

CSC 431

Dashboard

System Architecture Specification (SAS)

<Member Name>	<Role>
<Member Name>	<Role>
<Member Name>	<Role>
<Member Name>	<Role>

Version History

1			First Draft

Table of Contents

- 1. System Analysis
 - 1.1 System Overview
 - 1.2 System Diagram
 - 1.3 Actor Identification
 - 1.4 Design Rationale
 - 1.4.1 Architectural Style
 - 1.4.2 Design Pattern(s)
 - 1.4.3 Framework
- 2. Functional Design
 - 2.1 Diagram Title
- 3. Structural Design
- 4. Behavioral Design

Table of Tables

<Generate table here>

Table of Figures

<Generate table here>

1. System Analysis

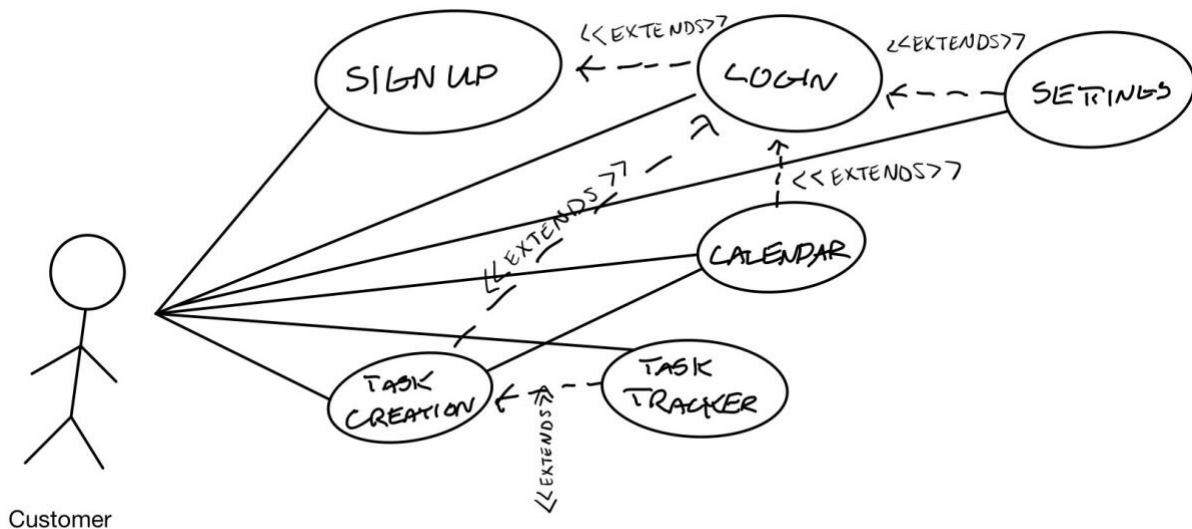
1.1 System Overview

The System will be comprised of four main parts: a calendar, task creation / management system, and settings page. Task creation / management and settings will be directly connected to the database. Data will be stored in the database and sorted / retrieved according to the user's needs. The database will store data via firebase using no-SQL database format. The calendar will display ongoing tasks and allow the user to view details on a specific task, including its main properties (description, tags, etc.) and whether it's marked as completed or not.

This pipeline is as follows: (1) the user submits a task creation request which triggers the task creation modal, (2) the controller initiates the process of querying data from the database based on the list of layers provided by the browser request, (3) the task is created and the calendar display the data to the user requesting it.

1.2 System Diagram

The following is our system diagram for Dashboard:



1.3 Actor Identification

There are two human actors involved in the use of Dashboard: users who have an account and users who do not yet have an account. Users who do not have an account are only able to access the signup and login pages until they

create an account. Users with accounts are able to access the login page as well as the calendar view and list view pages.

