Requirements Presentation

SplitSmart | Ryan Klingensmith & Dominic Baughman | CIS 376

Introduction

Project Summary

Project Name:

- SplitSmart Expense Sharing System

Project Managers:

- Dominic Baughman, Head Architect, CIS 376
- Ryan Klingensmith, Head Architect, CIS 376

Project Goal

SplitSmart is a necessary application for making payments and sharing expenses among peers an easy, versatile, transparent, and secure experience.

Background

- When splitting expenses amongst a group, most people will use verbal agreement or an honor system, which can and does lead to many unnecessary issues. This remains a constant problem among friends and family groups and is an issue that SplitSmart seeks to remedy.
- As online banking and payment applications become more common and people become more willing to utilize these programs, SplitSmart seeks to fill this niche moving forward.

Background

- The software is very robust and highly customizable, providing the flexibility that is necessary for something as important and complex as monetary transactions
- The software is flexible and easy to use from anywhere, allowing people to use the application however they and their groups may require

Project Scope

- The scope for this project is a web application and a mobile application that facilitates the grouping, splitting, and paying of shared expenses. The actual paying of the expenses is not within the scope of this project.
- This software is a standalone product, and therefore has no reliance on any external systems being updated or overhauled to function correctly.

Functional Requirements

High Priority

- The software allows for individual users to create and manage their own accounts that hold and track all their expenses
- The software allows for users to create customizable groups which other users can be added to or removed from at any time
- The software allows for the creation, assignment, splitting, customization, and payment tracking of expenses

Medium Priority

- The software creates and pushes notifications to the user when a new expense is created
- The software automatically tracks the standing balance owed between users
- The software automatically tracks standing expenses owed to a group and notifies them when it has been paid
- The software generates a report on a prompt from the user using the data held within the app

Low Priority

- The software has a specific and consistent look and feel across its entirety
- The software allows the user to change the name of their account or group
- The software allows the user to make names and descriptions for payments

Non-functional Requirements

Non-functional Requirements (1)

Ease of Use

- Customer should be able to figure out how to use the application in x minutes

Resilience

- Application will never be down for longer than x hours
- Application will have a buffer of x% users in use before it goes down

Non-functional Requirements (2)

Security

- Application will prevent any uninvited users from joining groups
- Application allows users to sign in with password and refuses entry without

Supportability

- Application will be supported on every major device currently on the market
- Application will be supported by every major browser
- Application will be supported by every major mobile app store

Non-functional Requirements (3)

