Project Report

Expense Tracker

Submitted to Manipal University, Jaipur

Towards the partial fulfilment for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In Computers Science and Engineering

2022-2023

By

Prerna Dureja

229301453

Shriyam 229302281

Under the guidance of

Dr. Surbhi Sharma

Department of Computer Science and

Engineering School of Computer Science and

Engineering Manipal University Jaipur

Jaipur, Rajasthan

TABLE OF CONTENTS

INTRODUCTION

In today's paced world of advancements and the increasing digitalization of our lives, it has become crucial to efficiently manage our finances. Whether it's handling organizational expenses, the task can be quite overwhelming. To address this challenge, our minor project focuses on using Java and Object Oriented Programming (OOP) to create a user-friendly application called the Expense Tracker.

Our aim with the Expense Tracker is to develop an application that's not easy to use but also packed with features for effective expense management. The main objectives of our project are;

1. User-Friendly Interface: We strive to create a graphical user interface (GUI) that simplifies expense recording and management for users at any level of proficiency.

2. Data Persistence: Our focus is, on implementing a data storage mechanism that ensures the integrity and accessibility of expense records.

3. Expense Categories: We provide the flexibility for users to categorize their expenses, enabling analysis and reporting.

4. Search and Reporting: Users will have the ability to search for expenses and generate reports, allowing them to visualize their financial activities effectively.

5. Security: We prioritize safeguarding users' financial data by incorporating authentication and encryption mechanisms to maintain privacy and security.

By accomplishing these objectives, we believe that our Expense Tracker application will assist individuals or organizations in managing their expenses in a digital world.

The growing intricacy of transactions, along with the necessity for record-keeping, highlights the importance of a reliable system for tracking expenses. Individuals need tools to manage their budgets, while businesses require solutions to monitor and optimize their spending.

MOTIVATION

Several strong reasons drive the development of an expense tracking system:

- **Efficiency and Simplicity:** Traditional manual ways of tracking spending, like the use of paper receipts or complicated spreadsheets, take a lot of time and are errorprone. By automating the recording, categorization, and reporting of expenses, the major goal here is to streamline and simplify this process.

- **Data-Driven Decision-Making:** Access to correct financial data is essential in the age of data-driven decision-making, which is a critical component of today's society.
  An Expense Tracker system can provide customers with in-depth statistics and analytics, giving them the knowledge they need to make decisions about their financial objectives, investments, and budgetary modifications.

- **Customization and Scalability:** Providing users with a highly personalized, scalable, and customizable tool. Different people and organizations have different needs for spending and tracking. Utilizing OOP concepts, we can develop a system that changes to these requirements.

- **Better Financial Health:** The main goal is to assist users in increasing the state of their finances by including options like expense categorization, spending analysis, and budgeting. The Expense Tracker enables customers to manage their money, pay off debt, and pursue their financial objectives.

- **Privacy and Security:** Ensuring the security and privacy of financial data is a critical motivation. An Expense Tracker can implement robust security measures, such as encryption and authentication, to safeguard sensitive financial information, giving users peace of mind.

- **Knowledge of technology:** By making an expense tracker, students and developers have the chance to obtain crucial experience and skills in these technologies. It's a Java programming knowledge and OOP ideas applied practically fixing issues in the actual world.

- **Market Demand:** The demand for such applications in the market also drives the motivation. As individuals and businesses seek more sophisticated and userfriendly expense management tools, there is a growing opportunity to provide a feasible solution.

- **Educational Value:** A project in which you track your expenses can be a platform enabling users to learn about budgeting, personal finance, and financial preparation.

