SW Engineering CSC648/848
Spring 2021

Application Title: Milestone

Section 02 - Team 06

Team Lead: Maria Caravez (mcaravez@mail.sfsu.edu)

Bera Coskun - Front End Lead

Peter Ijeoma - Back End Lead

Jooyoung Kim - GitHub Master

Jeffrey Ye - Database Manager

Milestone 4

History Table

Submitted	**Revised**
4/20/2021	
Revision Summary	

1. Product Summary

Our talent pipeline application Milestone, is designed for both the passive and active users. We understand the complexities of the college student and are eager to provide a platform that is proactively matching students to employers. Our application's design is simple, which allows quick access and easy flow of navigation. We understand that equal opportunity is important in the workforce, which is why we have allowed users to share their demographic information to satisfy diversity in all industries. We're committed to establishing a product that will encourage users to showcase their various and unique skills.

Priority 1

1. First-time visitors shall be onboarded.
2. Guests (students, and headhunters) who do not want to sign up yet, shall be able to search.
3. Search capability for new graduates or students by graduation date, academic standing, major, demographics, and military code.
4. Users shall be able to login.
5. Students shall be able to create, view, and edit profiles.
6. Students shall be able to search for open positions of interest to them.
7. Users shall register to use the system.
8. Headhunters shall create job postings.

URL: http://ec2-18-188-8-216.us-east-2.compute.amazonaws.com/

2. Usability Test Plan

Test Objectives

Our team plans to perform a usability test on our student search functionality. With our student search functionality, we strive to create a platform where it is simple for headhunters to find an adequate match for their open positions. We understand headhunting isn’t easy, and can get time consuming, which is why we implemented an intuitive search functionality that’ll help find the suitable candidate. Our intent is to provide a service that is simple. By using our service, the headhunter is able to search broadly for students with ease, but is also able to refine the results afterwards to find the perfect candidates. For this reason, we aim for our users to reach satisfaction through the specified goals by providing them the essential resources and the functionality they need.

Test Background & Setup

System Setup

Our usability testing will be moderated remotely with screen-sharing via Zoom. The meeting will be recorded to analyze further after the test. During the meeting, the participant will be asked to share their video feed. A team member will assist and annotate any expressions that may be critical in pointing out confusion.This will allow us to read the participants body language for any confusion. A participant will be asked to use the specified URL, perform the task described, and is open to ask questions if needed. A team member will be ready to assist with any clarification needed.

Starting Point

Users will follow the URL to our landing page and perform the student search functionality from there.

URL: http://ec2-18-188-8-216.us-east-2.compute.amazonaws.com/

Intended Users

Users will be acting as potential guest headhunters. This means our platform is open to any employer even if they do not want to register for an account, however, in order to contact any potential candidates, the headhunter will be prompted to register -- we will not be testing this functionality portion.

What Is To Be Measured: User Satisfaction Evaluation

The user’s ability to complete the tasks and the ease of doing so will be measured. Our evaluation is being conducted through user participation via summative method.

Usability Task Description

1. Use the Search Students input field to search for students.
2. Search all students on the website.
3. Filter students by organizations/groups option.
4. Filter for a student with a specific graduation date, month, or year.
5. Search for a student by any attribute in the entire website: name, skill, work experience.
6. Filter for students by a different organizations/groups option

Measuring Effectiveness

We will measure the effectiveness through error rate and completion percentage. This area will be noted during the participation of the usability test. Questions in our Licker Questionnaire will help us further analyze the effectiveness by understanding if the participant struggled with any step of the process.

Measuring Efficiency

We will measure efficiency through resources used (time). Our recording will allow us to see how long the participant took to get from one step to another. We will be able to point out any hesitation that may have resulted in additional time taken.

Likert Subjective Test

The Likert Subjective will use a scale consisting of “Strongly Agree”, “Agree”, “Neither Agree nor Disagree”, “Disagree”, and “Strongly Disagree” for the following questions:

1. Overall, the website was easy to navigate.
2. The searching functionality was intuitive.
3. Finding a student was simple.
4. The search results were easy to read.