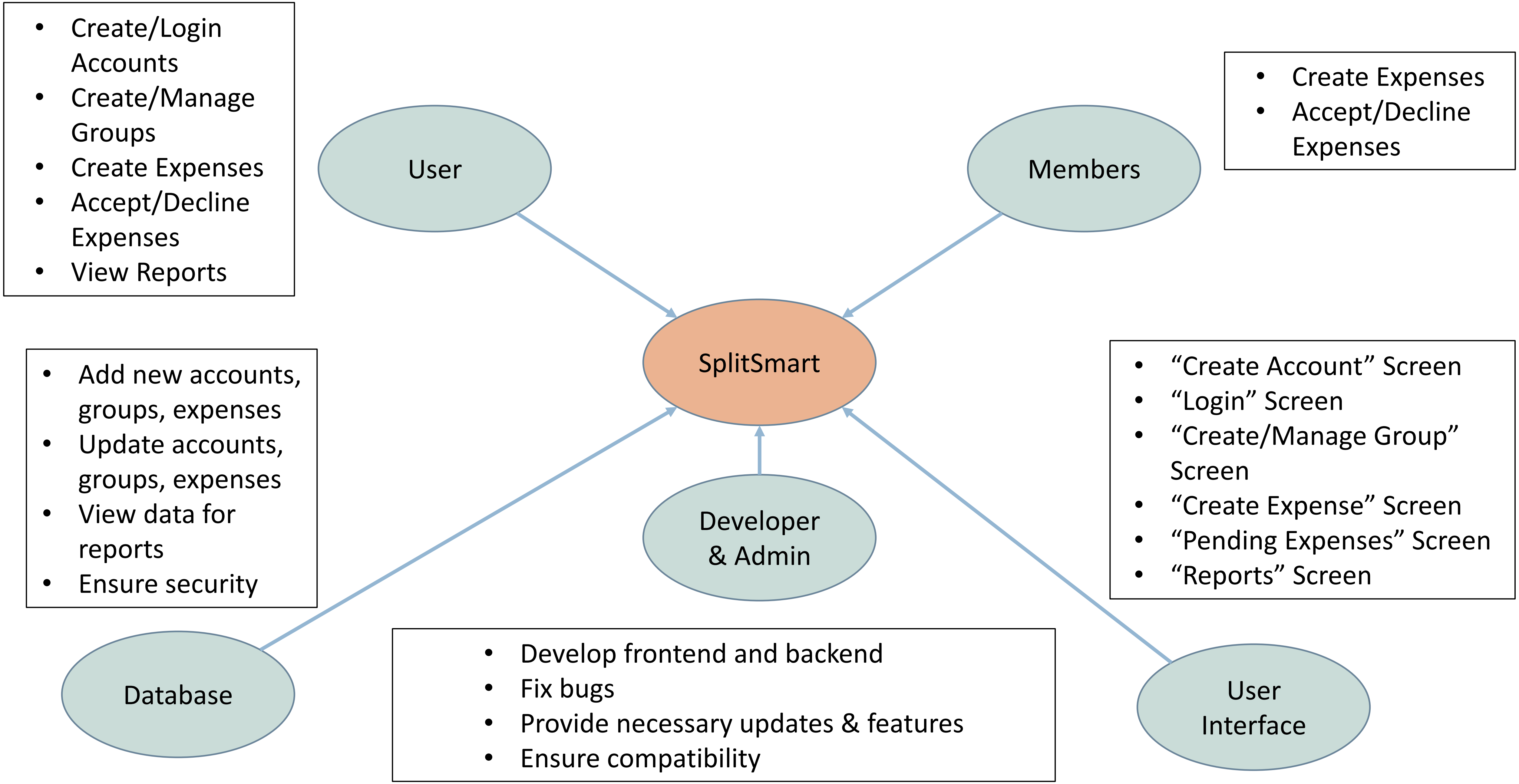
User Help

- Application will have built-in FAQ for questions and guidance

Customization

- Application will allow users to change their profile pictures
- Application will have light and dark modes

Context Model


• Create/Login Accounts
• Create/Manage Groups
• Create Expenses
• Accept/Decline Expenses
• View Reports
User
Members
• Create Expenses
• Accept/Decline Expenses
SplitSmart
• Add new accounts, groups, expenses
• Update accounts, groups, expenses
• View data for reports
• Ensure security
• “Create Account” Screen
• “Login” Screen
• “Create/Manage Group” Screen
• “Create Expense” Screen
• “Pending Expenses” Screen
• “Reports” Screen
Developer & Admin
Database
• Develop frontend and backend
• Fix bugs
• Provide necessary updates & features
• Ensure compatibility
User Interface

Use Cases

Create User Account

- Main Flow:
  1. User arrives at the “Create Account” screen
  2. System prompts user to enter their desired username and password
  3. User enters their desired username
  4. User enters their desired password
  5. User clicks “Create Account” button
  6. System verifies username isn’t already in use
  7. System verifies password satisfies requirements
  8. User is taken to the “Login” screen and prompted to login to account with new credentials
- The user will now be able to login to and use the application.

Login to User Account

- Main Flow:
  1. User arrives at the “login” screen
  2. System prompts user to enter their login username and password
  3. User enters their username
  4. User enters their password
  5. User clicks “Login” button
  6. System verifies login username and password are correct
  7. User is logged into the application and taken to the homepage
- The user can now progress throughout and use the application as intended.

Create Group

- Main Flow:
  1. User arrives at the “Create Group” screen
  2. System prompts user to enter their group name
  3. User enters their group name
  4. System prompts user to enter usernames for members and/or select usernames from shortlist
  5. User enters usernames for member and/or selects their username from shortlist
     a) Repeat until all desired members are added
  6. User clicks “Create Group” button
  7. Members are sent invitations to formally join new group, which they can select or decline
- The user will have created a group with desired members for them and others to create, manage, and share expenses within.

