High Level Design (HLD) of "Expenditure distributor Management"

Author: Aakash Bisen

HLD documentation version: 1.6

Last version updated: 23/03/2023

Document Version Control

Date Issued	Version	Description	Author
08-02-2022	1.0	Initial HLD Draft – V1.0	Aakash Bisen
25-03-2022	1.1	Added workflow process	Aakash Bisen
18-05-2022	1.2	Restructure and reformat HLD	Aakash Bisen
28-05-2022	1.3	Changed tool used and updated the same in HLD	Aakash Bisen
01-08-2022	1.4	Updated workflow process	Aakash Bisen
28-01-2023	1.5	Restructuring & Alignment	Aakash Bisen
23-03-2023	1.6	Final HLD	Aakash Bisen

Contents

Abstract

Expense tracking is utmost important these days especially in creating your financial budget & in investment strategies. Expense tracking helps you take control of your finances and stay on your budget. Keeping a daily record of your expenses by tracking receipts, invoices, and other outgoing expenses improves your financial health. Tracking expenses can also help you stay on top of your cash flow and prepare you for tax season. This application distributes the entire expenditure among the participants and track who is investing where. Each user can see the statistics for their past expenses.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	*Description*
EDM	Expenditure distributor Management
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
AWS	Amazon Web Services

2 General Description

2.1 Product Perspective

This application distributes the expenditure among participants and track who is investing where.

2.2 Problem statement

Create an application which distributes the expenditure among participants. Implement the below features in your application.

1) Allow users to create a group of people.
2) Allow each user to add their expense.
3) The expense must be approved among other group members.
4) Once a task is done allow group members to distribute expenses evenly.
5) Try to implement an online payment gateway if feasible so that members can easily return money to another member at the same portal.
6) Allow users to see the statistics for their past expenses.

2.3 Proposed solution

Basically this project requires you to build an application that allows users to create a group, track their expenses, get them approved by other group members, distribute expenses fairly, make payments to each other, and view past expenses statistics.

2.4 Further improvements

The application has enriched with online dedicated payment gateway feature, which allow members to pay each other at the same portal. This feature makes it easy for group members to repay each other for expenses without having to use a separate payment platform. Also this application can be used to allow users to see the statistics for their past expenses to manage & their financial plan or budget accordingly.

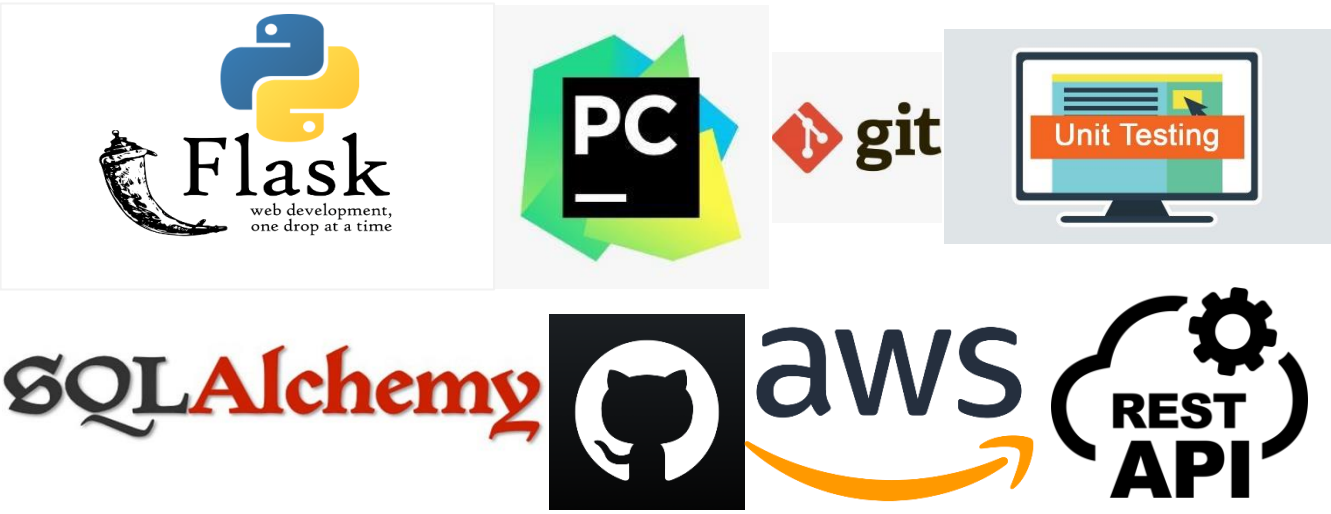
2.5 Requirements

The application requirements are as follows:

1) Allow users to create a group of people.
2) Allow each user to add their expense.
3) The expense must be approved among other group members.
4) Once a task is done allow group members to distribute expenses evenly.
5) Try to implement an online payment gateway if feasible so that members can easily return money to another member at the same portal.
6) Allow users to see the statistics for their past expenses.

2.6 Tools used

- PyCharm is used as IDE.
- Python Flask is used for backend development.
- Flask-Testing to provide unit testing utilities for Flask.
- unittest2 module to provide a rich set of tools for constructing and running tests.
- AWS is used for deployment of the model.
- SQLAlchemy is used to retrieve, insert, delete, and update the database.
- GitHub is used as version control system.
- The complete solution is exposed as an REST API.



2.7 Constraints

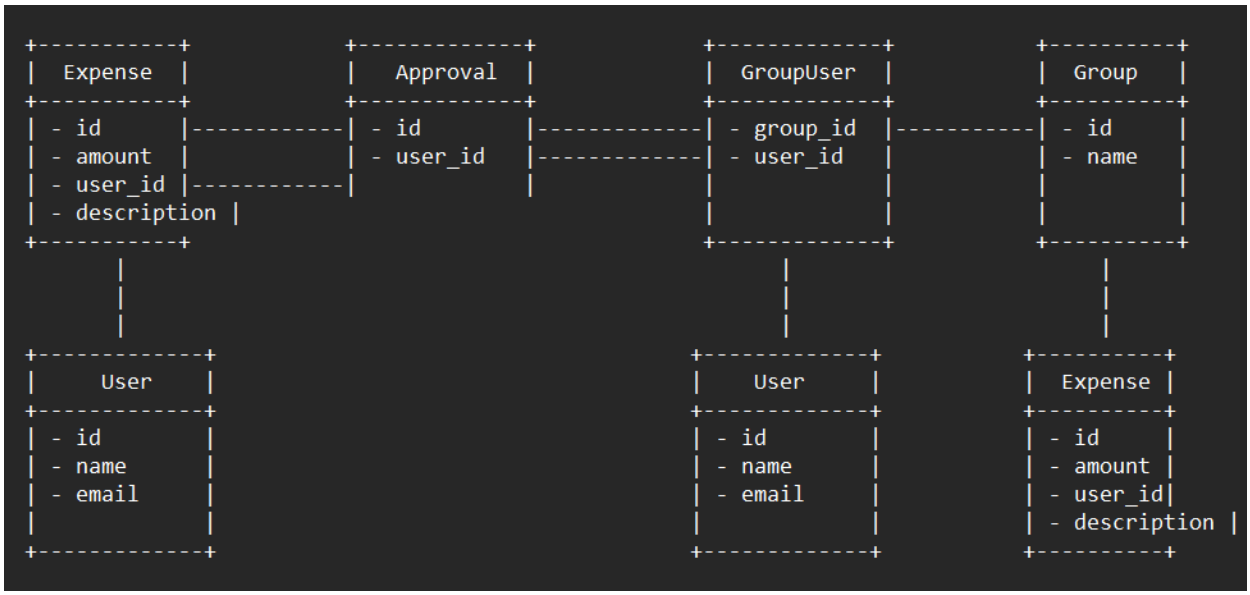
This solution system must be user friendly, as automated as possible and users should not be required to know any of the workings.

2.8 Assumptions

The main objective of this project is to implement distributor application (2.2 Problem Statement) for users where each user has permissions to create a group. Flask-Testing and unittest2 are used for running tests on it. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

3 Design Details

3.1 Process Flow:



In this diagram, the boxes represent the classes/entities, and the arrows represent the relationships between them. Here's a brief description of each entity:

- **User:** represents a user of the expense service. Each user has a unique id, a name, and an email.
- **Expense:** represents an expense that a user has incurred. Each expense has a unique id, an amount, a description, and a user_id field that references the User that incurred the expense.
- **Approval:** represents an approval for an expense by a user. Each approval has a unique id, an expense_id field that references the Expense being approved, and a user_id field that references the User giving the approval.
- **GroupUser:** represents the membership of a user in a group. Each GroupUser has a group_id field that references the Group that the user is a member of, and a user_id field that references the User that is a member of the group.
- **Group:** represents a group of users who share expenses. Each group has a unique id, and a name.

(**Note:** The relationships between the entities are represented by the arrows in the diagram. For example, each Expense is associated with a User, and this is represented by the arrow pointing from Expense to User. Similarly, each Approval is associated with an Expense and a User, and this is represented by the arrows pointing from Approval to Expense and User, respectively

3.2 Event log

The application code has implemented log of every event so that the user will know what process is running internally.

3.3 Error Handling

Errors are encountered with exceptions and explanation is displayed as to what went wrong. An error will be defined as anything that falls outside the normal and intended usage.

3.4 Performance

The application to distribute expenditure among participants and track who is investing where. Flask-Testing, unittest2 are used for testing the application to ensure that application work at optimized level.

3.5 Reusability

The code written and the components used has the ability to be reused with no problems.

3.6 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

3.7 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

3.8 Deployment

4 Conclusion

The application of distribution management has been successfully created. It'll distribute expenditure among participants and also allows users to create a group, track their expenses, get them approved by other group members, distribute expenses fairly, make payments to each other, and view past expenses statistics.