad. Do hereby declare that the project report **ANDROID BASED APPLICATION on TRACKING DAILY EXPENSES** is an original and bonafide work done by myself. This report is being submitted in partial fulfilment of requirement for the award in bachelor of degree in computer applications.

**SHAIK. SHAMEER VALI : 280121861044**

**ACKNOWLEDGEMENT**

As a student of St. Joseph’s degree college is affiliated to Osmania University. I express my heartful thanks to the **MOHAMMAND ILYAS,** organization for providing the necessary information related to my project.

I also express my sincere thankfulness to **Sri. SIRAJUDDIN**, Hon. Secretary & correspondent of St. Josephs Degree College and **Mrs. JYOTHI LAKSHMI** Principal of St. Josephs Degree College for providing me the opportunity and complete cooperation, and all the necessary infrastructure to complete my project.

Last but not the least it is my pleasure to have a dedicated faculty and guide for my project, their whole-hearted co-operation helped in completing the project in time and acquiring the knowledge.

**SHAIK SHAMEER VALI**

**2801-2186-1044**

**ABSTRACT**

Managing personal finances effectively is crucial for maintaining financial stability and achieving financial goals. In the contemporary digital era, mobile applications have become indispensable tools for tracking expenses conveniently and efficiently. This paper presents the design and development of an Android expense tracker application aimed at empowering users to take control of their finances effortlessly.

The proposed Android expense tracker offers a user-friendly interface coupled with robust features to streamline expense management. Key functionalities include expense categorization, real-time expense tracking, budget setting, and insightful data visualization. By categorizing expenses into predefined or customizable categories, users can easily classify their expenditures and gain insights into their spending patterns.

Real-time expense tracking enables users to record expenses as they occur, ensuring accuracy and completeness in financial records. Moreover, the application facilitates budget setting by allowing users to define spending limits for various expense categories. Timely notifications and alerts notify users when they approach or exceed their budgetary constraints, fostering better financial discipline.

Furthermore, the Android expense tracker employs data visualization techniques such as charts to present users with intuitive summaries of their financial activities. These visual representations offer valuable insights into spending trends, enabling users to make informed decisions and adapt their financial habits accordingly.

In conclusion, the proposed Android expense tracker offers a comprehensive solution for personal finance management, empowering users to track expenses efficiently, set budgets effectively, and make informed financial decisions.

**TABLE OF CONTENTS:**

DECLARATION………………………………………………………

ACKNOWLEDGEMENT………………………………………………

ABSTRACT…………………………………………………………

LIST OF ABBREVIATIONS………………………………………….

LIST OF FIGURES………………………………………………………

CHAPTER 1: INTRODUCTION ………………………………………

1.1 INTRODUCTION…………………………………………………

1.2 PROBLEM STATEMENT…………………………………………

1.3 OBJECTIVES………………………………………………………

1.4 METHODOLOGY AND MODULES………………………………

1.5 MANUAL TESTING PROCEDURE……………………………

CHAPTER 2: REQUIREMENT ANALYSIS AND SYSTEM REQUIREMENTS (Updated)

2.1 REQUIREMENT ANALYSIS

2.2 SYSTEM REQUIREMENTS

2.3 TECHNOLOGIES USED

CHAPTER 3: LITERATURE SURVEY……………………………….

CHAPTER 4: SYSTEM DEVELOPMENT……………………………

4.1 ANALYSIS…………………………………………………………

4.2 DESIGN OF THE APPLICATION……………………………….

4.2.1 ACTIVITY DIAGRAM

4.2.2 CLASS DIAGRAM

4.2.3 SEQUENCE DIAGRAM

4.2.4 USE CASE DIAGRAM

4.3 PROPOSED ARCHITECTURE…………………………………

4.4 DATA STORAGE AND MANAGEMENT……………………….

4.5 FEATURE IMPLEMENTATION…………………………………

CHAPTER 5: PERFORMANCE ANALYSIS…………………………

5.1 ANALYSIS OF SYSTEM DEVELOPED………………………

5.1.1 Screenshots of App Interface

5.1.2 Required Details

5.2 RESULTS AT VARIOUS STAGES………………………………

5.2.1 Screenshots of Key Features

5.2.2 Required Details

5.3 OUTPUT AT VARIOUS STAGES…………………………………

5.4 COMPARISON………………………………………………………

CHAPTER 6: MANUAL TESTING RESULTS………………………

6.1 TEST CASES

6.1.1 ADD EXPENSE

6.1.2 VIEW EXPENSES

6.1.3 DELETE EXPENSE

6.1.4 SELECT CATOGORY

6.1.5 ADD NOTE

6.2 TEST RESULTS

6.2.1 ADD EXPENSE

6.2.2 VIEW EXPENSES

6.2.3 DELETE EXPENSE

CHAPTER 7: CONCLUSIONS…………………………………………

7.1 CONCLUSIONS

7.2 FUTURE SCOPE

References………………………………………………………………

**CHAPTER 1**

**1.1 INTRODUCTION**

In today's fast-paced world, managing personal finances effectively has become increasingly crucial. Individuals often find themselves struggling to keep track of their expenses, leading to financial stress and mismanagement.

To address this challenge, the development of an innovative Expense Tracker App has been initiated.

This app aims to revolutionize the way individuals manage their finances by providing a user-friendly and efficient platform for tracking expenses on-the-go.

The Expense Tracker App is designed to cater to the modern lifestyle, where individuals often rely on mobile devices for various tasks.

With the app installed on their smartphones, users can effortlessly record and monitor their expenses in real-time, ensuring greater financial awareness and control.

Whether it's tracking daily expenditures, managing budgets, or analyzing spending patterns, the app offers a comprehensive solution to streamline personal finance management.

This initiative stems from the recognition of the common struggles faced by individuals in managing their finances effectively.

With the rise of digital banking and cashless transactions, there is a growing need for tools that empower users to take charge of their financial well-being.

By leveraging the capabilities of mobile technology, the Expense Tracker App seeks to provide users with a convenient and accessible platform to track, analyze, and optimize their spending habits.

Moreover, the Expense Tracker App is not just limited to personal use but can also benefit businesses, freelancers, and other professionals in managing their expenses more efficiently. By offering customizable features and intuitive functionalities, the app caters to a diverse range of users with varying financial needs and preferences.

In summary, the Expense Tracker App represents a significant advancement in personal finance management, offering users a powerful tool to take control of their money and achieve financial stability.

With its user-friendly interface, real-time tracking capabilities, and customizable features, the app is poised to redefine the way individuals manage their expenses in today's digital age.

The Expense Tracker App(Expensify) is like a digital wallet that helps people manage their money better. It's a handy tool you can use on your phone to keep track of your spending habits, set budgets, and see where your money goes.

I made this app because we know how hard it can be to stay on top of your finances. With so many bills to pay and things to buy, it's easy to lose track and end up spending more than you should. Our goal is to make managing money easier and more convenient for everyone.

The idea for the Expense Tracker App(Expensify) came from seeing how many people struggle with money management. From forgetting to pay bills on time to overspending on unnecessary things, we wanted to create something that could help people take control of their finances and make smarter decisions with their money.

With the Expense Tracker App, you can see all your transactions in one place, categorize your expenses, and set savings goals. It's like having a personal finance assistant in your pocket, guiding you towards better financial habits.

Not only is the app useful for individuals, but it can also benefit businesses and freelancers who need to track their expenses for tax purposes or budgeting. With customizable features and easy-to-use interface, it's a versatile tool that can adapt to different needs and situations.

In essence, the Expense Tracker App is here to make managing money simpler and less stressful. By giving people the tools they need to understand their finances better, we hope to empower them to make smarter decisions and achieve their financial goals.

**1.2 Problem Statement**

Many people find it difficult to keep track of their spending and manage their money effectively. They often end up spending more than they should, leading to financial stress and uncertainty. This is a common problem faced by individuals of all ages and income levels.

The problem is compounded by the lack of easy-to-use tools to help people manage their finances. Traditional methods, such as pen and paper or complex spreadsheets, are often cumbersome and time-consuming.

resulting in surprise bills, overdrafts, and difficulty in saving money for important goals like emergencies or future plans.

Furthermore, for individuals who are not accustomed to budgeting or financial planning, the lack of visibility into their spending habits can lead to a cycle of debt and financial insecurity. Without a clear understanding of where their money is going, they may struggle to make ends meet or achieve their long-term financial goals.

In addition, there is a growing need for a solution that caters to people from diverse backgrounds and financial literacy levels. Many existing tools are either too complex for beginners or lack the features needed for advanced users.

This leaves a gap in the market for a simple yet comprehensive solution that can help everyone manage their money better.

Overall, the problem statement revolves around the need for an accessible, user-friendly, and effective tool to help individuals track their expenses, budget effectively, and achieve financial stability. This is where the Expense Tracker App aims to make a difference.

**1.3 Objectives**

**Objective 1: Ease of Use**

1.1. Create an app that simplifies tracking daily expenses for everyone.

1.2. Organize expenses into easy-to-understand categories like food, bills, and entertainment.

**Objective 2: Customization and Personalization**

2.1. Allow users to customize the app to suit their preferences by adding their own categories or labels.

2.2. Provide options for personalization to enhance the user experience and adapt to individual needs.

**Objective 3: Security and Privacy**

3.1. Ensure users' financial information remains secure with robust security measures.

3.2. Implement privacy features to protect users' sensitive data from unauthorized access.

**Objective 4: Insights and Analysis**

4.1. Provide visual charts and reports to help users understand their spending habits.

4.2. Offer insights into spending trends to empower users to make informed financial decisions.

**Objective 5: Budget Management**

5.1. Enable users to set goals for their spending and receive reminders to stay within budget.

5.2. Help users track their expenses against predefined budget limits to manage finances effectively.

**Objective 6: Expense Sharing**

6.1. Allow users to split bills and expenses with others, simplifying group expenses.

6.2. Facilitate easy sharing of expense details to streamline financial transactions among friends or roommates.

**Objective 7: User-Friendly Design**

7.1. Design the app to be easy to use for people of different ages and backgrounds.

7.2. Create an intuitive user interface with clear navigation and minimal learning curve.

**Objective 8: Accessibility**

8.1. Ensure that the app is accessible to people with disabilities, making it easy for everyone to use.

8.2. Incorporate accessibility features such as screen reader support and adjustable font sizes for inclusivity.

**1.4 Methodology:**

**1.Iterative Development Process:**

Followed an iterative development process, incorporating feedback from stakeholders and users at each stage of development.

