Software Requirements Specification

for

ShareExpense

Version 1.0 approved
April 2, 2018

Prepared by:

Geetanshu Gupta
COE 3, 4th Semester
2016UCO1671

Raghav Kumar
COE 3, 4th Semester
2016UCO1640

Table of Contents

1. INTRODUCTION

1.1 PURPOSE

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the ShareExpense system. This document will cover each of the system's intended features and will also cover hardware, software, and various other technical dependencies. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 PROJECT SCOPE

The ShareExpense system is composed of two main components: a client-side web application, and a server-side application which will support and interact with various client-side features. The system is designed to facilitate the process of tracking and settling shared expenses. Potential scenarios include paying rent, splitting a check at dinner, sharing travel expenses, etc.

1.3 TERMINOLOGY USED

S.No.	Term	Definition
1.	User	A person registered on the platform.
2.	Transaction	A financial settlement between just two users noting the exchange of monetary consideration.
3.	Group Transaction	A financial settlement between two or more users where one user pays on behalf of other.
4.	One off Group Transaction	A group transaction which will occur only once.
5.	Prolonged Group Transaction	A group of group transactions where different group transactions occur on different dates or time.

1.4 REFERENCES TO OTHER DOCUMENTS

1. **ACM References**:

- Ozan Candogan, Asuman Ozdaglar and Pablo Parrilo. 2018. Pricing Equilibria and Graphing Valuations. ACM Trans. Economics and Computation 6, 1, Article 2 (March 2018), 16 pages. https://dl.acm.org/citation.cfm?id=3175495.
- Robert A. Elliott, Sr. and Edward B. Allen. 2013. A Methodology for Creating an IEEE Standard 830-1998 Software Requirements Specification

Document. Journal of Computing Sciences in Colleges 29, 2, 9 pages (December 2013). https://dl.acm.org/citation.cfm?id=2535437.

2. Software Engineering: A Practitioner's Approach by Roger S. Pressman, McGraw Hill International.
3. Software Engineering: Principles and Techniques by Sangeeta Sabharwal, New Age International.

1.5 OVERVIEW

The remainder of this document includes five chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter deals with the external interfaces required in the development of the product.

The fifth chapter highlights the other functional requiremnets like performance requirements, safety requirements, security requirements, etc.

The sixth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen.

The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The ShareExpense project is a new, self-contained product intended for use on the web platform. While the ShareExpense web application is the main focus of the project, there is also a server-side component which will be responsible for database and synchronization services. The scope of the project encompasses both server- and client-side functionalities, so both aspects are covered in detail within this document.

2.2 PRODUCT FEATURES

The following list offers a brief outline and description of the main features and functionalities of the ShareExpense system. The features are split into two major categories: core features and additional features. Core features are essential to the application's operation, whereas additional features simply add new functionalities. The latter features will only be implemented as time permits.

CORE FEATURES

1. User registration
2. Log in
3. Group creation
4. Standalone transactions
5. Group transactions
6. Final debt resolution
7. Display accounts

ADDITIONAL FEATURES

8. Member debt visualization
9. Paytm integration
10. Email/SMS notifications

2.3 USER CLASSES AND CHARACTERISTICS

The ShareExpense project is meant to offer a shared expenses solution that is faster, easier, and more convenient than manually calculating and handling debts. Consequently, the application will have little or no learning curve, and the user interface will be as intuitive as possible.

The following list categorizes the scenarios in which ShareExpense is expected to be utilized:

SHAREEXPENSE: POTENTIAL SCENARIOS

1. Long-term recurring expenses (e.g. rent, groceries, utilities)
2. Short-term recurring expenses (e.g. travel costs – gas, food, hotel)
3. Single expense (e.g., splitting a bill at dinner)

These groups are not meant to separate or categorize users, just the different situations in which ShareExpense is likely to be used. In fact, a user may utilize the application for all of these scenarios simultaneously. This is another defining feature of the ShareExpense system: support for multiple groups. This functionality allows a user to track expenses pertaining to several unrelated groups at the same time.

