

Java Expense Tracker Application

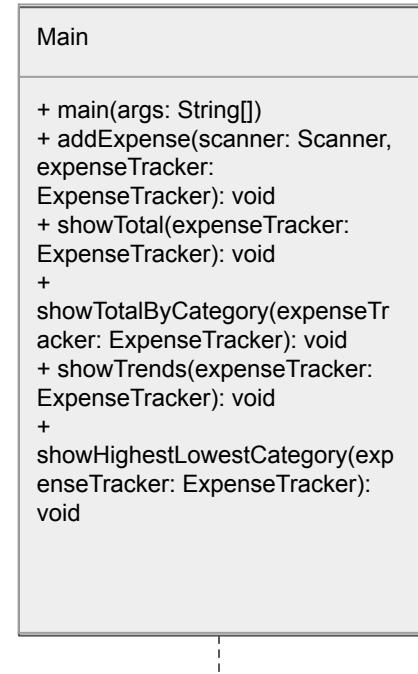
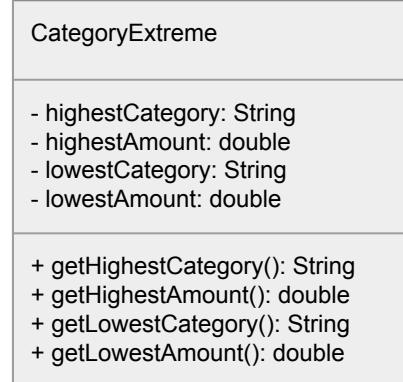
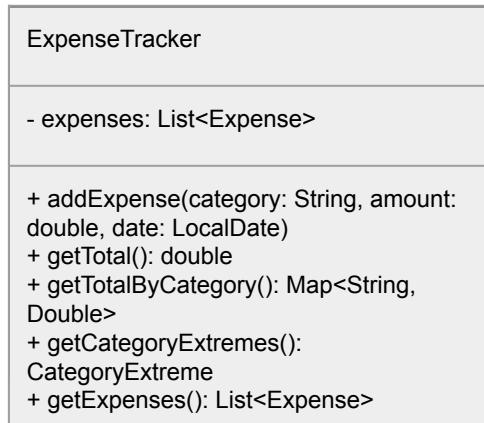
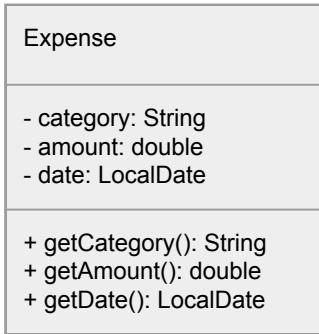


Presented By

Rovia Simmons

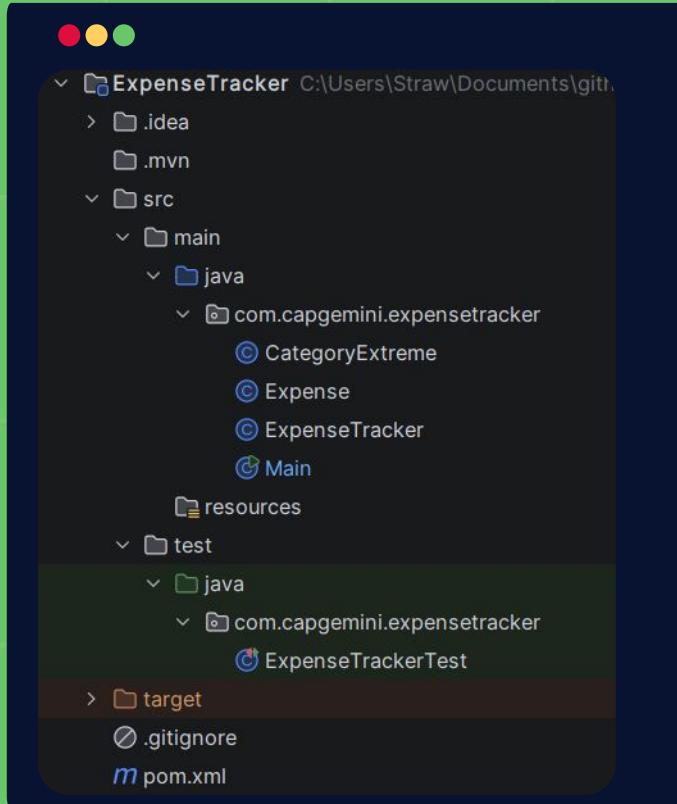
12/14/2025

Approach and Design



- **Expense** holds data; **ExpenseTracker** manages and analyzes expenses.
- **Main** handles user input/output (CLI).
- Data flows: user → **Main** → **ExpenseTracker** → results.

