A Project Report

On

Expense Manager

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

B.Tech

Session 2025-26 in

Computer Science and Engineering

By

RATNESH RAI

22SCSE1011240

MD. TABISH ZUBAIR

22SCSE1011255

Under the guidance of

Dr. Savita Kumari

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

MAY 2025

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project, entitled "EXPENSE MANAGER" in partial fulfillment of the requirements for the award of the B. Tech. (Computer Science and Engineering) submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of Feb, 2025 to Jun 2025, under the supervision of Dr. Savita Kumari, Department of Computer Science and Engineering, of School of Computing Science and Engineering, Galgotias University, Greater Noida.

The matter presented in this project has not been submitted by me for the award of any other degree of this or any other institution.

Name: RATNESH RAI

Admission No.: 22SCSE1010083

Name: MD. TABISH ZUBAIR

Admission No.: 22SCSE1010083

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Savita Kumari

Professor

CERTIFICATE

This is to certify that the Project Report entitled "EXPENSE MANAGER" which is submitted by Ratnesh Rai and Md. Tabish Zubair in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science and Engineering of School of Computing Science and Engineering, Galgotias University, Greater Noida, India is a record of the candidate's own work carried out by him under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Signature of Examiner(s)   Signature of Supervisor(s)

Signature of Program Chair   Signature of Dean

Date: June, 2025

Place: Greater Noida

ACKNOWLEDGEMENT

It gives me great pleasure to present the report of my B.Tech project undertaken during the final year. I owe special gratitude to Dr. Savita Kumari, Department of Computer Science & Engineering, Galgotias University, Greater Noida, India for his constant support and guidance throughout the course of my work. His sincerity and perseverance have been a constant source of inspiration.

I thank all the faculty members of the department for their assistance and cooperation. Lastly, I would like to thank my friends and peers for their encouragement during the project.

Signature:

Name: Ratnesh Rai

Admission No.: 22SCSE1011240

Name: Md. Tabish Zubair

Admission No.: 22SCSE1011255

Date: May, 2025

ABSTRACT

Managing personal finances is a fundamental yet often overlooked skill in modern life. This project presents a mobile application, **Expense Manager**, designed to help users track their daily expenses, categorize spending, and maintain better financial discipline. Built using **Android Studio** with **Java**, the app allows users to add, view, and delete expense records stored in a local text file (expenses.txt) using internal storage.

The project emphasizes simplicity and offline functionality. It features user-friendly components like RecyclerView for listing entries, a FloatingActionButton for adding new expenses, and AutoCompleteTextView for selecting predefined categories. Through this application, users gain real-time insight into their financial behavior without relying on complex systems or internet connectivity.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 Problem Introduction

In the absence of organized tracking, individuals often lose control of their daily expenditures. Traditional methods such as notebooks or spreadsheets are either inconvenient or inefficient. A lightweight mobile application with offline support offers an optimal solution for day-to-day expense tracking.

1.2 Objectives

1. Develop a user-friendly app to log daily expenses.

2. Categorize expenses for better analysis.

3. Store data securely in internal storage.

4. Enable deletion of specific records.

5. Design a clean and intuitive user interface.

1.3 Scope of the Project

This app is intended for students, professionals, or anyone who wishes to manage personal finances efficiently on their Android device, even without internet access.

1.4 Organization of the Report

This report includes the introduction and objective behind the project (Chapter 1), requirement specification (Chapter 2), design and system diagrams (Chapter 3), implementation and output (Chapter 4), and finally the conclusion and future scope (Chapter 5).

CHAPTER 2 PROPOSED SYSTEM

2.1 Overview

This project proposes a lightweight, offline-only Android app for daily expense tracking. It stores user data using text file I/O and offers clean UI navigation using RecyclerView and FloatingActionButton.

2.2 Features

- Add new expense entries with amount, category, note, and timestamp
- View all saved expenses in a scrollable list
- Delete individual expenses
- Offline functionality using local file storage
- AutoComplete input for category selection

2.3 Advantages

- Works without internet
- Fast and responsive
- Minimal dependencies
- Simple yet effective interface for non-tech-savvy users

2.4 Hardware Requirements

- Android device with Android 6.0 (Marshmallow) or higher
- Minimum 2GB RAM

2.5 Software Requirements

- Android Studio (latest version)
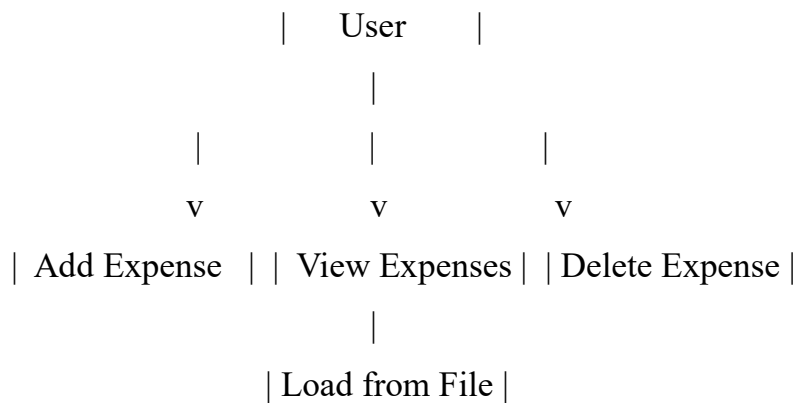- Java (Android SDK)
- Gradle build system

CHAPTER 3 SYSTEM DESIGN

6.1 Architecture Overview

The app follows a basic **Activity-based architecture**, with MainActivity handling expense viewing and deletion, and AddExpenseActivity managing new entries.

6.2 Use Case Diagram

- **Actor**: User
- **Use Cases**:
  - Add Expense
  - Delete Expense

```
                    |    User     |
                          |
        |              |            |
        v              v            v
| Add Expense  | | View Expenses | | Delete Expense |
                          |
                  | Load from File |
```
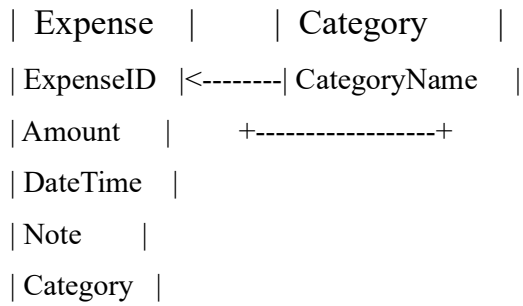
6.3 Data Flow

User Input → AddExpenseActivity → File Write → MainActivity → RecyclerView → User
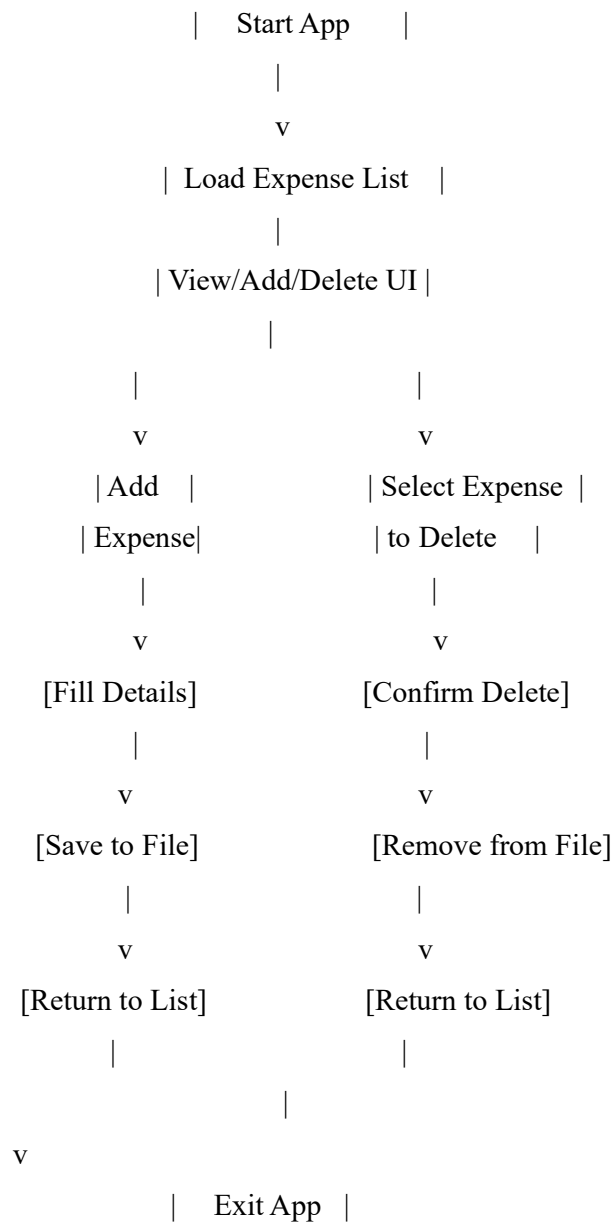
6.4 Database Design

There is no traditional SQL database. Instead, a text file (expenses.txt) stores each record in the format: <Category>|<Amount>|<DateTime>|<Note>

6.4 ER Diagram

```
|  Expense   |        | Category      |
| ExpenseID  |<--------| CategoryName  |
| Amount     |       +------------------+
| DateTime   |
| Note       |
| Category   |
```

6.5 Activity Diagram

```
                          |   Start App    |
                                 |
                                 v
                       |  Load Expense List   |
                                 |
                        | View/Add/Delete UI |
                                 |
                     |                    |
                     v                    v
                 | Add    |          | Select Expense  |
                 | Expense|          | to Delete       |
                     |                    |
                     v                    v
               [Fill Details]        [Confirm Delete]
                     |                    |
                     v                    v
               [Save to File]        [Remove from File]
                     |                    |
                     v                    v
              [Return to List]       [Return to List]
                     |                    |
                                 |
        v
                       |   Exit App   |
```

CHAPTER 4 Implementation and Results

7.1 Technologies Used

- Java (Android SDK)
- XML (UI Layouts)
- RecyclerView
- Internal Storage (File I/O)
- Material Design Components

7.2 Major Classes and Activities

a) MainActivity.java

- Initializes RecyclerView and loads data
- Allows deletion of expense records via deleteExpenseFromFile()
- Loads data from file using loadExpensesFromFile()
- Handles onActivityResult() to update UI after new entry

b) AddExpenseActivity.java

- Gathers user input
- Validates fields (amount, category)
- Saves new record to expenses.txt with timestamp
- Returns RESULT_OK to MainActivity

c) ExpenseAdapter.java *(Assumed)*

- Custom RecyclerView adapter to display expense list
- Implements delete callback via interface

d) Expense.java

- Plain Java class to model the Expense object with category, amount, datetime, and note

7.3 Sample Expense Entry

`Food | 250 | 01-06-2025 13:22:14 | Lunch at cafe`

7.4 Results and Output Screens

- Application loads and lists expenses successfully

- Records are written to and deleted from expenses.txt correctly

- UI remains responsive with multiple entries

- No crash or data corruption observed in tests
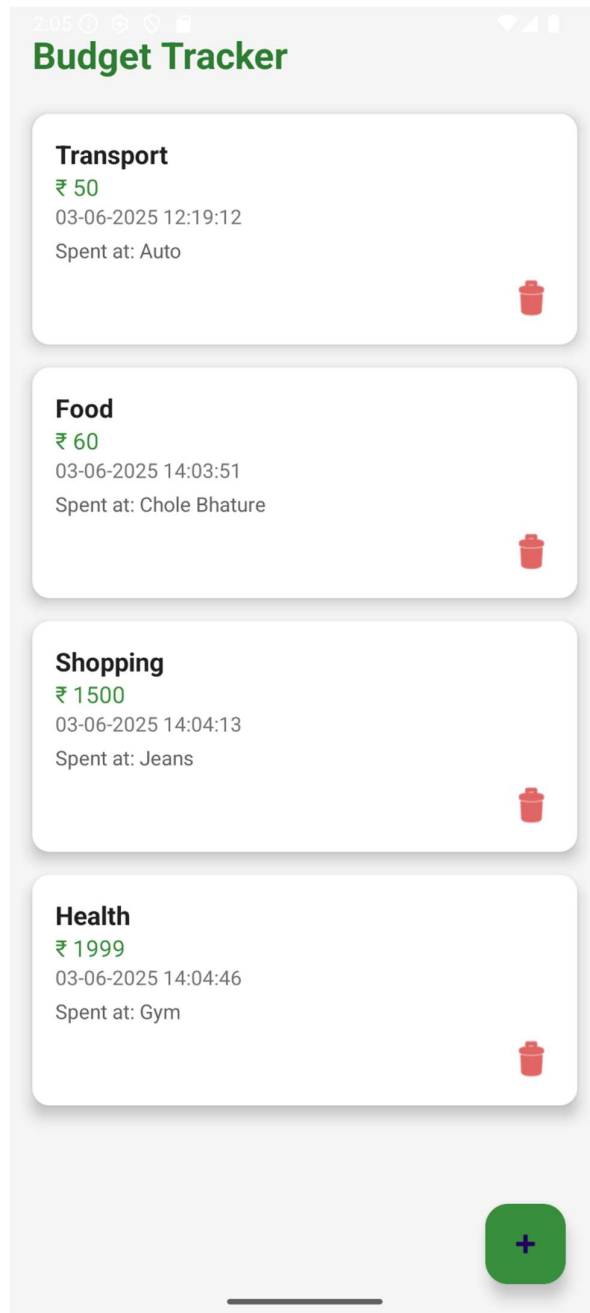
CHAPTER 5 CONCLUSION AND FUTURE SCOPE

This Expense Manager app successfully meets the goal of providing a minimal and efficient solution for personal expense tracking. It ensures offline usability, simplicity, and ease of access without depending on third-party APIs or cloud storage.

Future Enhancements:

- Add monthly budget limits with warnings
- Implement summary statistics and visual reports (charts)
- Use SQLite or Room DB for scalable storage
- Export data to Excel or PDF
- Add login for multiple user support
- Cloud backup/sync using Firebase

Appendix

Key Code Snippets

a) Saving Expense Entry

```
CopyEdit
File file = new File(getFilesDir(), "expenses.txt");
FileWriter writer = new FileWriter(file, true);
writer.append(record);
writer.close();
```

b) Loading Expense List

```
CopyEdit
BufferedReader reader = new BufferedReader(new FileReader(file));
String line;
while ((line = reader.readLine()) != null) {
    String[] parts = line.split(" \\| ");
```

```
    expenses.add(new Expense(...));
}
```

c) Deleting a Specific Entry

CopyEdit

```
if (category.equals(toDelete.getCategory().trim()) &&
    amount.equals(toDelete.getAmount().trim()) &&
    dateTime.equals(toDelete.getDateTime().trim()) &&
    note.equals(toDelete.getNote().trim())) {
    continue; // Skip writing this line
}
```

References

- Android Developer Docs – https://developer.android.com
- Stack Overflow
- TutorialsPoint Android File I/O
- Material Design Documentation
- Java I/O API