1.4 Design Rationale

1.4.1 Architectural Style

<Identify and briefly explain the architectural style e.g. 3-tier architecture>

1.4.2 Design Pattern(s)

<Identify the design pattern(s) you have deemed applicable to this architecture>

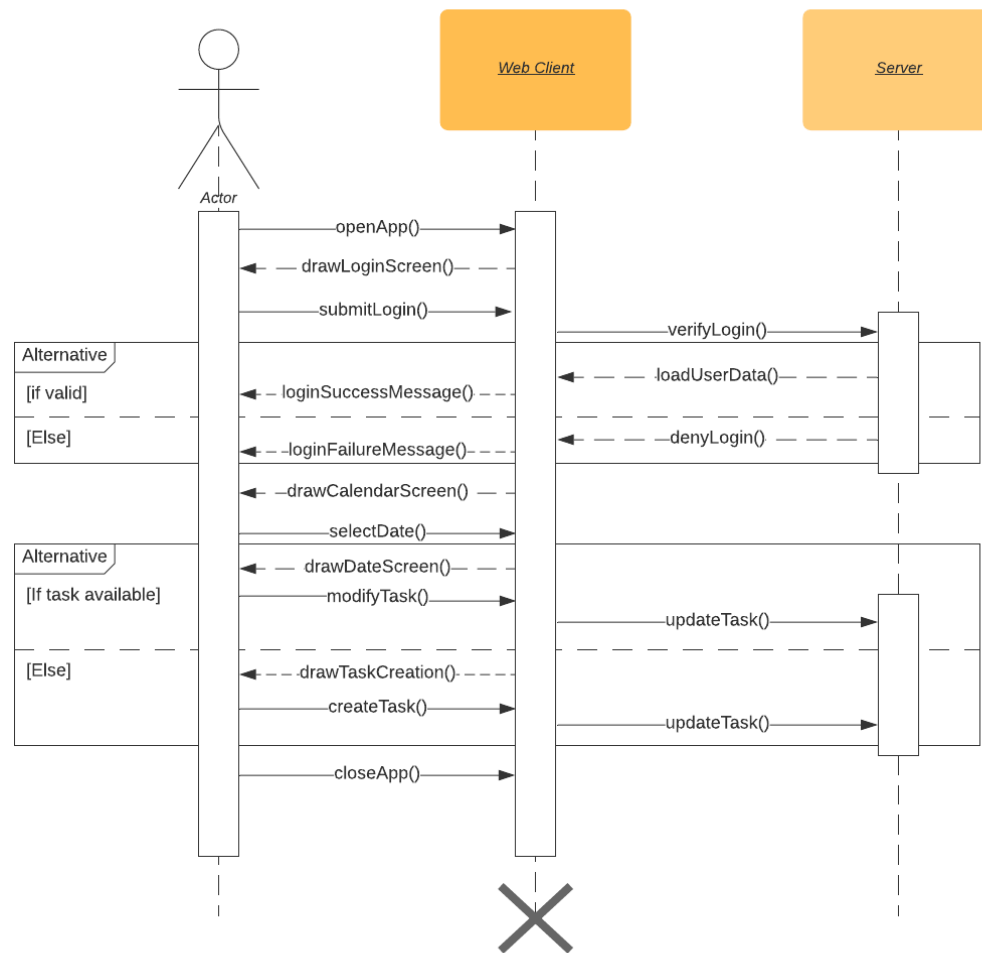
1.4.3 Framework

The web application will run with Responsive Web because it's lightweight and efficient.

2. Functional Design

2.1 Diagram Title

The following is our sequence diagram for use of Dashboard:



- When the user opens the application, the login screen opens.
- After submitting the login page, the login is verified and user data is loaded if the login was a success.
 - If the login was not a success, login is denied and a failure message is displayed.
- The calendar screen is then opened.
- Once a date on the calendar is selected, the date screen is opened.
- Within the date screen, if a task is available, it can be modified.
- If a task does not exist, the task creation modal opens.
- At the end of use, the application is closed (automatic logoff).

3. Structural Design

<Identify all components and model them using class diagrams>

The following is our class diagram for use of Dashboard:

