

MINI-PROJECT REPORT
ON

Expense Tracker

Submitted

In Partial Fulfillment of Requirement for the degree of

Bachelor of Technology

In

Information Technology

By

Abhinash Kumar Singh (2204220130001)

Manish Yadav (2204220130034)

Under the Supervision of

Mr. Vivek Singh



BANSAL INSTITUTE OF ENGINEERING AND TECHNOLOGY

NH24, Sitapur Road, Near Sewa Hospital, Lucknow (226201)

Session 2024-25



CERTIFICATE

Certified that Manish Yadav (2204220130034), Abhinash Kumar Singh (2204220130001), has carried out the research work presented in this report entitled “Expense Tracker” for the award of Bachelor of Technology from Uttar Pradesh Technical University, Lucknow under my supervision. The report embodies results of original work, and studies are carried out by the student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Place : Lucknow

Mr. Vivek Singh

Assistant Professor IT

ABSTRACT

The "Expense Tracker" is a Java-based desktop application designed to streamline personal financial management by enabling users to efficiently track and organize their expenses. Built with Java Swing for an intuitive graphical user interface and enhanced by the FlatLaf library for a modern and visually appealing design, the application delivers a seamless user experience. The SQLite database is utilized for secure and reliable data storage, ensuring persistence across user sessions.

This application offers several core features to enhance financial oversight, including the ability to add new expenses, categorize them, and sort expenses based on predefined categories. Additionally, it provides users with a comprehensive summary of their total expenditures, empowering them to make informed financial decisions. The "Expense Tracker" is easy to deploy and requires minimal prerequisites, including Java 17 or later and Maven for project building. With its user-friendly interface and robust functionality, the application caters to individuals seeking a practical solution for managing personal finances efficiently. By leveraging modern Java technologies, the Expense Tracker bridges the gap between financial tracking and usability, Tool for everyday financial managment. In conclusion, the "Expense Tracker" is a powerful yet simple tool for managing personal finances. By leveraging modern Java technologies, it bridges the gap between financial tracking and usability, making it an indispensable companion for individuals seeking better control over their spending habits and financial planning.

ACKNOWLEDGEMENT

We thank the almighty for giving us the courage and perseverance in completing the main project. This project itself is acknowledgements for all those people who have given us their heartfelt co-operation in making this project a grand success. Like most effective endeavors, preparing this project was a collaborative effort. I owe a great debt to many individuals who helped me in successful completion of this project. We are greatly indebted to our project guide **Mr. Vivek singh** for providing valuable guidance at every stage of this project work. We are profoundly grateful towards the unmatched services rendered by him.

Our special thanks to Head of Department **Mr. Abhishek Kumar Saxena** and all the faculty of Information Technology peers for their valuable advises at every stage of this work. Last but not least we would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the main project.

NAME: MANISH YADAV

ROLL NO: (2204220130034)

NAME: ABHINASH KUMAR SINGH

ROLL NO: (2204220130001)

CONTENTS

CHAPTER 1- INTRODUCTION	6-8
1.1 AIM	6
1.2 OBJECTIVE	6
1.3 PROBLEM STATEMENT	6
1.4 BACKGROUND	7
1.5 IMPLICATION	7-8
CHAPTER 2- LITERATURE REVIEW	09
CHAPTER 3- TOOLS & METHODOLOGY	10-14
3.1 REQUIRED TOOL	10
3.2 APPORACH USED	11
3.3 DESIGN	12
3.4 ENTITY RELATIONSHIP DIAGRAM	13-14
3.41 Entities	13
3.42 Diagram Description	14
3.43 Example Representation	14
CHAPTER-4 EXPERIMENTAL RESULT AND ANALYSIS	15-24
4.1 PROJECT USER INTERFACE	15
4.2 WORKING	15-24
4.21 User Inputs Data	16
4.22 Validation	17-18
4.23 Data Storage	19
4.24 Sorting Expenses Based on Categories	20-21
4.25 View Detailed Summaries of Total Expenses	22-24
CHAPTER-5 CURRENT &FUTURE SCOPE	25-26
5.1 CURRENT SCOPE	25
5.2 FUTURE SCOPE	26
CHAPTER-6 CONCLUSION	27-28
CHAPTER-7 REFERENCES	29

CHAPTER-1

INTRODUCTION

1.1 AIM

The primary aim of the "Expense Tracker" project is to develop a user-friendly application that allows individuals to record, categorize, and analyze their daily expenses. By providing clear insights into spending habits, the application empowers users to make informed financial decisions and improve budgeting practices. It aims to simplify expense tracking through an intuitive interface built with Java Swing and a reliable SQLite database for data storage. The project also focuses on enhancing user experience with a modern design powered by Flatleaf. Ultimately, the goal is to create a practical tool for better financial management.

1.2 OBJECTIVE

The objectives of the "Expense Tracker" project are as follows:

1. To develop a robust application for recording and managing daily expenses efficiently.
2. To enable categorization and sorting of expenses for better financial analysis.
3. To provide users with a clear summary of their total expenditures, helping them identify spending patterns.
4. To ensure data persistence and security using SQLite as the backend database.
5. To deliver a modern and user-friendly interface using Java Swing and FlatLaf.
6. To create a scalable solution that can be enhanced with additional features like analytics and multi-platform support in the future.

1.3 PROBLEM STATEMENT

The Expense Tracker project directly addresses the inefficiencies and challenges associated with traditional methods of financial tracking. Below are the key issues resolved by this project:

1. **Elimination of Manual Errors and Delays**
Traditional systems like handwritten records or spreadsheets are prone to errors and require significant manual effort to maintain and update. The application automates this process, ensuring accurate data entry, validation, and storage, saving users time and reducing the chances of errors.
2. **Simplified Data Management**
Users often struggle with organizing and retrieving financial data in traditional systems. The project solves this by integrating SQLite, which securely stores all records in an organized and easily retrievable manner. Users can view their financial history with just a few clicks.
3. **Category-Based Organization**
The Expense Tracker groups expenses into predefined categories such as Food, Housing, Utilities, and Transportation. This structure provides better visibility into spending habits, helping users identify areas where they may need to reduce expenses.
4. **Enhanced Data Analysis**
Traditional methods lack tools for analyzing spending patterns or summarizing expenses. The project includes features like summary generation, total expense calculation, and graphical representation of spending trends, which empower users to make informed financial decisions.
5. **Portability and Accessibility**
Unlike physical records or static spreadsheets, the digital nature of this project allows users to access their expense data from their systems easily. The user-friendly interface, built with Java Swing, ensures accessibility for users of all experience levels.
6. **Time Efficiency**
Recording, sorting, and summarizing expenses in traditional systems can be time-intensive. This project accelerates the entire process by streamlining data input, storage, and display.

