Requirements Document

By: Nidhi Patel
Elishbah
Zan
Pamela Shahu
Ahmed Alfaris

Table of Contents

1. Introduction

1.1 Purpose of Document

This is a SplitSmart requirements specification paper. Users of SplitSmart will be able to establish and keep track of spending as well as pay off any outstanding amounts for splitting costs with friends, roommates, family, coworkers, or other individuals or groups of people. The scope, goals, and objectives of the new system are laid forth in this document. This document represents the functional needs with use cases, interaction diagrams, and class models in addition to documenting the non-functional requirements. The goal of this paper is to provide guidance for the object-oriented language used in the target system's design and implementation.

1.2 Project Summary

Project Name: SplitSmart

Project Manager: Zhiwei Xu

Project Analysts: Nidhi, Elishbah, Pamela Shahu, Zan, Ahmed

Users: User and Group Manager

1.3 Background

SplitSmart is a web-based application with mobile apps for Android and IOS. The software has user accounts, group management, expense creation, notification system, approval workflow, balance tracking, payment tracking, and reporting features. Also allow users to create new expenses and add information such as amount, date, description.

1.4 Project Scope

The goal of this project is to provide a web-based tool that assists users in balancing their budgets and tracking their spending. This project excludes product advertising, inventory management, and account billing. The system will include an online search engine and a translator. This project includes online security concerns that go beyond the site's password protection.

1.5 System Purpose

1.5.1 Users

Customers:

Customers will find it simple to manage their finances once the system is in place.

Product Owners:

Product owners will be able to easily update the information about the website.

Customer Service:

Due to clients being able to get the information they want from the website, the new system results in a reduction in the workload of Customer Service.

Accounting:

Information about purchases will be delivered directly to Accounting, and get notification when a new payment is created.

Information technology:

This division is in charge of setting up the new database, hosting the website, and keeping everything running smoothly.

1.5.2 Location

The system will be available to any customer using the Internet. They can use this system from any location.

1.5.3 Responsibilities

Responsibilities of the new system:

- User needs to log in
- Guidance for website navigation
- Users to create new expenses
- Get notification when someone creates new expenses
- Account has information such as amount, date, description, etc.
- User track the balance owed by other user in the group
- User can track payments that have been made to settle balances
- Users will get reports of their expenses, balances, and payments.

- On-line help in website navigation
- Password protection scheme for non-public web pages

1.5.4 Need

This system is needed in order for customers to keep track of money they spend.

1.6 Overview of Document

Section 2: Functional Objectives

- Each goal outlines the system's expected behavior, and a test to see whether it has been effectively accomplished. These goals are arranged in order of priority. All goals must be achieved for the new system to be deemed a success.

Section 3: Non-Functional Objectives

- Each goal defines a technological need or restriction on the system's general features.

Section 4: Context Model

- It describes the external entities that communicate with the system.

Section 5: Use Case Model

- Use cases go into depth of the feature. Use case is describing the purpose of a new system. Each use case explains how it works in the system and helps users with application. The interactions between all use cases and system actors will be shown in the use case diagram.

Section 6: Class Model

- All analytical classes and their connections are displayed. A class is a group of system objects with similar data and behaviors.

Section 7: An appendix

2. Functional Objectives

2.1 High Priority

1. The system will allow users to get registered in the system and log in to access to use the system function of creating expenses and managing that.

2. The system will allow users to create and manage groups of people. Users can share the expenses with group members.

3. The system will allow users to create new expenses.
- Users will insert data about the expense in a form that consists of date, description, shared manner or split%, and image if any.
- Users can also add members who are related to this expense to adjust the balance.

4. The system will send a notification to all the users when a group member creates a new expense.

2.2 Medium Priority

1. This system will provide a function to track the balance owed by each user to every other user, taking into account all expenses that have been entered.

2. The system will provide a function that will allow users to record when payments have been made to settle balances owed so that the balances are accurately reflected in the system.

3. The system will provide a function that will provide users with reports and summaries of their expenses, balances, and payments.

3. Non-Functional Objectives

3.1 Reliability

- The system should be functional and available to the users 98% of the time, error prevention should be handled in this way.

