UnderGrad Assistant Portal

Design Document

10/18/2018

15Z

Team 15Z

Members: Staroverov, Singh, Kim J.M., Kim. Jwahoon

Course: Cpts 322 - Software Engineering Principles I
Instructor: Sakire Arslan Ay

TABLE OF CONTENTS

I. Introduction

The purpose for the Design Document is to provide more specific information about the design and construction process of our project. This is a slight revision of an earlier document, Requirements Document, so we will give a summary of the changes during the revision.
The description of our project is as follows. Our project is designed to aid students and teacher in more conveniently finding teacher's assistant positions and teacher's assistants, respectively. This will be done through a website app that will help match student TA's with instructors based on their qualifications. Since, this is usually done by hand, having an algorithm to help sort and match and have the professors hand pick students will allow for a more calculated decision based on the program and professors. These are the sections of our project:

Section II includes, the Architecture design. This is describing the layout of how our program components will interact with each other.
Section III includes the Design Details. This is describing the specific components to our project.
Section III.1 includes the Subsystem Design. This is describing the details of how our subsystem components will be put to use.
Section III.1.1 includes the [Client]. This is describing the details of the front end of our development. This pertains to the structure of our front end and how it intertwines with some features of the API.
Section III.1.2 includes the [Server]. This is describing the details of the back end of our development. This pertains to the JSON elements and how the program uses SQL alchemy to share data.
Section III.2 includes the Data Design of our project. This is describing the details of how our data is structured. This section has information on how our attributes are laid out.
Section III.3 includes the User Interface Design. This is describing how the design is structured for the user interface. It includes screenshots and highlights of what each portion of our UI does.
Section IV includes the testing plan for this project. This is describing which modules in our program will be tested and which framework will be used.
Section V includes the references used in this project.

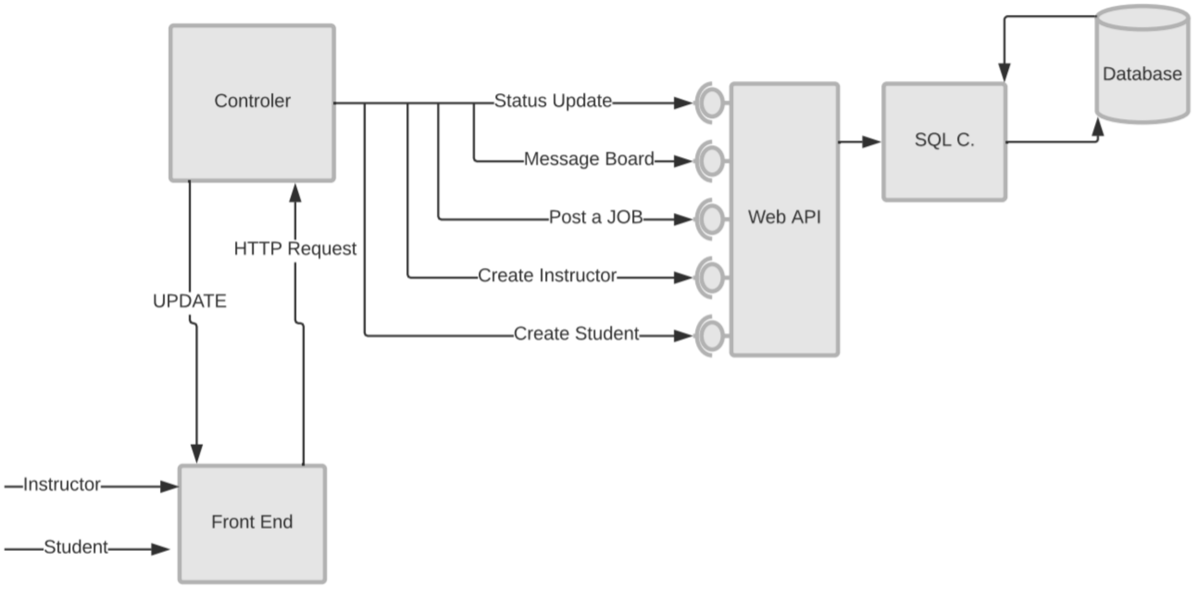
Document Revision History

Rev 1.0 <10/17/2018> <Introduction>
Rev 1.1 <10/18/2018> <User Interface, and sub systems>
Rev 1.2 <10/19/2018> < Data Design and Architecture Design>
Rev 1.3 <10/21/2018> <Finishing edits and sub systems>

II. Architecture Design

II.1. Overview

Three tier architecture will be used for the Architecture design. The three-tier architecture is great for web applications because it can be broken down to Web Browser, Web server, and a database parts. The Front-end block is the user interface for the Student or Instructor accounts. The front end sends a HTTP request to the controller, which is the javascript. The Controller analyzes the request and communicates with the Web API via routes. The Web API talks to the database and sends the response HTTP all the way back to the front end and updates the front end too. The BLock diagram is provided below.