Utilized Agile principles to prioritize features and adapt to changing requirements throughout the project lifecycle.

**Modules:**

**2.User Interface Design Module:**

Utilized Figma or similar design tools to create wireframes and mockups of the app's user interface (UI).

Designed intuitive and visually appealing UI elements using XML layouts in Android Studio, ensuring consistency with Material Design guidelines.

**3. Data Storage Module:**

Implemented SQLite database to store and manage expense data locally on the device.

Utilized Android's built-in SQLiteOpenHelper class to create and manage the database schema, tables, and CRUD (Create, Read, Update, Delete) operations.

**4.Expense Entry Module:**

Developed UI components and input forms for users to enter expense details, such as amount, date, category, and notes.

Implemented validation checks on user input to ensure data integrity and prevent errors

.

**5. Expense Categorization Module:**

Created functionality to categorize expenses into predefined or user-defined categories, facilitating organization and analysis.

Stored category data in SQLite database tables and provided user interface controls for category selection during expense entry.

**6. Budget Tracking Module:**

Designed features for users to set budget limits for different expense categories and track their spending against these limits.

Implemented logic to calculate and visualize budget progress using charts or progress bars within the app.

**7.Testing and Debugging Module:**

Conducted manual testing to validate the functionality and usability of the app's features.

Developed test cases and scenarios to cover various use cases, including expense entry, category management, budget tracking, and notification handling.

Recorded test results and identified issues during development iterations.

**1.5 MANUAL TESTING PROCEDURE**

**1. Expense Entry Testing**:

- I checked if I could add expenses correctly. This means entering the amount, selecting the date, picking a category, and adding any notes if needed.

- I tried entering different kinds of expenses, like groceries, bills, or entertainment, to see if the app handled them all properly.

**2. Expense Viewing Testing:**

- I tested if I could view my expenses after adding them. This meant going to the expense list and making sure all the expenses I added showed up there.

- I also checked if I could filter expenses by category or date to find specific ones easily.

**3. Expense Deletion Testing:**

- I made sure I could delete expenses if I needed to. This involved selecting an expense from the list and deleting it to see if it was removed correctly.

- I checked if the app asked for confirmation before deleting an expense to prevent accidental deletions.

**4. Budget Tracking Testing:**

- I tested if the app properly tracked my spending against my budget. This meant setting a budget for a specific category and then adding expenses to see if the app warned me when I was close to or over my limit.

- I also checked if the app accurately calculated my remaining budget after each expense.

**5. User Interface Testing:**

- I checked if the app's user interface was easy to understand and navigate. This involved looking at things like button placement, menu organization, and overall layout.

- I made sure the app looked good and worked well on different screen sizes and orientations.

**6. Security Testing:**

- I tested if the app kept my personal finance information safe. This involved making sure my data was encrypted and protected from unauthorized access.

- I also ensured that the app provided options for me to secure my data, such as enabling

app lock or using biometric authentication if available.

**CHAPTER 2**

**REQUIREMENTS AND SYSTEM ANALYSIS**

**2.1 Requirement Analysis for the Expense Tracker App(Expensify):**

During the requirement analysis phase of the expense tracker app development, an in-depth examination of user needs and project objectives was conducted to ensure the app's functionality aligns with the intended purpose. The analysis aimed to identify the key features and functionalities required to meet the project goals effectively. Here's a summary of the requirement analysis findings:

**1.Functional Requirements:**

**Expense Recording:** Users should be able to input their expenses seamlessly, specifying details such as the amount spent, date of the expense, relevant category, and optional notes.

**Expense Viewing:** Users should have the ability to view their recorded expenses conveniently, with options to filter and sort expenses based on various parameters like date and category.

**Expense Management:** The app should allow users to delete recorded expenses as needed, with a confirmation prompt to prevent accidental deletions.

**Budget Tracking:** Users should be able to set budget limits for different expense categories and receive notifications when nearing or exceeding these limits.

**Data Backup and Restoration**: The app should provide functionality for backing up user data securely and restoring it when necessary to prevent data loss.

**2.Non-Functional Requirements:**

**User Interface Design:** The app's interface should be intuitive and user-friendly, with clear navigation and visually appealing design elements.

**Performance Optimization:** The app should be optimized for performance, ensuring fast response times and smooth user interactions.

**Data Security**: Measures should be implemented to safeguard user data, including encryption of sensitive information and secure storage practices.

**Platform Compatibility:** The app should be compatible with Android devices running different versions of the operating system to maximize accessibility for users.

**3.Documentation and Validation:**

The identified requirements have been documented comprehensively, including detailed descriptions, priorities, and acceptance criteria.

Regular validation sessions were conducted to ensure that the requirements accurately reflect the project objectives and meet the needs of the intended users.

**2.2 System Requirements for the Expense Tracker App:**

**1. Operating System Compatibility:**

- The app should be compatible with Android operating systems, including versions 6.0 (Marshmallow) and above, to ensure widespread accessibility for users.

**2. Hardware Requirements:**

- The app should be optimized to run smoothly on a variety of Android devices, including smartphones and tablets, with varying screen sizes, processing power, and memory capacities.

**- Minimum hardware specifications may include:**

- Processor: ARM or x86 architecture with at least dual-core processing power.

- RAM: Minimum 1 GB of RAM for optimal performance.

- Storage: Sufficient internal storage space for app installation and data storage, with support for external storage options where available.

**3. Software Dependencies:**

- The app may require access to certain software components or libraries to facilitate its functionality, such as:

- Android SDK: The app should be developed using the Android Software Development Kit (SDK) to leverage platform-specific features and APIs.

- SQLite Database: The app may utilize the SQLite database engine for local storage and management of expense data.

**4. Network Connectivity:**

- While the app primarily operates offline for expense tracking, it may require intermittent network connectivity for certain features such as cloud synchronization, software updates, or accessing online resources like currency conversion rates or financial data feeds.

**5. Security Considerations**:

- The app should implement robust security measures to protect user data and ensure privacy, including:

- Encryption: Utilizing encryption techniques to secure sensitive data stored locally on the device.

- Secure Communication: Ensuring that data transmitted between the app and external servers is encrypted and protected from interception or tampering.

**6. Compatibility Testing:**

- The app should undergo comprehensive compatibility testing to ensure functionality across a wide range of Android devices, screen resolutions, and device configurations.

- Compatibility testing should also include validation against different versions of the Android operating system to identify and address any compatibility issues or platform-specific behaviors.

**2.3Tools & Technologies Used**

**Hardware**

**Processor**

Intel ® Core™ i5-1240p CPU @2.70GHz

**Installed Memory (RAM)**

8.00 GB

**System Type**

64 bit Operating System

**Software Interface**

**Client-Side**

Android Mobile

**Software**

Android Studio,Figma

**Database Server**

SQLite Database

**CHAPTER 3**

**Literature Survey**

1) "Expensify: An expense tracker and report creator for mobile apps" (2018) authored by L. Zhang and Y. Wang Expensify is a mobile app based spending tracker and report maker, and this essay assesses its efficacy. According to the authors, Expensify is a helpful tool for both people and organisations to manage their costs and produce reports for tax and reimbursement purposes.

2) "Personal Finance Management: The Design and Assessment of a Mobile Expense Tracker" by J. M. Clark et al. - An app called Cashew for tracking mobile expenses is designed and evaluated in this study. Cashew is a helpful tool, according to the developers, for managing personal finances while on the go and keeping track of expenditures.

3) "Comparing manual and automatic 4) expense tracking," L. V. Ackerman and M. Weigel (2020). - In this study, th e effectiveness of manually documenting spending in a notebook or spreadsheet vs automatically recording expenses is compared (i.e., using a software tool to automatically track expenses). The researchers discovered that automatic spending tracking is faster and more accurate than manual tracking .

4) M. S. Islam et al., "An intelligent expense tracking system employing machine learning approaches," 2019 - The design and deployment of an intelligent expense monitoring system that employs machine learning methods to forecast future expenses based on historical data are described in this study. The authors discovered that their method accurately forecasts costs and enables consumers to manage their money more effectively.

5) "A survey of expense tracking applications" by A. M. Al-Naser and others (2021) - The popular expense-tracking applications Mint, Pocket Guard, and Personal Capital are reviewed in this essay. These apps, according to the authors, offer a variety of functions for managing personal money, such as budgeting, goal-setting, and investment tracking.