PROJECTOBJECTIVE

Objective	Description	Pros of Existing Methods	Cons of Existing Methods

Automate Expense Recording	Create a system that automates the process of recording expenses, reducing manual effort.	Manual entry allows for customization. Users have direct control over data.	Time-consuming and prone to errors. Manual entries can be misplaced.
Expense Classification	Enable users to classify expenses, allowing detailed financial analysis and budget planning.	Flexibility to categorize expenses as per personal preferences. Users can define their own categories.	Requires consistent manual effort to classify expenses. Categories may become jumbled or inconsistent over time.
Real-time Expense Tracking	Provide real-time tracking of expenses to money is give users a current view of their financial overspending.	Users can instantly see where their going. Immediate detection of situation.	Requires constant manual entry for realtime updates. Users may not always have the time to update expenses regularly.
Detailed Reporting and Analytics	Generate inclusive reports and analytics to help users make informed financial decisions.	Allows for in-depth financial analysis. Supports long-term financial planning.	Manual generation of reports can be timeconsuming. Data analysis may require advanced spreadsheet skills.
Data Security and Privacy	Implement high-security measures to financial data and protect user privacy.	Users have control over their data. Data can be protect stored securely offline.	Possibility of data loss (e.g., paper receipts) or unauthorized access (e.g., shared spreadsheets). Security measures may not be applied consistently.
User-friendly Interface	Design an automatic and user-friendly graphical interface to ensure accessibility for all users.	Appeals to users with varying levels of technical proficiency. Reduces the learning process for new users.	It may lack modern features or customization options. User interface design may not serve specific user preferences.

Scalability and Customization	Create a system that can adapt to the different expense tracking needs different users and organizations.	It can be tailored to unique requirements. Scales to aid personal and business expenses.	Customization may require technical expertise. Scaling of existing methods can lead to complexity and inefficiency.

METHODOLOGY

Adding Entries

- Users can add financial entries by providing the following information:

-Date
-Description
-Amount
-Type (Expense or Income)
The application performs input validation to ensure accurate data entry.

- Viewing Entries

The application displays a table that shows the financial entries, including columns for date, description, amount, and type. Users can easily view and manage their financial records.

- Calculating Balance

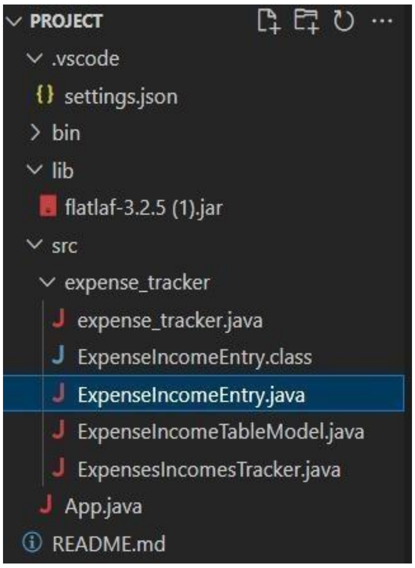
The application calculates the balance based on the sum of expenses and incomes, providing users with a realtime view of their financial status.

Usage Guide:
To use the Expense and Income Tracker application:

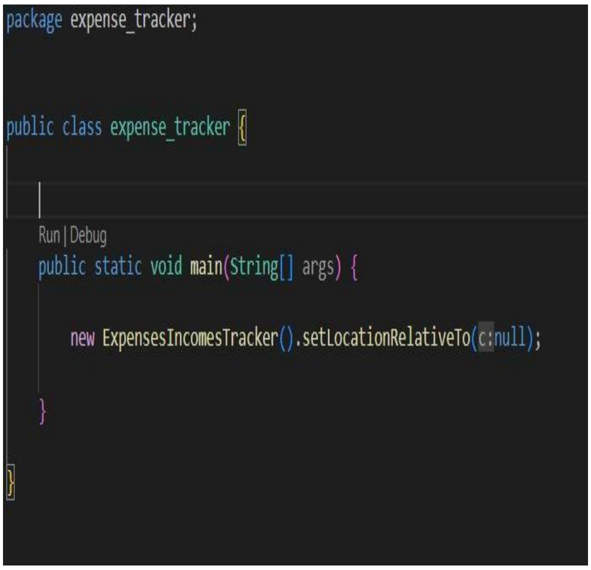
-Launch the application by running the Main Class (Expense_Income_Tracker).
-The main window will appear with input fields for adding new financial entries, a table for viewing existing entries, and a balance label.
-Enter the date, description, and amount, and select the type (expense or income).
-Click the “Add” button to add the entry. The table will be updated with the new entry, and the balance will be recalculated.
-To clear the input fields for the next entry, click the “Clear” button.

You can continue adding and managing financial entries as needed.

