Fundamentals of DATABASE MANAGEMENT SYSTEMS

EXPENSE REPORTING SYSTEM

A CLI-BASED SOLUTION

KARMANDEEP SINGH || PARTH GALA || LAIM SHAW || RISHIKARTHIK VELLIANGIRI

Project Overview

- Objective: A command-line Expense Reporting System using Python & SQLite.
- Why it’s useful: Helps users track expenses, analyze spending, and manage finances efficiently.

Features & Functionality

- Multi-user authentication with roles (Admin & User)
- CSV import/export
- Expense management: Add, update, delete, view expenses
- Detailed expense reports
- Robust error handling

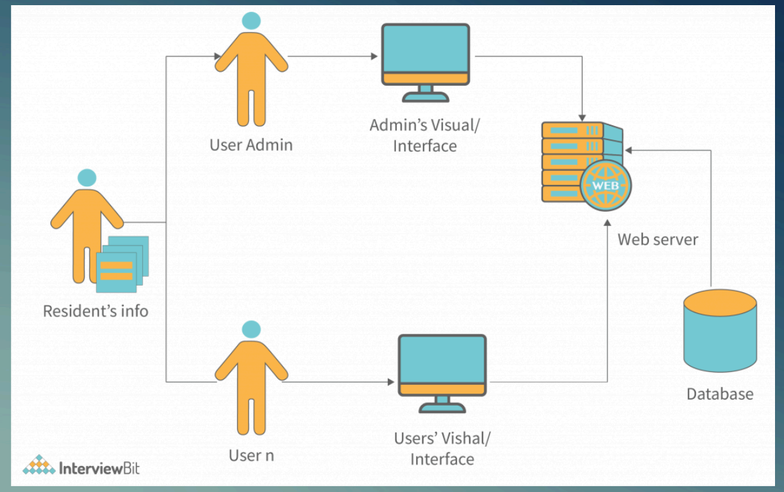
System Architecture

- **Tech stack:**

1. Python for backend logic
2. SQLite as the database
3. CLI for user interaction

- **Database structure:**

1. Tables: users, expenses, categories, payment_methods



Database Schema

1. User

Field	Type	Description
UserID	INT (PK)	Unique user id
Username	VARCHAR	Username (unique)
Password	VARCHAR	User password
Role	VARCHAR	User role (multi-valued: admin, user)

2. Expense

Field	Type	Description
ExpenseID	INT (PK)	Unique expense id
UserID	INT (FK)	References User(UserID); indicates which user recorded the expense
Category	VARCHAR	Expense category (stored as text; see relationship below for explicit relation)
Amount	DECIMAL	Expense amount
Date	DATE	Date of the expense (format: YYYY-MM-DD)
Tag	VARCHAR	Expense tag
Description	VARCHAR	Expense description (optional)
Payment_Method	VARCHAR	Payment method used (stored as text; see relationship below for explicit relation)

3. Category

Field	Type	Description
CategoryID	INT (PK)	Unique category id
Name	VARCHAR	Category name

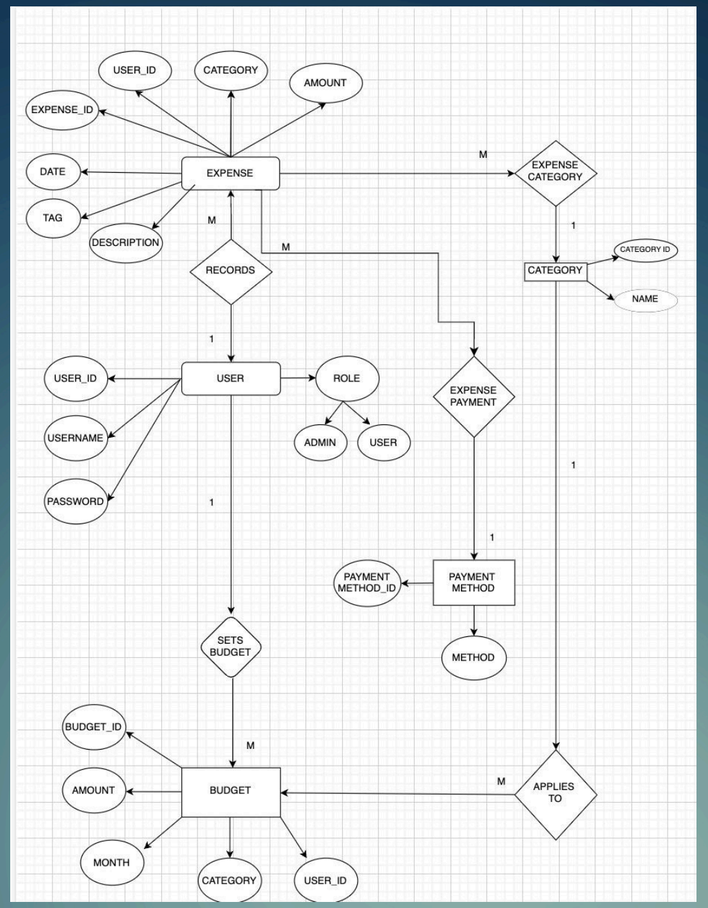
4. PaymentMethod

Field	Type	Description
PaymentMethodID	INT (PK)	Unique payment method id
Method	VARCHAR	Payment method name

5. Budget

Field	Type	Description
BudgetID	INT (PK)	Unique budget id
Amount	DECIMAL	Budget amount
UserID	INT (FK)	References User(UserID); indicates which user set the budget
Category	VARCHAR	Budget category (stored as text; see relationship below for explicit relation)
Month	VARCHAR	Month for the budget (format: YYYY-MM)

ER Diagram


USER_ID
CATEGORY
AMOUNT
EXPENSE_ID
DATE
EXPENSE
M
EXPENSE CATEGORY
TAG
DESCRIPTION
M
RECORDS
M
1
CATEGORY ID
CATEGORY
NAME
1
USER_ID
USER
ROLE
EXPENSE PAYMENT
USERNAME
ADMIN
USER
PASSWORD
1
1
1
PAYMENT METHOD_ID
PAYMENT METHOD
SETS BUDGET
METHOD
BUDGET_ID
M
AMOUNT
BUDGET
M
APPLIES TO
MONTH
CATEGORY
USER_ID

CLI Commands & Usage

```
login <username> <password>
add_expense <amount> <category> <payment_method> <date> <description>
list_expenses
report category_spending <category>
export_csv <file_path>
```

```
Expense Reporting System CLI. Type 'help' for available commands.
>> login admin admin123
Logged in as admin with role admin.
>>
>> add_expense 45.50 Food Cash 2024-04-01 "Lunch at cafe" meal
Expense added successfully!
>>
>> list_expenses
(1, 1, 'Food', 25.5, '2024-04-01', 'Lunch at cafe', 'meal', 'Cash')
(2, 1, 'Food', 100.0, '2024-04-05', 'Groceries', 'shopping', 'Cash')
(3, 1, 'Food', 150.0, '2024-04-10', 'Restaurant', 'meal', 'Cash')
(4, 1, 'Food', 12.5, '2024-04-01', 'Lunch at local cafe', 'meal', 'Cash')
(5, 1, 'Groceries', 75.0, '2024-04-03', 'Weekly grocery shopping at supermarket', 'groceries', 'CreditCard')
(6, 1, 'Transport', 25.0, '2024-04-05', 'Taxi ride for morning commute', 'commute', 'Cash')
(7, 1, 'Utilities', 120.0, '2024-04-07', 'Monthly electricity bill', 'home', 'BankTransfer')
(8, 1, 'Entertainment', 30.0, '2024-04-09', 'Movie tickets for a night out', 'entertainment', 'CreditCard')
(9, 1, 'Healthcare', 50.0, '2024-04-11', 'Monthly gym membership fee', 'fitness', 'CreditCard')
(10, 1, 'Education', 200.0, '2024-04-15', 'Fee for an online course', 'education', 'CreditCard')
(11, 1, 'Travel', 450.0, '2024-04-20', 'One-way flight ticket', 'travel', 'CreditCard')
(12, 1, 'Food', 45.5, '2024-04-01', 'Lunch at cafe', 'meal', 'Cash')
>>
>> report category_spending Food
Total spending for Food: 333.5
>>
>> export_csv data/exported_expenses.csv sort-on date
Data exported to data/exported_expenses.csv
```

Extra Features

Data Visualization

Convert expense reports into Pie Chart for better insights.
Use libraries like Matplotlib to visualize spending patterns.

Budget Tracking

Allow users to set monthly budgets for different categories.
Warn users when spending exceeds the budget with an alert

Expense Trends

Analyze spending patterns over time (weekly/monthly/yearly).
Identify categories where expenses are increasing or decreasing.

Key Challenges & Solutions

1. MULTI-USER ROLE MANAGEMENT & DATA RESTRICTION

CHALLENGE:

- INITIALLY, ALL USERS (INCLUDING REGULAR USERS) COULD ACCESS ALL REPORTS, EVEN THOSE MEANT ONLY FOR THE ADMIN. THIS VIOLATED ROLE-BASED ACCESS CONTROL (RBAC).

SOLUTION:

- WE UPDATED THE QUERY LOGIC IN OUR REPORTS.PY AND OTHER FUNCTIONS TO FILTER DATA BASED ON THE LOGGED-IN USER'S ID UNLESS THEY WERE AN ADMIN. NOW, NON-ADMIN USERS CAN ONLY SEE THEIR OWN EXPENSES.

2. EXPENSE DATA INTEGRITY & VALIDATION

CHALLENGE:

- USERS COULD ENTER INVALID DATA, SUCH AS NEGATIVE AMOUNTS, INCORRECT DATE FORMATS, OR NON-EXISTENT CATEGORIES, LEADING TO DATABASE INCONSISTENCIES.

SOLUTION:

WE ADDED:

- STRICT VALIDATION CHECKS BEFORE INSERTING DATA.
- ERROR HANDLING TO CATCH INCORRECT INPUTS AND GUIDE USERS TO ENTER VALID DATA.
- FOREIGN KEY CONSTRAINTS IN THE DATABASE SCHEMA TO PREVENT ORPHANED RECORDS.

Learnings & Takeaways

TECHNICAL SKILLS GAINED:

- IMPLEMENTING SQLITE WITH PYTHON FOR DATABASE MANAGEMENT.
- USING SQL QUERIES FOR CRUD OPERATIONS & REPORTS.
- HANDLING USER AUTHENTICATION & ROLE-BASED ACCESS CONTROL.
- IMPLEMENTING CSV IMPORT/EXPORT & DATA VALIDATION.
- ERROR HANDLING & DEBUGGING IN CLI-BASED APPLICATIONS

BROADER LEARNINGS:

- IMPORTANCE OF DATABASE DESIGN IN STRUCTURING INFORMATION.
- HOW REAL-WORLD EXPENSE MANAGEMENT SYSTEMS WORK.
- COLLABORATION & PROBLEM-SOLVING DURING DEVELOPMENT.
- WRITING MODULAR & MAINTAINABLE CODE.

THANK YOU