**CHAPTER 4**

**SYSTEM DEVELOPMENT**

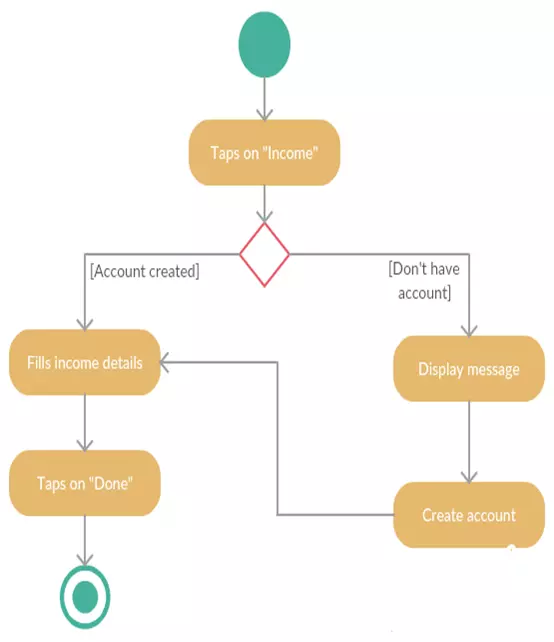
**4.1 ANALYSIS**

In this phase, I take a close look at the requirements I gathered earlier to understand exactly what my expense tracker app needs to do. I'm like a detective, examining each requirement carefully to ensure I'm not missing anything important. If I find any new needs or changes, I add them to my list.

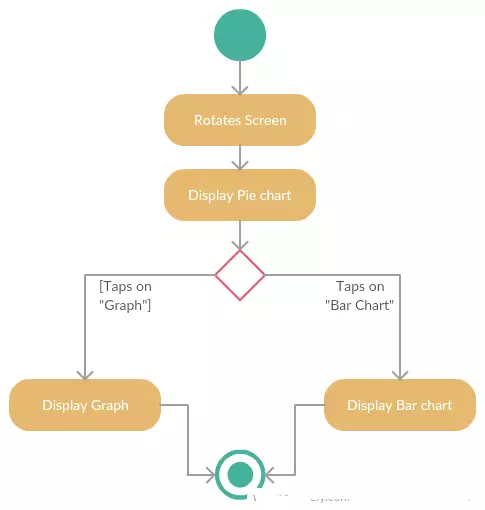
Once I finish analyzing, I have a clear picture of what my app will do. It's like having a map before going on a journey – I know where I'm headed and how I'll get there. This analysis is crucial because it guides everything I do next in building the app.

**4.2 ACTIVITY DIAGRAM**

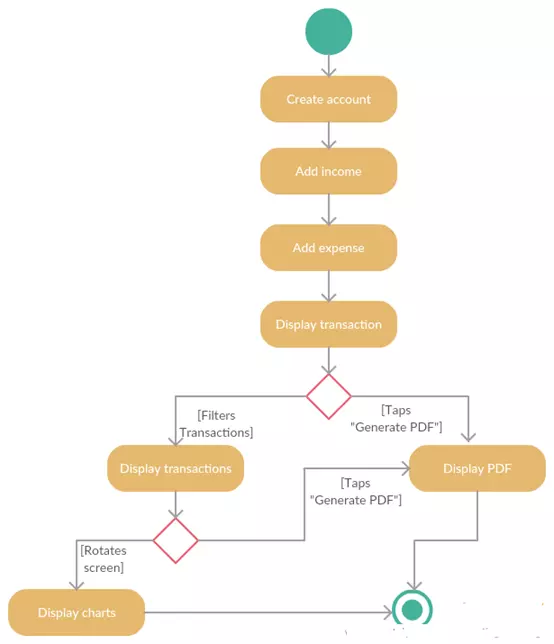
**-Sequence 1**

****

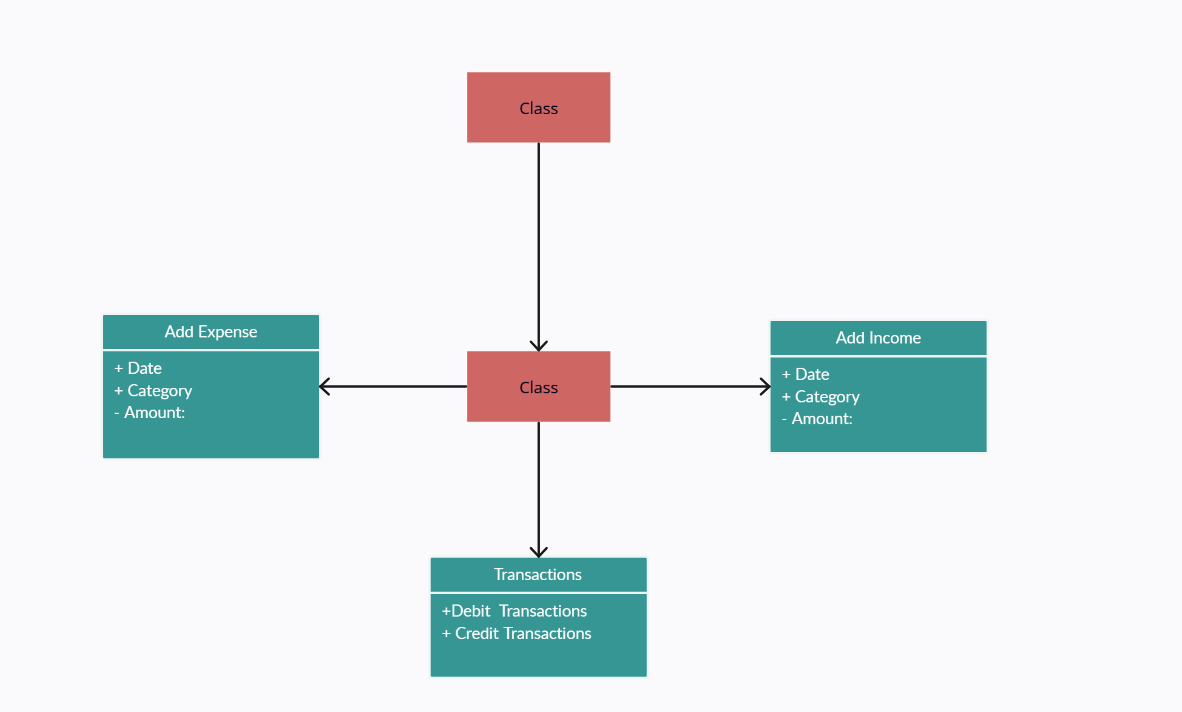
**-Sequence 2:**

****

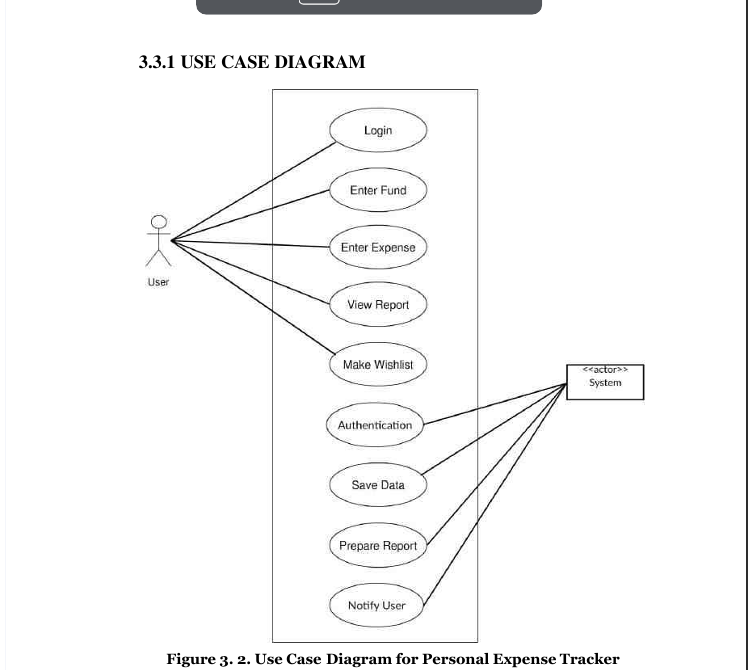
**-Sequence 3**

****

**4.2.2 CLASS DIAGRAM**

****

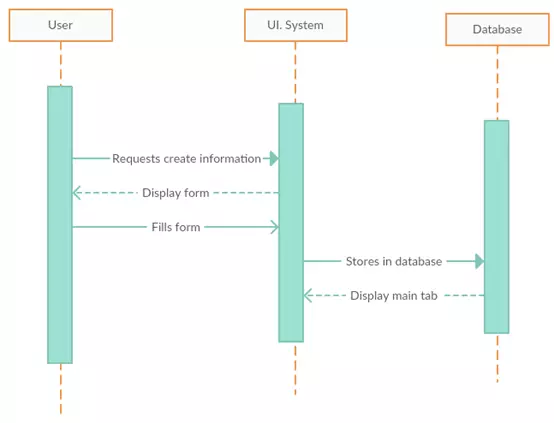
**4.2.4 USE CASE DIAGRAM**

****

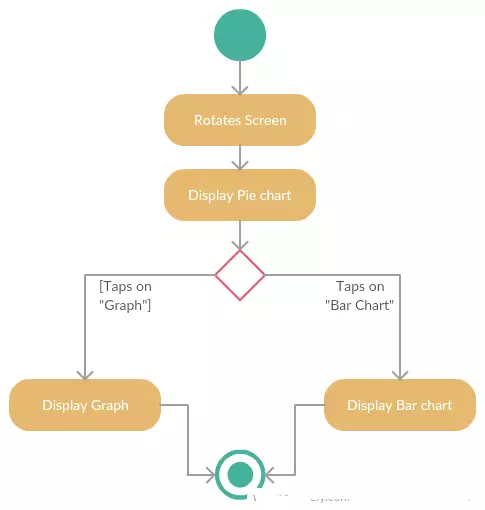
**Use Case Diagram**

* Personal Expense Tracker (PET) lets you have your privacy, you must login to open this application. Users are authenticated upon each entry so that they can enter their expenses and income which will be recorded by the system. System generates meaningful output form entered data.
* The saved data can later be altered if the user wants to do so. Altering here means adding description, changing wish list updating data etc.
* User can also view the result. They can also filter result to see the required content only.
* Upon using this application user are provided with three options for data entry namely –  Income, Expense and wish list. If he/she selects income or expense he/she would be provided with its types and subtypes. For wish list only items can be inserted. These data would be saved onto database according to their respective category.

