Teaching Assistant Selection App

Requirements Specifications

Git push master --force

Luke Brossman

Garrett Rudisill

Slater Weinstock

Course: CptS 322 - Software Engineering Principles I

Instructor: Sakire Arslan Ay

1. Introduction

Our application aims to provide a necessary service to both instructors and students. Finding teaching assistants can be a difficult process. Instructors could greatly benefit from a direct channel in which to advertise their available positions to the student body. This would greatly reduce the amount of time and effort that a instructor must allocate to filling these necessary positions. Additionally, a very complex user interface currently exists and this application seeks to improve the user experience from both the instructors and the students perspective. By providing a more simplistic UI, we can streamline the teaching assistantship selection process. This not only benefits the future teaching assistants and the professors, but also the students currently enrolled in these courses.

2. Requirements

Our program aims to meet the following requirements:

- Provides reasonable security for the database
- Program works through a synchronized and reasonably fast database
- Allows segmented database access through student and instructor accounts
- Allows instructors to easily acquire teaching assistant's without excess individual communication
- Will have student, faculty, and administration roles with granular access provided based on role
- Backend database will be written in python 3.x using flask with a RESTful architecture to allow administrators to easily extend the database to fit individualized needs and be platform independent
- Frontend will be written in HTML with Bootstrap for easy maintainability and extensibility

2.1 Customers, Users, and Stakeholders

Our stakeholder is Sakire Arslan Ay, whom is contracting out this project, and the school whom may choose to use the application following completion. Our user base is comprised of students who are applying for teaching assistantships and instructors who will select from the pool of student teaching assistants. Our customer is the EECS department at Washington State University.

2.2 Use Cases

Use Case 1:

Name	Database Initialization with Emails
Users	Administrators
Rationale	In order to upload and populate the database with a list of school email accounts for the instructors and the students. This will allow account creation by both students and instructors.
Triggers	Deployment of Backend
Preconditions	Database exists and administrator has up-to-date list of email accounts
Actions	Administrator uploads list of school email accounts
Post Conditions	Database will now be populated with student and instructor email accounts
Acceptance Tests	Verify the database based on given test email inputs, can attempt to make instructor and student accounts to see if successful.

Use Case 2:

Name	Creation of Account
Users	Students and Instructors
Rationale	Both the students and the instructors need to be able to create an account in order to access the TA selection app. This will allow the students to select classes to TA for, and the instructors to select and deselect students.
Triggers	Click “Create Account” link after visiting webpage
Preconditions	Database exists with school email accounts, instructor and/or student have clicked button
Actions	Student and/or instructor will click create an account link
Post Conditions	Student and/or instructor will be brought to a registration page where they will select what kind of user and fill in the details.
Acceptance Tests	Click link, see if brought to registration page.

Use Case 3:

Name	Account Registration (after account creation link)
Users	Students and Instructors
Rationale	In the account registration, both users will have to provide information about themselves, including first and last name, whether or not they are a student, their id number, current contact info, username, and password. We need this info to assign them to right type of user.
Triggers	Click “Create Account” link after visiting webpage
Preconditions	Database exists with school email accounts, instructor and/or student have clicked button
Actions	Student or instructor fill out the form with their current information. If they do not use their WSU email address, an error will be thrown as that email address is not in the database.
Post Conditions	An account will exist for that user, with a username and password. The user will be able to login
Acceptance Tests	Ensure that the user can login with their respective credentials, and view the applicable TA or instructor page.

Use Case 4:

Name	User Does Not Want To Be TA
Users	Student
Rationale	At any point, the student may opt to not be a TA anymore. There needs to be an ability for the student to cancel their TA application.
Triggers	Student clicks a link indicating they no longer want to be TA
Preconditions	That student has made an account successfully, and the student has clicked the link
Actions	The student will be prompted to insert their password as a confirmation, and then they will be deleted from the classes they initially selected to TA for.
Post Conditions	The student will no longer appear in the list for the courses they had previously selected, nor will the instructor be able to see them in their TA selection list.
Acceptance Tests	Make a student account, select various courses, and then click the link indicating they no longer want to be a TA. Verify they no longer appear in the list of courses they were going to TA for.