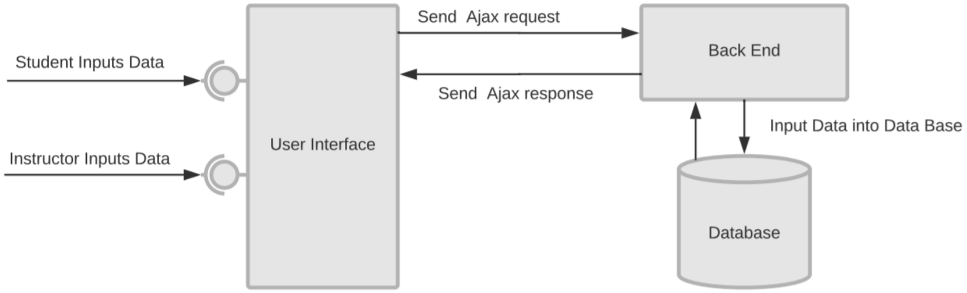
III. Design Details

III.1. Subsystem Design

There will be two subsystems in this project. One for the client and one for the server. The Client Subsystem will deal with all the front-end processes such as having user input data into the form and then send an AJAX request to the backend and then update the Backend if the command is Valid and then sends an AJAX response.

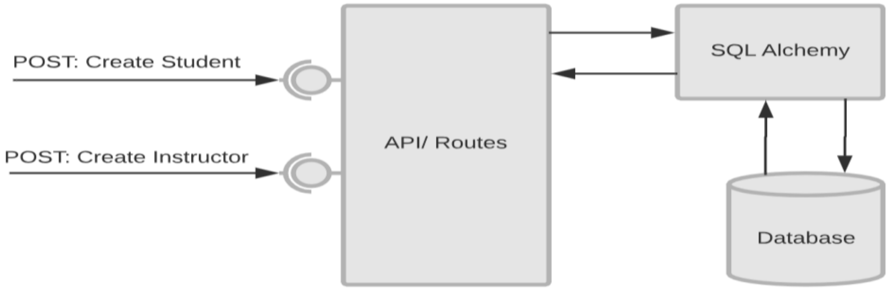
III.1.1. [Client]

The Client subsystem will be the all the front end development processes. At the moment, we only have the create Student and Instructor option available which will gather information from the user input and send an Ajax request to the back end and then receive an Ajax response if the command was valid. The server subsystem receives a POST request to make an account. The routes receive User data in JSON format and then the routes return User data for valid requests in JSON format. The routes communicate with SQL Alchemy by sending and receiving data from the database.



III.1.2. [Server]

The server receives a POST request to make an account. The routes receive User data in JSON format and then the routes return User data for valid requests in JSON format. The routes communicate with SQL Alchemy by sending and receiving data from the database.



III.2. Data design

There were two database tables created for the first iteration of our project; one for the student account information, and another for the instructor account information. Each of the two tables contained overlap as both are used to store user information. The following are lists of each attribute of each table created.

Student:

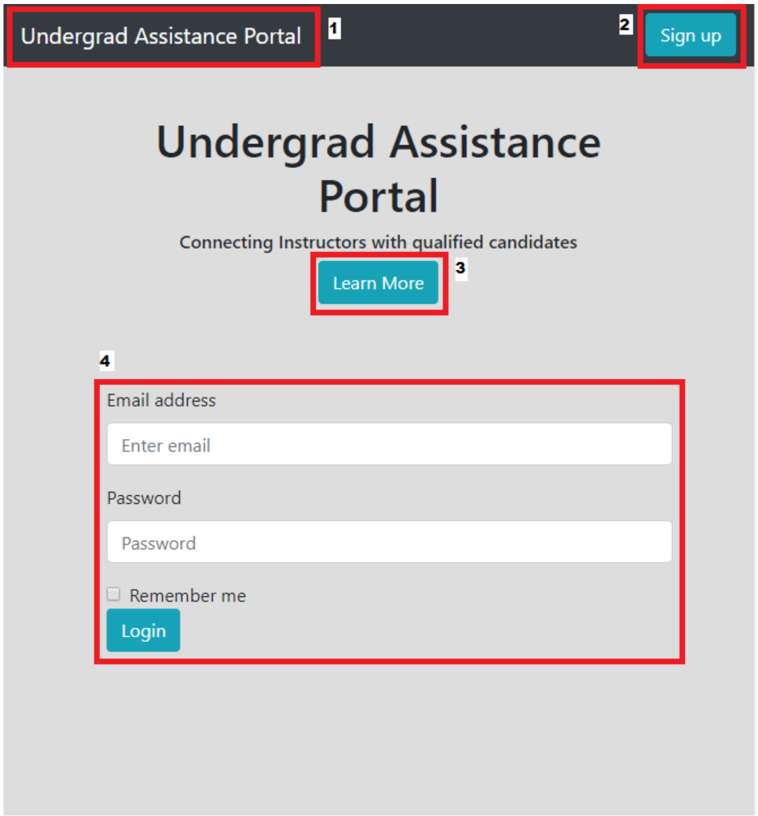
- Id
- Space
- First name
- Last name
- Email Address
- Password
- Date Created
- Major
- GPA
- Graduation Date

