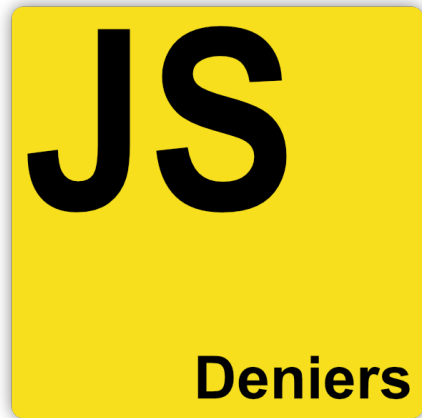# Research Outreach

## Design Document

10/25/2021

Iteration 1

**JavaScriptDeniers**

Bobby Templin, Dylan Tarlyn, Keagen Brendle, Wen Da Liu

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

TABLE OF CONTENTS

# I.  Introduction

This document serves to provide a blueprint for how we plan to design different aspects of our project. This includes things such as the models we plan to use and how they interact, the UI that the user will interact with, the subsystems of a project and how they interact, and various other diagrams and models that work to document the process of everything that goes into the design.

Our group aims to create a website that can be used by both students and faculty at Washington State University. The main purpose of this website is to display research positions within the college. Faculty can post their open research positions and elaborate on time commitments, eligibility, qualifications, and receive information from potential research assistants. And on the other end, students can view these posts regarding research positions and reach out if they feel inclined. Restrictions on who can view/post what will be crucial for appropriate user functions based on credentials.

Within this design document we will take you on an overview of the progress we have made for the first iteration of this project. We will begin with a description of the system architecture we used to develop this project accompanied by an illustration of the Model View Controller format. Additionally, from there we take a deeper dive into the individual components of MVC and explain the functionality of any subsystems within. We wrap up with an explanation of routes used for our URLS, a rundown of the visuals we used for View, and conclude with a brief restatement of our progress so far.

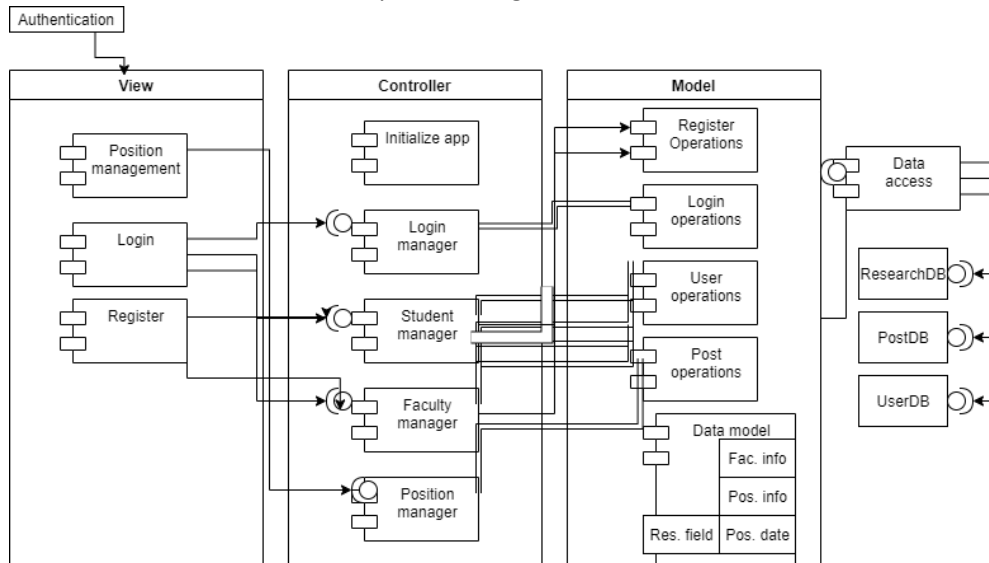Thanks in advance for your time

-JS Deniers

**Document Revision History**

Rev 1.0 - 10/27/2021 - Initial version

# II. Architectural and Component-level Design

## II.1. System Structure

UML Component Diagram - MVC Architecture



Our project is using the Model View Controller System Architecture (MVC). We have experience using MVC with our prior class projects, and it is a popular architecture in the industry as it "isolates the application logic from the user interface layer and supports separation of concerns" (TutorialsPoint). This site needs to handle alternate views and permissions based on whether a user is signed in, and further whether that user is a student or faculty. Separating forms, models, and controllers for these different instances lowers coupling and enables us to make changes to these databases as needed without messing up the rest of our subsystems. Additionally, each of these models only containing functions pertaining to their data set or purpose increases cohesion of our software. This is beneficial to us overall as our aim is for low coupling and high cohesion.

## II.2. Subsystem Design

### II.2.1. Model

The purpose of the post model is to store all the values associated with each post created by faculty members. These values are populated when a faculty fills them out through a form, and are displayed on the home page to students.

