Software Requirements Specification

for

Money Management System

for ExpenseManagement

Prepared by

Shubham Darak (43211)
Omkar Deshpande (43212)
Devesh Chandak (43213)
Alrik Fernandes (43215)

17 july 2020

Table of Contents

Table of Contents ii
Revision History ii
1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Product Scope 1
1.5 References 1
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Classes and Characteristics 2
2.4 Operating Environment 2
2.5 Design and Implementation Constraints 2
2.6 User Documentation 2
2.7 Assumptions and Dependencies 3
3. External Interface Requirements 3
3.1 User Interfaces 3
3.2 Hardware Interfaces 3
3.3 Software Interfaces 3
3.4 Communications Interfaces 3
4. System Features 4
4.1 System Feature 1 4
4.2 System Feature 2 (and so on) 4
5. Other Nonfunctional Requirements 4
5.1 Performance Requirements 4
5.2 Safety Requirements 5
5.3 Security Requirements 5
5.4 Software Quality Attributes 5
5.5 Business Rules 5
6. Other Requirements 5
Appendix A: Glossary 5
Appendix B: Analysis Models 5
Appendix C: To Be Determined List 6

Revision History

Name	Date	Reason For Changes	Version
v0.1	17/7/2020	Initial prototype	v.0.1

1. Introduction

1.1 Purpose

Purpose of this product is to enable users to keep track of their daily money transactions. The transaction may include money lent or borrowed from others, money spent on day to day transactions such as money spent on food, traveling, etc. This functionality allows users for efficient management of their money.

1.2 Document Conventions

The following abbreviations has corresponding meaning
DE - Daily Expense
FE - Food Expense
TE - Travelling Expense
GE - Group Expenses

1.3 Intended Audience and Reading Suggestions

This document is intended for all the stakeholders related to this product, including developers, testers, internal and external guides, coordinators and evaluators.
This document should be read in the given sequence i.e. Introduction, description, functionalities, requirements, features and use case diagram.

1.4 Product Scope

In our day to day life, we do many transactions that involve exchange of money, but at the end of the day we lose track of our transaction. For people living away from their home like students or laborers, it becomes crucial to use money safely. Thus comes into picture the money management system that allows users to keep track of their expenses. This product is basically like a digital diary.
The scope of the project is clear to give a simple and attractive application to simplify the work as well as to reduce the efforts while doing it offline or we can say by doing it with old methods. In this application we are able to save a database of all money based transactions present on the site in MySQL (in the form of tables). Various functions in jsp have been used like dealing with servlet, session, calling servlet within another servlet and similarly triggers, stored procedures, insert, update, delete etc. commands are used in MySQL to store the values in the database.

Following are some of the functionalities of product

1. *Note your FE*
2. *Note your TE*
3. *Note any money transaction between friends*
4. *These transaction gets reflected in other persons account also*
5. *Note down GE*

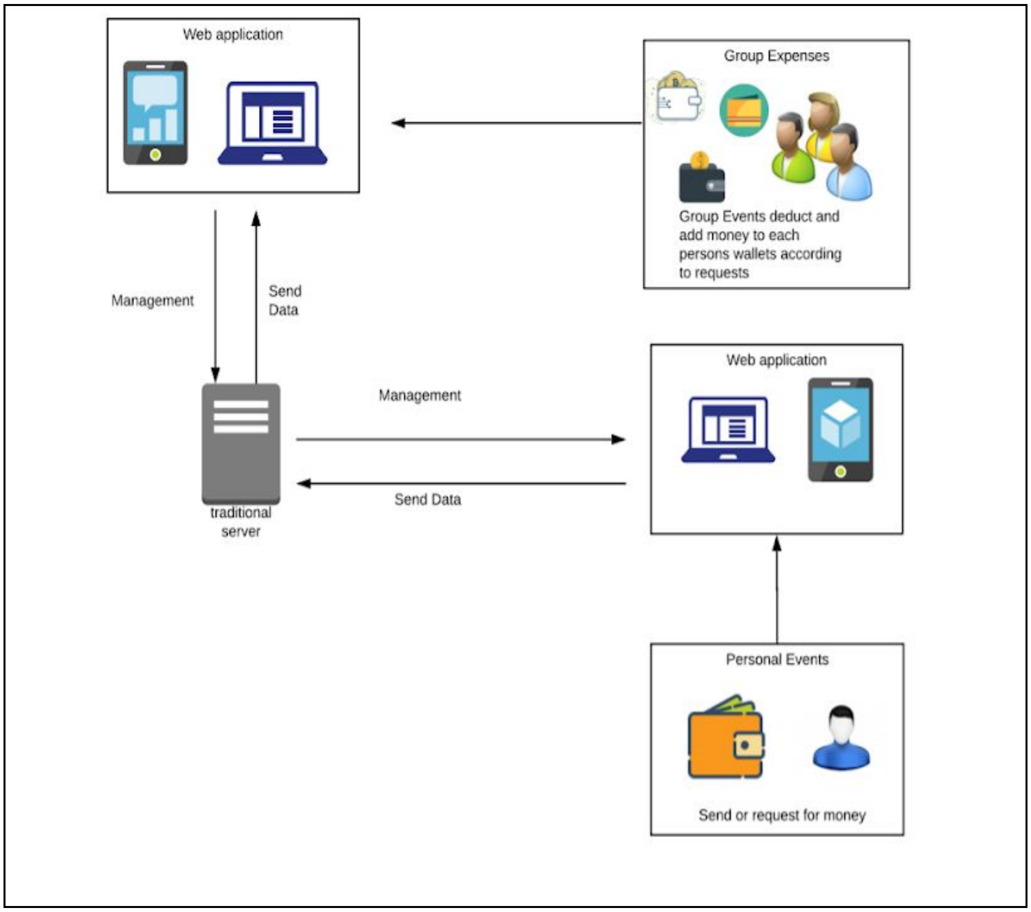
1.5 References

The idea of this product is closely related to an existing product named "Splitwise" available on google play store.

