Teaching Assistant Selection App

Requirements Specifications

GIT push –master force

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

Administrator: A administrator will deploy the backend of the application and initialize the database with a list of school emails. School emails are used as user identification and are combined with a user given password to create credentials for the database. During account creation a user is defined into what role they receive. An administrator can control the database at a granular level, adding classes with teaching assistant positions and adding users.

Preconditions:

- Upon initial deployment there will be a static administrator account. This account can create other administrator, faculty, and student accounts. Only administrator accounts can create other administrator accounts
- Administrators must upload to the database a list of valid university emails for account creation. A user will not be able to create a valid access token without an email from this list.

Postconditions:

- Administrators will have full access to the database.
- Administrators can modify and override faculty and student accounts for assigning position credentials.

Student: A student will create an account on the application. Following account creation, using their WSU email address, they will be able to login to the system. Upon logging in, the student will be prompted to fill out some additional information. The student will be prompted for their current contact information, including their email address, phone number, first and last name, and student ID number. Furthermore, the student will also need to enter information about their current cumulative GPA, major, and expected graduation date, and if they have ever been a teaching assistant for the EECS department before. Upon entering this information, the student will then be prompted to enter what course(s) they are interested in and what their grades were for those respective courses, as well as what semester(s) and year(s) they took the course(s). Within the student's account, they will not be able to see any information associated with other students who have applied for teaching assistant ships.

Precondition:

- Student must have a valid university given email that is contained in the system list

Postconditions:

- The email address that was used for account creation cannot be used to create an additional account.
- They can now view and select from a list of courses requiring teaching assistants.

Instructor: An instructor must create a account with a password to gain access to the database. The instructor may add any courses they are teaching to the database which require a teaching assistant. An instructor may also add additional lab sections and teaching assistant roles which are available to be assigned. The instructor will receive a list of students who have applied to the available position and then can grant or deny the position to a student. This includes the ability to revoke a teaching assistant position for any necessary reason.

Precondition:

- Instructor must have a valid university given email that is contained in the system list
- The instructors class must require a teaching assistant position be available.

Postcondition:

- Instructor can now view a list of applicants
- Instructor can assign positions to an applicant
- Instructor can revoke a position
- Instructor can add a course to the database
- Instructor can add or remove lab sections

Test cases:

Student test cases (given a functional database with 1 class with a position available):

- Can the student see the class position
- Can the student apply for the posting
- If the position is already assigned to another student is the application closed
- If the student is given said position is the application removed
- Once the student has applied, is the application hidden to prevent duplication applications

Instructor test cases (given a functional database with 1 class with a position available):

- Can the instructor see the course that he/she is instructing
- Can the instructor see the students that have selected their course
- Can the instructor approve students who have applied to TA for their course

Administrator test cases

- Can the admin create and delete accounts?
- Can the admin change account permissions?

- Can the admin perform all actions to statuses below administration level

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
- User page will provide an interface that holds all actions available per the users permission level

4. References

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