Use Case 5:

Name	User selects courses to TA for
Users	Students
Rationale	The students need to have an ability to select what classes they want to TA for.
Triggers	After having logged in, they will see a list of courses they can TA for. They will click which courses they want.
Preconditions	The student account exists and the student is logged in. A list of courses that are in need of TAships also exists.
Actions	The student will be brought to a page listing which courses have TAships available. The student will select from the list of courses every course they are interested in. The student will then click a submit button.
Post Conditions	The student will be able to see which courses they have selected to TA for, and the instructor will also be able to see which students have indicated they want to TA for their class.
Acceptance Tests	Have a student account select courses to TA for, and ensure that the student

	and the instructor can see that the student has indicated they want to TA.

Use Case 6:

Name	Approval of TAs for Courses
Users	Instructors
Rationale	The instructors will need to be able to approve various students for their courses.
Triggers	Instructor sees list of students who are interested in TAing for their course, and selects from that list.
Preconditions	Students have indicated that they want to TA for that course, and their name appears in the list
Actions	The instructor will have to select from that last the total number of TAs they need, and then save that list.
Post Conditions	The students will be notified of what course they were selected for, and the students name will be removed from other courses that they wanted to TA for.
Acceptance Tests	Login to an instructor account, attempt to add a student as a TA, and ensure that the student gets notified, and that they student cannot be selected for any other courses.

Use Case 7:

Name	User Removes TA
Users	Instructors
Rationale	The instructor needs to be able to also remove TAs for their courses, in the event they don't need as many TAs, etc.
Triggers	The instructor will deselect the students
Preconditions	The student has already been approved to TA for the instructors course by the instructor
Actions	The instructor will deselect the student and submit that change
Post Conditions	The student will be notified that they were removed from a course, and they can then re add courses that they are interested in TAing for.
Acceptance Tests	Have an instructor remove a student TA, and ensure the instructor no longer sees the student in their list, and the student can add more courses to TA for.

Use Case 8:

Name	Adding Courses
Users	Instructors
Rationale	The instructor needs to be able to add courses into the database
Triggers	The instructor will select a link to add courses, and will fill in the information associated with that course.
Preconditions	The instructor account exists, and is logged in.
Actions	The instructor will be able to add courses that they are instructing into the database. When they select the link and enter the course info and submit, the instructor will have access to that course.
Post Conditions	The instructor will be able to see the course, and the student users will be able to see the new course and be able to apply for it.
Acceptance Tests	Have the instructor add a new course, verify it exists in the database, and that the student users can see the new course added and apply for it.

Use Case 9:

Name	Deletion of Course
Users	Students and Instructors
Rationale	In the rare event that their is not enough enrollment, and the course is removed from that semester, the instructor needs to be able to remove the course
Triggers	Click delete course
Preconditions	The course exists in the database
Actions	The instructor will delete a course by clicking a deletion link. The students will then be informed that a course was removed that they had applied to TA for/or had been selected for.
Post Conditions	The course will be removed from the database, the students that were previously in that course will be notified they can apply for new TAships, the instructor will no longer see the course nor will the students.
Acceptance Tests	Delete a course, ensure students who were registered now can apply for TAships, ensure that the course is removed from the database, and that the instructor no longer has access to the course.

2.3 Non Functional Requirements

- The program will align with the university's color scheme
- The program will operate within the guidelines of FERPA guidelines
- The program will operate within the guidelines of the schools privacy policy
- The program will be completed within given budgetary constraints
- The program will be completed within given time constraints