By addressing these challenges, the Expense Tracker project provides an effective and modern solution for managing personal and household finances. It not only replaces outdated methods but also introduces features that empower users to take control of their financial habits with minimal effort.

1.4 BACKGROUND

In life cycle of human after birth the need of materials and belongings is obvious. In order to fulfill our needs and desire we buy goods. The rule of earth is that you must have money in order to buy desired good. So in this way the process of earning and spending goes on in our life. People in order to track their expenses use traditional paper system to keep the record of their income and expenditures. This type of traditional system is burdensome and takes more time. So there must be a management system which must help us to manage our daily earnings and expenses easily, and also helps us to analyze records efficiently. So we figured out a way to eliminate the traditional system with digital, portable, easier and simple way to record these data in just few clicks with our application.

In today's digital age, where technology plays a pivotal role in simplifying daily tasks, it is essential to adopt tools that make managing personal finances seamless. Traditional methods, though effective in the past, cannot keep up with the speed and complexity of modern lifestyles. Tracking expenses and income manually often leads to errors, data loss, and difficulty in generating insightful reports.

With the rise of financial management applications, individuals now have access to tools that not only save time but also provide valuable insights into spending patterns. Digital systems ensure accuracy, enable portability, and offer features like data visualization and categorization. These systems are designed to empower users with actionable data, helping them make informed financial decisions.

The Expense Tracker application is built to address these challenges effectively. It serves as a comprehensive solution for individuals looking to replace outdated methods with a user-friendly, reliable, and efficient system. By integrating features like real-time categorization, secure data storage, and insightful reporting, this application becomes an indispensable tool for financial management. It is not just a tool to record data but a gateway to better financial planning and smarter budgeting for individuals of all ages.

This shift from traditional to digital aligns with the broader trends of modernization and sustainability. By reducing reliance on paper and improving data accessibility, the application contributes to a greener, more organized future.

1.5 IMPLICATION

Understanding our spending habit is a challenging work unless we keep the proper record of each and every transaction we perform. Personal Expense Tracker (PET) is a way to analyze our spending habit on certain time interval .PET is the easiest and most user friendly personal finance Android application. The system attempts to free the user with as much as possible the burden of manual calculation and to keep the track of the expenditure .Instead of keeping a dairy or a log of the expenses on the smartphones or laptops, this system enables the user to not just keep the track on the expenses but also to plan ahead keeping the past budget in mind. The simple fact is, by tracking our expenses we will be able to stick to a budget and therefore save money.The "Expense Tracker" is a desktop application developed in Java to help individuals manage and monitor their financial expenditures effectively. By leveraging Java Swing for an interactive user interface and SQLite for seamless data persistence, the application simplifies the process of tracking daily expenses, organizing them into categories, and providing insightful summaries. Designed for ease of use and scalability, it serves as a practical tool for personal financial management, catering to users with diverse needs and technical backgrounds.

This application offers several core features to enhance financial oversight, including the ability to add new expenses, categorize them, and sort expenses based on predefined categories. Additionally, it provides users with a comprehensive summary of their total expenditures, empowering them to make informed financial

decisions. By offering sorting and categorization features, the application enables users to identify spending patterns and areas where they can cut costs effectively.

The "Expense Tracker" stands out due to its simplicity, usability, and focus on essential functionalities without overwhelming users with unnecessary complexities. Its modern look and feel, powered by the FlatLaf library, ensures that the application is not only functional but also visually appealing, making it a pleasure to use for individuals from all age groups and technical backgrounds.

Technical Details

The application is built with Java Swing, ensuring a lightweight yet robust graphical interface. SQLite has been chosen as the database for its ease of integration and ability to handle data persistently. Maven is utilized for building and managing dependencies, simplifying the development and deployment process. The modular design of the project allows for easy maintenance and potential feature expansion in the future.

Use Cases

The "Expense Tracker" is ideal for individuals, freelancers, or small business owners who want to maintain control over their finances. It allows users to track daily expenses, monitor spending habits, and generate reports for budgeting purposes. By providing real-time insights into financial health, it helps users stay on top of their financial goals and avoid unnecessary debt.

Deployment and Prerequisites

Deploying the "Expense Tracker" is straightforward, requiring Java 17 or later and Maven for building the project. The application can be run on any operating system with Java support, making it accessible to a broad audience. With minimal setup, users can get started quickly and focus on tracking their finances. The "Expense Tracker" has significant potential for further development. Features like integration with online banking, multi-user support, and advanced analytics can be incorporated to enhance its utility. Additionally, the application can be adapted for mobile platforms, providing users with on-the-go access to their financial data.

In conclusion, the "Expense Tracker" is a powerful yet simple tool for managing personal finances. By leveraging modern Java technologies, it bridges the gap between financial tracking and usability, making it an indispensable companion for individuals seeking better control over their spending habits and financial planning.

CHAPTER-2

LITERATURE REVIE

We found various similar product that have already been developed in the market. Unlike all those products Personal Expense Tracker (ET) provides security and graphical results. We provide the users to enter their wish-list After any purchase. It generates to notify user about their timely expense. In order to complete our task, we used Android platform to build a portable handy product that can run in Android phone. We used Android studio to build our application. The major language we used for scripting was Java and XML for producing better layout. We used SQLite to implement database. On comparing our project with other similar applications we came to conclude with the following:

Apps Features	Expense Manager by Bishinews	Hello Expense by Alan L	Personal Expense Tracker
Ease of Use	Medium	Low	High
Backup/Restore	Yes	Yes	No
Expense/Income	Yes	Yes	Yes
Wish List	Yes	No	Yes
Decision Making	No	No	Yes
Notification	Yes	No	Yes
Focus Group	Personal	Finance & Business	Personal
Password Protected	No	No	Yes
Detailed Report	No	No	Yes

CHAPTER-3

TOOLS AND METHODOLOGY

3.1 REQUIRED TOOLS

1. Java

Java is the core programming language used in this project. It is an object-oriented, platform-independent language that enables seamless development of applications across different systems. The robust libraries and APIs in Java helped implement essential functionalities like event handling and data processing. Java's extensive community support and documentation made troubleshooting and feature enhancement efficient. The use of Java ensures the application's scalability and long-term maintenance.