- Post(db.Model)
    - id: An integer value that will store ids as a primary key for each post on our main page
    - title: A string value that will represent the title of each post
    - date1: A DateTime value that will display the start time for each position
    - date2: A DateTime value that will display the end time for each position

- time: An integer value that will display the time required for each position
- research_field: A multiselect value that will hold the research fields available to apply to the position
- requirements: An string value that will represent the requirements of each position
- faculty_info: A string value that will hold the faculty contact information for each position

The purpose of the user model is to store all of the information associated with the users. These values are populated when a user registers for the site using the registration form. Users can either register as students or faculty. Faculty are able to create research positions, and students are able to apply for these positions.

- User(db.Model):
  - id: An integer value that will store ids as a primary key for each user on our main page
  - username: An string value that will hold the username
  - email:An string value that will hold the email
  - password_hash:An string value that will hold the password
  - usertype: A string value that will determine the usertype for the account

The purpose of the research model is to store the values for the different tags that a research position can have. The purpose of these tags is so faculty are able to identify the research areas that their positions are associated with, and so that students can apply to relevant positions.

- Research(db.Model):
  - id: An integer value that will store ids as primary key for each research type
  - field: A string value that will store each predefined research type

The purpose of the research_tag table is to associate the above tags with posts. The values are stored as foreign keys to link them to the research and post models.

- research_tag(db.Table):
  - post_id: An integer value that will store ids as foreign key to link the research model to posts
  - research_id:An integer value that will store ids as foreign key to link predefined tags to the research model

## II.2.2. Controller

The controller component is what allows for the connection between View and Model, it is the "middle man" in some senses. It reads in user input and then instructs the model what actions to take with the data objects which can be presented to the user within view.

Subsystems
- __init__.py

Design Document 4

- ○ Responsible for launching the website and importing necessary extensions. Interacts with the forms and models and the imported libraries are what are interdependent between them.
- errors.py
    - ○ Enables easy display of errors to users. Errors.py connects with the database and the templates to receive error data and to flash messages.
- forms.py
    - ○ Instantiates data members for use in forms on the site. Enables user input for information pertaining to posts they create with details regarding positions. Interacts with the templates within view to communicate information that has been entered by the user to be displayed to the site.
- routes.py
    - ○ Services the URLs for different paths within our site. This allows us to dictate whether information is sent and/or received on each page, and directs to new pages when necessary. Routes.py interacts with the forms and the templates so that any data received can be sent to the proper display page and utilizes imported Flask libraries to verify this data when appropriate.

| Methods | URL Path | Description |
|---|---|---|
| [GET], [POST] | /editprofile | A user may want to update the research topics they are interested in, or update any of their personal information. |
| [DELETE],[POST] | /delete | A faculty member may want to delete a research position that is no longer available, or is no longer being offered. |
| [GET] | /myapplications | A student will be able to view all of the applications that they have applied to. |
| [GET],[POST] | /mypositions | A faculty member will be able to view all of the positions that they have created. Included on this page is options to update the status of positions |
| [GET],[POST] | /apply | A student will be taken to a new route where they can apply for the position. This is where an application form will be rendered. |
| [POST] | /withdraw | A student will be able to withdraw from positions |
| [GET] | /qualifications | A faculty member will be able to the qualifications of the applicants to positions |

**II.2.3.  View and User Interface Design**


The role of the view in this project is to render pages using HTML. Our current user interfaces involve using CSS classes applied to HTML elements which include a theme that centers around using WSU brand images in order to provide a consistent layout. In the future we plan to revise our user interfaces to use Bootstrap in order to render elements.