- There should be a backup plan so that in case of any failure service doesn't remain suspended.

3.2 Usability

- There should be a user-friendly interface so users learn it quickly and they should be able to create an expense and user groups after 2 days of use.

- The system should be responsive and should give errors and responses to users against each action so users can easily learn how they can perform desired tasks.

- Design should be simple regardless of complexities and ambiguity in steps to perform tasks.

3.3 Performance

- The system should be able to handle 1000 users at a time and performance should not go down with the increasing number of users.

- The system’s response should be fast and should not exceed 3 seconds to load a web page.

- The download time and to view a pdf file of 56kbps should not exceed 10 seconds.

3.4 Security

- The system should have a secure user authentication system that stores passwords and keys in encrypted form so unauthorized users may not access the system.

- Payment details should be protected and stored in encrypted form.

3.5 Supportability

- The system should be capable of adding new features in the future to enhance the functionality of the system.

- The system should be compatible with all the browsers like Google Chrome, Internet Explorer, and Microsoft.

3.6 Online user Documentation and Help

- There should be a user guide that has complete information to perform the task and explain all the steps to perform that task.

- Link of this user should be in the top bar menu so it should be accessible for users all the time.

3.7 Purchased Components

None as we are utilizing free software available to us to program this project, the software we are utilizing is Android Studios.

3.8 Interfaces

Built in Database provided by Android Studios known as SQLite Database.

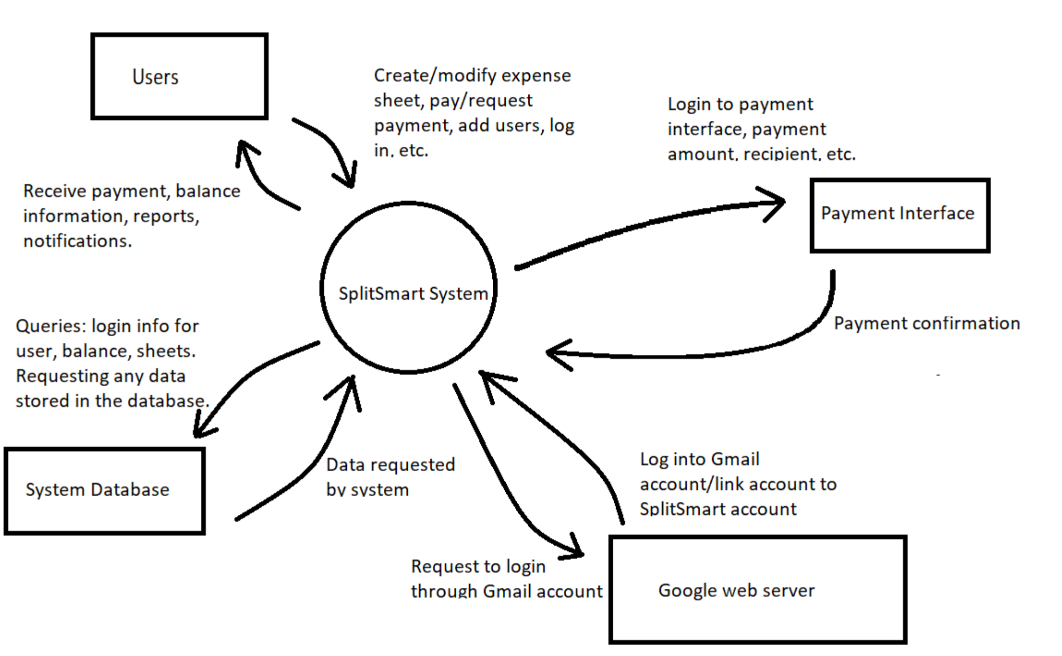
4. The Context Model

4.1 Goal Statement

The goal of the system is to provide a way for users to track expenses and plan events with groups of people by:

- Allowing users to share a SplitSmart expense sheet
- Providing a payment interface to connect to different payment systems
- Fast and up-to-date information on payment tracking and transactions

4.2 Context Diagram



4.3 System Externals

User

- A user is any individual who uses the SplitSmart software. A user can create a SplitSmart expense sheet and share it with other users, pay and receive payment from other users, and add/remove expenses from the sheet. The users will receive a notification for different events within the system. The system will also provide a summary of the expenses.