3. QA Test Plan

Test Objectives

Our quality assurance test will test the feature of student search. With specified tasks and results, a participant will identify whether the results match our expected output. This quality assurance test will also help us determine any bugs that we may currently have that are unaware of.

HW & SW Setup

Our QA testing will be moderated remotely with screen-sharing via Zoom. The meeting will be recorded to analyze further after the test. Our participant will be asked to follow the tasks in our table below. After completion of each task, the participant will annotate whether the expected output was true and rate with a pass or fail.

URL: http://ec2-18-188-8-216.us-east-2.compute.amazonaws.com/

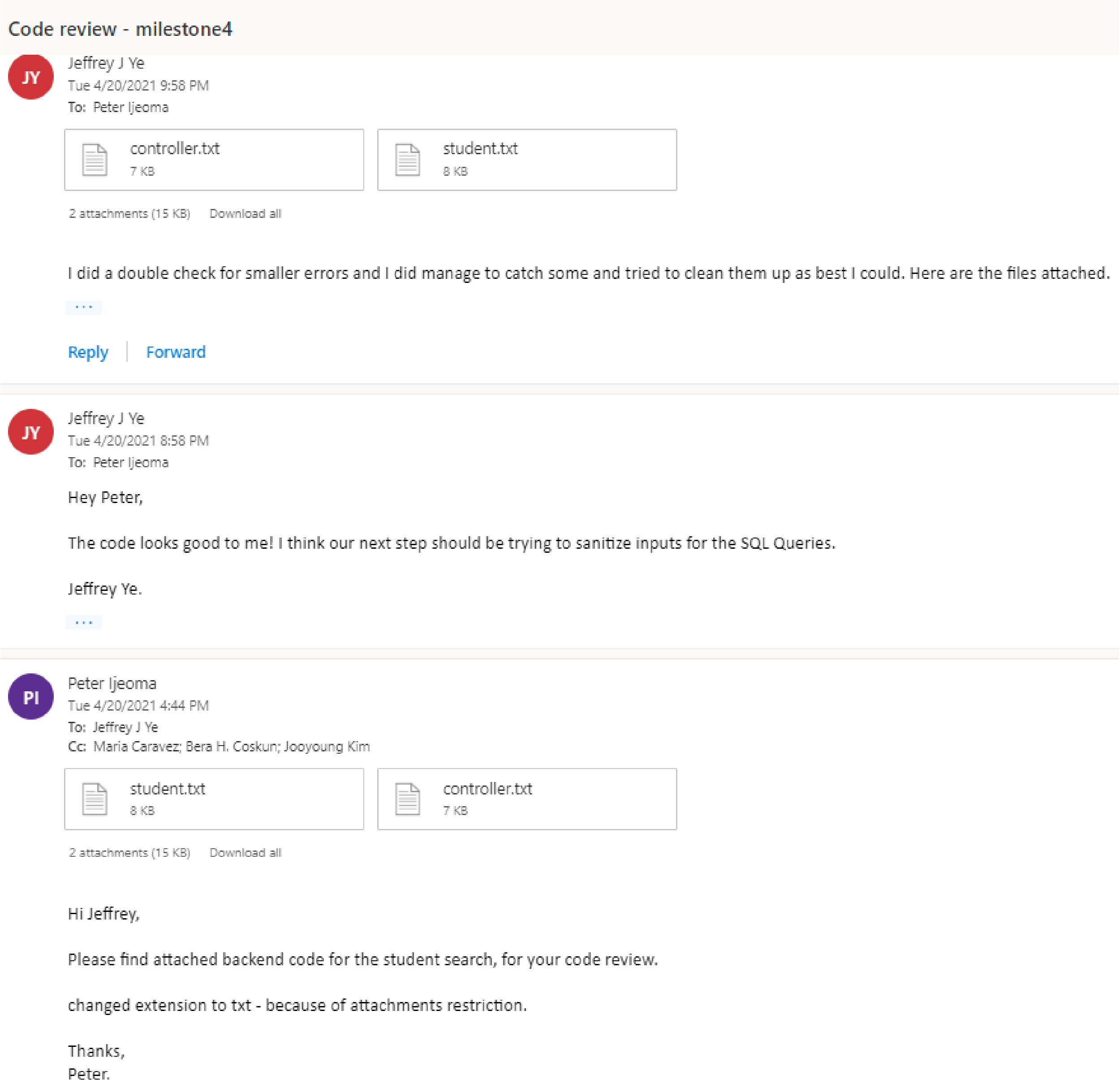
Feature to be Tested

Searching and filtering of students will be tested.

QA Test Plan Table

Test No.	Title	Description	Input	Expected Output	Test Results
Number	Title	Description	Test Input	Expected Output	Pass/Fail
1	Searching Specific Name	Input % in the search field.	“ Bera”	Get 1 results, prioritized by name containing “Bera”.	
2	Narrowing Search	Input % in the search field and add % as a skill.	“Maria”, “Javascript”	Get 3 results, containing “Maria” in any field and “Javascript” as a skill.	
3	Narrowing by Organization	Input % in the search field and add % as a Groups/Organizations.	“”, “AWIS”	Get 3 results, containing “AWIS” or “Association for Women in Science” as an organization.	
4	Narrowing by School Year	Input % in the search field and select % as a School Year.	“MEAN Stack”, “Senior”	Get 4 results, containing “MEAN Stack” or “MEAN” in any field and “Senior” as the school year.	

4. Code Review



Code review - milestone4

Jeffrey J Ye
Tue 4/20/2021 9:58 PM
To: Peter Ijeoma

I did a double check for smaller errors and I did manage to catch some and tried to clean them up as best I could. Here are the files attached.

Jeffrey J Ye
Tue 4/20/2021 8:58 PM
To: Peter Ijeoma

Hey Peter,

The code looks good to me! I think our next step should be trying to sanitize inputs for the SQL Queries.

Jeffrey Ye.

Peter Ijeoma
Tue 4/20/2021 4:44 PM
To: Jeffrey J Ye
Cc: Maria Caravez; Bera H. Coskun; Jooyoung Kim

Hi Jeffrey,

Please find attached backend code for the student search, for your code review.