**4.2.3 Sequence Diagram**

****

**-Sequence Diagram 2**



**4.3 PROPOSED ARCHITECTURE**

In this section, I outlined how I planned to organize the expense tracker app. It was akin to drawing a map before embarking on a journey, helping me understand how different parts of the app would work together.

**1. Overview of My Plan:**

- I explained how I intended to organize the app, delineating its various components and their communication channels.

- I chose a style for building the app, akin to assembling building blocks, to maintain organization.

**2. Breaking Down the Pieces:**

- I divided the app into different parts and explained the functionality of each.

- I elucidated how these parts would synergize to achieve the desired app functionality.

**3. Keeping Things Scalable and Fast:**

- I considered scalability to ensure the app could handle numerous users without performance degradation.

- I also deliberated on maintaining smooth operation in case of unexpected errors.

**4. Making Sure It's Safe and Reliable:**

- I discussed strategies to safeguard the app from security breaches and protect users' privacy.

- I prioritized reliability to prevent crashes or data loss.

**6. Getting It Ready to Use**:

- I planned the deployment process to ensure the app could be easily installed and updated.

- I explored strategies for smooth app updates and maintenance.

**7. Explaining Why I Made Certain Choices:**

- I provided smothness behind design decisions, such as balancing speed and complexity.

- I justified my choices based on project requirements and objectives.

**4.4 DATA STORAGE AND MANAGEMENT**

I used stored data for the expense tracker app using SQLite. SQLite is a lightweight, file-based database engine that's easy to use and suitable for mobile applications like mine.

**1. Choosing SQLite for Data Storage:**

- I selected SQLite for its simplicity, efficiency, and seamless integration with Android applications.

- SQLite's lightweight nature makes it ideal for storing structured data without the overhead of a client-server architecture.

**2. Designing the Database Schema:**

- I designed the database schema to represent the various entities and relationships in the expense tracker app.

- This involved creating tables for expenses, categories, and any other relevant entities, along with defining their attributes and relationships.

**3. Handling Data Storage:**

- I utilized SQLite's data types to store different types of data, such as text, numbers, and dates, appropriately.

- Data storage operations, including inserting, updating, deleting, and querying data, were implemented using SQL queries within the app's code.

**4. Implementing Data Validation:**

- I incorporated data validation mechanisms within the app to ensure that only valid and consistent data is stored in the SQLite database.

* I used Room library of Android Studio to make sure data save securely.

- This helped maintain data integrity and prevent errors or inconsistencies in the stored data.

**5. Ensuring Data Security:**

- While SQLite databases are stored locally on the device, I implemented encryption mechanisms to protect sensitive data.

- Encryption techniques such as SQLCipher were employed to encrypt the SQLite database file, adding an extra layer of security.

**6. Managing Data Backup:**

- I implemented a data backup mechanism to safeguard against data loss due to device failure or app uninstallation.

- This involved periodically exporting the SQLite database to a backup file stored in a secure location, such as the device's external storage or cloud storage services.

**7. Monitoring and Maintenance:**

- Regular monitoring of the SQLite database's performance and health was conducted to identify any potential issues.

- Maintenance tasks such as vacuuming the database to optimize performance and periodic integrity checks were performed to ensure data consistency.

**4.5 Features**

1. Create multiple budget
2. Live Data
3. Customizable views.
4. Select Transaction type(cash/online).
5. Background color.
6. Modify Transactions.
7. Offline datastore.
8. Selecting budget mode(Weekly/Monthly).
9. Generate reports as PDF files.
10. Fully customizable categories.
11. Cash flow(Pie chart).
12. Expenses percentage.
13. Show transactions.
14. Weekly, Monthly , yearly Analysis.

**CHAPTER 5**

**ANALYSIS OF SYSTEM DEVELOPED**

**5.1 Analysis of system developed**

Now we will discuss the functions of each class and package created for the expense tracker app.

**Common**

* Created a package named **com.myapp.expensify.**
* This package facilitates data sharing among different components of the app.
* Created Required activities of different kind and linked with Main.Activity.
* Imported required packages which may be problem at compile time.
* Implemented functions to handle data formatting and manipulation.
* Taken API from an open source website, called Anychart, which is a library which shares data regarding the different charts view to developers.
* Functions include conversion of Unix timestamp to date format and retrieval of image links.
* Made a live Transaction view , where changes can be visible on state change.

**Helper**

* Introduced a new package named helper containing a class package.
* Implemented utility functions to assist in app functionalities, such as making network requests.

**Model**

* Designed model classes to represent expense data and related entities.
* Included constructors and getters/setters in model classes for data encapsulation.

**Main Activity**

* Integrated location services to retrieve the current device position for expense tracking.
* Implemented runtime permission handling for accessing device resources.
* Managed activity lifecycle methods for proper resource management.
* Utilized asynchronous tasks for background operations, such as data fetching and processing.
* Parsed JSON data using Json library for populating UI elements with expense information.
* Linked different activities and fragments.
* Downloaded and imported icons and image assets.
* Configured Gradle properties and set SDK minimum version and updated Dependencies.

**Results at various stages**

* **Steps taken during development**
* Developed an intuitive user interface for seamless expense tracking and management.
* Leveraged SQLite database for efficient storage and retrieval of expense data.
* Used Room library(Android) to implement storing of data in local database in secure manner.
* Implemented features like budget setting, expense categorization, and data visualization to enhance user experience.
* Made Different view where user can see his/her expenditure.
* Conducted rigorous testing procedures to ensure app functionality and reliability.
* Enforced continuous monitoring and maintenance practices to address any issues and improve app performance.
* Even if device is not connected to internet the user can make transactions.

**Proposed work**

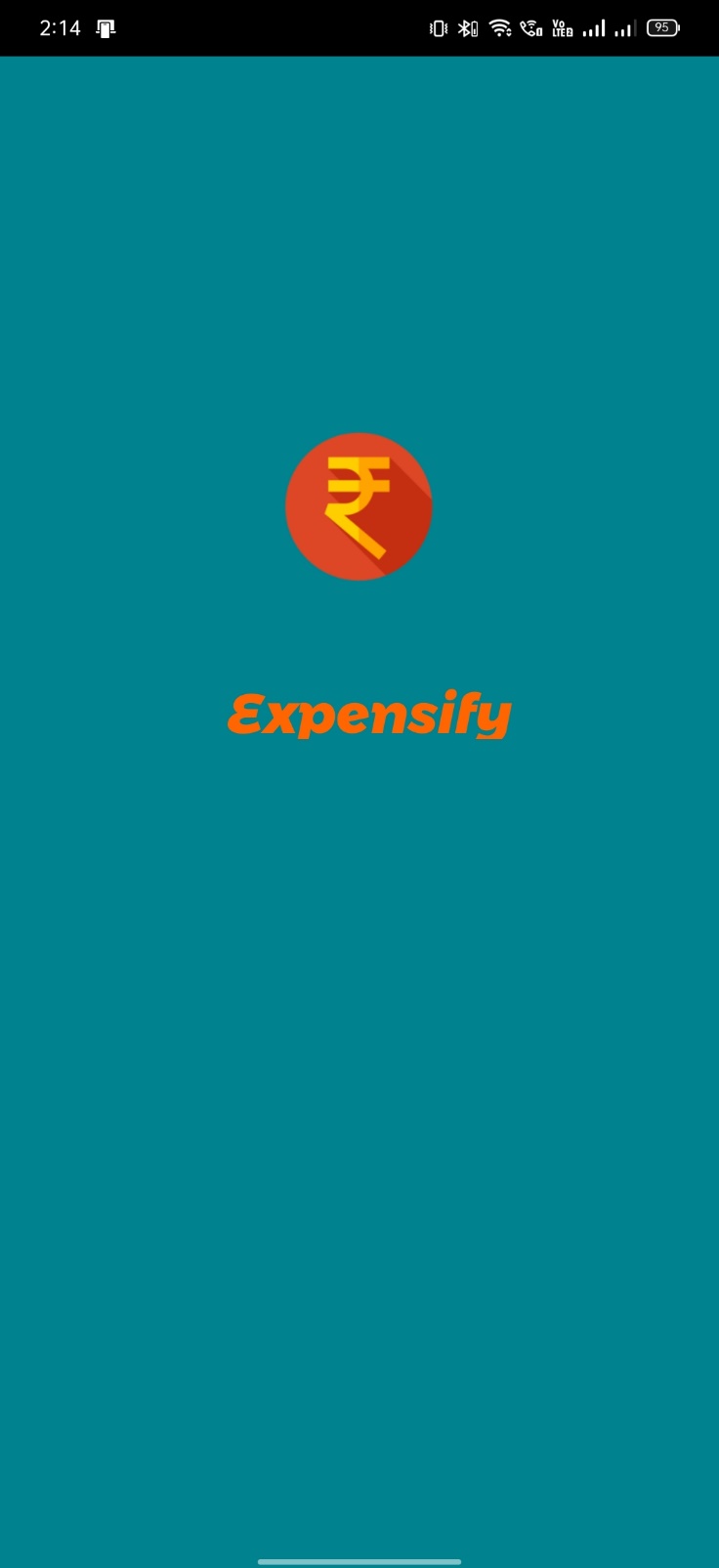
The Expense Tracker project uses iterative methodology to manage tasks effectively. iterative breaks the project into small steps and involves working closely with others to improve it step by step. This method focuses on planning, doing, and checking the work regularly to make sure it meets everyone's needs.

