Muhammad Huzaifa

SP 23691

BSSE-05-AFTERNOON

Software Construction and Development

Project Name: Expense Management System

Functional Requirements

1. Add Expense

- **FR1.1**: The system shall allow users to input expense details, including:
  - Amount (positive numeric value)
  - Category (e.g., Food, Travel, Rent)
  - Date (defaulting to current date, with manual override)
  - Optional description (text field).
- **FR1.2**: The system shall validate inputs (e.g., non-empty amount, valid date format).
- **FR1.3**: The system shall display a confirmation message upon successful expense addition.

2. View Expenses

- **FR2.1**: The system shall display a list of all expenses sorted by date (newest first).
- **FR2.2**: The system shall allow filtering expenses by:
  - Date range (e.g., last 7 days, monthly).
  - Category.
- **FR2.3**: The system shall display total expenses for the selected filter period/category.
- **FR2.4**: The system shall support sorting expenses by amount (ascending/descending).

3. Delete Expense

- **FR3.1**: The system shall allow users to select and delete an expense from the list.
- **FR3.2**: The system shall prompt for confirmation before deletion.
- **FR3.3**: The system shall update the expense list and totals after deletion.

4. Set Budget

- **FR4.1**: The system shall allow users to set a monthly budget for a specific category.
- **FR4.2**: The system shall validate budget input (positive numeric value).
- **FR4.3**: The system shall display a warning when expenses exceed 80% of the budget.
- **FR4.4**: The system shall highlight overspent categories in red.

5. Generate Report

- **FR5.1**: The system shall generate a visual report with:
- **FR5.2**: The system shall include summary statistics (total spent, budget comparisons).

6. Exit

- **FR6.1**: The system shall automatically save all data to a persistent storage file upon exit.
- **FR6.2**: The system shall close gracefully without errors.

Non-Functional Requirements

1. Usability

- **NFR1.1**: The user interface shall be intuitive, with clear labels and navigation (e.g., buttons for Add/Delete).
- **NFR1.2**: Error messages shall be descriptive (e.g., “Invalid amount: must be a positive number”).

2. Performance

- **NFR2.1**: The system shall load/display expenses within 2 seconds for up to 10,000 entries.
- **NFR2.2**: Reports shall generate within 5 seconds for datasets under 1,000 entries.

3. Reliability

- **NFR3.1**: The system shall auto-save data every 5 minutes to prevent data loss.
- **NFR3.2**: The system shall handle concurrent user actions without crashing (e.g., adding while generating a report).

4. Security

- **NFR4.1**: Expense data shall be stored in an encrypted file (if applicable).
- **NFR4.2**: User sessions shall time out after 15 minutes of inactivity.

5. Compatibility

- **NFR5.1**: The application shall run on Windows, macOS, and Linux (if desktop-based).

6. Maintainability

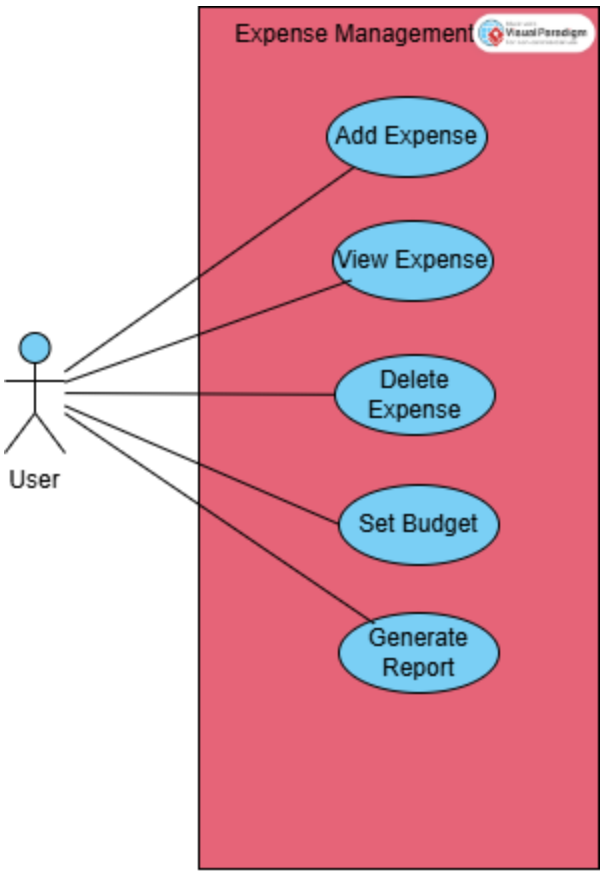
- **NFR6.1**: The code base shall follow modular design principles for easy updates (e.g., separate classes for UI, logic, and data).
- **NFR6.2**: Documentation shall include inline comments and a README file for setup instructions.