changed extension to txt - because of attachments restriction.

Thanks,
Peter.

5. Best Practices for Security

1. Password

- Passwords are encrypted before being stored in the database.
- Check for password strength before the user account is created and a set maximum number of characters checked in the backend.
- Login is a POST request, avoiding a persisting URL, passing in the password through the body.

2. Email

- Emails are encrypted before being stored in the database. Users cannot access emails of other users even if logged in.

3. Database Access

- We’ve prevented login as root into MySQL database from a client. Only users we’ve created with proper permissions can do so. The backend works as the middleman between the client and database.

4. Search Box Validation

- All inputs - sanitization to prevent SQL injection attacks before database query and a set maximum number of characters checked in the backend. In the front end, data is also validated as a user experience measure.
    i. “Everything” Search Box
    ii. Major Input
    iii. Skills Input
    iv. School Year Dropdown
    v. Groups Input

5. Adherence to Non-functional Specifications

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO). DONE
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers - ON TRACK
3. Selected application functions must render well on mobile devices - ON TRACK
4. Data shall be stored in the team's chosen database technology on the team's deployment server. - DONE
5. No more than 100 concurrent users shall be accessing the application at any time - ON TRACK
6. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users. - ON TRACK
7. The language used shall be English. - DONE
8. Application shall be very easy to use and intuitive. - DONE
9. Google maps and analytics shall be added - ON TRACK
10. No e-mail clients shall be allowed. You shall use webmail. - ON TRACK
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. - ON TRACK
12. Site security: basic best practices shall be applied (as covered in the class) - ON TRACK
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development. - DONE
14. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2021. For Demonstration Only" at the top of the WWW page. (Important so not to confuse this with a real application). - ON TRACK