Manage Group

- Main Flow:
  1. User arrives at the "Manage Group" screen
  2. User is prompted to select one of the following:
     a) Adjust settings
     b) Add users
     c) Remove users
  3. User selects one of the above choices
  4. User clicks "Save Changes" button
  5. Added members are sent invitations to formally join new group, which they can select or decline
- The user will have adjusted the settings within a group, added additional member(s), or removed existing member(s)

Create Expense

- Main Flow:
  1. User arrives at the “Create Expense” screen
  2. User is prompted to enter a description, price, selects the applicable group(s) and applicable member(s), and how they would like to split the expense based on amounts or percentages.
  3. User enters necessary information
  4. User clicks “Add Expense” button
  5. Applicable member(s) are sent notifications of new expense to accept or decline
     a) If accepted; payment will be automatically deducted from associated bank account or application balance
     b) If declined; original expense creator will be notified, and expense will be subject to further evaluation

- The user will have created an expense that is shared amongst desired group(s) and member(s) to be accepted or declined

Accept Expense

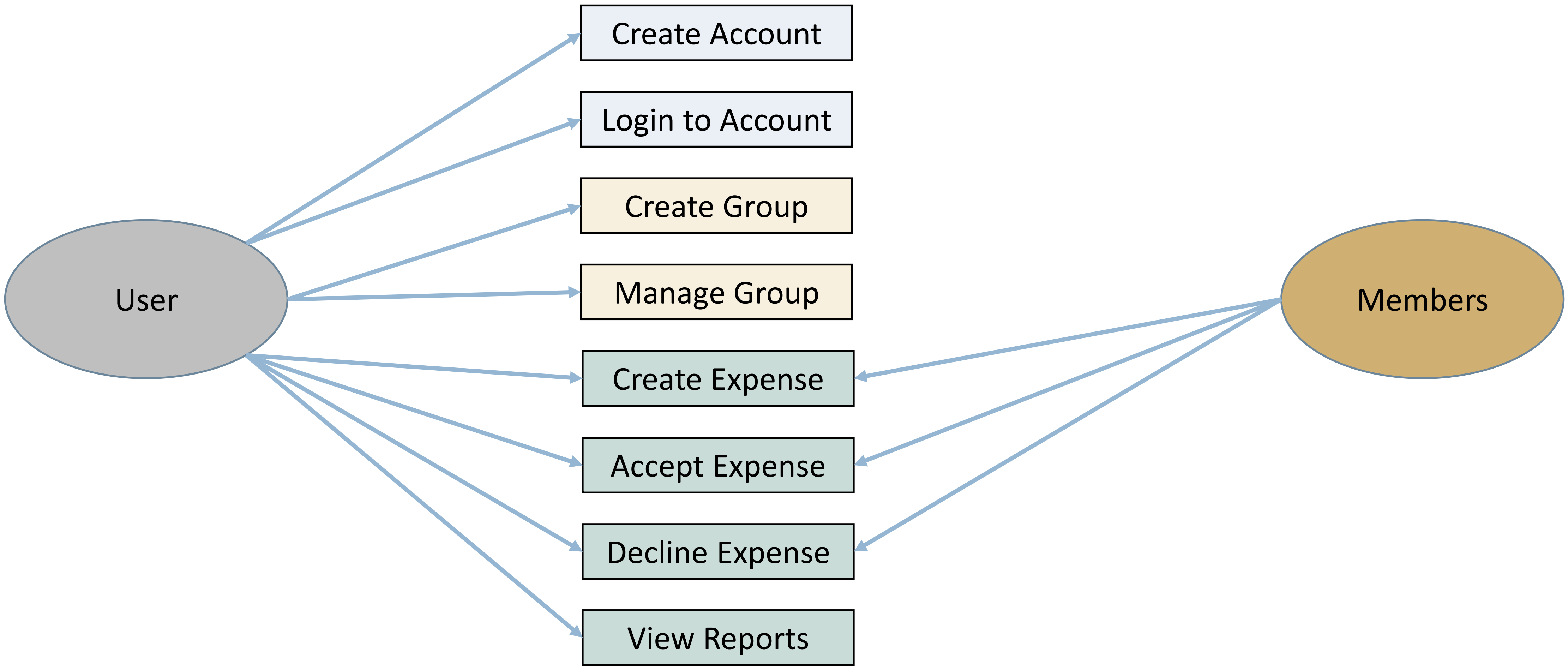
- Main Flow:
  1. User arrives at the “Pending Expenses” screen
  2. User selects the necessary expense to open details
  3. User selects “Accept Expense” button
  4. Expected amount will be automatically deducted from associated bank account or existing application balance

- The user will have accepted a shared expense and the payment will automatically be processed and sent

Decline Expense

- Main Flow:
  1. User arrives at the “Pending Expenses” screen
  2. User selects the necessary expense to open details
  3. User selects “Decline Expense” button
  4. Original expense creator will be notified, and expense will be subject to further evaluation

- The user will have declined a shared expense and the expense will be subject to further review by the original creator


User
Create Account
Login to Account
Create Group
Manage Group
Create Expense
Accept Expense
Decline Expense
View Reports
Members