Instructor:

- Id
- Space
- First name
- Last name
- Email Address
- Password
- Date Created
- Phone Number
- Office Location

III.3. User Interface Design

<Home Page>



Undergrad Assistance Portal

Connecting Instructors with qualified candidates

1. “Homepage” button

   - This button letting you go back to the main page

2. “Sign up” button (use case #1 - Create)

   - This button pop up the sign up page

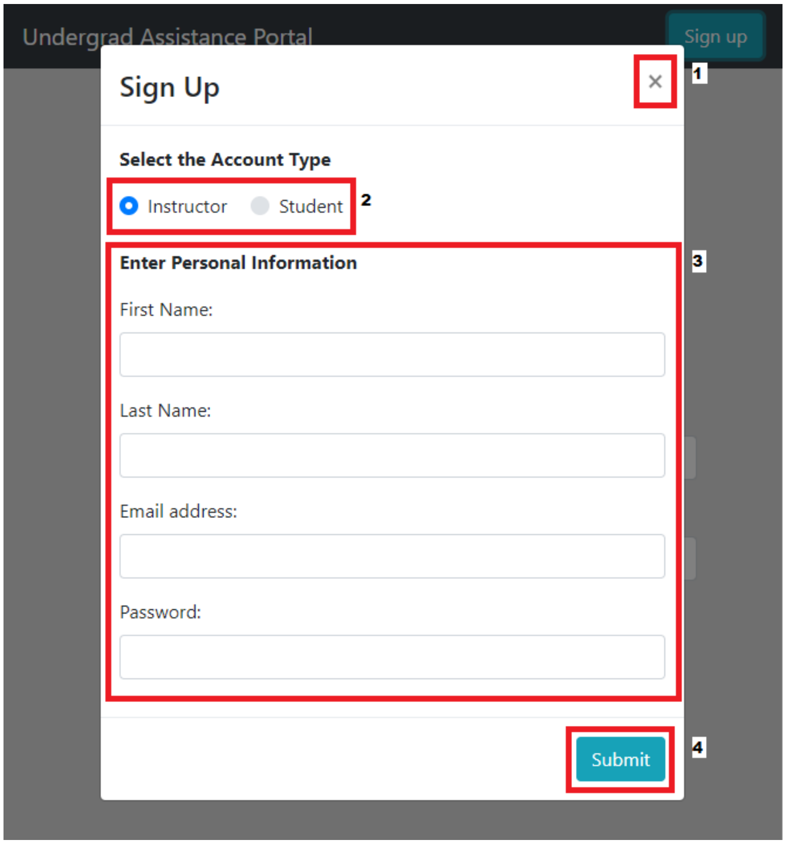
3. “Learn More” button

   -

4. “Login” section (use case #2 - Login)

   - In this section, you can type the account information to access in to the portal
   - Requests the user’s Email address and Password.
   - The users input their Email address and Password.
   - Clicking “Login” button, users can access to their account.