We chose iterative because it lets us adapt quickly to changes in the market or what users want. Instead of planning everything upfront, we make small improvements often. This helps us get feedback quickly and make changes easily. It's like building a puzzle one piece at a time, making sure each piece fits well with the others.

* The Expense Tracker app has two main parts: one for storing data about money, and another for users to see and interact with that data. Even if you're not connected to the internet, you can still use the app.
* You can choose what you spent money on, how much, and how you paid for it.
* The app also helps you understand where your money goes. It looks at what you've entered and tells you which categories you spend the most money on. This helps you see where you might want to spend less.
* You can also look back at your past spending to see how it's changed over time. This helps you make better decisions about your money in the future.
* Overall, the Expense Tracker project follows iterative principles to make sure the app works well for everyone. We keep improving it based on feedback, so it's always getting better and easier to use.
* In this project first we collect need from the different users that exactly what they want. Then we start develop the project. First I develop the Splash screen. After few seconds of time delay user redirect on main activity that is dashboard page. Here user can see the various expenses done by them.
* Also they get the option to add, delete, update expenses on the same page. On the second page user can get the summary of their expenses in the form of pie chart.
* Also in this app user can add their investment, get notification about their payments, also set reminder. The user can get message when the expense limit got exceeds.
* In the transaction, the tab user has an option available for creating a report in PDF. Users click on the PDF button then PDF report will be generated to the user and also user could view that report and that report will be automatically saved in the device.
* On the menu bar user can go through different fragments and can customize viewing of the data.

**5.1.1 ScreenShots of expense tracker system**

* This screen is the start up screen for the expense tracker app(Expensify).Whenever user click on the app this is the first screen that appears.

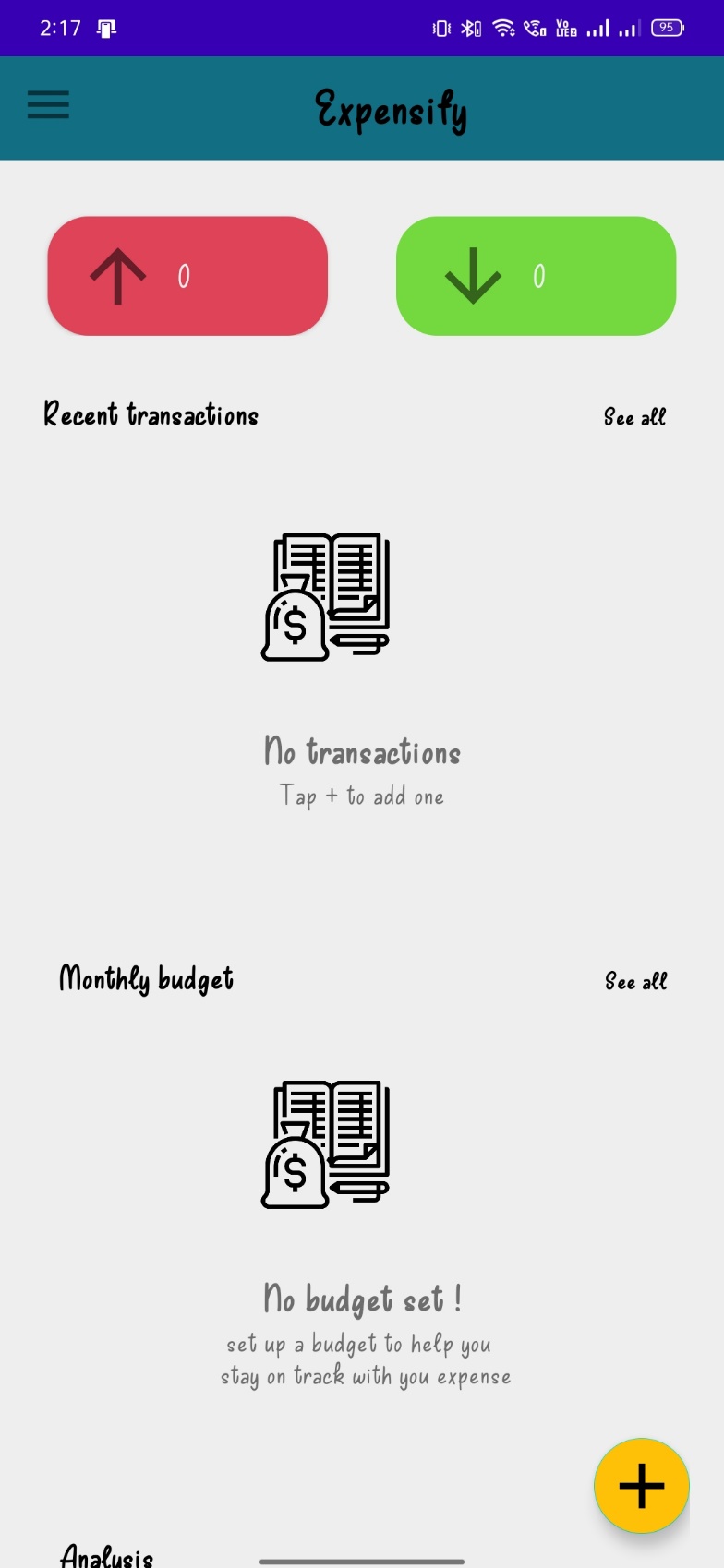
****

**1.Splash Screen for Expense tracking activity.(fig 1)**

**Sample code for splash screen🡪**

**package com.myapp.Expensify;//app package  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
  
*//import androidx.activity.EdgeToEdge;*import androidx.appcompat.app.AppCompatActivity;  
  
import com.example.Expensify.R;  
  
public class SplashScreen extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle *savedInstanceState*) {  
 super.onCreate(*savedInstanceState*);  
*// EdgeToEdge.enable(this);* setContentView(R.layout.*activity\_splash\_screen*);  
*// This is for compat view  
// ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
// Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
// v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
// return insets;  
// });* new Handler().postDelayed(new *Runnable*() {  
 @Override  
 public void run() {  
 startActivity(new Intent(SplashScreen.this, MainActivity.class));  
 finish();  
 }  
 },2500);  
 }  
}**

* After few milliseconds the **dashboard** activity is loaded. In the dashboard there are different buttons for different options user will be able to make changes according to his need.
* If the user downloads the app for the first time then empty dashboard is visible. And there is scroll bar by which user can see different chart view



**2.Dashboard Screen(MainActivity) for Expense tracking activity.(fig 2)**

**Sample code for dashboard(Main activity)--🡪**

package com.myapp.Expensify;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.cardview.widget.CardView;  
import androidx.core.view.GravityCompat;  
import androidx.drawerlayout.widget.DrawerLayout;  
import androidx.recyclerview.widget.LinearLayoutManager;  
  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity implements View.*OnClickListener* {  
 DrawerLayout dl;  
 ImageView iv,empty\_content,empty\_budget;  
 FloatingActionButton fap1;  
 *Animation* fapopen,fapclose;  
 TextView textView;  
 Boolean isOpen=false;  
 RecyclerView recyclerView,show\_data\_budget;  
 ArrayList<Integer> amount;  
 ArrayList<String> date;  
 ArrayList<String> type;  
 *Cursor* cursor,cursor1,cursor2,budgetcursor;  
 int i=0;  
 Button expense, income;  
 int expens\_amt = 0;  
 int income\_amt = 0;  
 LinearLayout linearLayout,HideLayout\_budget;  
 ArrayList<Integer> image;  
 ArrayList<String> paymentmode,transaction\_id,spinner,note;  
 TextView getTextView,view,addhomebudget,no\_budget,tap\_budget;

All transactionTextView

MyTransactionHomeAdapter myTransactionHomeAdapter = new MyTransactionHomeAdapter(spinner, note, transaction\_id,paymentmode,amount,date,type,image, MainActivity.this);

if(myTransactionHomeAdapter.getItemCount() == 0) {

linearLayout.setPadding(0,0,0,150);

empty\_content.setVisibility(View.VISIBLE);

empty\_content.setImageResource(R.drawable.calculate);

empty\_content.setAdjustViewBounds(true);

getTextView.setVisibility(View.VISIBLE);

view.setVisibility(View.VISIBLE);

linearLayout.setPadding(0,0,0,150);

}

recyclerView.setAdapter(myTransactionHomeAdapter);

recyclerView.setLayoutManager(new LinearLayoutManager(MainActivity.this));

// Monthly Budget

while (budgetcursor.moveToNext()){

budgetid.add(budgetcursor.getString(0));

budgetdate.add(budgetcursor.getString(1));

budgetamount.add(budgetcursor.getString(2));

totalamount.add(budgetcursor.getInt(3));

}

BudgetAdapter budgetAdapter =new BudgetAdapter(MainActivity.this,budgetid,budgetdate,budgetamount,totalamount);

if(budgetAdapter.getItemCount() == 0){

HideLayout\_budget.setPadding(0,0,0,150);

empty\_budget.setVisibility(View.VISIBLE);

empty\_content.setImageResource(R.drawable.budget);

empty\_budget.setAdjustViewBounds(true);

no\_budget.setVisibility(View.VISIBLE);

tap\_budget.setVisibility(View.VISIBLE);

HideLayout\_budget.setPadding(0,0,0,150);

}

show\_data\_budget.setAdapter(budgetAdapter);

show\_data\_budget.setLayoutManager(new LinearLayoutManager(MainActivity.this));

// month Analysis

month\_card.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent();

intent.setClass(MainActivity.this,MonthAnalysis.class);

startActivity(intent);

}

});

year Analysis

year\_card.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent();

intent.setClass(MainActivity.this,YearAnalysis.class);

startActivity(intent);

}

});