Payment interface

- A payment interface is a system such as PayPal, Zelle, etc. that allows users to pay, request, or receive payment from other users. Users should be able to connect a payment system to their account. The system will have to interface with this system to send queries.

Google web server

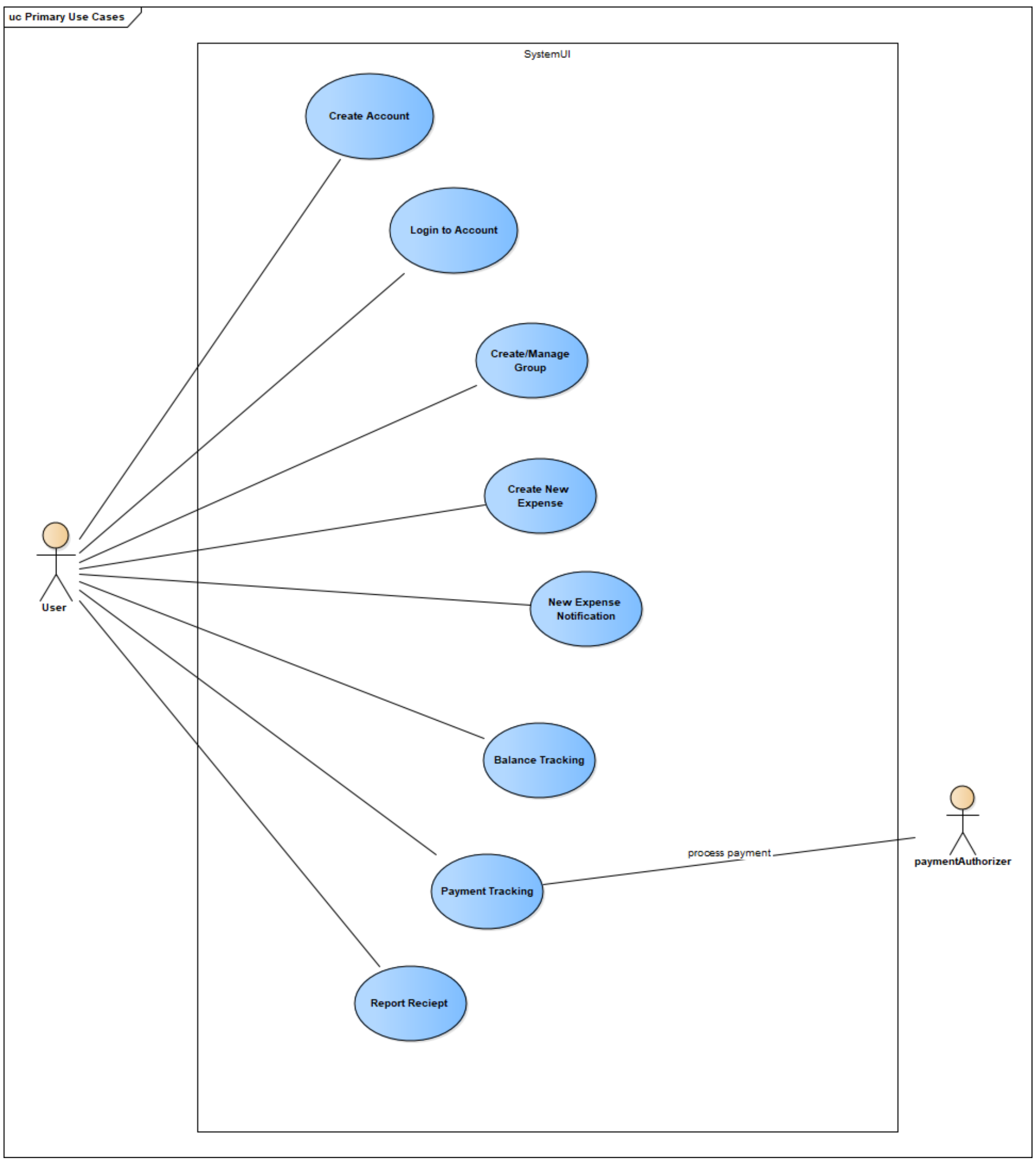
- The SplitSmart software needs to interact with the Google web-server in order to allow users to create accounts through existing Gmail accounts.This interaction is needed so that users can link their Gmail accounts rather than creating an account with SplitSmart if they prefer.