<Sign Up page>



Sign Up ×

Select the Account Type

● Instructor ○ Student

Enter Personal Information

First Name:

Last Name:

Email address:

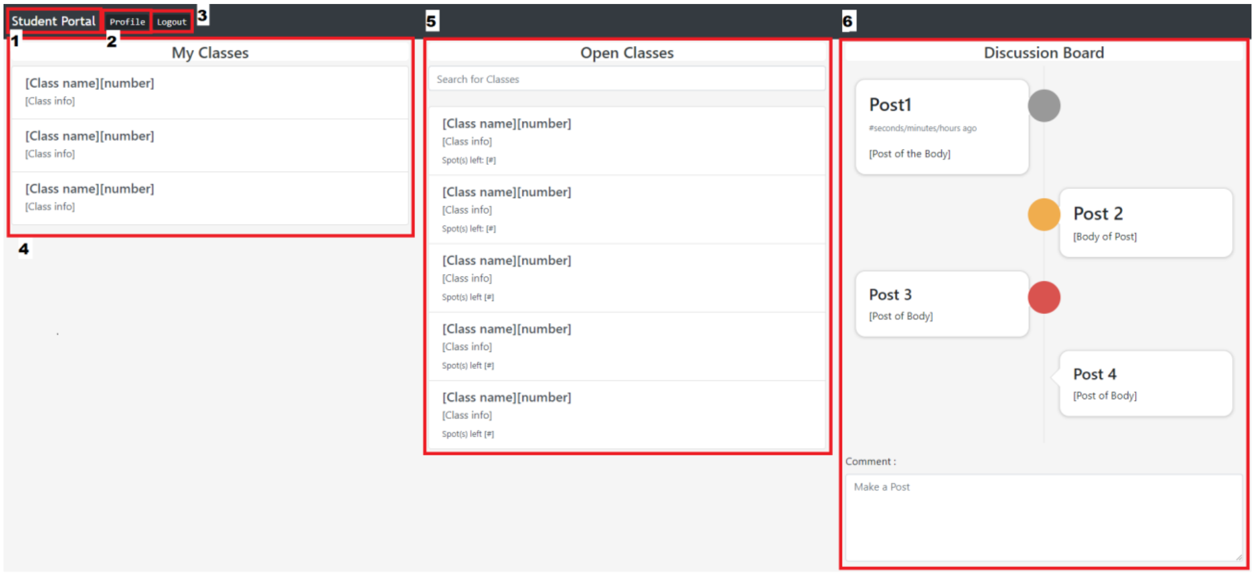
Password:

Once you click the “Sign up” button from main page, this Sign Up page pop up (use case #1 - Create)

1. “Exit” button
   - By clicking the button users can close the popped up screen.
2. “Select Account type” section
   - Users can select whether sign up for instructor or student.
3. “Personal Information” section
   - Users are required to enter personal information.
4. “Submit” button
   - By clicking the button account is created successfully.

Once user login in the main page, user can access to their portal page.

<Student Portal Page>



1. "Portal" button

   - By clicking "Portal" button, user can come back to main user portal page from wherever user was on

2. "Profile" button

   - By clicking "Profile" button, user can access to personal profile page, where user can see their profile which can be added, edit or delete

   - (Use Case #5 Enter Contact Info) In the profile user can enter and edit contact info. (user's first name, last name, WSU ID, email, and phone number)

   - (Use Case #6 Enter Student Info) In the profile user can enter and edit student info. (GPA, graduation date, prior TA experience, grade received, etc.)

3. "Logout" button

   - By clicking "Logout" button, user can log out their account from the portal and directly go back to the main page where user can login.

4. "My Classes" section

   - In this section, user can see which classes they added for TAship preference.

   - (Use case #7 Awaiting Status) User can see current status of the TAship, by clicking on the status option. (assigned, waiting, or declined)

