Expense Tracker

A Python Project

Made by Aryan Sharma

Reg. No. 25BAI11359

Problem Statement

- **Manual Error Prone:** Traditional methods like pen and paper are susceptible to calculation errors.
- **Data Loss:** Physical receipts and notebooks can be easily lost, damaged, or misplaced.
- **Lack of Insights:** It is difficult to manually analyze spending habits or visualize where money is going.
- **Time Consuming:** Recording every transaction manually is tedious and often neglected.

Project Objectives

Digitization

To create a persistent digital record of all financial transactions, eliminating physical clutter and ensuring data safety.

Visualization

To provide clear visual insights into spending patterns through charts and reports, aiding in better financial decisions.

Control

To empower users to set budgets, track expenses against limits, and maintain control over their personal finances.

Functional Requirements

- **User Authentication:** Secure login and registration functionality to protect user data.
- **Expense Management:** Ability to Add, Edit, View, and Delete expenses (CRUD operations).
- **Categorization:** Assign expenses to specific categories (e.g., Food, Transport, Utilities).
- **Budgeting:** Set monthly or category-wise budget limits and receive alerts.
- **Reporting:** Generate text-based or visual reports summarizing spending over a period.
- **Data Persistence:** Automatically save all records to a database or file system.

Non-functional Requirements

Performance & Reliability

The system should respond to user inputs within `1 second`. It must ensure data integrity, preventing corruption during save operations, and handle exceptions gracefully without crashing.

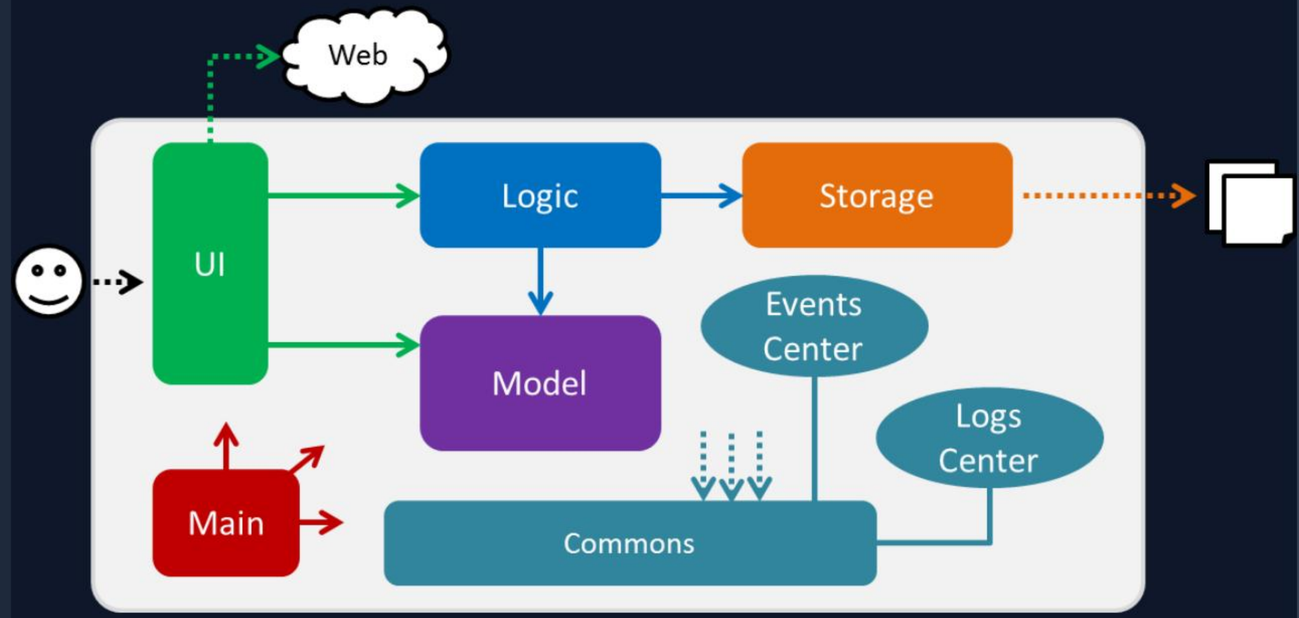
Security & Usability

User passwords must be hashed before storage. The interface (whether CLI or GUI) should be intuitive, requiring minimal training for a new user to navigate and log expenses.