System database

- The system database is a database that holds information about user accounts, SplitSmart expense sheets, etc. The system must interface with this database in order to retrieve information about users such as login information, associated expense sheets, and other information. The system will send queries to the database and receive back data.

5. The Use Case Model

5.1 System Use Case Diagram



5.2 Use Case Descriptions

ID:	UC-001
Title:	Create Account
Description:	This feature will allow user to create an account for the SplitSmart app
Primary Actor:	General user
Preconditions:	SplitSmart application is open and ready to use
Postconditions:	A new SplitSmart account is generated
Main Success Scenario:	1. General user opens app on mobile device/web browser 2. General user selects Create new account tab 3. General user fills sign up form and creates account through email 4. Systems Creates new user account and stores in database 5. System displays "successful"
Extensions:	4.1 Submitted information is incorrect 4.1a Systems displays "Unsuccessful, Try Again"
Frequency of Use:	Whenever a new user wants to use the application
Status:	Pending Review
Owner:	
Priority:	High

ID:	UC-002
Title:	Login to account
Description:	This feature will allow user to log in to personal SplitSmart account and access own expenses and balance information
Primary Actor:	Account user
Preconditions:	SplitSmart account already created
Postconditions:	Personal account access is provided, and user is able to use app functions
Main Success Scenario:	1. Account user selects Login Tab 2. System prompts user to fill in account credentials 3. Account user submits correct account credentials 4. System provides access to user's personal account 5. Account user navigates app to view own expenses and/or balance information
Extensions:	3.1 General user submits incorrect account credentials 4.1 Systems denies access to user's personal account.
Frequency of Use:	Whenever an Account user wants to use app to access account and use application functions
Status:	Pending Review
Owner:	
Priority:	High

ID:	UC-003
Title:	Create/ Manage Group
Description:	This feature will allow a user to create and manage a group of participants to track shared expenses/group balance
Primary Actor:	Account user
Preconditions:	SplitSmart application is on, and user is logged into account
Postconditions:	A group module is created, and user can manage group expenses and balance information
Main Success Scenario:	1. Account user selects Create New Group Tab 2. System displays Group form and prompts user to fill out new group form 3. Account user fills out group information and selects create 4. Systems creates new group module with all newly added participants 5. System displays information on users account for user to access
Extensions:	5.1 Account user selects group module 5.1a Account user edits group information
Frequency of Use:	Whenever an Account user wants to create new group module to track group shared expenses and balance information
Status:	Pending Review
Owner:	
Priority:	High

ID:	UC-004
Title:	Create New Expense
Description:	This feature will allow user to create a new expense
Primary Actor:	Account user
Preconditions:	SplitSmart application is open, and user is logged into account
Postconditions:	A new expense is created
Main Success Scenario:	1. Account user selects Create New Expense Tab on account 2. System will display New Expense Form and prompt user to fill out the New Expense form 3. Account user fills out information for the new expense 4. Systems creates new expense and displays "Successful" message 5. System allows access for user to manage new expense
Extensions:	
Frequency of Use:	Whenever an account user wants to create a new expense
Status:	Pending Review
Owner:	
Priority:	Moderate