2. Overall Description

2.1 Product Perspective

This is a stand alone product available on both web and android.



2.2 Product Functions

1. Add Personal Expenses
2. Add money to wallet
3. Set Expense limit
4. Create Transaction (request), accept transaction..
5. Create a group.

2.3 User Classes and Characteristics

Users - General Users (All Users have same privilege)

1. *Enter FE*
2. *Enter TE*
3. *Enter GE*
4. *Enter borrow money*
5. *Enter lent money*
6. *Settle pending transactions*

2.4 Operating Environment

1. *Operating system - Windows/ Linux/ Android (Version > 4.0)*
2. *Web Browser - Any*
3. *Database - Mysql*
4. *Connection - JDBC*
5. *IDE - Eclipse*
6. *Technologies - java/ JSP*

2.5 Design and Implementation Constraints

Developer limitations -

1. *Mysql Workbench version 7.0 or above is required*
2. *Server technology to host the project is required.*

2.6 Assumptions and Dependencies

The idea of project is based on following assumptions

1. *The user of this product is well familiar with using electronic devices*
2. *There is no involvement of real currency*
3. *This is just the reflection of users daily transactions as a digital diary*

3. External Interface Requirements

3.1 User Interfaces

The application is very user friendly and uses a GUI interface implemented in CSS, HTML, BOOTSTRAP, JavaScript to Communicate with the user. Various features are self – explanatory. Forms are easy to fill in and components can be added, removed and updated very easily through a Single dialog box. The application includes tool-tip hints to give a brief description of the particular input Field. Drop down lists to display categories of particular fields.

List boxes are used to display all the components at once so that users can see all the components of a Particular type at once. One can just select the component and modify and remove the component. (based on the access control of the person).

Hover effects are provided to the sign-in and login buttons for appealing effect. Theme is selected in such a way that it can be changed as per convenience of the user. Default selected is dark theme ,but it can be changed to white background or any other combinations provided. Proper alignment of buttons are done,so that users can easily navigate through them. Notification icon is provided with red colour so it is visible to the eye instantly as soon as the user logs in.

3.2 Hardware Interfaces

The hardware interface mainly consists of any device that supports web page rendering. So in today's world basically any and almost every device is capable of using our web app. Our web application is responsive in nature so it modifies the sizes of the entities according to the resolution of the devices that it is being used on. Any android phone and computer device of any kind is capable of rendering our application.

3.3 Software Interfaces

1. *Operating system - Windows/ Linux/ Android (Version > 4.0)*
2. *Web Browser - Any*
3. *Database - Mysql*
4. *Connection - JDBC*
5. *IDE - Eclipse*
6. *Technologies - java/ JSP*

3.4 Communications Interfaces

Supported for all browsers which eventually use HTTP / HTTPS.

4. System Features

4.1. Personal Event:

If user-A borrows money from user-B then user-B can send the request to the user-A that he/she had borrowed from him/her & this request will be stored in a pending personal request table. Now user-B(borrower) will approve or reject the request. If he/she approves , the request will be stored in a personal event table & a "PAY" button will be available to the borrower to pay the borrowed money. But if the borrower rejects the request,the request will be deleted from the pending personal request table. Every action of the borrower will be notified to the lender through the notification table and vice versa. Every action of the user in a personal event will be stored in a personal event log table.

4.2. Daily Expenses:

Users can add daily expenses and his/her expenses of the whole day will be stored in the daily expense table with the total amount of each day. All the categories with their amount will be stored in a daily category table. Users can search daily expenses according to the date provided by the user by matching the date field in the daily expense table.

4.3. Group Event:

If user creates a group event a group event will be created and it will be stored in group event table & for all the participants that are added in the group event by the owner, a group event request will be created and it will be stored in pending group request table & all the participants will get notified and the request will be shown in pending group request block of dashboard with amount distributed equally among the participants including owner. If a participant accepts group request then the respective request will be stored in user in group table and that participant will be able to 'PAY' for that respective group event (i.e. owner will be paid), hence the respective changes will be made in the wallets of all the participants including owner in user table. Participants and the owner will be notified with respect to each other's actions.

4.4. Notification:

For every action of the user that makes changes to the respective wallet will be notified to the respective person. All-log table contains every task performed by the particular user.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

There aren't any rare or tedious performance requirements for our application. Just a stable and reliable internet connection would do the job.

5.2 Security Requirements

The user has to keep in mind that anyone with access to their credentials can modify the transaction and logs of transactions. In worst case scenario if the intruder enters your account they may change the amounts related to a certain trans

5.3 Software Quality Attributes

Our application has been rigorously tested by us for all sorts of ambiguities and shortcomings. Our observations show complete read after consistency i.e. all changes show up in the user accounts in real time. We have also tested the application for the circumstances of unexpected shutdowns like power failure.

5.4 Business Rules

In case of group events the person who makes the expenses over who can edit the transaction amount and debt on each candidate involved in the group event.

In case of personal events only the person himself is responsible for making changes and adding new logs.