System Architecture

The system follows a modular design pattern:

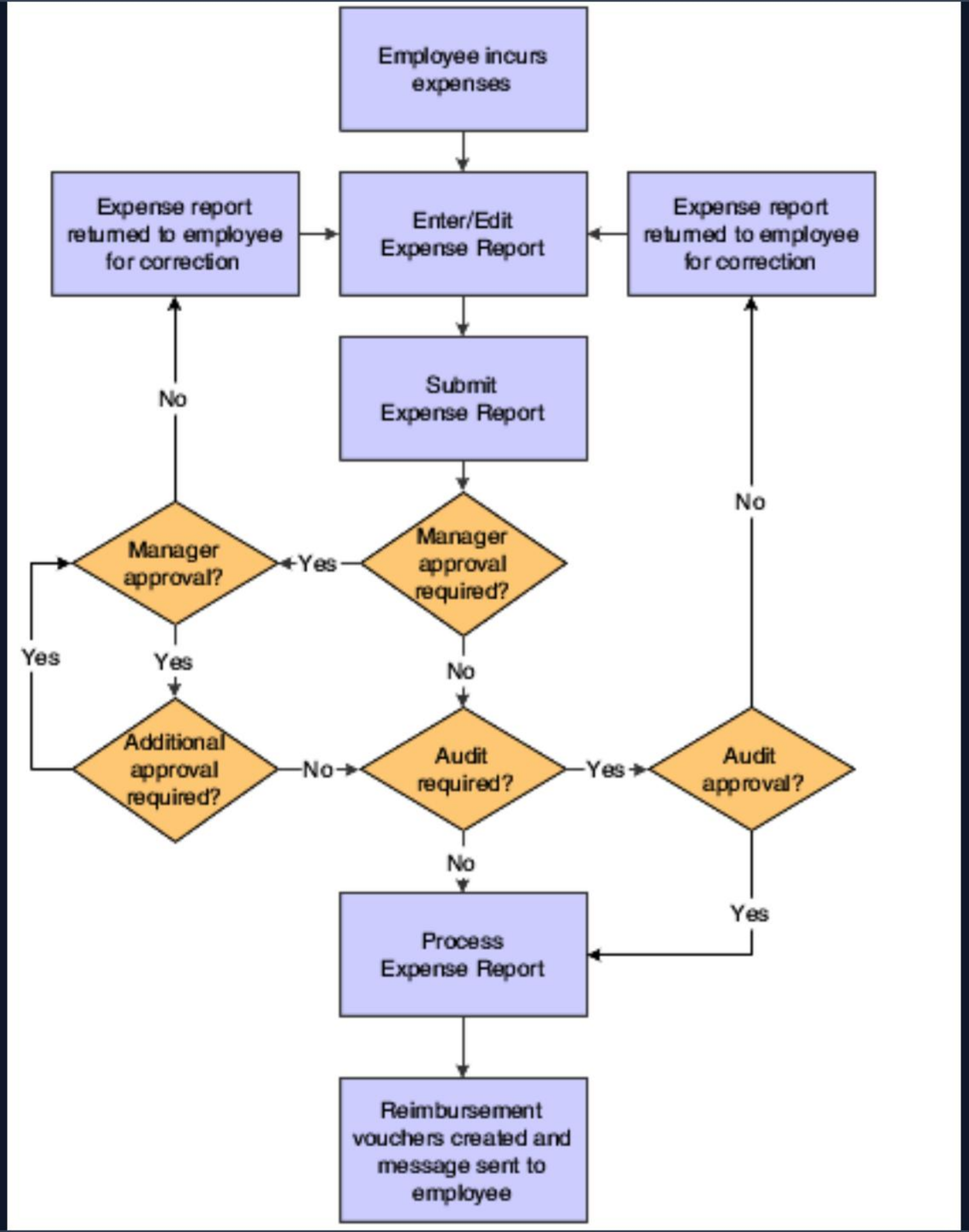
- **Presentation Layer:** Handles user interaction via CLI or GUI.
- **Application Layer:** Contains the core logic (Python classes) for processing data.
- **Data Layer:** Manages storage and retrieval using SQL or file systems.