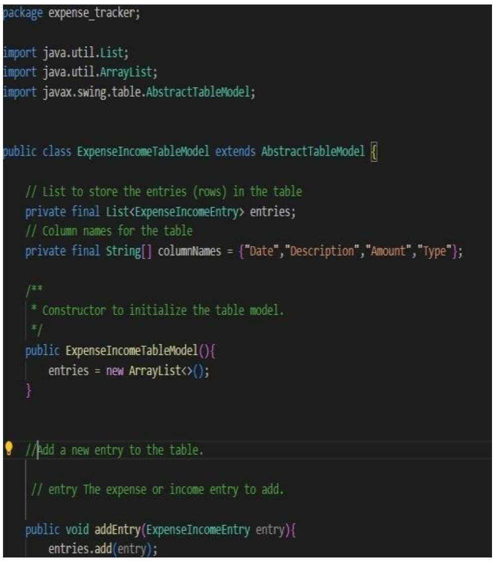
PLANNING OF WORK



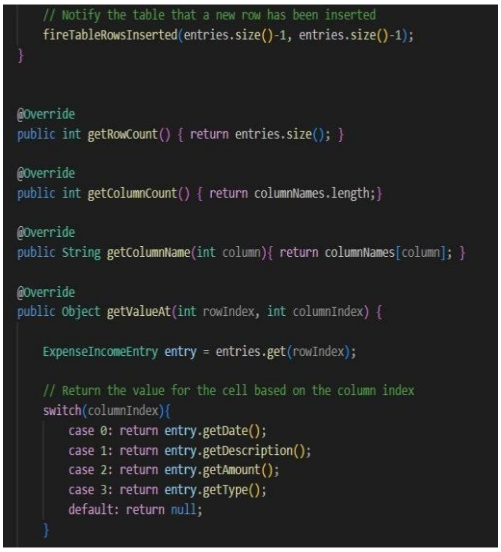
```
.vscode
  settings.json
bin
lib
  flatlaf-3.2.5 (1).jar
src
  expense_tracker
    expense_tracker.java
    ExpenseIncomeEntry.class
    ExpenseIncomeEntry.java
    ExpenseIncomeTableModel.java
    ExpensesIncomesTracker.java
  App.java
README.md
```



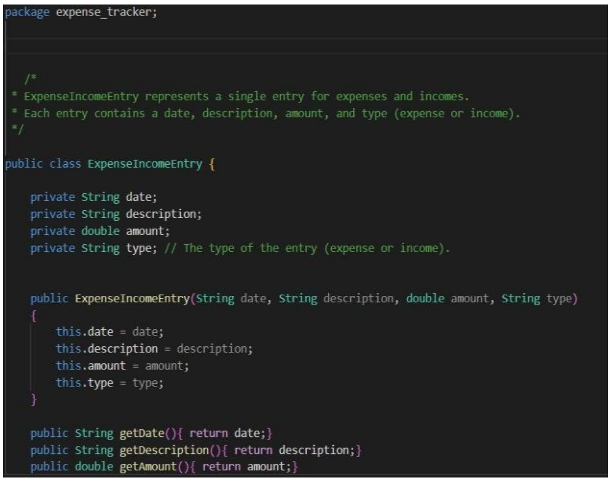
```java
package expense_tracker;


public class expense_tracker {


    Run | Debug
    public static void main(String[] args) {

        new ExpensesIncomesTracker().setLocationRelativeTo(c:null);

    }

}
```