2. Java Swing

Swing is a part of Java's Standard Library and was used to create the graphical user interface (GUI) for the application. It provides lightweight, platform-independent components that enable the development of a dynamic and interactive interface. Swing's features, such as customizability and event-driven programming, allowed the creation of responsive input forms and display panels. By leveraging its layout managers and built-in components, the application offers a user-friendly experience. The flexibility of Swing enabled the integration of modern styling with the help of external libraries like FlatLaf.

3. FlatLaf

FlatLaf is a modern look-and-feel library for Java Swing applications, designed to enhance the GUI's aesthetics. It provides a clean, flat design similar to contemporary software, which improves user engagement and satisfaction. The library seamlessly integrates with Swing components, requiring minimal changes to the codebase while delivering a professional appearance. FlatLaf supports dark mode, which can be optionally implemented for better user comfort. This tool also ensures the visual consistency of the application across different operating systems.

4. SQLite

SQLite is a lightweight database management system used to store and manage user expense data locally. It eliminates the need for a separate server, making the application self-contained and easy to deploy. The database was used to handle operations like adding, retrieving, and categorizing expense data. SQLite is highly efficient and reliable, which ensures fast read/write performance even with a growing dataset. Its simplicity in setup and use makes it ideal for desktop applications like the Expense Tracker.

5. Maven

Maven is a project management and build automation tool, essential for managing dependencies and building the project. It simplifies the setup by automatically downloading required libraries and ensuring version compatibility. Maven's lifecycle capabilities were used to compile, test, and package the application into a runnable JAR file. Its dependency management feature ensured that tools like FlatLaf and SQLite were seamlessly integrated into the project. Using Maven also standardized the development workflow, making it easier for others to contribute to or build upon the project.

3.2 APPROACH USED

Having a proper track to our expenses helps us to organize our finance which is actually a very difficult task. Looking through our expenses at regular interval allow us to review our spending habit Writing records of our expense in paper is a traditional way. In the world driven by technology, it is not a right measure or choice indeed. It is difficult to relate our present expenses with past when data need to be reviewed and it is even not secure.

So with this particular application Personal Expense Tracker, we have tried to feature several concepts that the user can fully track all of his/her expense as compared to present system in practice. Personal Expense Tracker is a technology driven approach where user have to log in to track their spending and earnings.

1. Requirement Analysis

- Identified the core functionalities, such as adding, categorizing, and viewing expenses.
- Gathered user requirements for features like intuitive UI and reporting.
- Considered scalability and performance for handling increasing data volumes.

2. Modular Design

- Followed a modular approach to separate concerns:
 - **Frontend (UI):** Developed using Java Swing with FlatLaf for modern styling.
 - **Backend (Logic):** Handled data processing and operations such as CRUD (Create, Read, Update, Delete).
 - **Database:** Designed SQLite tables for efficient data storage and retrieval.
- This separation made the application easier to debug and enhance.

3. Database Design

- Used ER modeling to structure the database with normalized tables for users, expenses, and categories.
- Ensured relationships between entities (e.g., User to Expense, Expense to Category) for data consistency.
- SQLite was chosen for its lightweight nature, suitable for local desktop applications.

4. Object-Oriented Programming (OOP) Principles

- Encapsulation: Kept data secure by defining clear boundaries for classes.
- Inheritance: Used base classes for common functionalities, minimizing redundancy.
- Polymorphism: Simplified dynamic operations like categorization and filtering.

5. Agile Development Practices

- Developed the project incrementally by implementing and testing one feature at a time.
- Gathered feedback after each stage to refine the application.
- Used iterative development to introduce new features such as dark mode and advanced reporting.

6. User-Centered Design

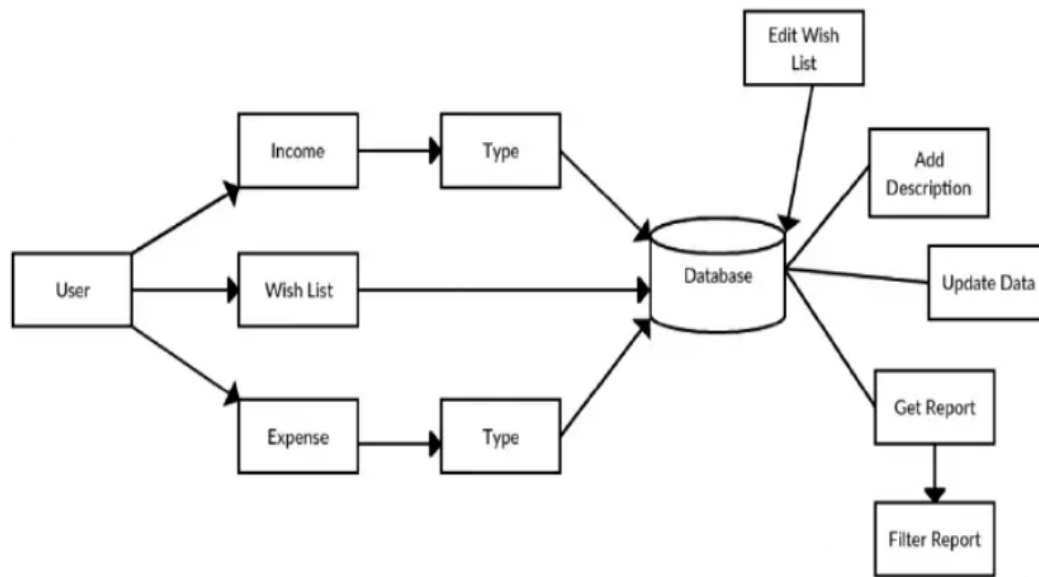
- Designed the UI with the user in mind, ensuring intuitive navigation and usability.
- Integrated FlatLaf to provide a visually appealing experience.
- Included error messages and validations to guide users and prevent incorrect input.

7. Testing and Optimization

- Conducted unit testing for individual components like database operations and UI responsiveness.

- Performed integration testing to ensure smooth interaction between modules.
- Optimized queries to enhance the performance of data retrieval for large datasets.