**Landing page:** The landing page is the index route and the default page when visiting the site. This page simply displays the links to register if a user would like to create a new account, or login if they already have an account.
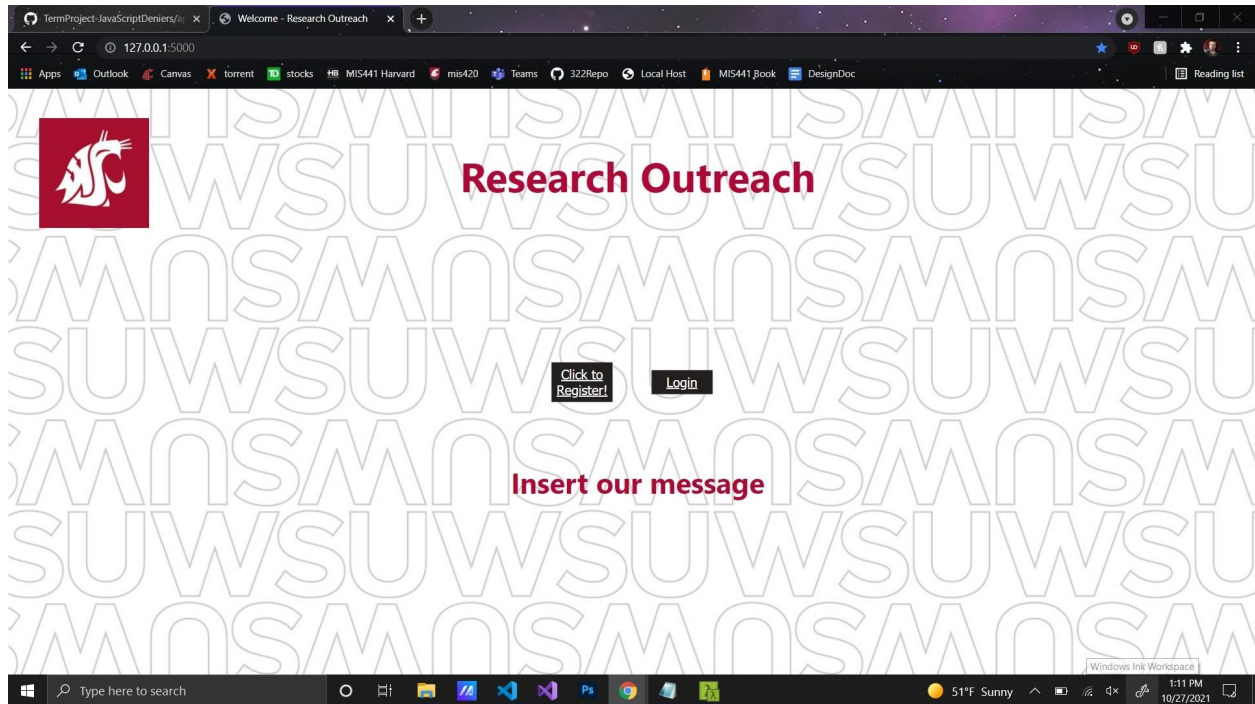


**Home page:** The home page is the default page when a user is logged in. The home page will contain all of the research positions that have been created, as well as a navigation bar at the top for users to navigate the site. We plan to update this UI to match the rest of the theme.
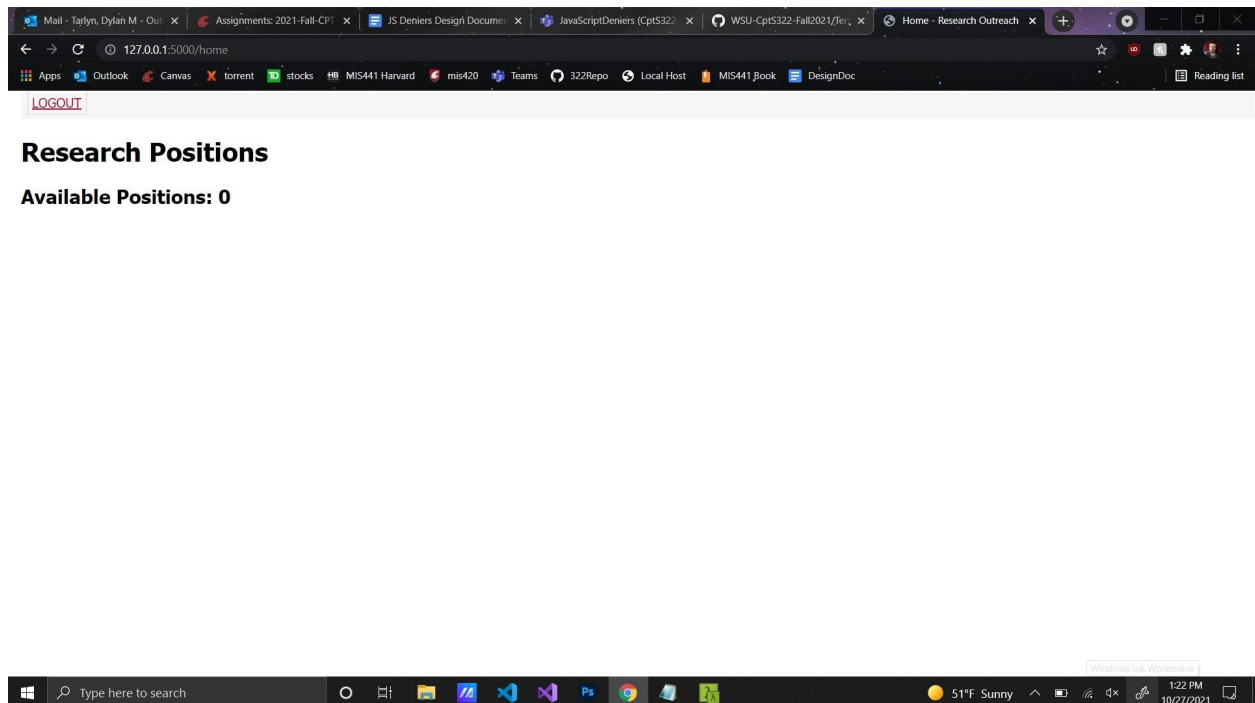
**Login/Register**: The login and register pages contain the same basic UI where users can input their information accordingly to either register a new account, or log into their existing account.
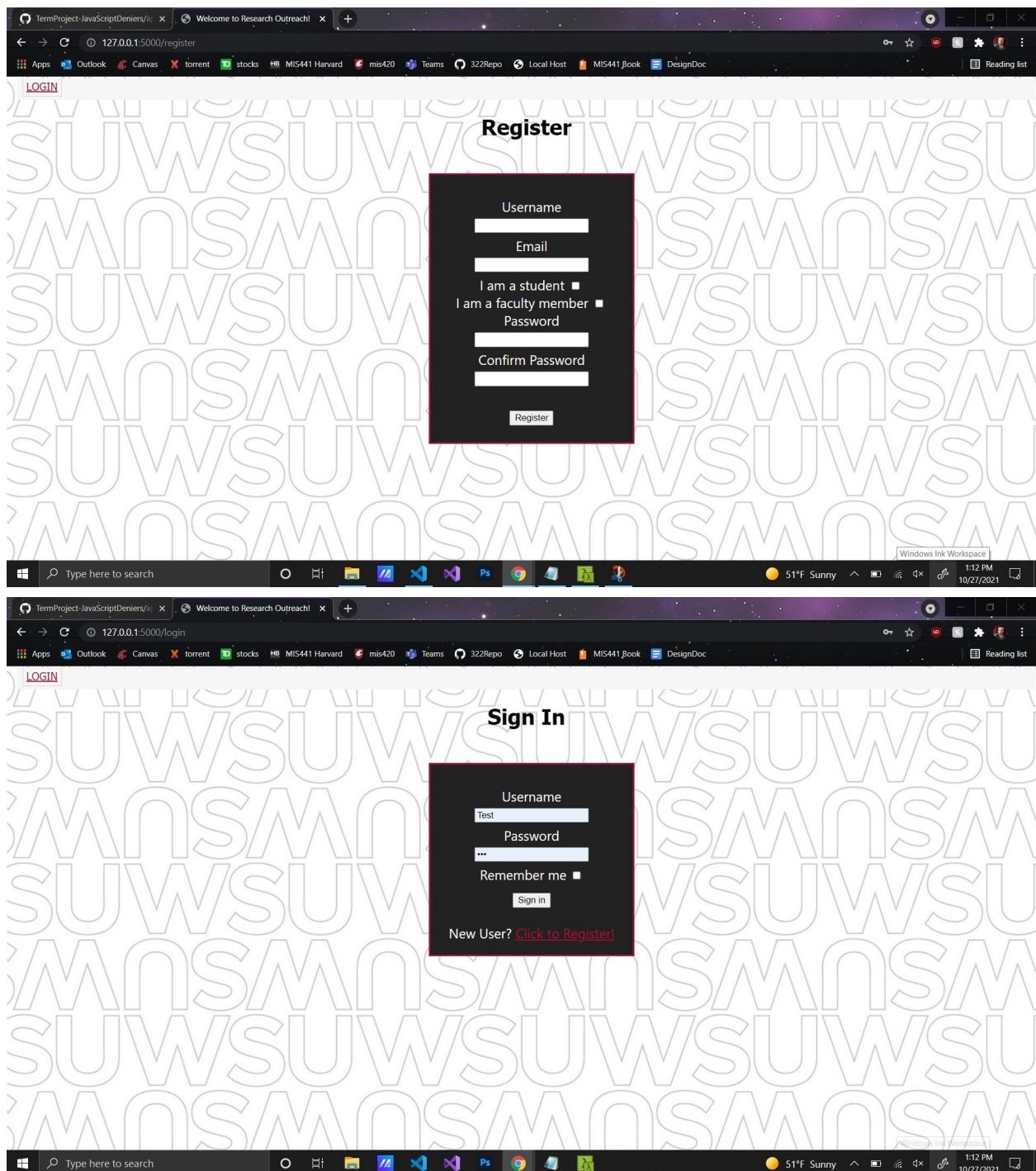




**Post:** The post page contains a simple layout where a faculty member can fill out the post form in order to create a new research position.

## III. Progress Report

In our iteration1 we added a page where the user can create an account where they can choose if they are a student or a faculty. We also added a page where only the faculty can access the page to create a position for the students to see. We also added some CSS styling to the pages. Based on this work we have a framework and foundation to build upon for future iterations including both new aspects of the site, as well as expanding on existing elements and improving them and adding additional functionality.

## IV. References

"Basic MVC Architecture." TutorialsPoint, https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm.