Design Document

ShareWize: Expense Sharing Web App

Group 24

Arun Rathaur

Ayaz Lockhat

Mubtasim Rahman

Veeral Patel

Jan 24, 2024

ShareWize

SHARING MADE SIMPLE

Revision History

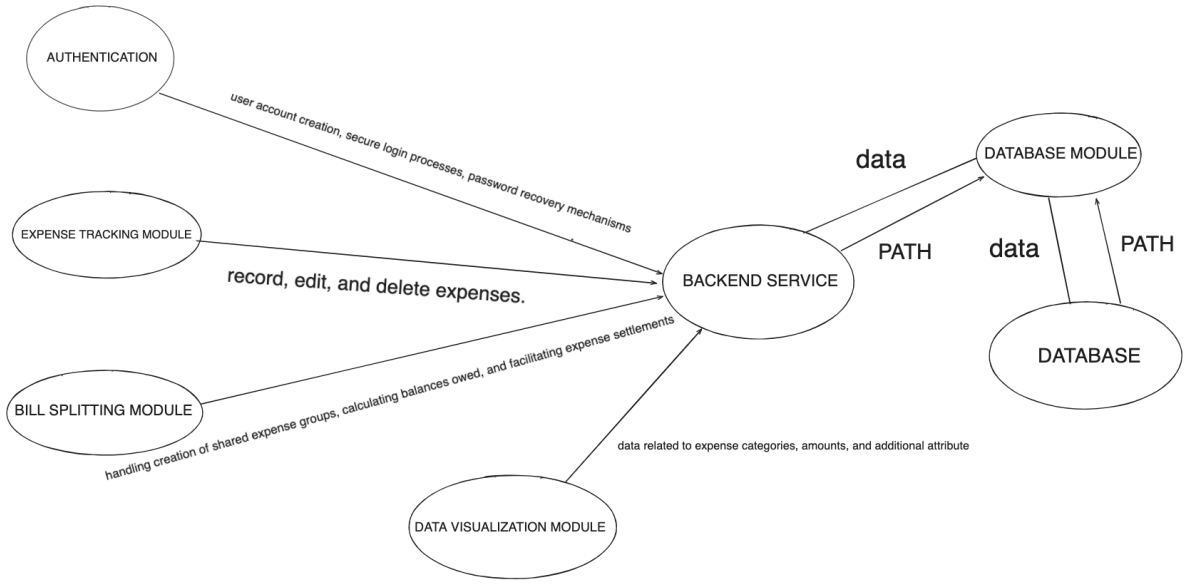
Version	Date	Change	Revision
0	Jan 24, 2024	-	-

1. Purpose of Project

1.1. Goal of the Project and document

The project aims to create an efficient and user-centric expense-sharing solution, streamlining bill-splitting and expense tracking for enhanced financial transparency within social circles. The primary goal is to empower users in managing shared finances seamlessly, with a potential exploration of revenue streams for sustainable service enhancement. This document serves to communicate the system's design, detailing its interactions with the environment, component breakdown, timing constraints, potential build issues, and definitions of normal operation and undesired events, providing stakeholders with a clear understanding of the project's overall design and objectives.

1. Component Diagram



2. Components Relationships

2.1. Authentication:

2.1.1. Rationale

- The Authentication component is responsible for user account creation, secure login processes, and password recovery mechanisms, ensuring the fulfilment of the authentication requirement. Additionally, it manages user roles, distinguishing between regular users and administrators.

2.2. Bill Splitting Logic:

2.2.1. Rationale

- The Bill Splitting Logic is integral to the Expense Sharing feature, handling the creation of shared expense groups, calculating balances owed among group members, and facilitating expense settlements. It ensures fairness in expense distribution, meeting the requirements of the Expense Sharing feature.

2.3. Data Visualization:

2.3.1. Rationale

- The Data Visualization Module is responsible for presenting expense data in visual representations such as pie graphs. It utilizes data related to expense categories, amounts, and additional attributes for graph generation, aligning with the Data Visualization requirement.

2.4. Expense Tracking Module:

2.4.1. Rationale

- The Expense Tracking Module fulfils the Expense Tracking requirement by allowing users to record, edit, and delete expenses. It also supports the generation of expense summaries and the attachment of images or receipts.

2.5. Database:

2.5.1. Rationale

- The Database component meets the Information Storage requirement by securely storing user profiles, authentication data, expense records, and group information. It ensures data integrity, compliance with data protection regulations, and provides backup mechanisms.

3. Components Description:

3.1. Authentication(P0)

3.1.1.

- **Normal Behaviour:** Users can create accounts, log in securely, recover passwords, and have different roles assigned (e.g., regular user, administrator).
- **API:**
  - **Input:** User credentials (username, password), role designation.
  - **Output:** Authentication status (success/failure), user roles.
- **Implementation:**
  - **Classes:** GoogleOAuthProvider, axios

    - **Methods:** Login, handleLoginSuccess, jwtDecode, axios.post, googleLogin
    - **Relationships:** User interacts with AuthenticationManager for account-related actions. The google authentication will give us a response with many attributes that will allow us to identify the user and secure the user's information
- **Undesired Outcomes:**
    - Incorrect role assignment leading to inappropriate access.
    - Password recovery failures.

3.2. Bill Splitting Logic:

3.2.1.