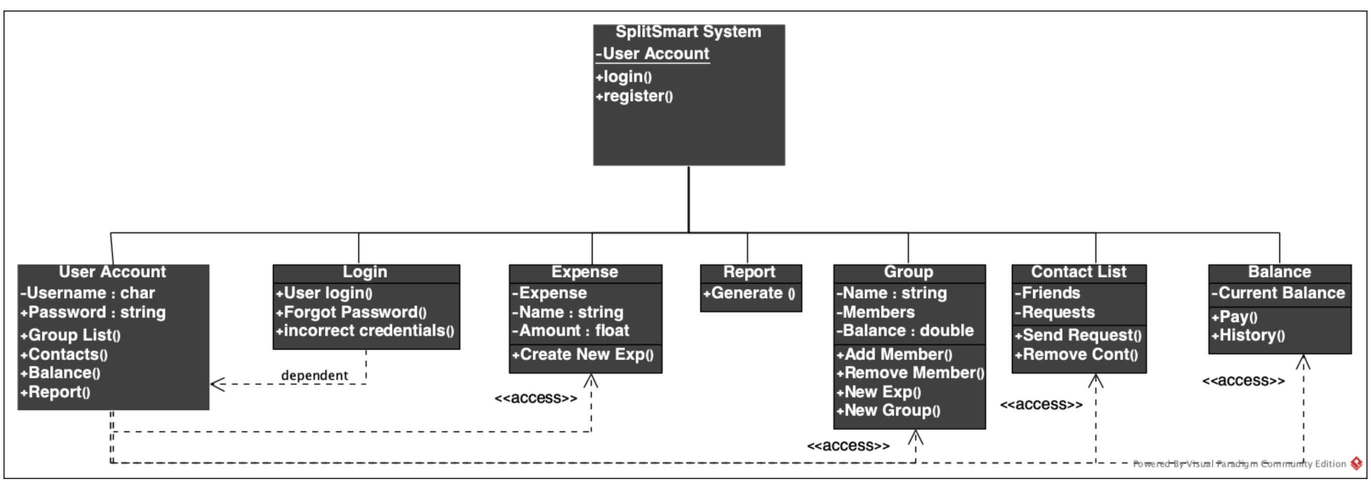
ID:	UC-005
Title:	New Expense Notification
Description:	This feature will allow Notify users whenever other group members create a new expense
Primary Actor:	Account user
Preconditions:	A group is created, and expenses are shared among participants
Postconditions:	User is notified that a group participant has created a new expense
Main Success Scenario:	1. A Group participate creates a new expense under group expenses 2. System sends message to all participants that a new expense has been created.
Extensions:	
Frequency of Use:	Whenever a new expense within a group is created
Status:	Pending Review
Owner:	
Priority:	Low

ID:	UC-006
Title:	Balance Tracking
Description:	This feature will allow user to track the balance owed by each user to every other user
Primary Actor:	Account user
Preconditions:	Group has been created and new expenses have been created as well
Postconditions:	A tracked balance of the money owed by each user is available
Main Success Scenario:	1. Group expenses are entered 2. System automatically updates database as new activity is made on balance
Extensions:	
Frequency of Use:	Whenever a group is created, and expenses have been entered
Status:	Pending Review
Owner:	
Priority:	Moderate

ID:	UC-007
Title:	Payment Tracking
Description:	This feature will allow user to track payments made to settle balances owed
Primary Actor:	Account user
Preconditions:	Group is created and a payment has been made to the balance owed
Postconditions:	Balance is accurately reflected into the system
Main Success Scenario:	1. Group user makes a payment to total balance 2. Systems approves payment 3. System updated balance owed and displays new balance
Extensions:	2.1 System does not approve payment 2.1a System does not update balance
Frequency of Use:	Whenever a Group user makes a payment to balance owed
Status:	Pending Review
Owner:	
Priority:	Moderate

ID:	UC-008
Title:	Report receipt
Description:	This feature will provide users with reports/summaries of their expenses, balances, and payments
Primary Actor:	Account user
Preconditions:	Expense has been created and paid for
Postconditions:	A detailed report of all expense activity
Main Success Scenario:	1. Account user selects Reports Tab 2. Systems displays Reports module with all expenses available 3. Account user chooses which expense a report is needed on 4. System displays Report information of chosen expense
Extensions:	2.1 System displays Report module with no expenses available
Frequency of Use:	Whenever an Account user wants to see an expense report
Status:	Pending Review
Owner:	
Priority:	Moderate

6. The Class Model



7. Appendix Glossary

Whitepaper
Technical paper containing detailed product specifications.