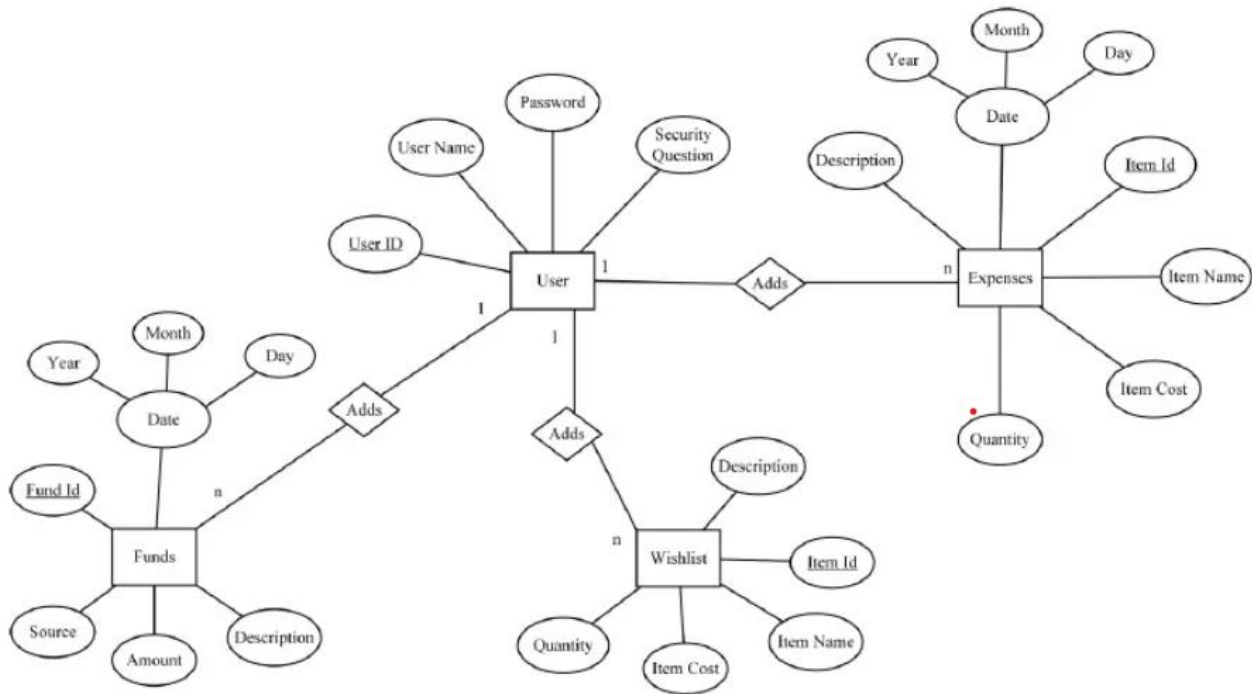
3.3 DESIGN



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. 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.

3.4 ENTITY RELATIONSHIP DIAGRAM (ERD)



The **Entity-Relationship (ER) Diagram** for the Expense Tracker project represents the database structure and the relationships between various entities involved in the system.

Here's the description of the ER diagram for the project:

3.4.1 Entities

- **Expense:** Represents an individual expense record. Attributes include:
 - Expense-ID (Primary Key)
 - Amount
 - Category
 - Date
 - Description
 - Quantity
- **Category:** Represents predefined or custom categories for expenses. Attributes include:
 - Category-ID (Primary Key)
 - Name
 - Description
- **funds:** Represents the fund using the Expense Tracker application. Attributes include:
 - Fund-ID (Primary Key)
 - Source
 - Amount
 - Description

3.42 Diagram Description

The Fund entity is connected to the Expense entity through the Fund-ID attribute. This ensures that each expense is linked to a specific fund.

The Expense entity is connected to the Category entity via the Category-ID attribute, allowing expenses to be grouped into logical categories (e.g., Food, Travel, Utilities).

The attributes in each entity are designed to ensure proper normalization and avoid data redundancy.

3.43 Example Representation

- **Tables Generated:**
 - Users Table (storing user information)
 - Expenses Table (storing expense details linked to users and categories)
 - Categories Table (storing predefined or user-defined categories)

This ER diagram ensures data consistency, efficient retrieval, and scalability of the database system for the Expense Tracker.

CHAPTER-4

EXPERIMENTAL RESULT AND ANALYSIS

4.1 Project User Interface

The interface of our project that are visible to the user ,and by using which the user perform operation like adding their Daily expense and making record of our Expenses:



4.2 WORKING

This project perform some basic operation like:

Data Flow Diagram:

- 1.User inputs data via UI.
- 2.Backend processes and validates input.
- 3.Data is stored in the database.
- 4.Backend retrieves processed data for insights.
- 5.Data is visualized on the dashboard.

4.21 User Inputs Data

Explanation:

The first step involves users entering details about their expenses. These details typically include the amount spent, the date of the expense, and the category it belongs to (e.g., Food, Transport, Entertainment). The application provides an intuitive graphical interface (GUI) built with Java Swing, allowing users to input this data seamlessly through text fields and dropdown menus.

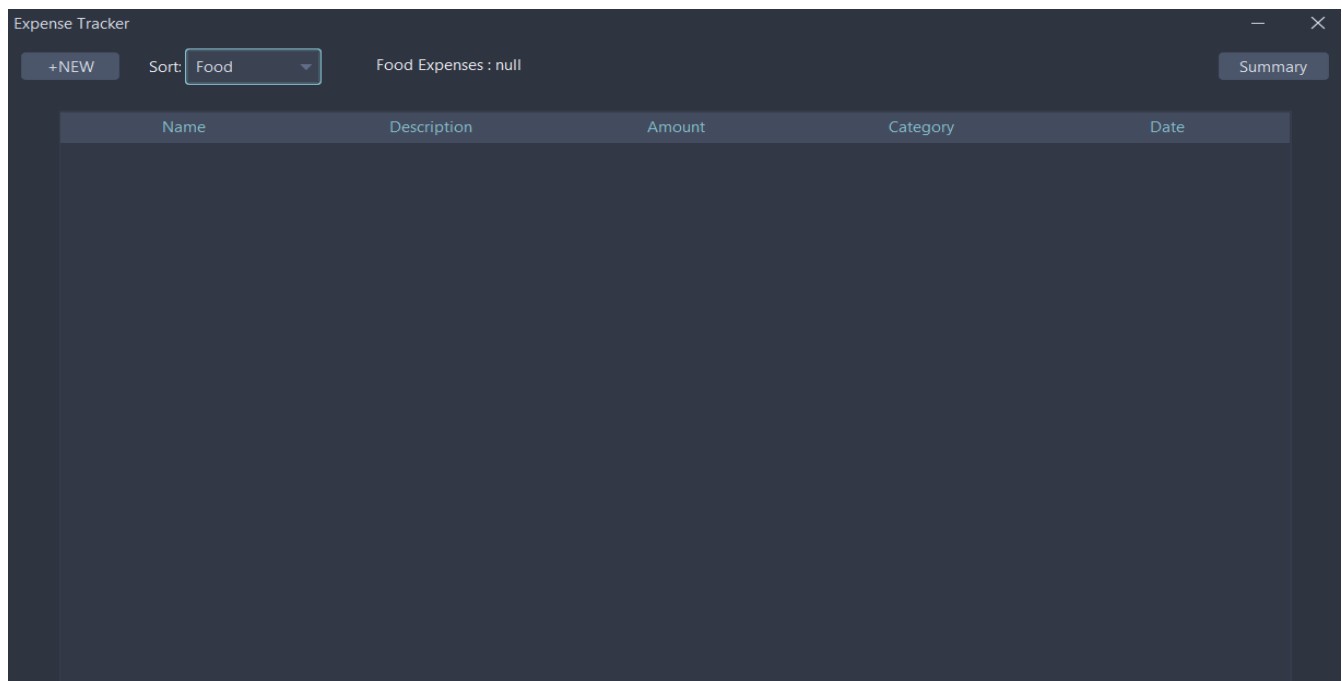
In addition to the basic input mechanism, the application includes an " **New**" **button** that enables users to add detailed descriptions or create new categories dynamically. This feature ensures flexibility, allowing users to personalize the application according to their unique financial tracking needs.