5. "Opened Classes" section

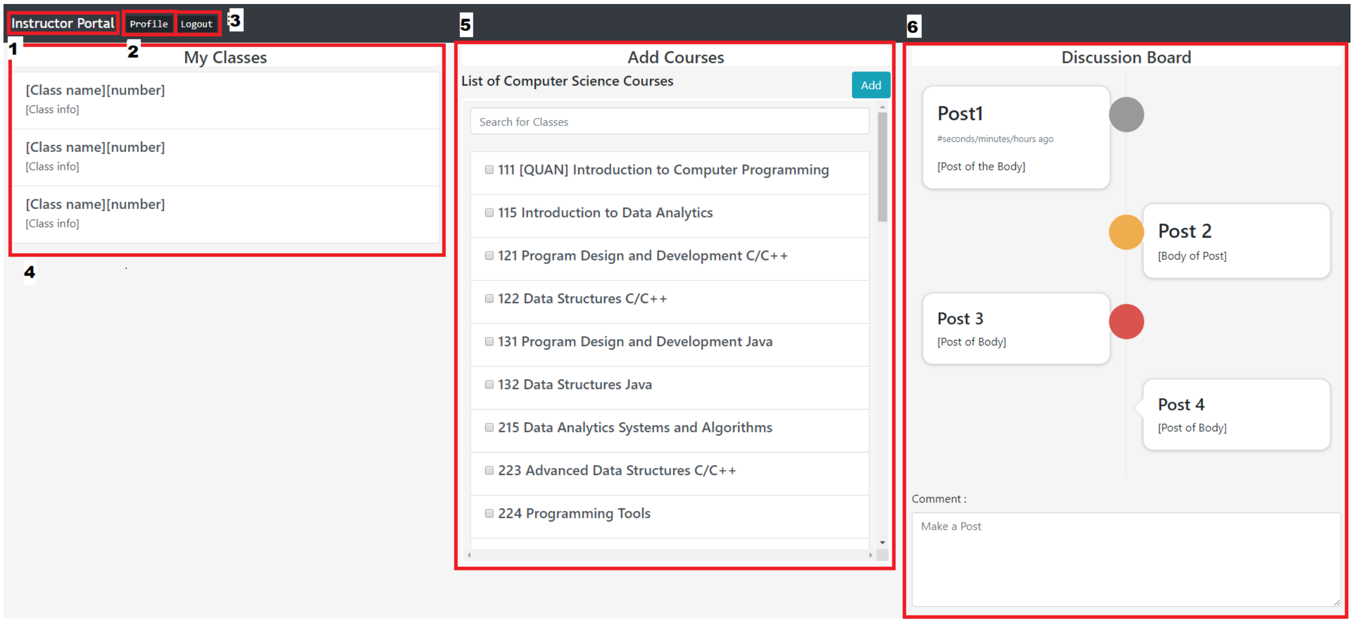
   - In this section, user can search and find which classes are opened for TAship.

   - (Use Case #3 Search Courses) There is search courses, where user can search for preferred course.

6. “Discussion Board” section

   - (Use Case #9 Instructor and TA message board) In this section, students and instructor can have discussion freely.

<Instructor Portal Page>



My Classes

[Class name][number]
[Class info]

[Class name][number]
[Class info]

[Class name][number]
[Class info]

Add Courses

List of Computer Science Courses

Search for Classes

- □ 111 [QUAN] Introduction to Computer Programming
- □ 115 Introduction to Data Analytics
- □ 121 Program Design and Development C/C++
- □ 122 Data Structures C/C++
- □ 131 Program Design and Development Java
- □ 132 Data Structures Java
- □ 215 Data Analytics Systems and Algorithms
- □ 223 Advanced Data Structures C/C++
- □ 224 Programming Tools

Discussion Board

Post1
#seconds/minutes/hours ago
[Post of the Body]

Post 2
[Body of Post]

Post 3
[Post of Body]

Post 4
[Post of Body]

Comment :

Make a Post

1. “Portal” button

   - By clicking “Portal” button, user can come back to main user portal page from wherever user was on

2. “Profile” button

   - By clicking “Profile” button, user can access to personal profile page, where user can see their profile which can be added, edit or delete

   - (Use Case #5 Enter Contact Info) In the profile user can enter and edit contact info. (user’s first name, last name, WSU ID, email, and phone number)

   - (Use Case #6 Enter Student Info) In the profile user can enter and edit student info. (GPA, graduation date, prior TA experience, grade received, etc.)

3. “Logout” button

   - By clicking “Logout” button, user can log out their account from the portal and directly go back to the main page where user can login.

4. “My Classes” section

   - In this section, user can see which classes they currently teach.

   - (Use Case #8 Instructor assigns TA to course) Instructor can assigns TA to the course when they click their class from this section. (list of students will show up after clicking)

   - (Use Case #10 Instructor removes TA from course) Instructor can removes or edit when they click their class from this section. (list of students will show up after clicking)

5. "Add Courses" section

   - (Use Case #4 Post a TA position) Instructor will add their courses which need TA position.

   - List of all the Computer Science is in the box and instructor can check the box then add.

6. "Discussion Board" section

   - (Use Case #9 Instructor and TA message board) In this section, students and instructor can have discussion freely.

IV. Testing Plan

Unit Testing: The team will plan automated unit testing to mainly the back-end aspects of the project. This will include testing our functions that require responses from the back end in order to update or store data. Although unsure at this point in the project, we are contemplating using HtmlUnit as our framework for testing.

Functional Testing: The team will go through the possible use cases manually. We will conduct running tests on the program from the perspective of the user. This is to understand fully how the use cases need to be, before the final iteration of the project.

UI Testing: The team will go through all the possible aspects of the user interface manually. We will run through each of the components in the UI to ensure that everything is working properly. We will also use the UI in perspective of the user so that we can best estimate the quality needed before the final iteration of the project.

V. References