2.4 DESIGN AND IMPLEMENTATION CONSTRAINTS

The primary design constraint is the web platform. Creating a user interface which is both effective and easily navigable will pose a difficult challenge. Other constraints such as limited memory and processing power are also worth considering. ShareExpense is meant to be quick and responsive, even when dealing with large groups and transactions, so each feature must be designed and implemented with efficiency in mind.

2.5 ASSUMPTIONS AND DEPENDENCIES

TIME DEPENDENCIES

As mentioned previously, the features of ShareExpense are divided into two groups: core features and additional features. Core features are crucial to the basic functionality of the ShareExpense application. These features must all be implemented in order for the application to be useful. Optional features, however, are not critical to the function of the application. They are usability improvements and convenience enhancements that may be added after the application has been developed. Thus, the implementation of these features is entirely dependent upon the time spent designing and implementing the core features. The final decision on whether or not to implement these features will be made during the later stages of the design phase.

EXTERNAL DEPENDENCIES

Several of the features presented in this document rely on the existence and maintained operation of several APIs. A non-exhaustive list follows.

1. **EMAIL NOTIFICATIONS**
   The Web platform is not suited for sending mass emails. Thus, the central server will be responsible for this feature of the application. The client will notify the server when messages need to be sent using a custom API that is to be created. This API will use standard HTTP messaging to facilitate client-server communications. The API will be implemented using Node.js.
2. **SMS NOTIFICATIONS**
   The client side application can request the server to send SMS through some API to the user informing him/her of any activity.

3. PAYTM WEB API

We will use the Paytm API in order to facilitate payment of debts that users may have incurred and wish to pay using the software.
Reference: http://paywithpaytm.com/developer/paytm_api_doc/

3. System Features

ShareExpense's system features are divided into two main categories: core features and additional features. Core features form the body of the application and include any features that are essential to the functionality of the order to have a fully-functioning application. Additional features, however, are not required for the app to function. They include any features which, if time permits, will be added to the application in order to provide extra functionality.

CORE FEATURES

3.1 USER REGISTRATION

When the user registers on the application for the very first time, the user is presented with an initial registration screen. This screen prompts the user to create an account on the ShareExpense server using his/her email address. The user also enters a **unique** "Username", which will be the name that is shown as their handle within the groups, contact number and their full name. Completing this process will create and store an account for the user on the ShareExpense server, enabling all of the application's synchronization capabilities.

3.1.1 USER REQUIREMENTS

VALID EMAIL ADDRESS
The user cannot proceed until a valid email address is entered. The application will verify that the user's input is consistent with the format of an email address (i.e. xxxx@xxxxx.xxx).

3.1.2 SYSTEM REQUIREMENTS

SECURE DATABASE SYSTEM
The application must insure that the user's information is encrypted and safely stored.

3.2 LOG IN

The user needs to supply in his/her registered username and password to see his/her account. If he/she forgets his/her password he/she can request a change, which will send a link to the recovery procedure to his/her email address. A user can choose to let the browser remember the logged in state. The client webpage sends the combination of username and password to the server, which then validates the pair and replies with a suitable response. A valid username password pair logs in the user to the account.

3.2.1 USER REQUIREMENTS

VALID USERNAME
The user must be registered with the system before logging in.

3.2.2 SYSTEM REQUIREMENTS

SECURE DATABASE SYSTEM
The application must insure that the user's information is encrypted and safely stored.

3.3 GROUP CREATION

The "Groups" screen will be the main screen of the application. From this screen, users will be able to view existing groups. Groups may be created by adding members from the user's contacts or by manually entering an email and name. The creator of a group is designated as the "Leader" of that group. The Leader is responsible for confirming transactions submitted by other members of the group and will also have the ability to stop or disband the group.

3..1 USER REQUIREMENTS

VALID EMAIL ADDRESS(ES)
The group leader must enter valid email addresses for group members in order to obtain the full functionality of the application.

3..2 SYSTEM REQUIREMENTS

DATA STRUCTURE(S) FOR REPRESENTING
Bills must be discrete objects that the application/server can store and manipulate.

3.4 STANDALONE TRANSACTIONS

A standalone transaction is a situation where a bill is split between two or more users and the payment has been done by one group member on a single occasion. Users must input a display name (e.g., "Groceries"), the associated cost, and select the individuals responsible for the expense. Once the Bill is confirmed, it is synced to the server, enabling all other ShareExpense users in the group to view it.

3..1 USER REQUIREMENTS

NONE
Only an active internet connection is required.

3..2 SYSTEM REQUIREMENTS

COST-DIVISION ALGORITHM
The algorithm must divide the cost and add the correct amount to each user's current balance.

3.5 GROUP TRANSACTIONS

A group transaction is a situation where a bill is split between the members of an existing group and the payment has been done by one group member. Bills created within the application are meant to represent expenses incurred by the group in real life. Users must input a display name (e.g., "Groceries"), the associated cost, and select the individuals responsible for the expense. Once the Bill is confirmed, it is synced to the server, enabling all other ShareExpense users in the group to view it.

3..1 USER REQUIREMENTS

NONE
Only an active internet connection is required.

3..2 SYSTEM REQUIREMENTS

COST-DIVISION ALGORITHM
The algorithm must divide the cost and add the correct amount to each user's current balance.

3.6 FINAL DEBT RESOLUTION

This feature represents a real-world transaction between two or more members of a group. A common scenario would involve the user resolving his/her debt by making payments to other group members. The resulting changes in balances are calculated automatically and displayed for the users involved.

3..1 USER REQUIREMENTS

NONE
Only an active internet connection is required.

3..2 SYSTEM REQUIREMENTS

REAL-TIME BALANCING
The application must be able to update each user's balance on the fly.

3.7 DISPLAY ACCOUNTS

This is a global feature that will enumerate an individual's debt across all groups which he/she is a member. Visual cues will be used to provide a distinction between positive, negative, and even balances within each group. There will be an option to reconcile all debts from this screen. Selecting this option will show the user a list of transactions which are required before his/her debts can be resolved.

3..1 USER REQUIREMENTS

NONE
Only an active internet connection is required.

3..2 3.7.2 SYSTEM REQUIREMENTS

NONE
Only an operational server is required.

ADDITIONAL FEATURES

3.8 MEMBER DEBT VISUALIZATION

This feature allows users to access group balances as bar graphs, rather than the default text only form. These visuals can be obtained at any time by selecting the Member Debt Visualization option from any group. The visuals are updated any time a group member’s balance changes.

3..1 USER REQUIREMENTS

NONE
Only an active internet connection is required

3..2 SYSTEM REQUIREMENTS

GRAPHICS IMPLEMENTATION
The application must have a method of converting raw data into rich, dynamic visuals.

3.9 PAYTM INTEGRATION

This feature will offer users the option to resolve their debts through using Paytm. This will enable users to make real financial transactions through the app, which will then update all balances accordingly.

3..1 USER REQUIREMENTS

PAYTM ACCOUNT
Only users with a Paytm account will be able to utilize its functionalities. Obtaining a Paytm account is done independently of the ShareExpense System.

3..2 SYSTEM REQUIREMENTS

PAYTM INTEGRATION
The application must utilize the Paytm API to provide the appropriate functionalities.

3.10 E-MAIL/SMS NOTIFICATIONS

Since groups may involve individuals without ShareExpense (i.e., offline users), there must be an alternative method of notifying these individuals. The simplest way to achieve this is via email or text message (SMS). This will allow these individuals to enjoy some of the same functionalities as ShareExpense users. Also the details of every new activity will be mailed/messaged to the existing users.

3..1 USER REQUIREMENTS

A VALID E-MAIL ADDRESS OR PHONE NUMBER
The offline user must have some method of being contacted electronically.

3..2 SYSTEM REQUIREMENTS

E-MAIL AND SMS CAPABILITIES

The application must be able send a request to the server to compose and send emails/SMS.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACE

The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla or Google Chrome by which user can access to the system.

The user interface shall be implemented using any tool or software package like Node.js, PHP, Ruby, etc.

4.2 HARDWARE INTERFACES

No hardware interfaces have been identified.

4.3 SOFTWARE INTERFACES

The ShareExpense app is to be developed using Node.js and the client side web browser must support HTML5 and JavaScript.

4..1 INCOMING AND OUTGOING ITEMS

Outgoing data consists of group information, bills, transactions, and confirmations sent by users to the server. Incoming data consists of updates from the server regarding member balances, as well as any notifications deemed necessary.

4..2 SERVICES AND COMMUNICATIONS

ShareExpense relies on server push and pull protocols to be fully functional Communication will occur in occasional, short bursts between client and the server in the following situations:

1. Whenever a user creates/confirms a new bill or transaction
2. Whenever the server finishes processing group/member debts to update users
3. The application will notify the server when it successfully receives an update

4.4 COMMUNICATIONS INTERFACES

The ShareExpense application will have a network server that is web-based and created using Node.js. The server exists to retrieve information from the database and calculate payments. The product also calls for a database system that stores user information and transaction history between users. Whenever a user opens the ShareExpense application, a pull protocol will be used to retrieve and sync the latest transaction updates from the server.

5. OTHER FUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

Performance should not be an issue because all of our server queries involve small pieces of data. Changing screens will require very little computation and thus will occur very quickly. Server updates should only take a few seconds as long as the system can maintain a steady signal. The cost-division algorithms used by in application will be highly efficient, taking only a fraction of a second to compute.

5.2 SAFETY REQUIREMENTS

ShareExpense will not affect data stored outside of its servers nor will it affect any other applications installed on the user's system. It cannot cause any damage to the system or its internal components. The only potential safety concern associated with this application applies to virtually all handset apps: ShareExpense should not be used while operating a vehicle or in any other situation where the user's attention must be focused elsewhere.

5.3 SECURITY REQUIREMENTS

This application assumes that only the user or whoever he/she allows will have access to his/her system. With that being said, only a Google email address is required to verify the identity of the user upon opening the app. The Paytm API provides all of the security checks needed to ensure that no fraudulent transactions occur.

5.4 SOFTWARE QUALITY ATTRIBUTES

The graphical user interface of ShareExpense is to be designed with usability as the first priority. The app will be presented and organized in a manner that is both visually appealing and easy for the user to navigate. There will be feedbacks and visual cues such as notifications to inform users of updates and pop-ups to provide users with instructions. To ensure reliability and correctness, there will be zero tolerance for errors in the algorithm that computes and splits expenses between group members. To maintain flexibility and adaptability, the app will take into account situations in which a user loses internet connection or for whatever reason cannot establish a connection with the server. Overall, the app balances both the ease of use and the ease of learning. The layout and UI of the app will be simple enough that users will take no time to learn its features and navigate through it with little difficulty.

6. Other Requirements

This section contains information of a general or explanatory nature that may be helpful, highlighting the highly key requirements and factors of the project, including requirements' apportioning, backend build and execution plan, and finally, the schedule of the project along with the budget plan of the product.

6.1 APPORTIONING OF REQUIREMENTS

The versions of the system are only available in English, with major functions of the project in play by the end of the development period. In case the project is delayed, there may be some requirements that could be transferred to the upcoming versions. The development of project will be taken care of by the authors of this SRS document, with no apportioning of requirements amongst external hired help to be done at any point of time during the schedule of the project.

6.2 DATABASE

A database for ShareExpense calls for a server side implementation that holds information for the users, groups, bills, transactions, as well as all the relationships involved. The database will be using MySQL. The server will be configured on a Linux platform, and through use of Node will allow interaction and processing in conjunction with the database. Processes to be done on the server include: pushing/pulling data, updating data, and generating notifications.

6.3 SCHEDULE AND BUDGETS

The schedule is such that the project shall be completed through three months of building phase, it being a semester-long project. The budget of the project highlights expenditures with regard to initial costs incurred by the creator of project, and finally, registration payments by users.

The budget plan involves purchasal of API keys:
Mailgun - API for Email
Msg91 - API for SMS

Testing version of the project includes utilisation of the free-trial based running of the aforementioned APIs for a period of 14 days. For operation thereafter, purchasal of API keys is a compulsion, expenditure of which is recovered from payments made by the user at the time of registration.