- **Normal Behaviour:** Users can create shared expense groups, add expenses, specify participants, and settle balances within the group.
- **API:**
    - **Input:** Group details, expense details, participant information.
    - **Output:** Balances among group members, settlement confirmations.
- **Implementation:**
    - **Classes:** ExpenseGroups, Expense, GroupMember
    - **Methods:** addExpense, settleBalance, calculateShares,
    - **Relationships:** An ExpenseGroup has multiple Expense instances. An Expense has a list of Participants who owe a share of the expense. Each ExpenseGroup maintains a list of groupMembers (participants).
- **Undesired Outcomes:**
    - Incorrect calculations leading to imbalances.
    - Group creation or settlement failures.
    - Participants not properly accounted for.

3.3. Data Visualization:

3.3.1.

- **Normal Behaviour:** Presents expense data in visual representations (e.g., pie graphs) based on categories and amounts.
- **API:**

    - **Input:** Expense categories, amounts, additional attributes for graph generation.
    - **Output:** Visual representations of expense data.
- **Implementation:**
    - **Classes:** Visualization,
    - **Methods:** generatePiChart(data), generateBarChart(data), generateLineChart(data), displayVisaulization(visualization)
    - **Relationships:** Other components interact with Visualization to request visual representations of expense data.
- **Undesired Outcomes:**
    - Inaccurate visualisations.
    - Graph generation failures.
    - Visuals not reflecting the actual expense data.

3.4. Expense Tracking Module:

3.4.1.

- **Normal Behaviour:** Users can record, edit, and delete expenses. The module provides summaries of expenses and supports attaching images or receipts.
- **API:**
    - **Input:** Expense details (date, description, category, amount), edit/delete requests, image files.
    - **Output:** Visual confirmation of record/edit/delete actions, expense summaries.
- **Implementation:**
    - **Classes:** ExpenseTracker (group specific)
    - **Methods:** recordExpense(expenseDetails), editExpense(expenseId, updatedDetails), deleteExpense(expenseId), getExpenseSummaries(), attachImageToExpense(expenseId, imageFile)
    - **Relationships:**The ExpenseTracker class is responsible for managing and manipulating expense data. Other components in the system may interact with the ExpenseTracker to record, edit, delete expenses, and retrieve expense summaries.
- **Undesired Outcomes:**
    - - Inaccurate recording or editing of expenses.
    - - Image attachment failures.
    - - Loss of expense data.

3.5. Database:

3.5.1.

- **Normal Behaviour:** Secure storage and retrieval of user profiles, authentication data, expense records, and group information.
- **API:**
  - **Input:** User data, authentication data, expense records, group information.
  - **Output:** Retrieved data for user authentication, expense tracking, and group management.
- **Implementation:**
  - **Classes:**db
  - **Methods:**storeUserProfile(userData), authenticateUser(username), storeExpenseRecord(expenseData), retrieveExpenseRecords(userId), storeGroupInformation(groupData), retrieveGroupInformation(userId)
    Relationships: Other components interact with db for data storage and retrieval.
- **Undesired Outcomes:**
  - Data corruption or loss.
  - Unauthorised data access.
  - Slow data retrieval affecting system performance.

4. UI Details:

The user interface for our financial application is designed to provide an intuitive and visually appealing experience. When launching the app, users can see a sleek home screen featuring a modern logo and a clean navigation menu icon for easy access. The top section of the screen displays a detailed overview, including the user's total budget, recent transactions, and a visually engaging representation of their expense categories. Quick actions that can be taken by our users are conveniently placed with buttons, allowing users to easily add expenses, create groups, or access detailed reports.

1. Dashboard:
 - Description: The dashboard serves as the central hub for users to view a summary of their expenses and group activities.
 - Components:
  - Expense summary charts (pie chart, bar chart).
  - Quick links to recent expenses and groups.

2. Expenses:
  - Description: Users can record, edit, and delete expenses through a dedicated section.
  - Components:
    - Form for adding a new expense (fields for date, description, category, amount).
    - List of recent expenses with options to edit or delete.
    - Attachment option for receipts/images.

3. Groups:
  - Description: Users can create, join, and manage expense-sharing groups.
  - Components:
    - Create a new group form.
    - List of existing groups with options to view details or join.
    - Group details page showing members and shared expenses.

4. Profile:
  - Description: Users can view and edit their profile information.
  - Components:
    - Profile picture and basic information.
    - Edit profile form.

Focusing on group management, a dedicated "Create Group" button takes center of their attention, encouraging users to initiate new expense-sharing circles effortlessly. Existing groups are showcased, featuring the group name, a distinctive avatar representing the group, and a concise summary of shared expenses. This design not only prioritizes functionality but also aims to enhance the user experience through a harmonious blend of aesthetics and usability.

One of the prominent colours in our application is green because many people associate green with money and since the idea behind our applications revolves around finances, we thought it was a good fit for the colour. We paired this colour with dark grey because it creates a strong contrast and adds professionalism to the website. We chose our font design to be simple so it could be easily read without distractions. Our goal is to create an easy interaction for the user. We created our website the way it is so the users can seamlessly interact with it, meaning they can login and then go to their expense group or create expenses. This adds continuity to the application and the directions for the user.

- Responsive Design: Ensure the UI is responsive for various screen sizes and devices.
- Intuitive Navigation: Use clear navigation patterns, such as a top bar or sidebar, for easy access to different sections.
- Charts: Make the charts interactive for a better user experience. Users can hover over or tap on chart elements for more details.