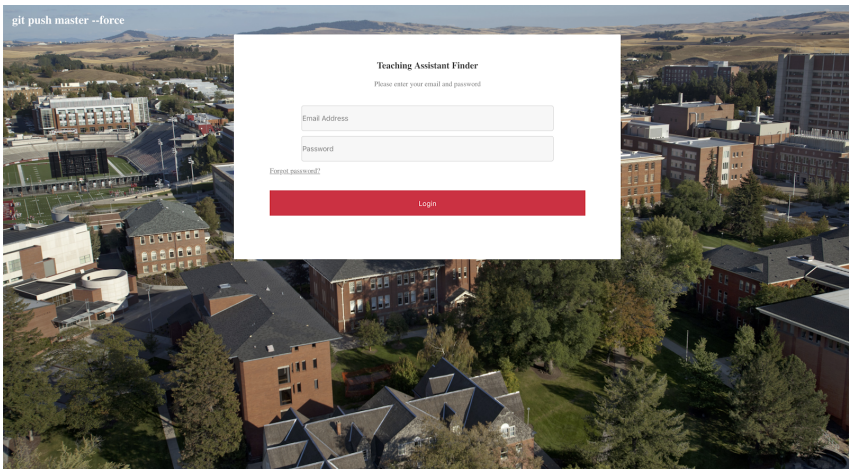
3. User Interface Requirements

- User interface will adhere as much as possible to the universities styling standards, including but not limited to fonts, colors, design language, and etc.
- Initial website page will be a login page with an option to create an account if one does not already exist for the user
    - This will ask for the email and password
    - Contain a redirect to the account creation page if the user must make an account to proceed
- User page will provide an interface that holds all actions available per the users permission level
- Administrator page will contain info on:
    - Registered accounts
    - Account roles
    - Active classes
- Administrator actions will include:
    - Create an account
    - Remove an account

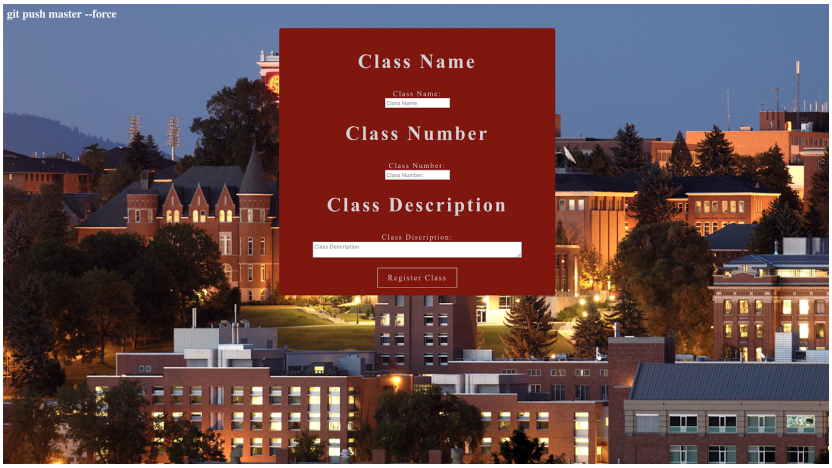
    - Edit an account
    - Create a class
    - Remove a class
    - Edit a class
    - Reset database
    - Logout
- Student page will contain info on:
    - Current applications
    - Approved applications
- Student actions in the UI will include:
    - Apply for a TAship
    - Cancel and application
    - Edit account information
    - Logout
- Instructor page will contain information for:
    - Current classes with status of TAships
    - Applications for a class they teach if one is not assigned
- Instructor actions in the UI will include:
    - Add a class
    - Add a TA position
    - Select a TA (per class)
    - Edit account information
    - Logout

Mockup UI:

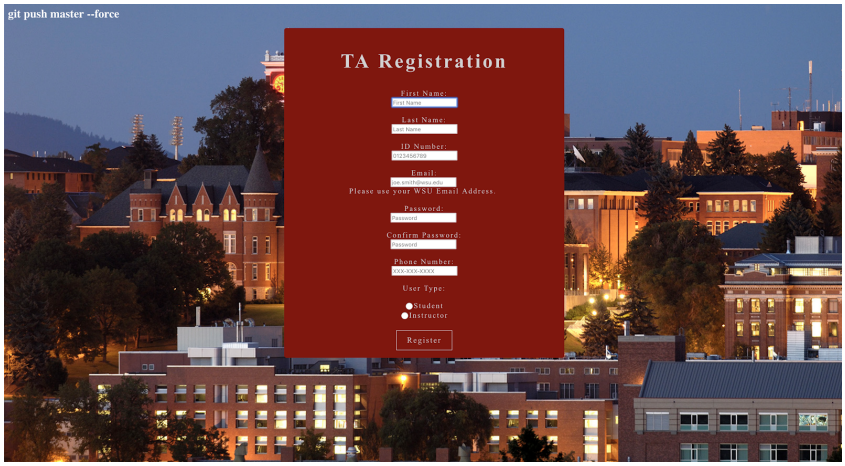
Login Page



Class creation page



Account creation page



4. References

Application Repository - https://gitlab.eecs.wsu.edu/322-fall2018-termproject/TeamgitPush