Key Files and Folders



Project structure showing main source code and test files



This project uses Maven structure

Main Files:

- `Main.java`: The application's entry point.
- `ExpenseTracker.java`: Contains the core logic for expense handling.
- `Expense.java`: The data model for an individual expense.

Test Files: `ExpenseTrackerTest.java`: Contains the unit tests to verify the functionality

Process to Run, Test, and Verify



Run the application

Locate the folder with the POM file in the command line

```
mvn clean compile
```

```
mvn exec:java -Dexec.mainClass="com.capgemini.expensetracker.Main"
```



Run the tests and verify ExpenseTracker works

```
Mvn test
```



Add sample/seed data

Enter expenses via CLI prompts

Or pre-populate in Main.java:



Add sample/seed data

```
public class Main { & Rovia Simmons *
    public static void main(String[] args) { & Rovia Simmons *
        System.out.println("Welcome to the Expense Tracker App!");

        Scanner scanner = new Scanner(System.in);
        ExpenseTracker expenseTracker = new ExpenseTracker();

        // pre-populate here!
        expenseTracker.addExpense( category: "Food", amount: 12.50, LocalDate.of( year: 2025, month: 12, dayOfMonth: 13));
        expenseTracker.addExpense( category: "Transport", amount: 3.75, LocalDate.of( year: 2025, month: 12, dayOfMonth: 14));
        expenseTracker.addExpense( category: "Entertainment", amount: 25.00, LocalDate.of( year: 2025, month: 12, dayOfMonth: 15));
```

Test Data and Seed Data



Manual data:

```
1 Food 11.60 2025-12-12
1 Transport 1.79 2025-12-12
1 Transport 3.79 2025-12-14
1 Entertainment 25.00 2025-12-13
```

OR pre-populate Main.java:

```
public class Main { & Rovia Simmons
    public static void main(String[] args) { & Rovia Simmons
        System.out.println("Welcome to the Expense Tracker App!");

        Scanner scanner = new Scanner(System.in);
        ExpenseTracker expenseTracker = new ExpenseTracker();

        // pre-populate here!
        expenseTracker.addExpense( category: "Food", amount: 12.50, LocalDate.of( year: 2025, month: 12, dayOfMonth: 13));
        expenseTracker.addExpense( category: "Transport", amount: 3.75, LocalDate.of( year: 2025, month: 12, dayOfMonth: 14));
        expenseTracker.addExpense( category: "Entertainment", amount: 25.00, LocalDate.of( year: 2025, month: 12, dayOfMonth: 15));
```

Welcome to the Expense Tracker App!

Select an option (1-5)

1. Add expense
2. View total expense
3. View total expense by category
4. View expense trend
5. View highest and lowest spend category
6. Exit

1

Enter a category

Food

Enter an amount

11.60

Enter a date (YYYY-MM-DD)

2025-12-12

Successfully added expense.

Test Data and Seed Data (Cont'd)

```
Select an option (1-5)
1. Add expense
2. View total expense
3. View total expense by category
4. View expense trend
5. View highest and lowest spend category
6. Exit
2
Total expenses: 83.43

Select an option (1-5)
1. Add expense
2. View total expense
3. View total expense by category
4. View expense trend
5. View highest and lowest spend category
6. Exit
3
== Total Expenses by Category ==
Entertainment : 50.00
Transport      : 9.33
Food          : 24.10

Select an option (1-5)
```

```
Select an option (1-5)
1. Add expense
2. View total expense
3. View total expense by category
4. View expense trend
5. View highest and lowest spend category
6. Exit
2
Total expenses: 83.43

Select an option (1-5)
1. Add expense
2. View total expense
3. View total expense by category
4. View expense trend
5. View highest and lowest spend category
6. Exit
3
== Total Expenses by Category ==
Entertainment : 50.00
Transport      : 9.33
Food          : 24.10

Select an option (1-5)
```

Date	Category	Amount
2025-12-12	Food	11.60
2025-12-12	Transport	1.79
2025-12-12	Transport	3.79
2025-12-13	Food	12.50
2025-12-13	Entertainment	25.00
2025-12-14	Transport	3.75
2025-12-15	Entertainment	25.00

4

5

Category with highest expense:
Entertainment : 50.00

Category with lowest expense:
Transport : 9.33

Select an option (1-5)
1. Add expense
2. View total expense
3. View total expense by category
4. View expense trend
5. View highest and lowest spend category
6. Exit
6

Thank you for using the Expense Tracker App!
Exiting...