For example, if a user wishes to add an expense under a new category, like "Pet Supplies," they can use the " New" button to create this category on the spot. Similarly, detailed notes can be included for individual expenses, such as specifying "Dog food purchase."

Example:

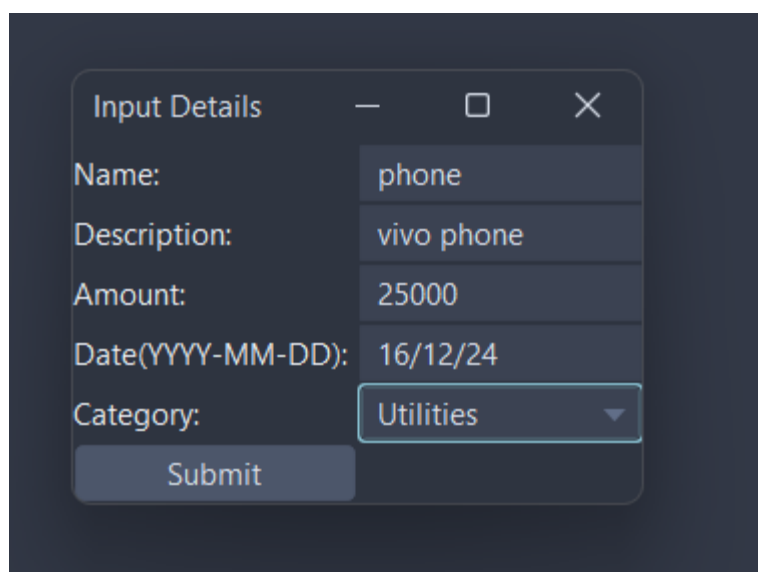
A user records ₹25000 spent on Phone on 16nd December 2024 and categorizes it as "utilities." If " utilities " is not a predefined category, the user clicks the "Add New" button, inputs " utilities " as a new category, and assigns the expense to it.

Before:



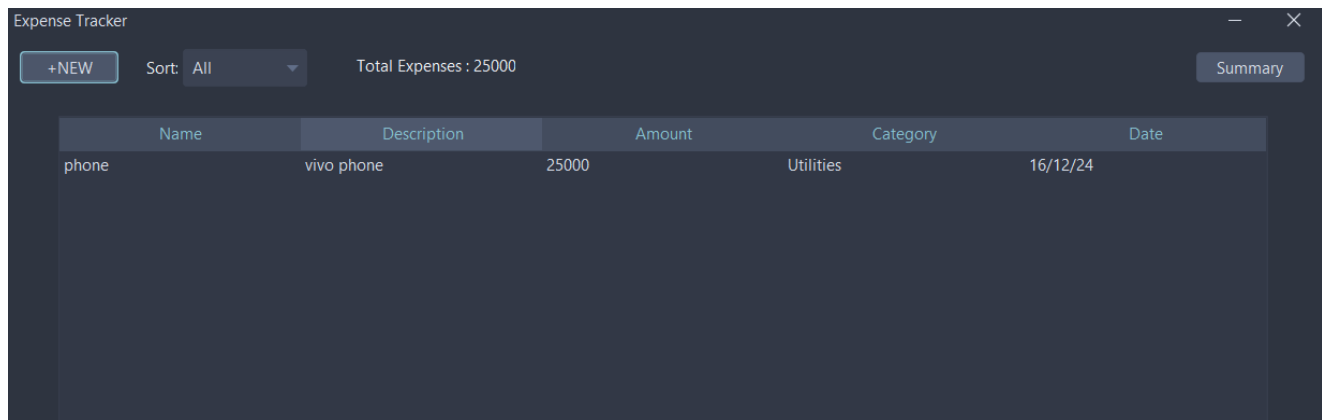
The screenshot shows the 'Expense Tracker' application window. At the top, there is a '+NEW' button, a 'Sort' dropdown menu currently set to 'Food', and a text display 'Food Expenses : null'. A 'Summary' button is located in the top right corner. Below this header is a table with five columns: 'Name', 'Description', 'Amount', 'Category', and 'Date'. The table body is currently empty.

Adding:



The screenshot shows a modal dialog box titled 'Input Details'. It contains five input fields: 'Name' with the value 'phone', 'Description' with 'vivo phone', 'Amount' with '25000', 'Date(YYYY-MM-DD):' with '16/12/24', and 'Category' with a dropdown menu showing 'Utilities'. A 'Submit' button is at the bottom of the dialog.

After Adding:



Name	Description	Amount	Category	Date
phone	vivo phone	25000	Utilities	16/12/24

4.22 Validation

Explanation:

Validation is a crucial step to ensure the integrity and accuracy of the data submitted by the user. It acts as a checkpoint to verify that the input adheres to the expected format and standards before storing it in the database. This step minimizes errors, enhances user experience, and ensures that only valid, usable data is processed.