7. Scalability

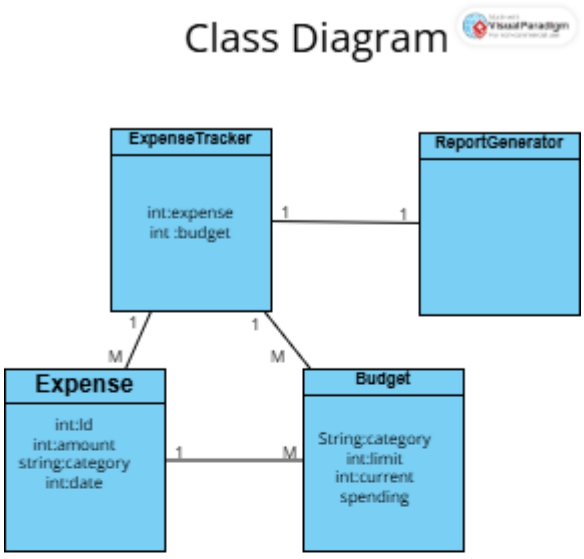
- **NFR7.1**: The system shall support up to 50,000 expenses without performance degradation.

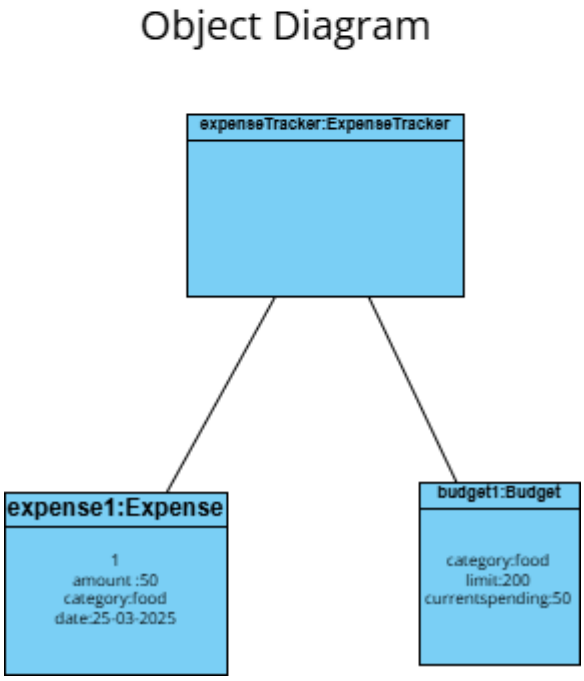
Use Case Diagram:

Expense Management System

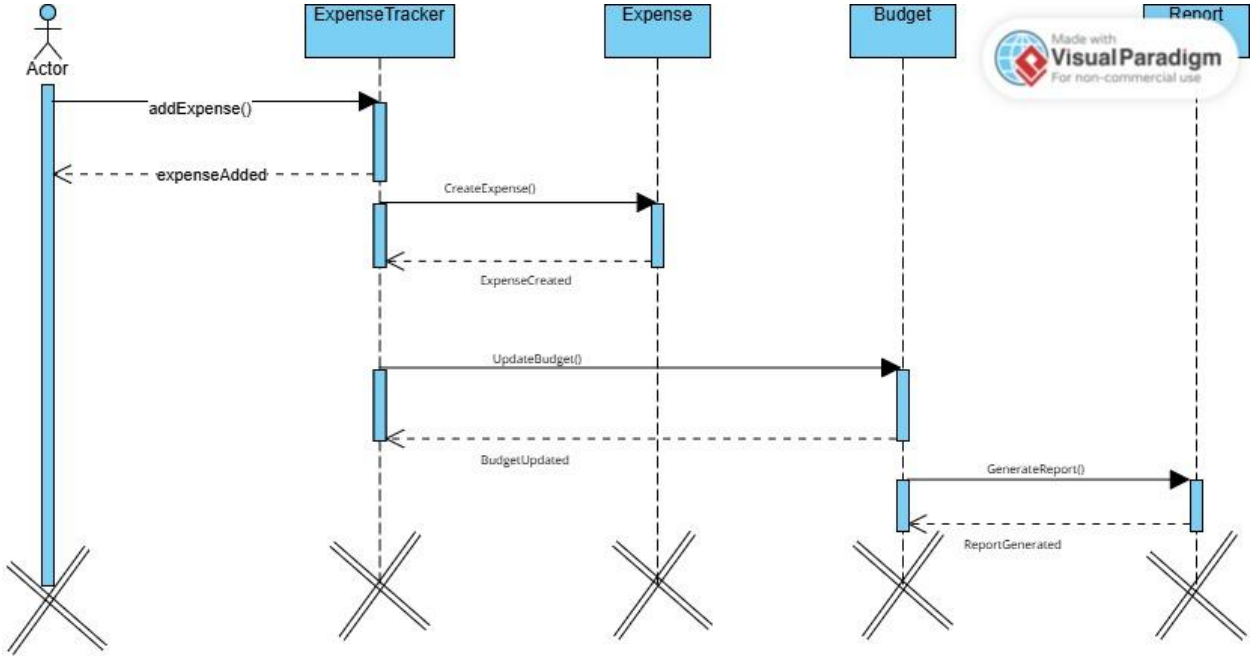


Class & Object Diagram:





Sequence Diagram:



Communication Diagram: