MIAMI

CSC 431

# Medicollab

System Architecture Specification (SAS)

**<Team #6>**

| | |
|---|---|
| Arindham Aneja | Class Diagram, Design Rationale |
| Yiran Sheng | System Diagram |
| Peiyu Jiang | Sequence Diagram |

## Version History

| Version | Date | Author(s) | Change Comments |
|---|---|---|---|
| **1** | April 6th, 2021 | Arindham Aneja, Yiran Sheng, Peiyu Jiang | First Draft |
| **2** | May 5th, 2021 | Arindham Aneja, Yiran Sheng, Peiyu Jiang | Final Draft |

# Contents

# 1. System Analysis

## 1.1. System Overview

The purpose of the system is to provide a collaborative platform to medical students and professors. The intended audience of this project is medical students and professors. By using ‘Medicollab’, the students can search for their respective professors and can add them to their list. The “Medicollab” will be an Android and Iphone based application. To create a ‘Medicollab’, we will use Java as a backend programming language, XML as a frontend language, and SQL as a database. The purpose of this project is to provide the medical students a complete platform where they can see all their assignments, grades and collaborate with teachers. The students can save a lot of their time by managing all the things in one application and there is no need to use multiple other third-party social media applications. The application will be developed using the Model View Controller (MVC) architecture. By using MVC architecture, we will be able to add the new features in future easily. All the user interface related functionality will be implemented in View component, whereas the core logic or business logic of the application will be implemented in Model. The controller will be responsible for maintaining communication and data passing between view and model.

## 1.2. System Design

The system design is comprised of Model, View and Controller.

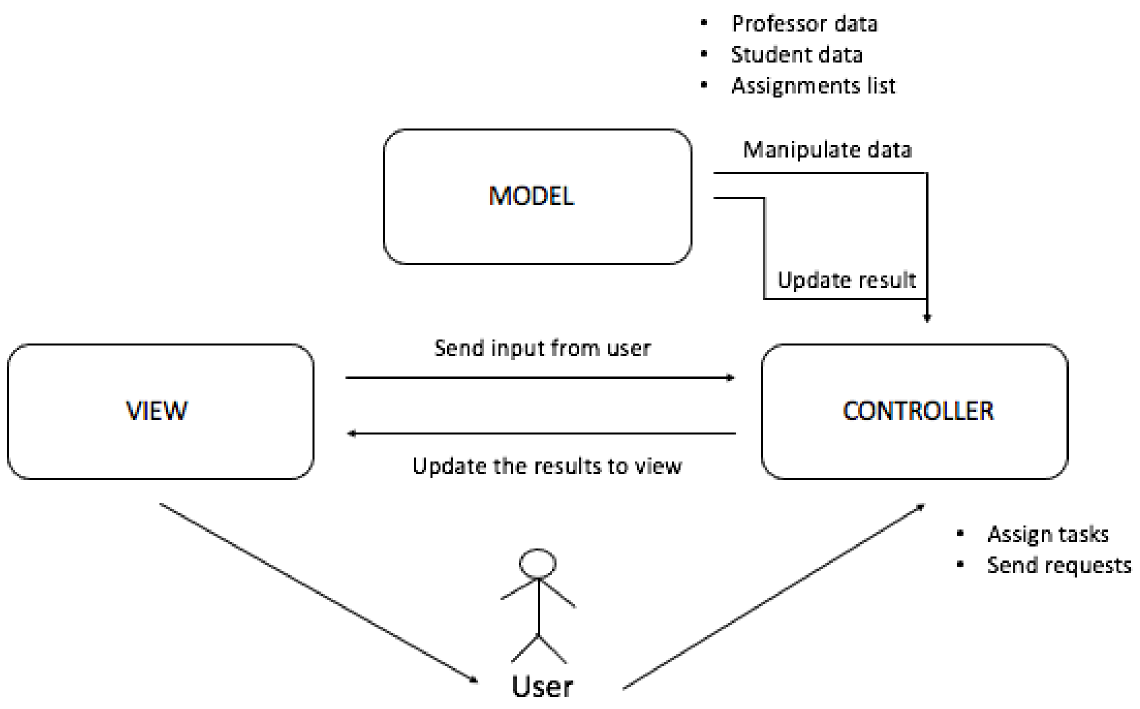


**Figure 1: System Design of Medicollab**

### 1.3. Actor Identification

The actors of this system are medical students and professors. Both the actors will sign up and Login in the system. The students can search for their respective professors and can add them to their list. A student will send a request to the professor. The professor can accept or reject the student request. The professor can assign different tasks or assignments to students. Moreover, the students can view their pending assignments, the grades of their completed assignments, and a schedule of their classes.