Detailed Steps of Validation in the Project:

1. Amount Field Validation

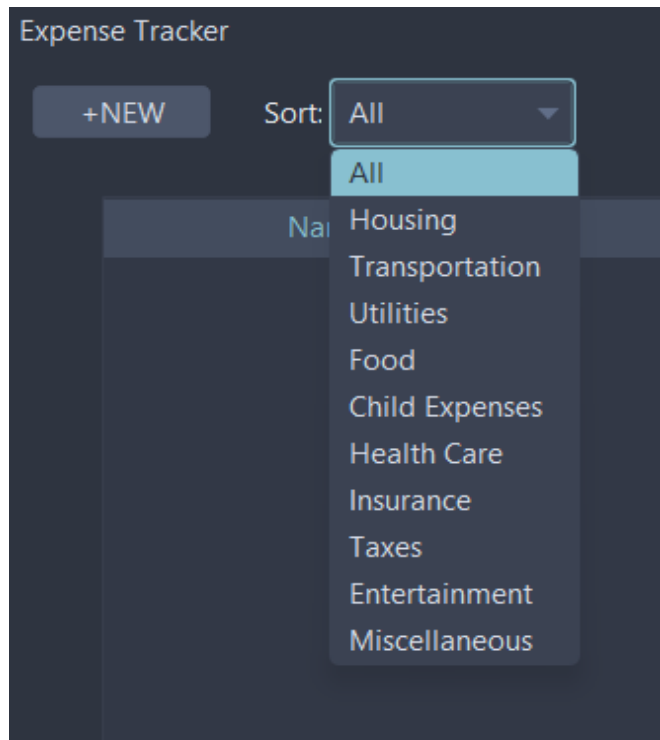
- The application checks whether the "Amount" field contains a valid numeric value.
- Input should be positive numbers only, with no alphabetic or special characters (e.g., entering "500" is valid, but "five hundred" or "500\$" is not).
- Real-time validation: If an invalid value is detected, the application immediately highlights the field and displays an error message, prompting the user to correct it.

2. Date Field Validation

- Ensures the date follows the specified format (e.g., DD/MM/YYYY).
- Checks for invalid or incomplete dates (e.g., "32/12/2024" or "2024-12-22" would be flagged).
- Validates that the date is not empty and belongs to a valid calendar range (e.g., dates cannot be in the distant future or unreasonably far in the past).

3. Category Selection Validation

- The application ensures that a valid category is selected from the dropdown menu.
- If users wish to input a new category, the system checks for duplicate or empty entries to maintain consistency in the database.



4. **Description Validation** (Optional)

- If a description field is included, the system verifies that it contains only valid characters and is within a reasonable length limit (e.g., 250 characters max).

5. **General Error Handling**

- Fields cannot be left blank. The system prompts the user to fill all mandatory fields before submission.
- Any error in validation is communicated clearly to the user with descriptive messages (e.g., "Please enter a valid amount" or "Date format must be DD/MM/YYYY").
- If all fields pass validation, only then is the data submitted to the database for storage.

Benefits of Validation in the Application:

- **Prevents Incorrect Data Entry:** Ensures that incorrect, incomplete, or inappropriate data does not enter the database.
- **Improves User Experience:** Real-time feedback helps users correct mistakes instantly, reducing frustration.
- **Maintains Data Integrity:** Validation rules ensure uniformity, making future data retrieval and analysis accurate and efficient.
- **Reduces Debugging Efforts:** By catching errors early, validation reduces the likelihood of system crashes or inconsistencies during data processing.

4.23 Data Storage

Explanation:

After validation, the expense data is securely stored in the SQLite database. SQLite is a lightweight, serverless database engine that ensures efficient data handling and storage, making it ideal for local applications like your Expense Tracker. The database is designed with a structured schema to maintain data integrity and allow for seamless data retrieval and manipulation.

Database Design and Classification

1. SQLite Tables:

- **Expense Table:**
 - **Purpose:** This table stores details about each expense.
 - **Columns:**
 - Expense-ID: A unique identifier for each expense entry (Primary Key).
 - Amount: The value of the expense (Numeric).
 - Date: The date when the expense occurred (Text/Date).
 - Category-ID: Links to the associated category in the Category Table (Foreign Key).
 - Description: Additional notes about the expense (Text).
 - **Example Entry:** Expense-ID: 1, Amount: 500, Date: 16/12/2024, Category-ID: 1, Description: "Groceries for the week."
- **Category Table:**
 - **Purpose:** This table stores predefined and custom expense categories.
 - **Columns:**
 - Category-ID: A unique identifier for each category (Primary Key).
 - CategoryName: The name of the category (e.g., Food, Travel, Shopping).
 - **Example Entry:** Category-ID: 1, CategoryName: "Food"
- **Relationships**
 - **One-to-Many Relationship:**

Each category in the Category Table can be associated with multiple expenses in the Expense Table.

Example: The "Food" category may link to multiple expenses like "Lunch", "Groceries", or "Dinner."
 - **One-to-One Relationship:**

Each expense in the Expense Table corresponds to exactly one category from the Category Table.

Example: An expense for ₹500 on "Groceries" is linked to a single category, "Food."

How Data is Stored and Retrieved:

- **Storage:**

- When users input expense details, an SQL INSERT query saves the data into the Expense Table, with the associated Category-ID linking to the Category Table.

- Example SQL Query:

```
INSERT INTO Expenses (Amount, Date, Category-ID, Description)
VALUES (25000, '16/12/2024', 1, Utilities for the week');
```

- **Retrieval:**

- To display all expenses along with their categories, an SQL JOIN query is used.

- Example SQL Query:

```
SELECT e.Expense-ID, e.Amount, e.Date, c.CategoryName, e.Description FROM Expenses e
JOIN Categories c ON e.Category-ID = c.Category-ID;
```

Benefits of This Structure:

1. **Scalability:** Easily add or modify categories without disrupting expense data.
2. **Efficiency:** Relational structure allows for quick lookups and efficient queries.
3. **Data Integrity:** Foreign Key constraints ensure that each expense is linked to a valid category.
4. **Flexibility:** Users can define custom categories in the Category Table while maintaining consistency across expenses.
5. **Organized Analysis:** Enables generating detailed reports by categories or dates using the relationships.

4.24 Sorting Expenses Based on Categories

Explanation: In the Expense Tracker application, sorting expenses based on categories allows users to efficiently view and organize their expense data. This feature enhances usability and provides insights into spending patterns. The categories available in the application, such as Housing, Transportation, Food, Entertainment, and others, enable users to classify their expenses systematically.

How Sorting Works in the Project

1. **Dropdown Menu for Category Selection:**

- A dropdown menu, as shown in the interface, allows users to filter expenses by specific categories or view all expenses.

- Example Categories:
 - **Housing:** Rent, repairs, or home-related costs.
 - **Food:** Groceries, dining out, etc.
 - **Transportation:** Travel expenses like fuel or public transport.
 - **Entertainment:** Leisure activities such as movies or games.

2. Filtering Logic:

- When a user selects a category (e.g., "Food"), the application queries the SQLite database for all expenses associated with that category.
- The filtered results are displayed in the GUI, helping users focus on relevant expenses.
- Example SQL Query:

```
SELECT * FROM Expenses WHERE Category = 'Food';
```

3. Backend Implementation:

- The dropdown menu sends the selected category as a parameter to the database query.
- If "All" is selected, the application retrieves all expense records without filtering.

4. Display Results:

- The retrieved data is displayed in a tabular format within the application.
- For example, selecting "Transportation" shows only transportation-related expenses with details like date, amount, and description.

5. User Interaction:

- The **+NEW** button allows users to add new expense entries, which are automatically assigned to a selected category.
- Sorting can also be combined with other features like date range selection or expense analysis for more granular control.

Expense Tracker

+NEW Sort: All Total Expenses : 25000 Summary

Name	Description	Amount	Category	Date
phone	vivo phone	25000	Utilities	16/12/24

Expense Tracker

+NEW Sort: Food Food Expenses : null Summary

Name	Description	Amount	Category	Date
------	-------------	--------	----------	------

Benefits of Sorting by Categories

- **Better Financial Analysis:** Users can track how much they are spending in specific areas (e.g., monthly food expenses).
- **Improved Decision-Making:** Helps identify categories where expenses can be reduced.
- **Simplified Record-Keeping:** Organizes data for easy reference and reporting.
- **User-Friendly Experience:** Intuitive design makes it easy to filter and find relevant information quickly.

4.25 View Detailed Summaries of Total Expenses

Introduction:

The "Expense Summary" feature of the Expense Tracker application is a crucial tool for users to understand their financial activities comprehensively. It provides a breakdown of expenses across multiple categories while also presenting the overall total spending. This functionality enhances the user experience by offering a clear and concise analysis of financial data.

Details of the Feature:

1. Data Aggregation:

- The application aggregates expense data by summing the total amounts within each predefined category (e.g., Housing, Food, Utilities, etc.).
- It simultaneously calculates the overall total expenses, giving users a complete view of their financial status.
- SQL Example:

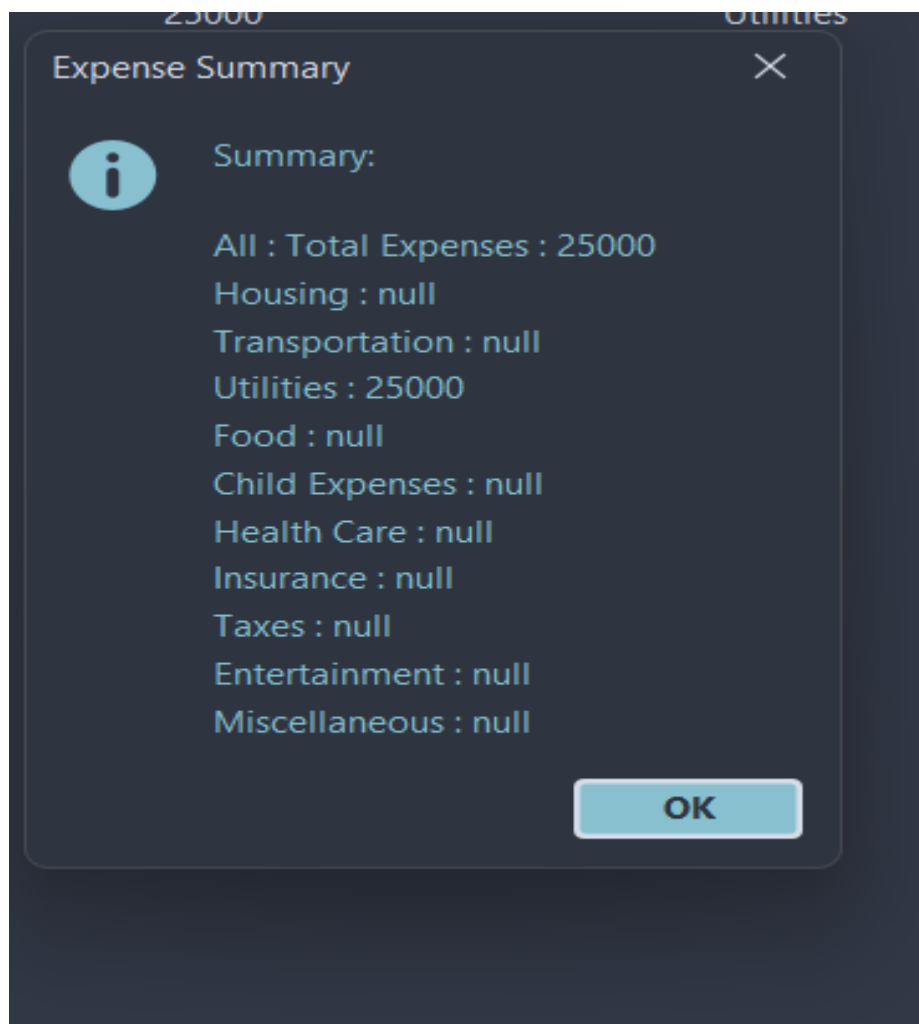
```
SELECT Category, SUM(Amount) AS Total FROM Expenses GROUP BY Category;
```

2. Display Summary:

- A popup window or summary panel displays the data in a structured format. Each category is listed with its respective total.
- Categories with no recorded expenses are marked as "null," indicating that no data exists for those entries.
- Example Summary:
 - All: ₹25,000
 - Utilities: ₹25,000
 - Housing, Food, etc...: "null"

3. Dynamic Updates:

- The summary dynamically updates whenever the user adds, modifies, or deletes an expense.



- This ensures real-time accuracy, making it a reliable tool for tracking finances.

4. User Interaction:

- The **OK** button on the summary box allows users to close the window, while the organized display makes it easy to interpret financial data at a glance.

5. Error Handling:

- Categories with no entries are explicitly displayed as "null," prompting users to review their data and possibly add missing expenses.

Benefits of the Expense Summary:

- **Comprehensive View:** Users can analyze their spending across different categories, providing a holistic financial overview.
- **Enhanced Decision-Making:** By identifying categories with high spending, users can plan their budgets more effectively.
- **Real-Time Tracking:** Instant updates ensure that users always have an accurate picture of their finances.
- **User-Friendly Design:** A single, intuitive display makes the feature accessible and easy to use.