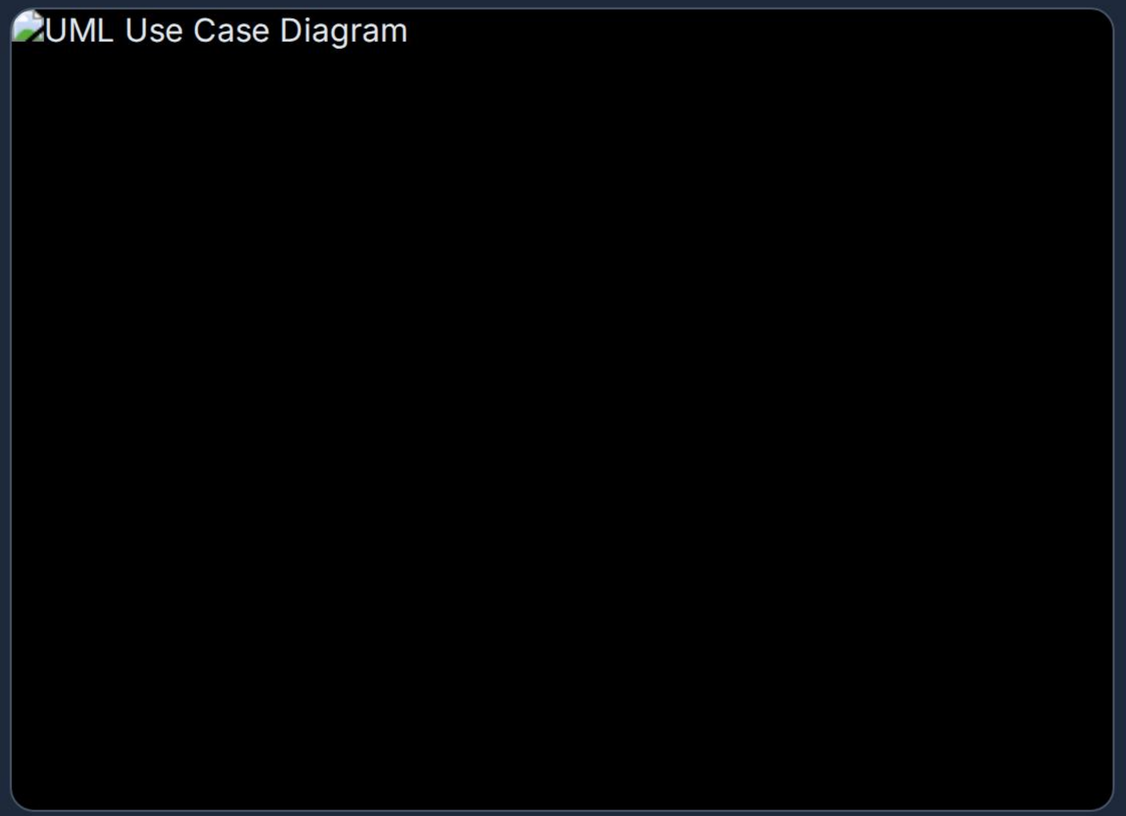
Process Flow

A visual representation of the user's journey through the application:

From the initial authentication to the main menu selection, users execute specific actions which are then processed and stored, returning feedback to the interface.



Use Case Diagram



Actor: User

The primary actor interacts with the system to perform the following use cases:

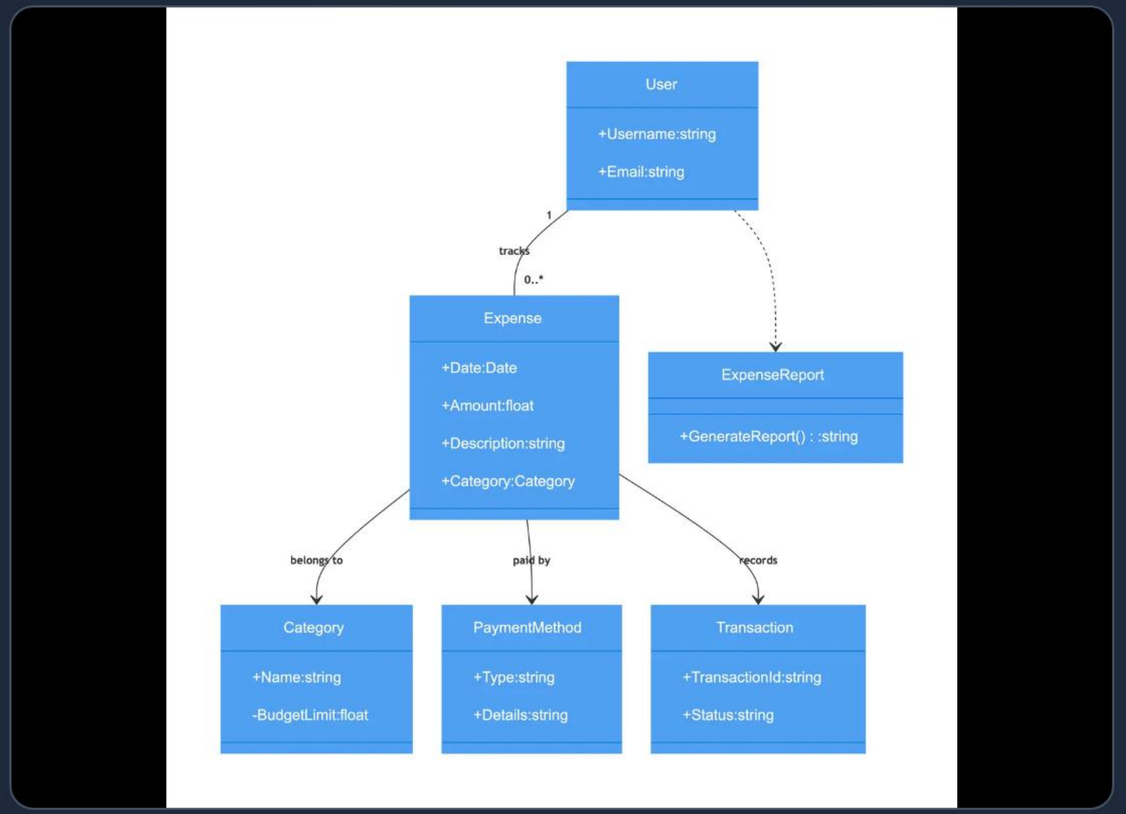
- **Manage Expenses:** Includes adding, modifying, and deleting records.
- **View Reports:** querying the system for summaries.
- **Manage Budget:** Setting and updating limits.
- **Authentication:** Logging in and out.

Class Diagram

Object-Oriented Structure

The system utilizes classes to encapsulate data and behavior:

- `User`: Manages credentials and preferences.
- `Expense`: Represents a single transaction entity.
- `Category`: Defines classification types.
- `DatabaseManager`: Handles database connections and queries.

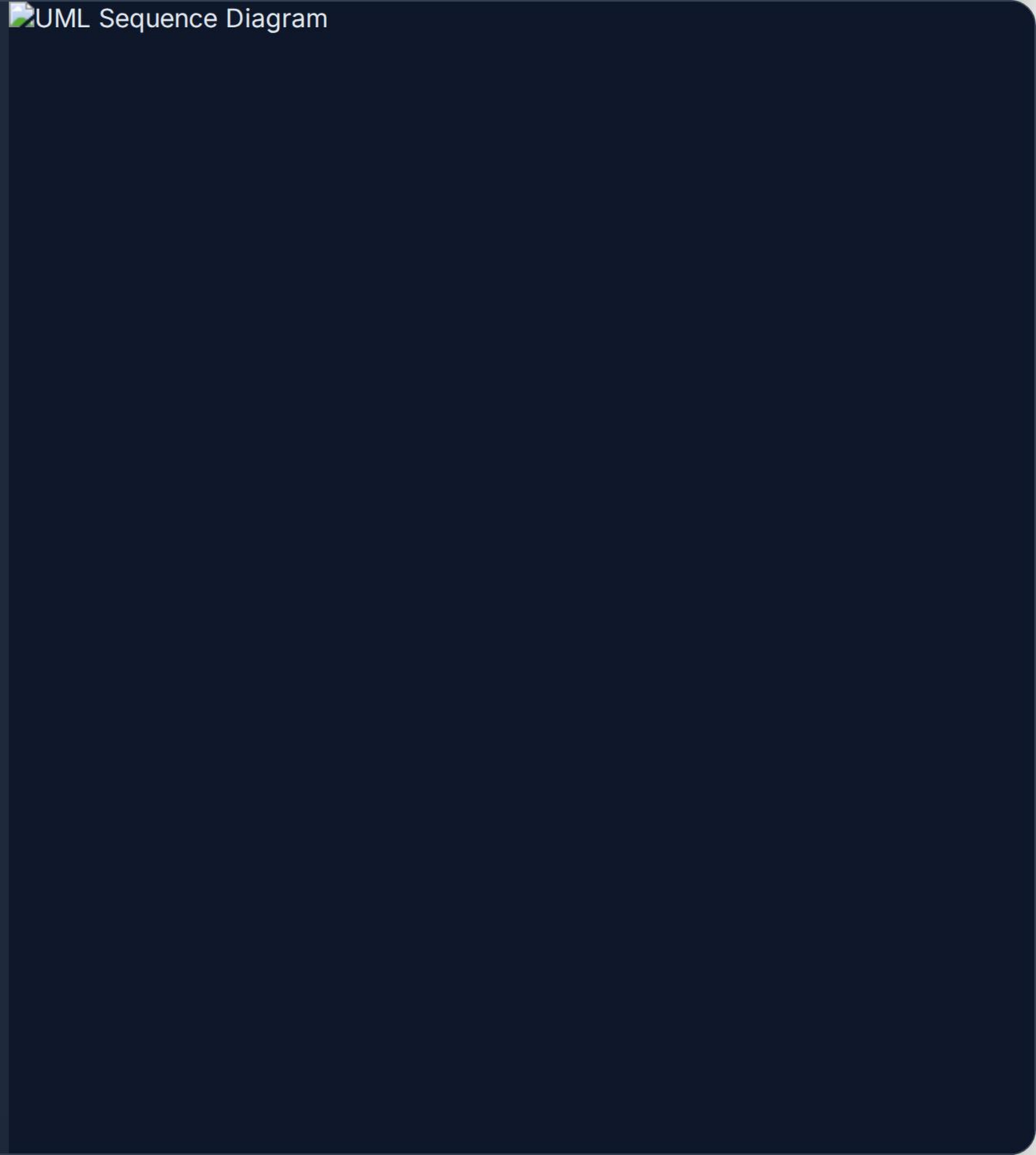


Sequence Diagram

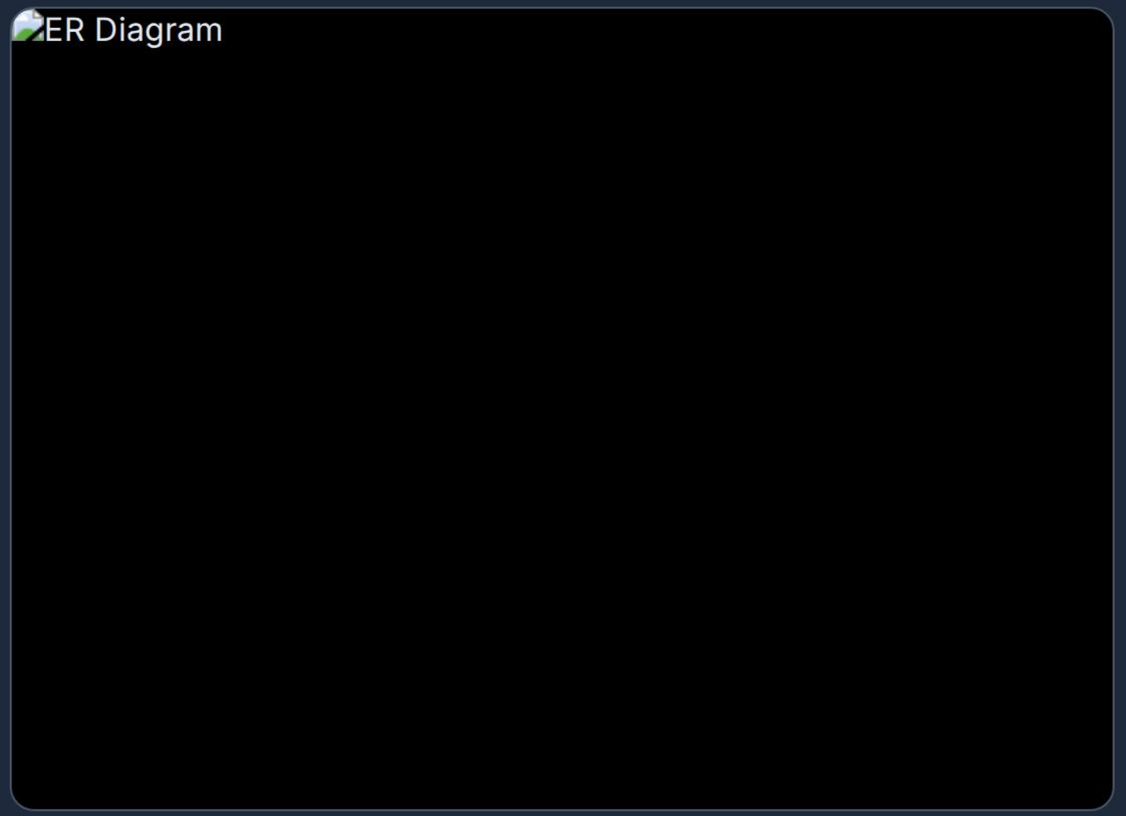
Add Expense Scenario

This diagram illustrates the time-ordered sequence of messages during an 'Add Expense' operation:

1. User inputs expense details.
2. Interface validates input.
3. Controller creates Expense object.
4. Database Manager persists object.
5. Confirmation returns to User.



Database & Schema Design



Schema Overview

The database consists of relational tables connected by foreign keys:

`Users Table`

ID (PK), Username, Password_Hash, Email

`Categories Table`

ID (PK), Name, Description

`Expenses Table`

ID (PK), User_ID (FK), Category_ID (FK), Amount, Date, Description

Questions?

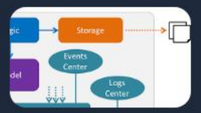
Thank you for reviewing the Expense Tracker Project.

Aryan Sharma | 25BAI11359

Image Sources

https://www.carsonthorncpa.com/wp-content/uploads/2013/09/Organizing-Your-Financial-Statements.jpg

Source: www.carsonthorncpa.com



https://cs2103-ay1819s1-t12-1.github.io/main/images/Architecture.png

Source: cs2103-ay1819s1-t12-1.github.io

https://docs.oracle.com/en/applications/jd-edwards/financials/9.2/eoaem/images/exp_mgmt_proc_flow_90.gif

Source: docs.oracle.com

https://i.sstatic.net/YPo4w.jpg

Source: stackoverflow.com

https://images.doclify.net/gleek-web/d/5f99beb1-c978-48a9-b2b7-d4219869c244.png?w=1200&format=webp

Source: www.gleek.io

https://cdn-us-04.visual-paradigm.com/node/on/w/wxapdleo/rest/diagrams/shares/diagram/db261d5a-fcf0-4596-a917-0e8c340ceefd/preview.png

Source: online.visual-paradigm.com

Image Sources

https://svg.template.creately.com/inabzh9t1

Source: creately.com