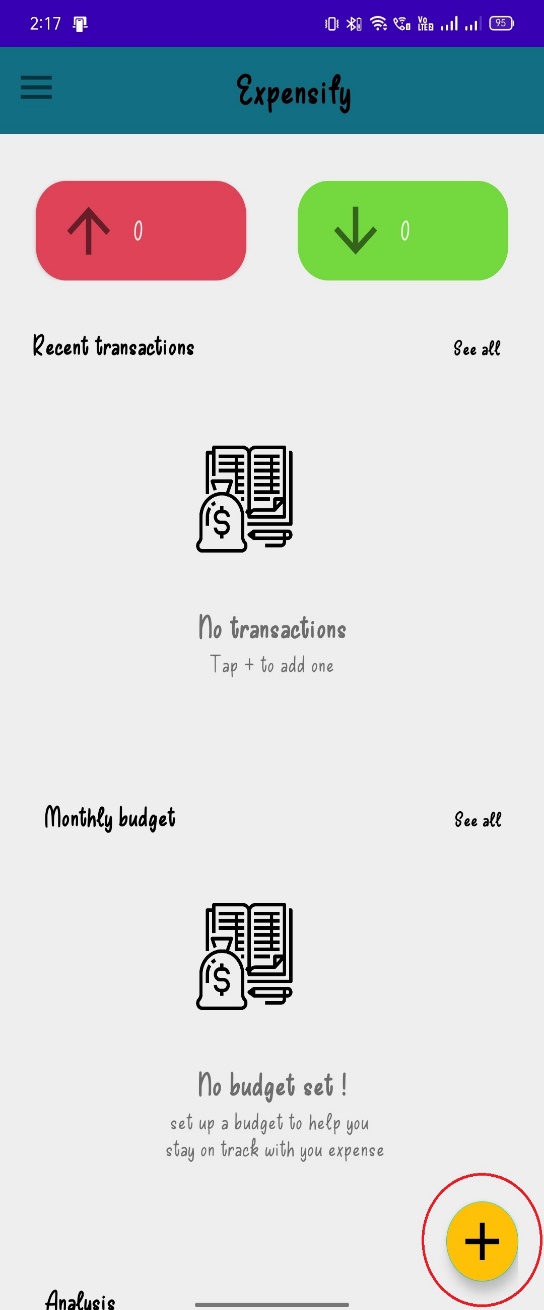
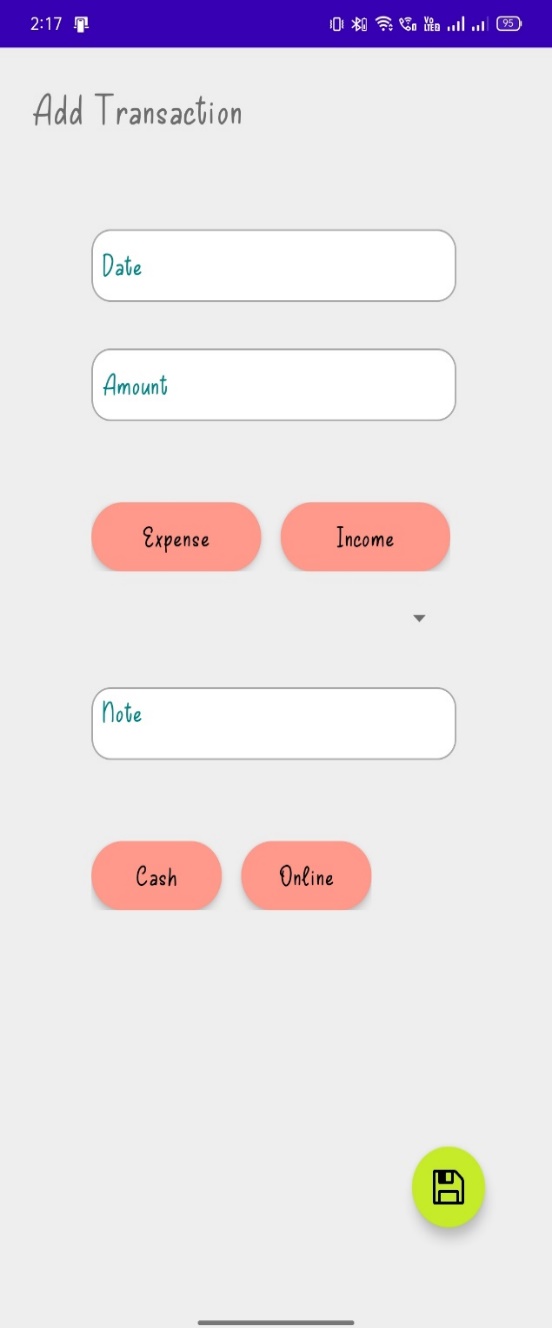
fap1.setOnClickListener(new View.OnClickListener() {

@Override

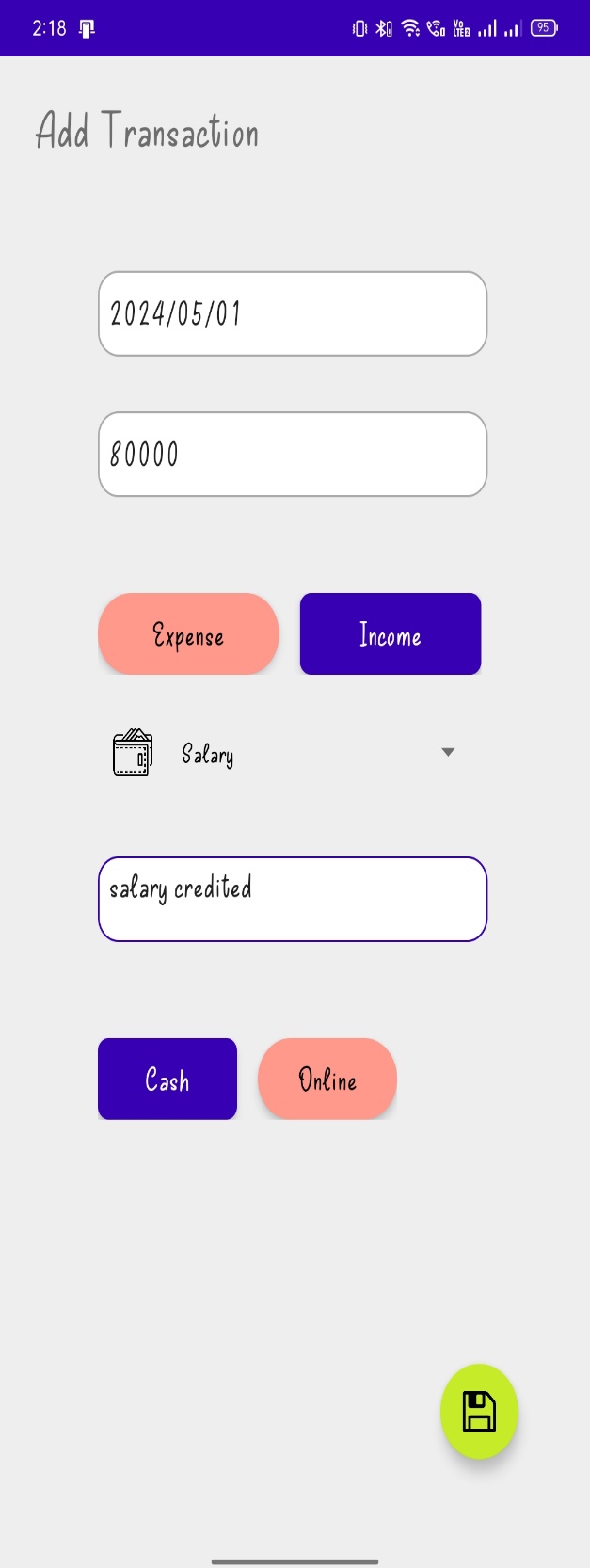
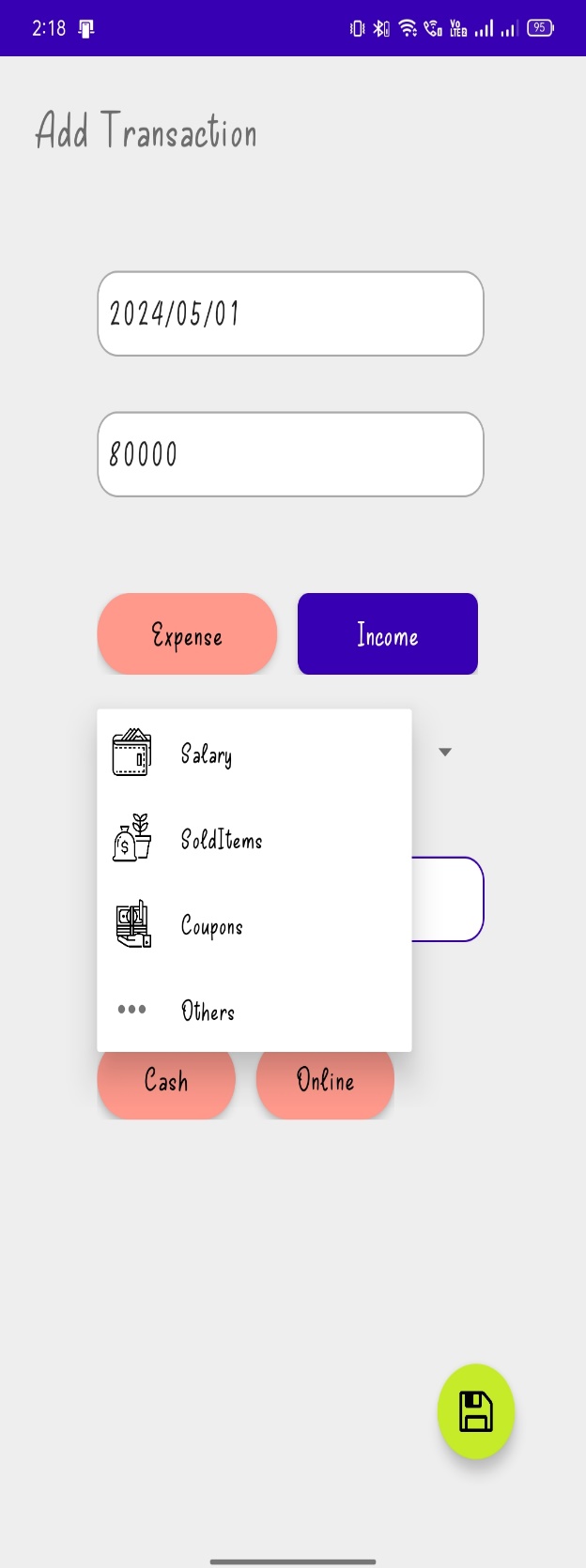
public void onClick(View v) {

//this is sample code only not the original one.

