Software Architecture and Design Document

Nick Azzouz, Christian Balan, Leonardo Diaz, Xavier Evans, Zane Richards

Table of Contents

1. Introduction

1.1. Purpose of Document

The purpose of this document is to serve as a comprehensive guide that outlines the structure, components, and behavior of the SmartSplit system. It will provide a clear and detailed description of how the software will be designed and implemented to meet the specified requirements.

1.2. Scope of Document

The document will cover the fundamental organization of the software system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution as per the IEEE definition of software architecture.

2. Architectural Representation

This document presents the architecture as a series of views; use case view, logical view, and deployment view. These views are presented using the Unified Modeling Language (UML).

3. Architectural Goals and Constraints

Below are some of the main requirements as well as constraints that must be kept creating the smart split system.

Goals:

- Be able to manage high traffic in seasons where group travel is more prevalent such as summer.

Constraints:

- Dev team is limited to only 5 members with only limited experience, thus there are limitations on what is achievable when creating the system.

4. Use-Case View

Below is a list of the most important use cases for the system as seen in the Software Requirements Specification document

:

- Create account
- Account login
- Manage account
- Manage payment method
- Manage transactions
- Make payment
- Manage group
- Manage friends

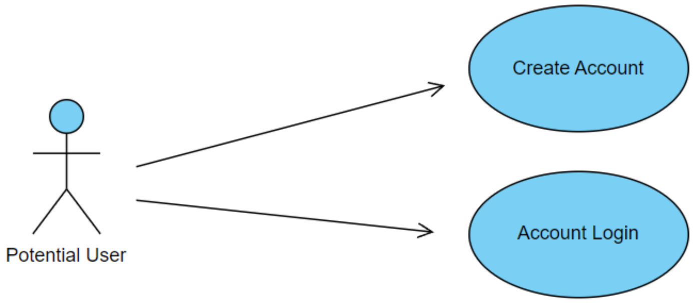


Fig 1. Unregistered User Use Cases

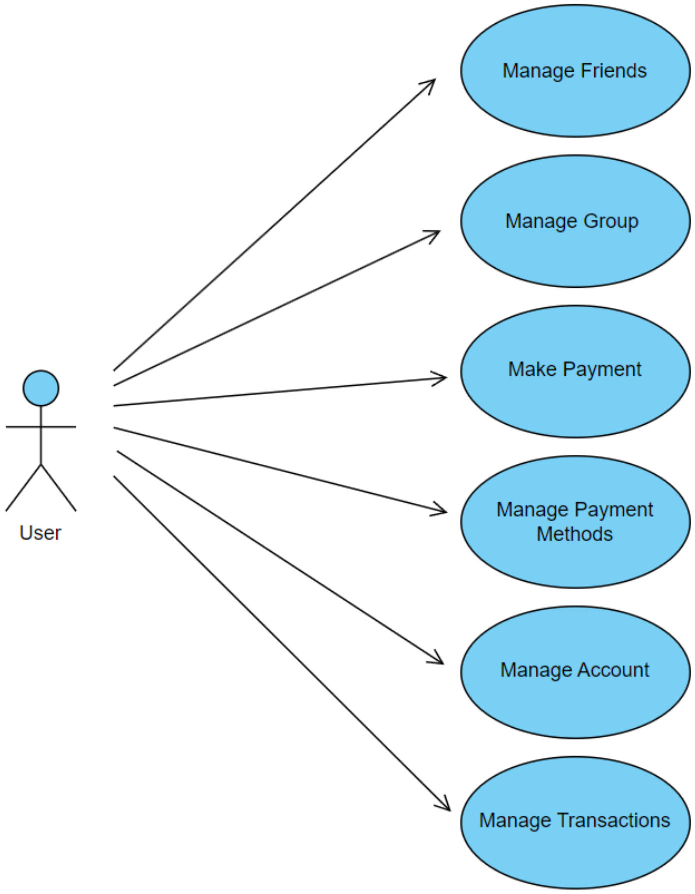


Fig 2. Registered User Use Cases

Use Case Descriptions:

1. **Create account:**
   This use case occurs when an unregistered user selects the create account option where they will be prompted to enter an email address, sufficient password, and complete a recaptcha.

2. **Account login:**
   This use case occurs when a user has created a valid account and received a confirmation email. They will be able to login to their account using the credentials they provided.

3. **Manage account:**
   This use case occurs once a user has logged in. They can manage their account information and make any edits needed. (Note any changes made to security settings will require a password confirmation prior.)

4. **Manage payment method:**
   This use case occurs when a user wishes to manage their payment methods. They can link bank accounts or other online payment systems such as Paypal or venmo to their account to be used for transactions.

5. **Manage transactions:**
   This use case occurs when a user has active transactions they have made. This allows the user to make changes to the transaction and change features like cost, splitting method, etc.

6. **Make payment:**
   This use case occurs when a user would like to settle an outstanding payment they have. The customer selects a transaction, confirms the amount owed and pays via their defined payment method.

