# 

# 

# 

# 

# **CSC 431**

# Alpha System Architecture Specification (SAS)

**<Team number>**

|  |  |
| --- | --- |
| Borui Wang | Developer |
| Morgan Ritter | Developer |
| Ross Pierson | Developer |
| Maya Miller | Developer |
| Hayeon Oh | Team Leader, Developer |
| Zhenyuan Wang | Developer |
| Dorel Last | Developer |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1 | 3/30/2020 | ALPHA | Creation of the document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[**1. System Analysis 6**](#_heading=h.3dy6vkm)

[1.1 System Overview 6](#_heading=h.1t3h5sf)

[1.2 System Diagram 6](#_heading=h.4d34og8)

[1.3 Actor Identification 6](#_heading=h.2s8eyo1)

[1.4 Design Rationale 6](#_heading=h.17dp8vu)

[1.4.1 Architectural Style 6](#_heading=h.3rdcrjn)

[1.4.2 Design Pattern(s) 6](#_heading=h.26in1rg)

[1.4.3 Framework 6](#_heading=h.lnxbz9)

[**2. Functional Design 7**](#_heading=h.35nkun2)

[2.1 Diagram Title 7](#_heading=h.1ksv4uv)

[**3. Structural Design 8**](#_heading=h.44sinio)

[**4. Behavioral Design**](#_heading=h.2jxsxqh) 9

# Table of Tables

<Generate table here>

# Table of Figures

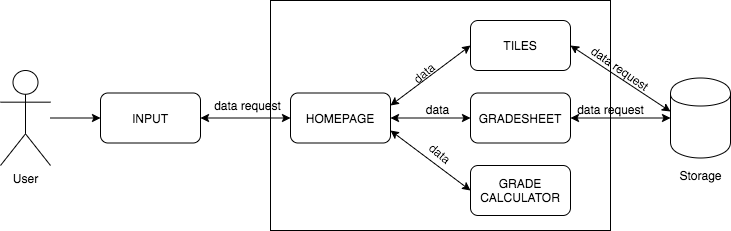
<Generate table here>

### System Analysis

### 1.1 System Overview

The system will help the user manage their courses by providing a grade calculator and organizational features. The system will be an application containing course tiles, an editable grade sheet, and a database containing all of the user’s inputs and information necessary to save the data and operate the system. The system’s primary function is to calculate the user’s class grades and display them in an organized manner. The database will display the grade sheet to the user, as well as update the grade sheet after the user has made changes. The web browser will fetch the data for the tiles and display them to the user via the homepage. The database also holds the external link, inputted by the user, updates the link when it is changed, added, or deleted, and displays it within the course tiles.

#### 1.2 System Diagram



#### 1.3 Actor Identification

There are two types of actors: authorized users and administrators. The authorized user is able to add, edit, or delete any information that they have uploaded. Administrators are able to manage the accounts of the users. The system must also support interaction from non-human actors to retrieve data from the databases.

#### 1.4 Design Rationale

##### 1.4.1 Architectural Style

This project primarily calls for a three-tier architectural style, where the client-server software architecture pattern will have the user interface, functional process logic, computer data storage and data access developed and maintained as independent modules on separate platforms.

##### 1.4.2 Design Pattern(s)

The design pattern deemed most applicable to our project was a Layered Pattern setup, including primarily three tiers. At the top is the presentation layer where the UI will be held, in the middle is the application layer housing all of the applications internal commands, and most abstractly is the data access layer to access relevant information pertaining to each user.

##### 1.4.3 Framework

For this project, we will primarily be using Angular, React and NodeJS because they best fit our requirements as they do streamline a lot of development mechanisms to allow continuous delivery and rapid deployment of the completed code.

### 2. Functional Design

*<Identify all significant workflows as sequence diagrams using the following format>*

#### 2.1 ALPHA - Sequence diagram

### 

### 3. Structural Design

#### 3.1 ALPHA - Class diagram