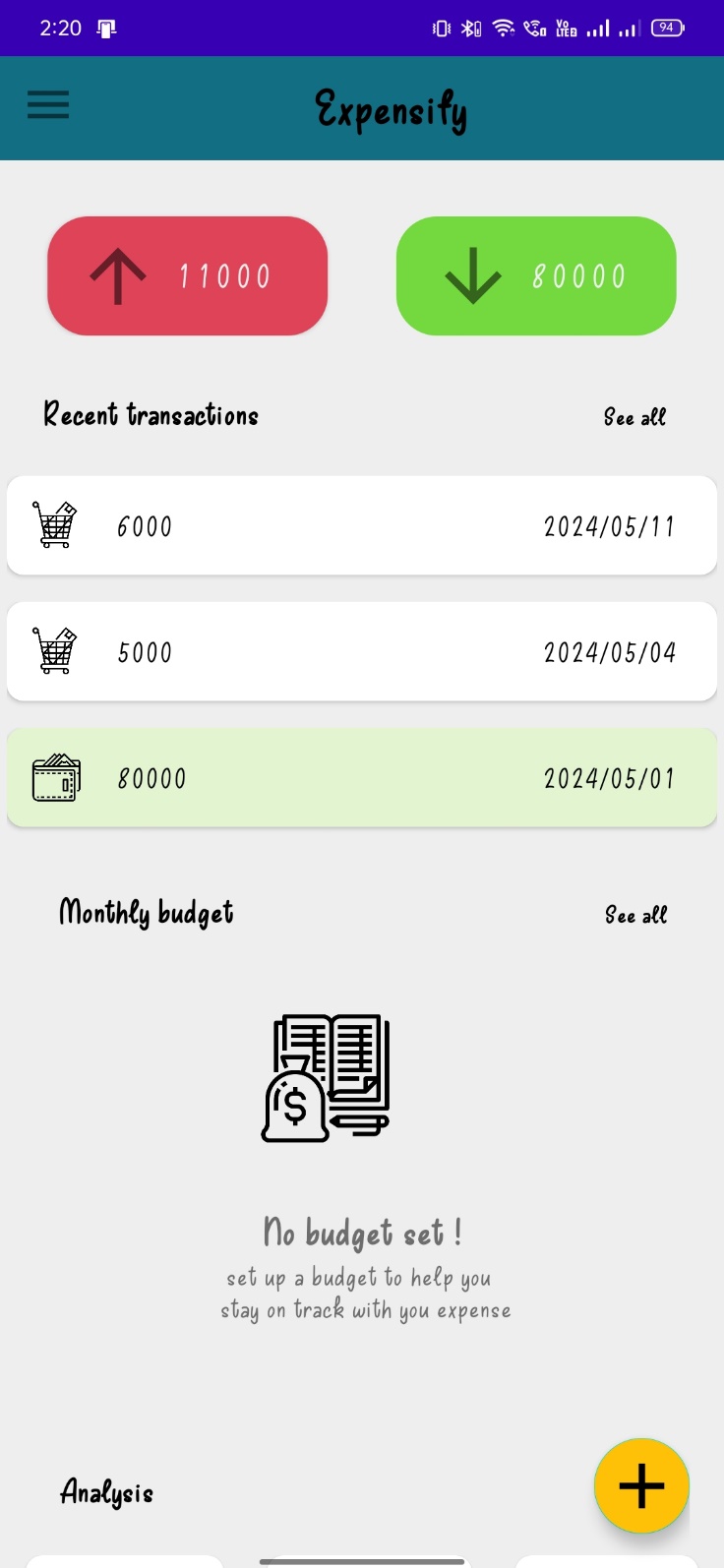
* The user can add the transaction by clicking **(+)** on thedashboard and a new window pops up and user can give the data related to the transaction.
* After entering the details press on the **save** button then transaction will be saved.

* The user can select different catagories.

**** 

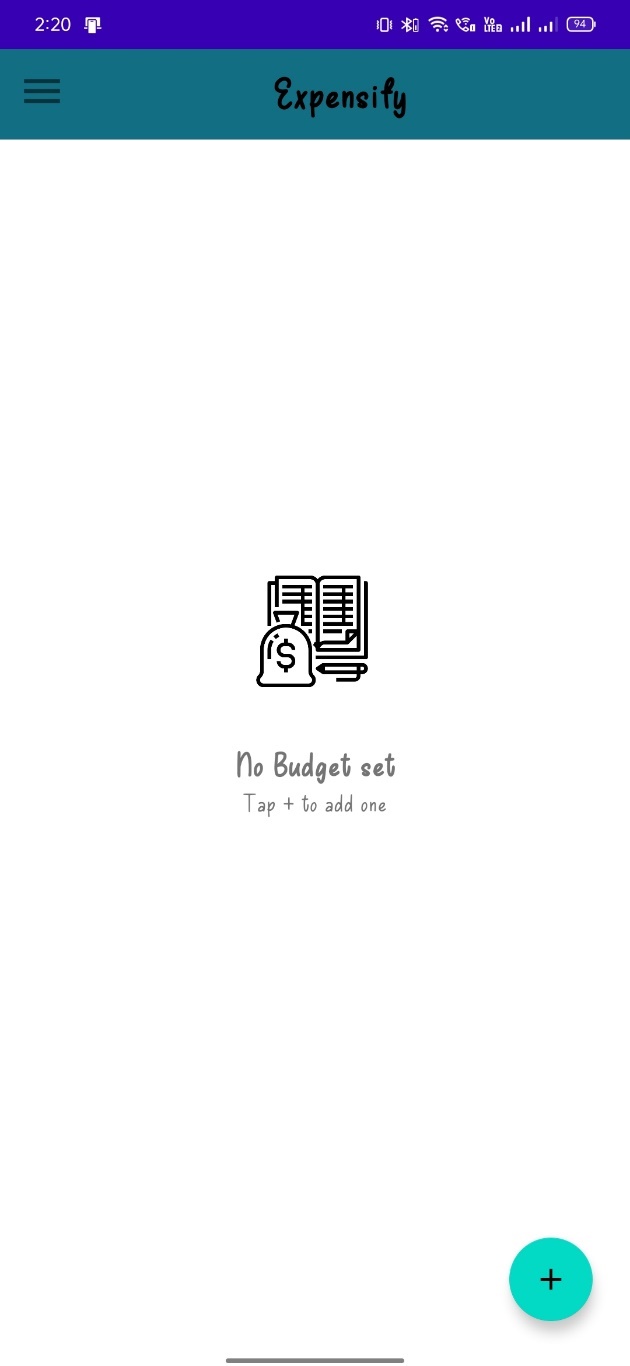
1. By Continue Adding Transactions: Repeat the above steps to add more transactions as needed. The Expense Tracker app makes it easy to input and manage multiple transactions to keep your financial records up-to-date.