7. **Manage group:**
   This use case occurs when a user would like to create a group with members of their friend list. The user selects the create group option, selects friends from their friend list to add to the group, and names the group.

8. **Manage friends:**

This use case occurs when a user would like to manage their friend list. The user will look up a friend by their username and have the option to add or delete them.

5. Logical View

Overview

A description of the logical view of the architecture. Describes the most important classes, their organization in service packages and subsystems, and the organization of these subsystems into layers. Also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes, subsystems, packages and layers.

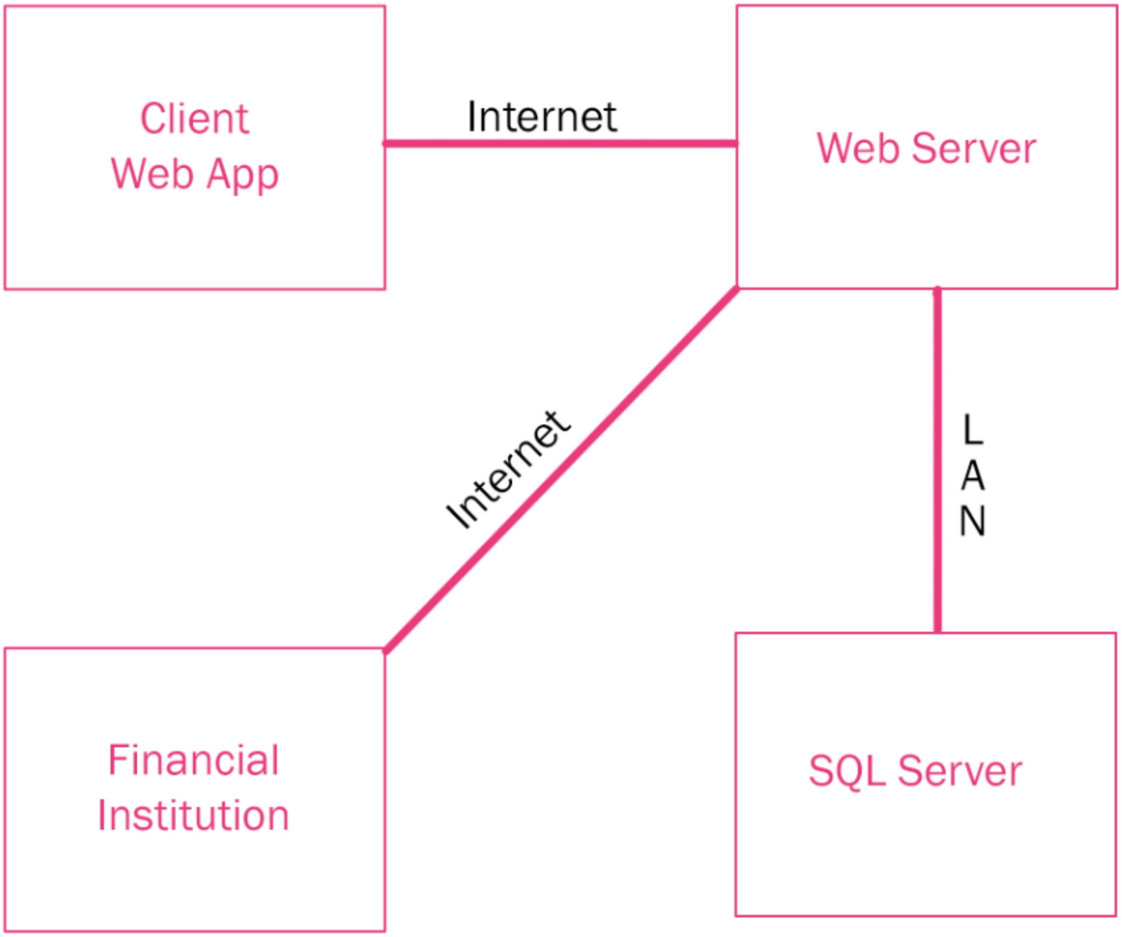
The logical view of the SmartSplit is comprised of 2 main packages:

- **Presentation**
  - Contains boundary classes to support the maintaining of profiles, payment viewing, balance viewing, viewing transaction reports, and viewing friends list/groups .
- **Application**
  - Contains control classes for major processing functionality within the system. Supports profile validation, payment processing, payment validation, payment disputes, report generation, and back-end friend/group management.

Link to Diagram: https://imgur.com/N0J3T6o

6. Deployment View

This section describes the physical network configurations in which the software will run. The client app will run on a modern web browser provided by a PC or smartphone. The Financial institution will communicate with the SmartSplit application on the web server to process transaction requests to bank accounts. The SQL server serves as the database link for holding user account information.



7. Size and Performance

The software as described in this document will support 100,000 concurrent users. Should it need to scale beyond this threshold, more server hardware will need to be deployed in covered regions.

8. Quality

The software described in the document will support the latest secure connection standards as required by external systems operated by financial institutions.