Example Usage:

A user logs monthly expenses into the tracker. After entering the data, the Expense Summary feature shows:

- **Total Expenses:** ₹25,000
- **Utilities:** ₹25,000
- **Housing, Food, and other categories:** "null" (indicating no recorded expenses).

This helps the user immediately spot which categories have been used and identify untracked areas, encouraging more effective expense management.

CHAPTER-5

CURRENT & FUTURE SCOPE

5.1 CURRENT SCOPE:

1. Expense Management:

- The project currently provides an efficient solution for tracking daily expenses, replacing traditional methods like paper or spreadsheets with a user-friendly digital application.
- Users can input, categorize, and organize expenses easily, gaining a clear understanding of their financial habits.

2. Data Analysis and Summaries:

- The feature for viewing categorized summaries and total expenses enables users to analyze spending patterns effectively.
- Sorting and filtering options enhance usability and allow for focused financial tracking.

3. Personal Finance Tool:

- Acts as a reliable, portable, and simple tool for managing finances, particularly for individuals, small business owners, and families.
- Encourages better budgeting and financial discipline by offering insights into spending behaviors.

4. Local Storage System:

- The SQLite database provides an effective solution for data storage, ensuring users can manage their expense records offline without relying on cloud-based systems.

5. Scalable Design:

- The modular structure allows for future integration of additional features like custom reports, advanced filtering, and trend analysis without requiring a major redesign.

5.2 FUTURE SCOPE:

1. Cloud Integration:

- Transitioning from local storage to cloud-based databases for real-time data syncing across multiple devices, allowing users to access their records anywhere.

2. Mobile Application Development:

- Expanding the project to mobile platforms (Android/iOS) to provide a more portable and widely accessible solution for expense tracking.

3. Machine Learning for Expense Prediction:

- Integrating machine learning algorithms to predict future expenses based on past spending patterns.

- Features like automated categorization of expenses and spending alerts could be implemented.

4. Advanced Data Visualizations:

- Adding visual tools such as graphs and charts to present users with a more comprehensive overview of their financial data.
- Monthly and yearly comparisons can help users track progress towards financial goals.

5. Integration with External Services:

- Linking the app with payment systems, bank accounts, or e-wallets to automate data entry.
- Providing users with real-time tracking of their transactions and a holistic view of their financial activities.

6. Multi-User Support:

- Extending the application to support multi-user accounts, which would make it suitable for families or small businesses.
- Adding role-based permissions for data access and management.

7. Expense Sharing and Notifications:

- Features for shared expenses or group tracking, particularly for events or households.
- Notifications or reminders for overdue bills, upcoming expenses, or financial milestones.

8. Custom Reporting:

- Providing users with the ability to generate detailed reports based on custom time periods, categories, or specific filters.
- These reports can be exported to formats like PDF or Excel.

CHAPTER-6

CONCLUSION

The **Expense Tracker** project is a comprehensive solution for managing personal financial activities, developed to address the inefficiencies of traditional expense recording systems. This project introduces a digital, user-friendly platform that enhances financial management through features like real-time expense tracking, organized categorization, and detailed data summaries. Its scope and functionality cater to both individual users and small-scale businesses, making it a valuable asset in today's digital age.

Key Highlights of the Project

1. Objective Fulfillment:

The primary goal of the project was to simplify the process of tracking daily expenses and income. By replacing manual methods such as pen-and-paper or spreadsheets, the application provides a portable and efficient alternative, reducing human error and saving time.

2. Technology Utilization:

- **Frontend:** The graphical user interface (GUI) is built using Java Swing, offering an intuitive and visually appealing platform for users to interact with the system.
- **Backend:** Data storage is managed with SQLite, ensuring secure and efficient handling of financial records. Its lightweight structure and reliability make it ideal for local applications.
- **Programming Language:** Java serves as the backbone of the application, providing cross-platform compatibility and robust functionality.

3. User-Centric Design:

The application prioritizes user convenience through features such as:

- Categorized expense tracking (e.g., Housing, Food, Utilities).
- Summarized reports and real-time data updates.
- An easy-to-use "Add New" button for seamless data entry.

These features ensure that users can manage their financial data with minimal effort and maximum efficiency.

4. Data Analysis and Insights:

The Expense Summary feature provides users with an organized overview of their spending, broken down by categories. This functionality helps users identify spending trends and make informed decisions to optimize their budgets.

Impact and Benefits

• Enhanced Financial Discipline:

By offering detailed insights into spending habits, the project encourages users to maintain better control over their finances. Tracking expenses in real-time enables users to identify areas where they can save money.

- **Accessibility and Portability:**

The local SQLite database ensures that users can access their financial data offline, making the application highly reliable and portable.

- **Scalability:**

The modular design allows for the integration of future enhancements such as mobile app versions, cloud synchronization, and predictive analytics.

- **Error Reduction:**

The validation mechanism prevents invalid data entries, ensuring data integrity and accuracy.

Future Potential

This project lays a strong foundation for future development. Possible enhancements include:

- Integration with online banking or payment systems for automated data entry.
- Advanced features like data visualizations, AI-powered expense predictions, and notifications for budgeting goals.
- Expansion to mobile platforms (Android/iOS) to increase accessibility. These advancements will further solidify the application's relevance in a competitive digital landscape.

REFERENCES

- **SQLite Official Documentation**

Provides detailed guidance on SQLite database creation, management, and integration, which was crucial for the backend database of the Expense Tracker project.

URL: <https://sqlite.org/docs.html>

- **Java Swing Official Documentation**

Served as the primary source for designing the user interface with Java Swing, explaining components such as text fields, buttons, and dropdown menus.

URL: <https://docs.oracle.com/javase/tutorial/uiswing/>

- **IntelliJ IDEA**

A robust development environment used for coding and debugging the project, offering tools for Java programming and database connectivity.

URL: <https://www.jetbrains.com/idea/>

- **Oracle Java Tutorials**

Essential for learning core Java concepts such as file handling, data validation, and database interaction. These tutorials helped in developing the project's backend logic.

URL: <https://docs.oracle.com/javase/tutorial/>

- **UX Design Insights for Finance Applications**

Articles and guidelines on user experience design for financial applications inspired the layout and usability of the Expense Tracker interface.

URL: <https://uxdesign.cc/designing-personal-finance-apps/>

- **Investopedia Article: Why Expense Tracking is Important**

This resource provided foundational knowledge on the importance of expense tracking in financial management, aligning the project's objectives with real-world needs.

URL: <https://www.investopedia.com/>