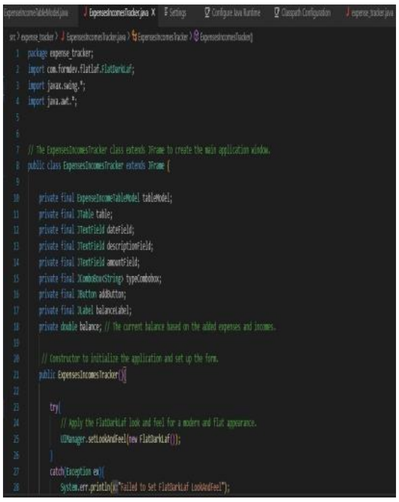
```java
package expense_tracker;

import java.util.List;
import java.util.ArrayList;
import javax.swing.table.AbstractTableModel;


public class ExpenseIncomeTableModel extends AbstractTableModel {

    // List to store the entries (rows) in the table
    private final List<ExpenseIncomeEntry> entries;
    // Column names for the table
    private final String[] columnNames = {"Date","Description","Amount","Type"};

    /**
     * Constructor to initialize the table model.
     */
    public ExpenseIncomeTableModel(){
        entries = new ArrayList<>();
    }


    //Add a new entry to the table.

     // entry The expense or income entry to add.

    public void addEntry(ExpenseIncomeEntry entry){
        entries.add(entry);
```



```java
        // Notify the table that a new row has been inserted
        fireTableRowsInserted(entries.size()-1, entries.size()-1);
    }


    @Override
    public int getRowCount() { return entries.size(); }

    @Override
    public int getColumnCount() { return columnNames.length;}

    @Override
    public String getColumnName(int column){ return columnNames[column]; }

    @Override
    public Object getValueAt(int rowIndex, int columnIndex) {

        ExpenseIncomeEntry entry = entries.get(rowIndex);

        // Return the value for the cell based on the column index
        switch(columnIndex){
            case 0: return entry.getDate();
            case 1: return entry.getDescription();
            case 2: return entry.getAmount();
            case 3: return entry.getType();
            default: return null;
        }
```



```java
package expense_tracker;



  /*
 * ExpenseIncomeEntry represents a single entry for expenses and incomes.
 * Each entry contains a date, description, amount, and type (expense or income).
 */

public class ExpenseIncomeEntry {

    private String date;
    private String description;
    private double amount;
    private String type; // The type of the entry (expense or income).


    public ExpenseIncomeEntry(String date, String description, double amount, String type)
    {
        this.date = date;
        this.description = description;
        this.amount = amount;
        this.type = type;
    }

    public String getDate(){ return date;}
    public String getDescription(){ return description;}
    public double getAmount(){ return amount;}
```



ExpenseIncomeTableModel.java
ExpensesIncomesTracker.java X
Settings
Configure Java Runtime
Classpath Configuration
expense_tracker.java
src > expense_tracker > ExpensesIncomesTracker.java > ExpensesIncomesTracker > ExpensesIncomesTracker()

```java
package expense_tracker;
import com.formdev.flatlaf.FlatDarkLaf;
import javax.swing.*;
import java.awt.*;


// The ExpensesIncomesTracker class extends JFrame to create the main application window.
public class ExpensesIncomesTracker extends JFrame {

    private final ExpenseIncomeTableModel tableModel;
    private final JTable table;
    private final JTextField dateField;
    private final JTextField descriptionField;
    private final JTextField amountField;
    private final JComboBox<String> typeCombobox;
    private final JButton addButton;
    private final JLabel balanceLabel;
    private double balance; // The current balance based on the added expenses and incomes.

     // Constructor to initialize the application and set up the form.
    public ExpensesIncomesTracker(){

        try{
           // Apply the FlatDarkLaf look and feel for a modern and flat appearance.
           UIManager.setLookAndFeel(new FlatDarkLaf());
        }
        catch(Exception ex){
           System.err.println(x:"Failed to Set FlatDarkLaf LookAndFeel");
```

FACILITIES

The applications in the expense tracking system are built on the technologies described below:
- Java Programming Language Java for GUI (Graphical User Interface)
- Swing Java for GUI (Graphical User Interface)
- FlatDarkLaf for a modern and flat look

The Expense Tracker Application provides a modern and visually appealing user experience. It achieves this through the integration of the FlatDarkLaf look and feel, which not only enhances the aesthetics of the application but also ensures a contemporary and flat design that is in line with current design trends. FlatDarkLaf is a modern and sleek look and feel library for Java applications. It replaces the standard Swing look and feel with a more stylish and visually pleasing design. This integration contributes to a more engaging and enjoyable user interface, making the process of managing finances a more aesthetically pleasing experience. The use of FlatDarkLaf demonstrates the application's commitment to providing a cutting-edge and visually satisfying user experience, making it stand out among financial management tools. The Expense Tracker Application is more than just a digital ledger. It is a platform for informed financial decision-making, helping users understand their spending patterns, set budgets, and make adjustments where necessary. With this application, users can effortlessly create records of their financial transactions, categorize them, and visualize their financial history, providing a clearer picture of their monetary health.

The Expense and Income Tracker application is built using Java and the Swing library for creating the graphical user interface. It offers the following key features:

-Adding Entries: Users can add new financial entries, providing information such as the date, description, amount, and type (expense or income).

-Viewing Entries: The application displays a table that allows users to view their financial entries.

-Calculating Balance: The application calculates the balance based on the added expenses and incomes, providing a quick overview of the user's financial status.

Project Structure

The project consists of the following classes:

-ExpenseIncomeEntry: This represents a single financial entry with attributes for date, description, amount, and type (expense or income).

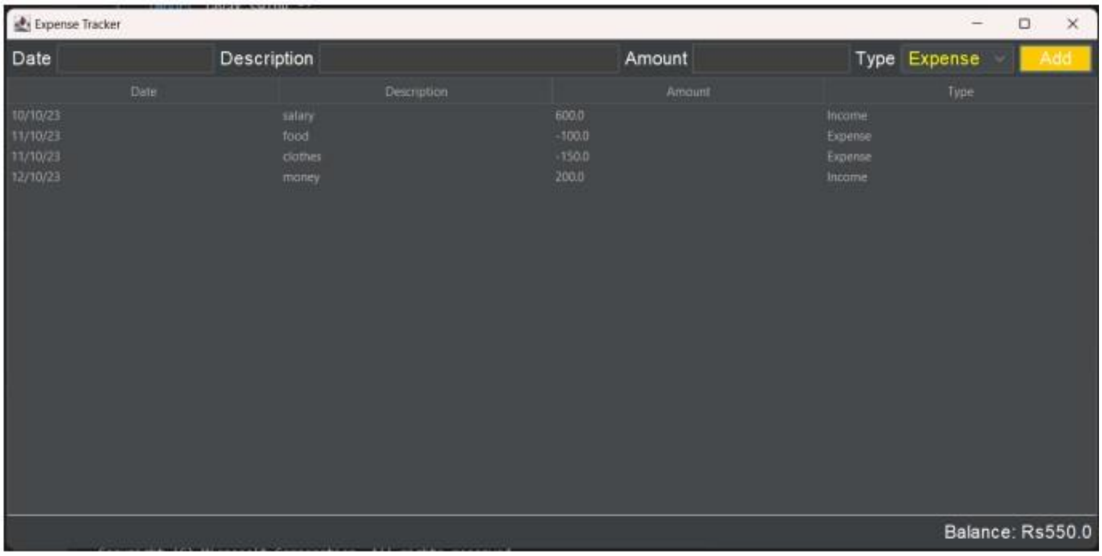
-ExpenseIncomeTableModel: Manages the data and structure of the table used to display entries.

-ExpensesIncomesTracker: The main application class that extends JFrame and creates the user interface. It handles the addition of new entries and updates the balance.

-Main Class (Expense_Income_Tracker): The entry point of the application, which initializes the main application window.

USER INTERFACE

It is intended that the user interface will be clean and pleasant to use. It includes the following screens for Inputting the date, description, amount, and type (expense or income) for new entries. Using JTable, you can display a tabular view of the financial statements. A picture of the current balance.



Type: Expense

Date	Description	Amount	Type
10/10/23	salary	600.0	Income
11/10/23	food	-100.0	Expense
11/10/23	clothes	-150.0	Expense
12/10/23	money	200.0	Income

Balance: Rs550.0

CONCLUSION

It is intended that the user interface will be clean and pleasant to use. It includes the following screens for Inputting the date, description, amount, and type (expense or income) for new entries. Using JTable, you can display a tabular view of the financial statements. A picture of the current balance.

There might be further improvements to the expense tracker application that can be made: Exports finance data in various formats, such as CSV or Excel. Authentication of the user for secure access.

For better data visualization, create charts and graphs.

The Expense Tracker Application is a valuable tool for tracking expenses and incomes efficiently. It provides users with an easy-to-use interface to manage their financial activities and calculate their current balance.

BIBLIOGRAPHY/REFERENCES

1. Title: ‘Personal Finance Management: Current Practices and Future Opportunities’

- Author: Scott D. Dyreng, Jeffrey L. Hoopes, Brian J. William

- Source: Journal of Accounting Research, Volume 53, Issue 1 (2015)

2. Title: ‘Importance of Expense Tracking in Personal Finance Management’ • Author: Richard Barrington

- Source: The Balance, an online financial publication

- URL: https://www.thebalance.com/importance-of-expensetracking-2386008

3. Title: ‘Software Requirements Specification (SRS) Template’

- Author: IEEE Computer Society

- Source: IEEE Computer Society, a leading organization in the field of computer science and engineering

- URL: https://www.computer.org/education/docs/srs.pdf

- Java Documentation: https://docs.oracle.com/en/java/

- FlatLaf Look and Feel: https://www.formdev.com/flatlaf/