Software Requirements Specification

for

Fair Share

Version 1.0 approved

Prepared by

Aditya Kirti, Himanshu Jangid, Divya Kesarkar

Thakur College of Engineering & Technology

February, 2025

Table of Contents

Table of Contents i

Revision History i

[1. Introduction 2](#_Introduction)

1.1 [Document Purpose 2](#_Document_Purpose)

1.2 [Product Scope 2](#_Product_Scope)

[1.3 Definitions and Acronyms 2](#_Definitions_and_Acronyms)

[1.4 Document Conventions 1](#_Document_Conventions)

[1.5 References and Acknowledgement 2](#_References_and_Acknowledgements)

[2. Overall Description 3](#_Overall_Description)

2.1 [Product Description 2](#_Product_Description)

[2.2 Product Functionality 2](#_Product_Functionalities)

[2.3 Users 2](#_Users)

[2.4 Operating Environment 3](#_Operating_Environment)

[2.5 Design and Implementation Constraints 3](#_Design_and_Implementation_Constrain)

[2.6 Assumptions and Dependencies 3](#_Assumptions_and_Dependencies)

[3. Specific Requirements 3](#_Specific_Requirements)

[3.2 Functional Requirements 4](#_Functional_requirements)

[3.3 Behavioral Requirements 7](#_Behavioral_requirements:)

[4. Other Nonfunctional Requirements 7](#_Other_Nonfunctional_Requirements)

[4.1 Performance Requirements 7](#_Performance_Requirements)

[4.2 Safety and Security Requirements 7](#_Safety_and_security_Requirements)

[4.3 Software Quality Attributes 8](#_Software_Quality_Attributes)

[Appendix A: Glossary 8](#Gloassary)

Appendix B: Data Dictionary 9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| - | - | - | - |

# Introduction

## Document Purpose

The purpose of this document is to define the functional and non-functional requirements of the Expense Splitter application. The system enables users to split expenses efficiently, manage debts, and track payments among friends, family, or colleagues. It ensures ease of use with multiple features designed to simplify expense management.

## Product Scope

Expense Splitter is a web application that allows users to share and settle expenses fairly. It supports multiple currencies, integrates with payment applications, and offers social connectivity features. The application is designed for groups who need to manage shared expenses, such as roommates, travel groups, and event organizers.

## Definitions and Acronyms

* **User**: An individual who registers to use the Expense Splitter application.
* **Debt Management**: The feature that keeps track of how much each user owes or is owed.
* **Multi-currency Support**: The ability to handle transactions in different currencies with automatic conversion.
* **Recurring Expenses**: Periodic expenses that are automatically added to the system.
* **PDF Export**: A feature that allows users to download expense reports in PDF format.

## Document Conventions

Throughout the document (formatted in MS Office ‘03) font used for:

### Topics are Times New Roman formatted in ‘Heading 2’ style. Font size for headings is 18

### Sub topics are Times New Roman formatted in ‘Heading 2’ style. Font size for sub-topics is also 14

### Text is Times New Roman formatted in ‘Normal’ style. Font size for text is 12

### Italics have been used for laying special emphasis on certain information.

### All references to the websites used are hyperlinked in Times New Roman, Normal + Times style and size 12.

## References and Acknowledgements

* <http://en.wikipedia.org/wiki/Non-functional_requirement>
* http://www.mrl.nott.ac.uk/~sdb/g52hci/topic%206%20-%20requirements/SRSTemplate.doc

# Overall Description

## Product Description

The **Expense Splitter** is an intuitive software for managing group expenses, offering features like multi-currency support, debt tracking, and payment gateway integration. Unlike traditional tools, it emphasizes social connectivity, group event tracking, and a flexible interface. Designed for efficiency, it works well on devices with low memory capacity while providing an adaptable, user-centric platform for seamless expense management.

## Product Functionalities

### The application shall allow users to securely and easily access their accounts, enabling them to manage their expenses and perform transactions.

### The system should allow interaction with external devices and services via networking capabilities, such as integration with mobile wallets and payment systems.

### The application shall be user-friendly and designed for ease of access, offering features like notifications, expense history, and reminders for timely payments.

### The application aims to offer free, online financial management services to users with internet access, ensuring they can efficiently manage group expenses across various contexts.

## Users

The users of the **Expense Splitter** application are expected to have a basic understanding of financial terminology related to transactions and expenses, such as "debt," "expense splitting," and "currency conversion." Users should also be familiar with digital payment systems and how to use them to settle shared costs effectively.

## Operating Environment

The **Expense Splitter** application will be designed to work across multiple platforms and will be accessible on web browsers such as Chrome, Firefox, Safari, and Internet Explorer (version 5 and above). It is intended for use on both personal computers and mobile devices. The system will be optimized for use in homes, workplaces, or on-the-go, ensuring that users have access to all necessary tools for managing group finances in one place.

## Design and Implementation Constraints

The **Expense Splitter** app will adhere to strict privacy and security guidelines, ensuring that all user data and transaction records are securely stored and cannot be easily shared or accessed without proper authorization. The application will be designed using **HTML, JavaScript, PostgreSQL, and other web technologies**, ensuring compatibility across devices and browsers. Only registered users will be allowed to access certain features, such as creating and managing expense groups, and making payments or adjustments. The application will be available in **English** by default, though future localization is possible based on demand.

## Assumptions and Dependencies

The application assumes that the user has a stable internet connection, as it relies on online transactions and cloud-based services to manage expenses. The system assumes that users have sufficient device storage space to save transaction history, notifications, and other essential data. Full functionality of the **Expense Splitter** app is dependent on the availability of the internet. Without an internet connection, users will not be able to sync their data or make real-time adjustments to their expenses.

# Specific Requirements

## External Interface Requirements

### Hardware Interfaces

The system should support various client devices, including desktops, laptops, and mobile devices (smartphones, tablets). It must be compatible with input devices like keyboards, mice, and touchscreens, and output devices such as displays and printers. Optional peripherals like barcode scanners and QR code readers should also be supported for specific use cases.

### Software Interfaces

The system must use web technologies like HTML5, CSS3, and JavaScript frameworks (e.g., React) for the frontend, and server-side technologies like Node.js for backend operations. It should integrate with payment gateways (e.g., Stripe), support file exports (PDF, CSV), and allow seamless data communication via RESTful APIs for external integrations.

### Communications Interfaces

Communication between the client and server should be secure over HTTPS, ensuring encrypted transactions. Real-time features should be enabled via Web Sockets, with API integrations for third-party services. The system should support email and push notifications for user alerts and reminders, while using web hooks for real-time updates from external services.

## Functional requirements

|  |  |
| --- | --- |
| Use Case Name: | Register |
| Actors: | User |
| Pre Condition: | The new user should not have a previously owned account. |
| Post Condition: | The user will have registered successfully. |
| Scenario: | 1. The User clicks on sign up. 2. The system provides the User with the registration form. 3. The User has to fill the correct details in the form. 4. If the information is valid, User is successfully registered. |
| Exception: | Occurs when the user has input an invalid entry into the form.  Can also occur if the user has chosen a ID already owned by an existing user. |

|  |  |
| --- | --- |
| Use Case Name: | Login |
| Actors: | User |
| Pre Condition: | The actor should be a registered user. |
| Post Condition: | The user shall be shown his/her home page. |
| Scenario: | The User clicks on the link to Sign In.The User enters username and password.The username and password are validated by the Account Manager.If the username and password are valid then the User is taken to his/her account. |
| Exception: | When the user has input ID and password that do not match. |

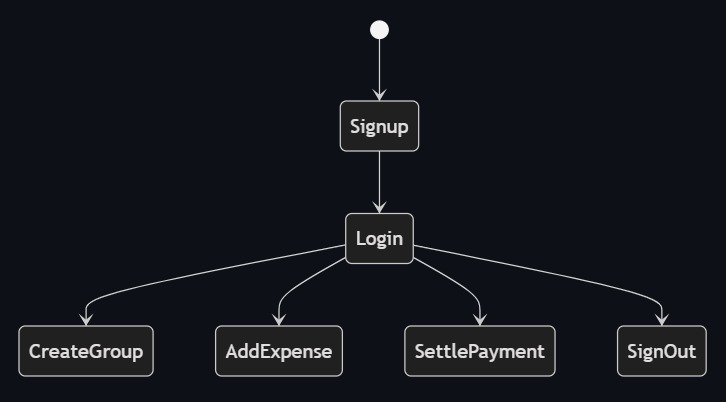
|  |  |
| --- | --- |
| Use Case Name: | Add Expense |
| Actors: | User |
| Pre Condition: | The user must be logged in |
| Post Condition: | The expense will be added to the system |
| Scenario: | 1. The User navigates to the add expense page. 2. The User enters details such as amount, description, category, and participants. 3. The User submits the expense. 4. The system records the expense and updates balances. |
| Exception: | If the user fails to provide required fields, an error message is displayed. |

|  |  |
| --- | --- |
| Use Case Name: | Create Group |
| Actors: | User |
| Pre Condition: | The User must be logged in |
| Post Condition: | A group is created for shared expenses |
| Scenario: | The User navigates to the create group section.The User enters a group name and adds participants.The system verifies the entered details.If valid, the group is successfully created and members are notified. |
| Exception: | If the participants are not registered, an error message is displayed. |

|  |  |
| --- | --- |
| Use Case Name: | Settle Payment |
| Actors: | User |
| Pre Condition: | The User must have outstanding balances |
| Post Condition: | The payment is recorded and balances are updated |
| Scenario: | The User navigates to the settle payment section.The User selects the person to settle with and enters the amount.The system records the payment and updates balances. |
| Exception: | If the participants are not registered, an error message is displayed. |

|  |  |
| --- | --- |
| Use Case Name: | Sign Out |
| Actors: | User |
| Pre Condition: | The User should be signed in to his/her account |
| Post Condition: | The User returns to the homepage of the website |
| Scenario: | 1. User clicks on the ’Sign Out’ link. 2. After successfully signing out user is taken to the home page. |
| Exception: | Server is down or login session has expired |

## 3.3 Behavioral requirements



# Other Nonfunctional Requirements

## Performance Requirements

The Expense Splitter application must be optimized to function efficiently across various devices, ensuring minimal processing power requirements while maintaining a smooth user experience. Performance benchmarks will be set to ensure seamless transactions and expense calculations, even when handling large groups and complex splitting scenarios. Additionally, data retrieval and synchronization with external payment systems must be optimized to minimize latency.

## Safety and security Requirements

User privacy and security are crucial. The system will comply with industry-standard security measures to protect user data, including encryption for sensitive information such as financial details and login credentials. User IP addresses and personal details will not be shared with third parties, ensuring compliance with data protection regulations. Authentication mechanisms such as two-factor authentication (2FA) will be implemented to prevent unauthorized access.

## Software Quality Attributes

The system must be reliable with high availability, ensuring minimal downtime. It should support multiple file formats for data interoperability and provide an intuitive user interface with built-in guidance. Data integrity should be maintained to prevent loss or corruption, while scalability must allow for increased user demand without performance issues. Audit logs will track modifications, and the system will persist the last transaction details for easy management of expenses.

Appendix A: Glossary

(an alphabetical list of terms)

1. **Expense Splitting** - The process of dividing a shared cost among multiple participants.
2. **Multi-currency Support** - A feature allowing expenses to be recorded in different currencies with real-time conversion.
3. **Debt Management** - A system for tracking who owes whom and suggesting repayment strategies.
4. **Recurring Expenses** - Automatic expense entries for repeated payments like rent or subscriptions.
5. **Payment Integration** - Connecting the system with third-party payment gateways for seamless transactions.
6. **Equal and Unequal Splitting** - The ability to divide expenses equally or in customized ratios.
7. **Audit Logs** - Records of all transactions and modifications for transparency and accountability.
8. **User Authentication** - A security feature ensuring only authorized users access the system.
9. **PDF Export** - Generating a downloadable PDF file of transaction summaries for record-keeping.
10. **Bill Reminders** - Automated notifications to remind users of pending payments.

Appendix B: Data dictionary

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them

# User Entity

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraint** |
| Username | Alphanumeric | Unique identifier for the user | Primary Key |
| Password | Alphanumeric | User's login password | Minimum 6 characters |
| Email Address | Char | User's email for communication | Must be valid format |
| Age | Number | Age of the user | Must be 18 or above |
| Name | Char | Full name of the user | Maximum 30 characters |

* 1. **Expense Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraint** |
| Expense ID | Alphanumeric | Unique identifier for each expense | Primary Key |
| Amount | Number | The total expense amount | Cannot be negative |
| Currency | Char | The currency in which the expense is recorded | Must be a valid currency |
| Split Type | Char | Equal or unequal split | Must be defined |
| Paid By | Alphanumeric | Identifier of the person who made the payment | Must be an existing user |
| Participants | List | List of users involved in the expense | Minimum 2 users required |

* 1. **Payment Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Constraint** |
| Payment ID | Alphanumeric | Unique identifier for each payment | Primary Key |
| Payer | Alphanumeric | User making the payment | Must be an existing user |
| Payee | Alphanumeric | User receiving the payment | Must be an existing user |
| Amount | Number | The amount being transferred | Cannot be negative |
| Date | Date | Date of the payment | Must be valid date format |
| Status | Char | Payment status (Pending, Completed, Failed) | Must be defined |