### 1.4. Design Rationale

#### 1.4.1. Architectural Style

The Medicollab will be an Iphone and Android based application. The Android application specifically follows the Model View Controller Architecture by default. The view component will handle all the user interface related functionality whereas the business logic of the system will be implemented in the Model component. The controller component ensures the communication between the view and model. The model component further interacts with the system database which store all the related information.

### 1.4.2. Design Pattern

The design pattern is reusable solution for commonly occurring problems.  The following design patterns will be used in the implementation of Medicollab application:

1- **Singleton:** We will use Singleton pattern to ensure that that there is only one instance of a class is created at one time. The Singleton pattern will be used in student and professor class to create only one instance of student and professor at one time.
2- **Factory:** Factory is a very commonly used design pattern. It is used to create the abstract objects that share the common properties. The factory pattern will applied in person class. The student and professor classes will extend the person class. Because the student and professor classes share some common properties, therefore, we will create the abstract objects of student and professor classes in person class by using the factory pattern.

### 1.4.3. Framework

We will use 'React Native' for Medicollab as it is slick, smooth and responsjve user Interface which can be applied to both Android and IOS platforoms. In the React Native Framework, we will use Java programming language to develop the application.

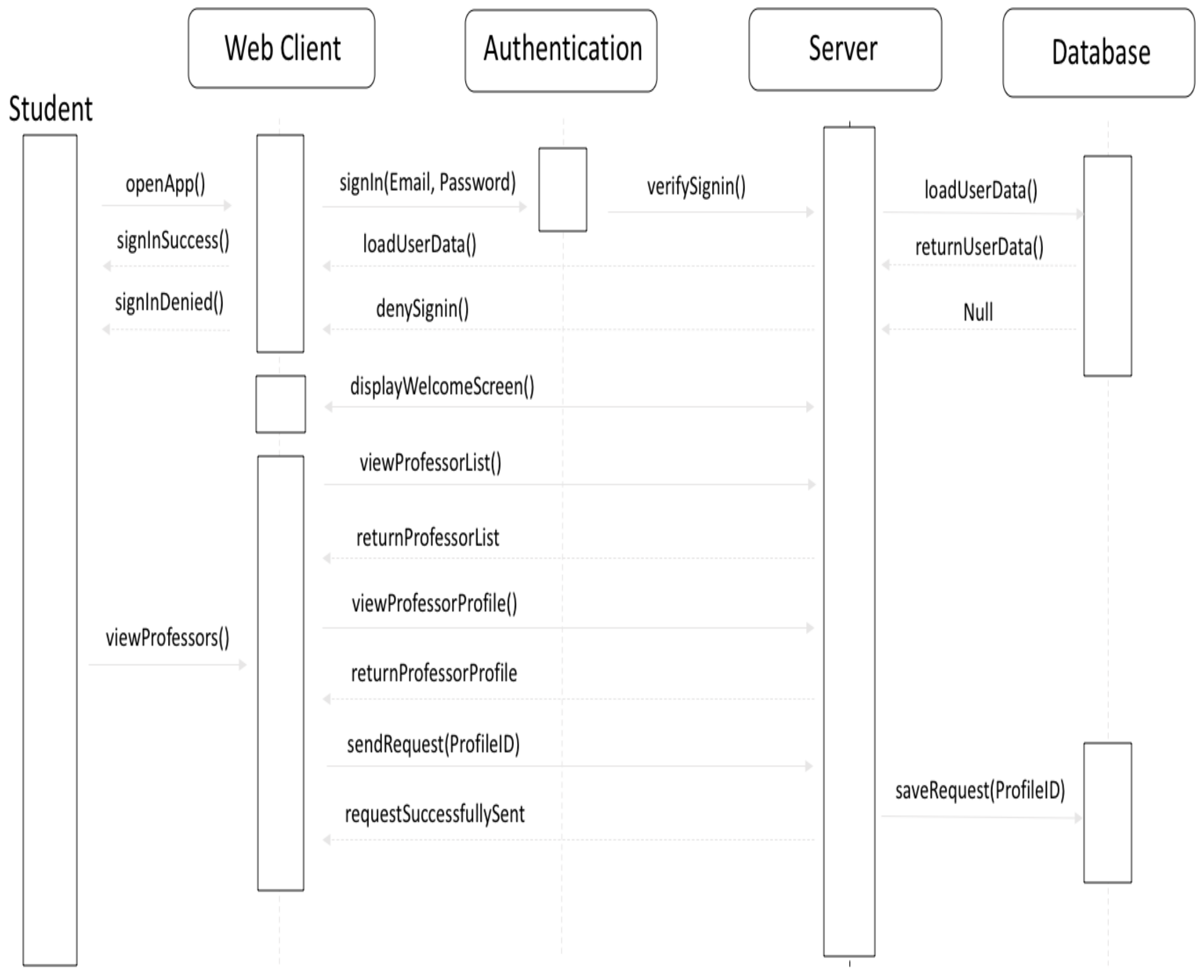
## 1.5. Cross-cutting Requirements

Following are the cross-cutting requirements of Medicollab application:

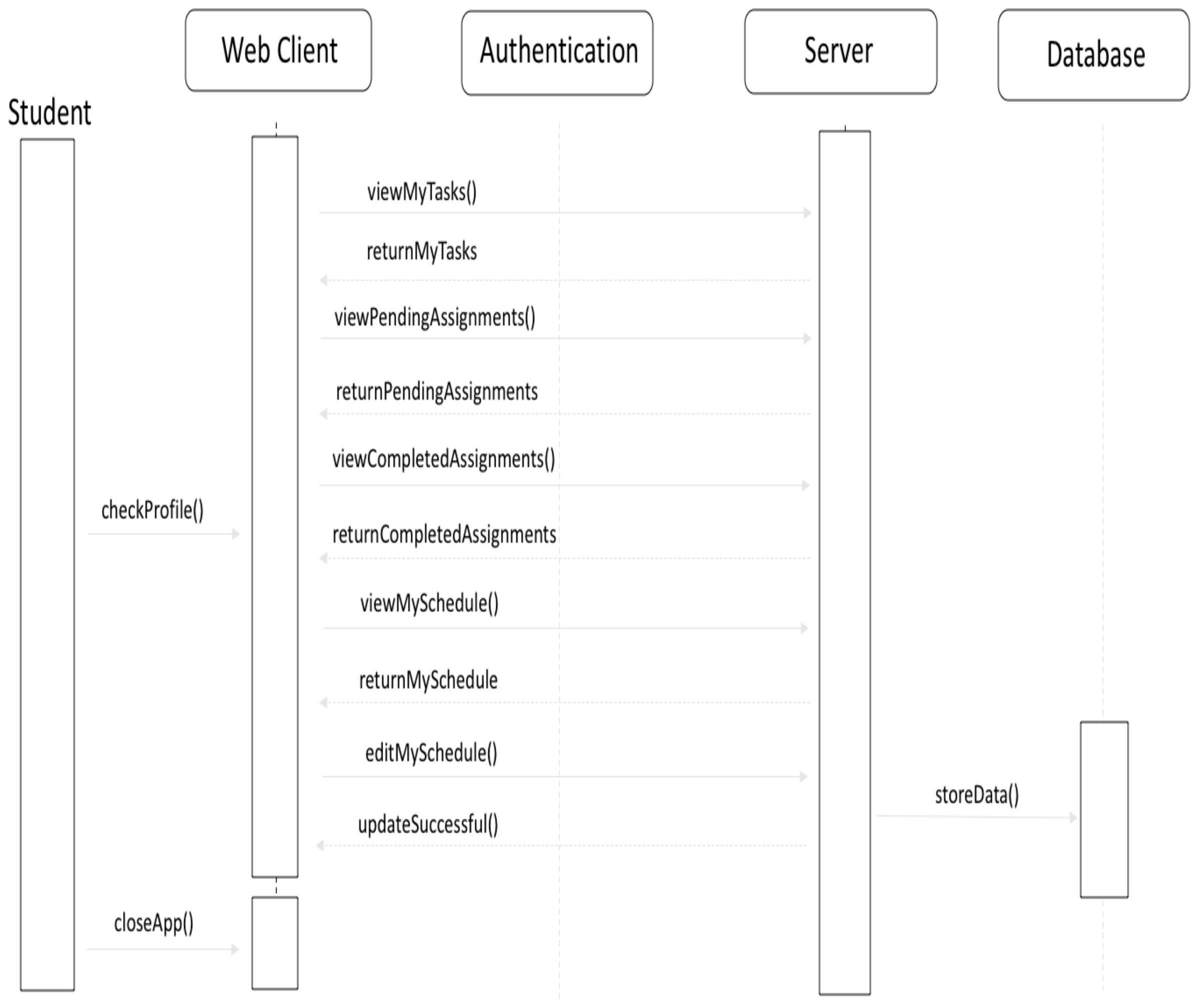
- **Sign up:** Signing up is a lengthy process in which the user fill out many required fields. It is a critical system requirement because a user cannot perform the other functionalities if the sign up is not performed.
- **Send Request to professor:** It is another cross-cutting requirement in which the student search for the available professors and send the request to professor. After that, the professor add the student in his/her student list.

# 2. Functional Design

In this section, the sequence diagram outlines all the necessary steps for sending a request to professor.



- In order to send request to the Professor, first, student opens the application and performs login with their email and password.
- The email and password is verified from the database which directs the user to the welcome screen or returns null which denies sign in.
- After the successful login, the welcome screen is displayed and the student can view professors list and professor's profile which is fetched from the server.
- When the professor's profile is displayed, the student can click on the send request button and all the related information is stored in the database.



- In the check profile class, a student can view their tasks, pending assignments and completed assignments which are all returned from the server and updated once the student completes another task.
- The student can also view and edit their schedule which gets updated in the database.
- After using the app, the user can close it which will exit from the app.

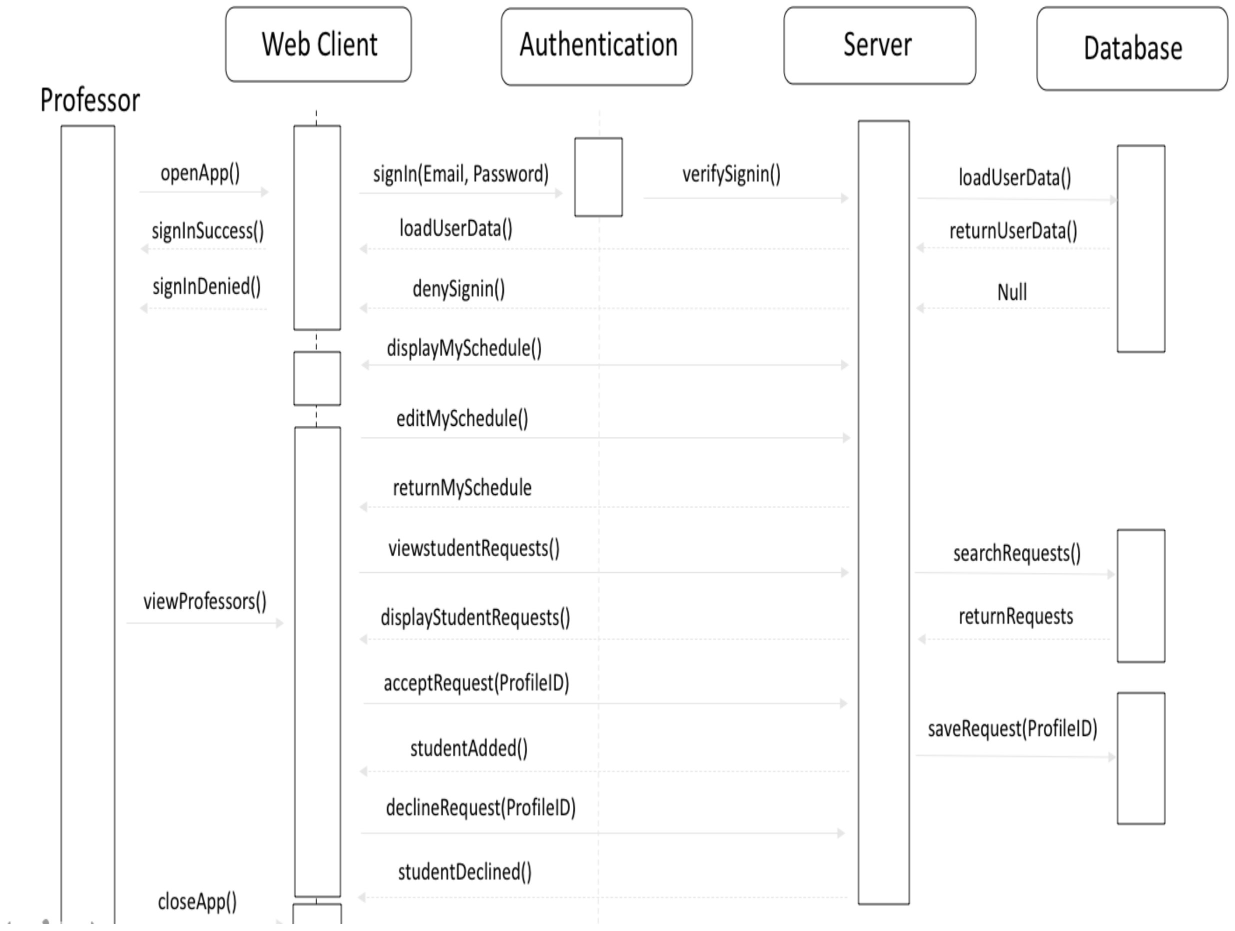


**Figure 2: Sequence Diagram for Medicollab**

- The second actor for this application, Professor, can open the app with their credentials which would require verification from the database and directs the user to the app if the credentials match or returns null and denies sign in if they don't match.
- The professor can fetch their schedule, which can be edited, and view student requests which are returned from the database.
- Furthermore, they can accept or reject requests which are updated in the database and notified on student's screen.
- Once done with the application, the professor can close the app which will exit from the system.

## 3. Structural Design

The below-given class diagram shows the essential classes and structural design of the Medicollab application.

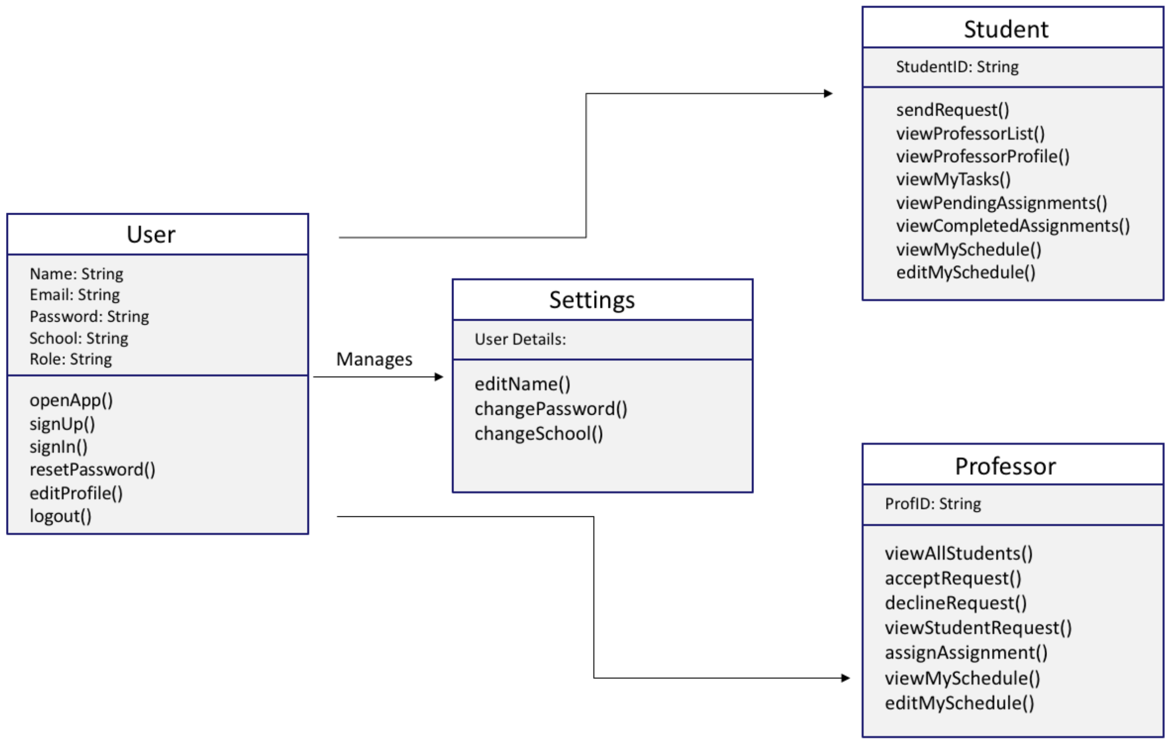


**Figure 3: Class Diagram of Medicollab**