****

**//sample code**

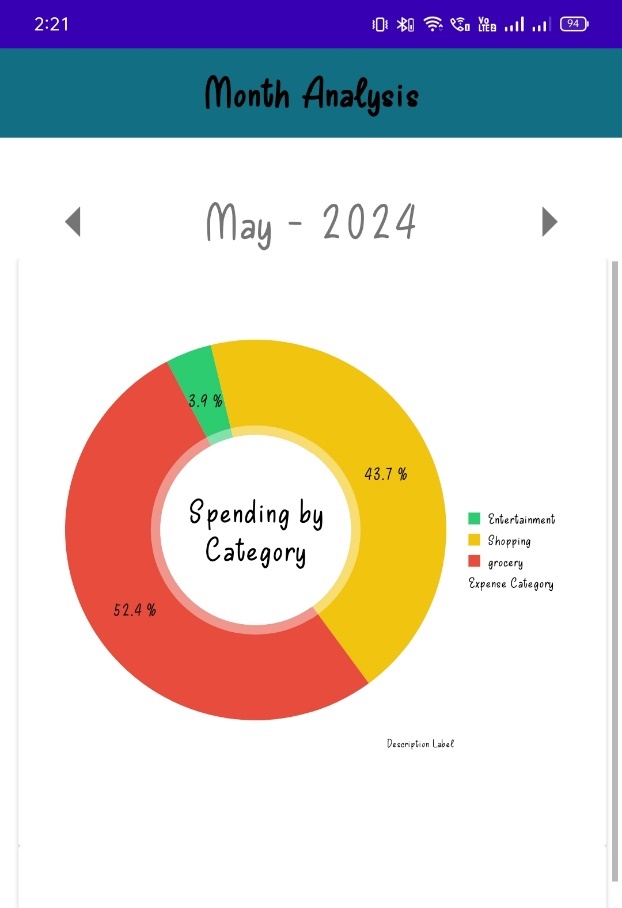
public class AllTransactions extends AppCompatActivity {  
 DrawerLayout dl;  
 RecyclerView recyclerView;  
 ArrayList<String> dates,amount,typeoftransaction,spinner,note,paymentmode;  
 String dates1,amount1,typeoftransaction1,spinner1,note1,paymentmode1,bd\_date;  
 int transactionid1,image1,remain\_bef\_delete,remain\_aft\_delete,total\_budget,budgetid;  
 ArrayList<Integer> transactionid,image,bid,remain\_amt,total\_amt;  
 ArrayList<String> bt\_date;  
 dbHelper myDB;  
 FloatingActionButton fab1;  
 ImageView imageView;  
 TextView textView;  
 boolean flag=true;  
 MyTransactionItemAdapter myTransactionItemAdapter;  
 @Override  
 protected void onCreate(Bundle *savedInstanceState*) {  
 super.onCreate(*savedInstanceState*);  
 setContentView(R.layout.*activity\_all\_transactions*);  
 dl = findViewById(R.id.*drawer\_layout*);  
 recyclerView = findViewById(R.id.*transaction*);  
 fab1 = findViewById(R.id.*addtrans*);  
 imageView = findViewById(R.id.*imgtrans*);  
 textView = findViewById(R.id.*txttrans*);  
 myDB = new dbHelper(AllTransactions.this);  
 transactionid = new ArrayList<>();  
 dates = new ArrayList<>();  
 amount = new ArrayList<>();  
 typeoftransaction = new ArrayList<>();  
 spinner = new ArrayList<>();  
 note = new ArrayList<>();  
 paymentmode = new ArrayList<>();  
 image = new ArrayList<>();  
 bid = new ArrayList<>();  
 bt\_date = new ArrayList<>();  
 remain\_amt = new ArrayList<>();  
 total\_amt = new ArrayList<>();  
 displayData();  
 fab1.setOnClickListener(new View.*OnClickListener*() {  
 @Override  
 public void onClick(View *v*) {  
 Bundle bundle = new Bundle();  
 bundle.putBoolean("key", true);  
 Intent i = new Intent();  
 i.putExtras(bundle);  
 i.setClass(AllTransactions.this,addincome.class);  
 i.setFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  
 startActivity(i);  
 }  
 });  
  
 myTransactionItemAdapter = new MyTransactionItemAdapter(this,transactionid,dates,amount,typeoftransaction,spinner,note,paymentmode,image);  
 recyclerView.setAdapter(myTransactionItemAdapter);  
 recyclerView.setLayoutManager(new LinearLayoutManager(AllTransactions.this));  
  
 new ItemTouchHelper(new ItemTouchHelper.*SimpleCallback*(0,ItemTouchHelper.*LEFT* ) {  
 @Override  
 public boolean onMove(@NonNull RecyclerView *recyclerView*, @NonNull RecyclerView.*ViewHolder viewHolder*, RecyclerView.*ViewHolder target*) {  
 return false;  
 }  
  
 @Override  
 public void onSwiped(@NonNull RecyclerView.*ViewHolder viewHolder*, int *direction*) {  
 int data = (Integer) *viewHolder*.itemView.getTag();  
 final int position = *viewHolder*.getLayoutPosition();  
 flag = true;  
 transactionid1 = transactionid.remove(position);  
 dates1 = dates.remove(position);  
 amount1 = amount.remove(position);  
 typeoftransaction1 = typeoftransaction.remove(position);  
 spinner1 = spinner.remove(position);  
 note1 = note.remove(position);  
 paymentmode1 = paymentmode.remove(position);  
 image1 = image.remove(position);  
 myTransactionItemAdapter.setCards(transactionid,dates,amount,typeoftransaction,spinner,note,paymentmode,image);  
 myTransactionItemAdapter.notifyItemRemoved(position);  
 Snackbar snackbar = Snackbar.*make*(recyclerView,"Transaction is Deleted",Snackbar.*LENGTH\_LONG*);  
 snackbar.setAction("UNDO", *v* -> { flag = false;  
 transactionid.add(position,transactionid1);  
 dates.add(position,dates1);  
 amount.add(position,amount1);  
 typeoftransaction.add(position,typeoftransaction1);  
 spinner.add(position,spinner1);  
 note.add(position,note1);  
 paymentmode.add(position,paymentmode1);  
 image.add(position,image1);  
 myTransactionItemAdapter.setCards(transactionid,dates,amount,typeoftransaction,spinner,note,paymentmode,image);  
 myTransactionItemAdapter.notifyItemInserted(position); });  
 **Budget Management** **:**

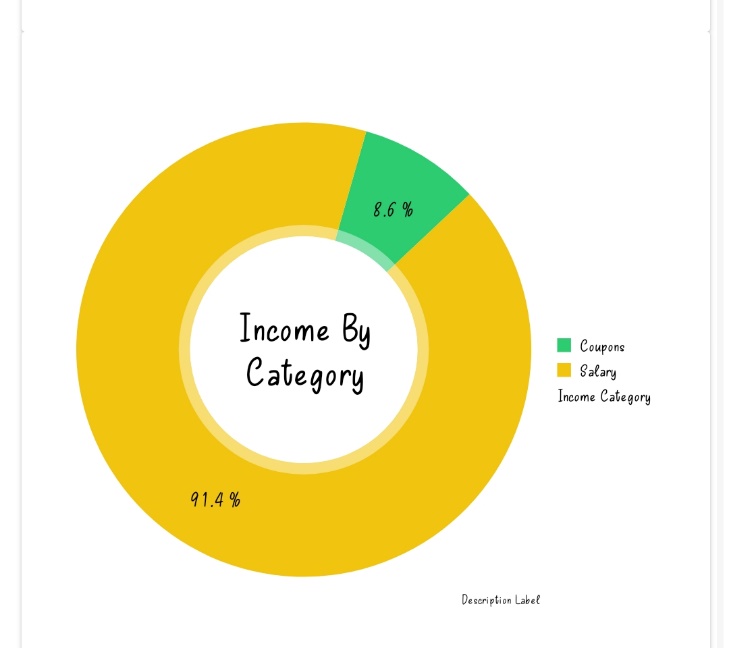
* User can set a budget for a month and according to the budget he’ll be aware of the transaction and the application will be sending the notifications timely.
* User can set budget by pressing **+** and can select date and set budget.

****

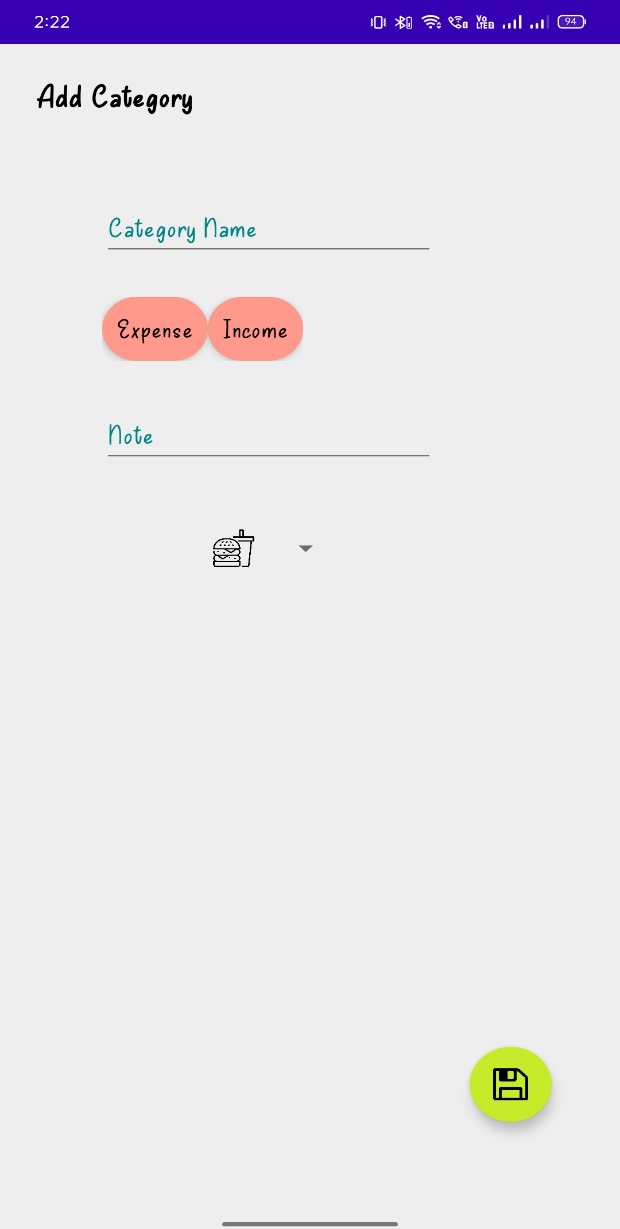
**Pie Chart view:**

* User can view the transactions in custom view according to the transaction the data is aligned in pie chart and because of **Anychart** library the data is fetched from database and displayed.

****



1. **Add a New Category**: Look for the "Add Category" button and tap on it to create a new category.
2. **Enter Category Details**: On the category creation screen, you'll see a field to input the name of the new category.
3. **Save the Category**: After entering the category name, tap on the "Save" or "Add" button to save the new category.
4. **Confirmation**: You may receive a confirmation message indicating that the category has been successfully added.
5. **View Updated Category List**: Optionally, you can navigate back to the categories screen to see the newly added category listed along with existing ones.
6. **Edit or Delete Category (Optional)**: If needed, you can edit or delete the category later. Most apps allow you to update category names or delete categories that are no longer needed.

****

**Settings:**