
CptS 322 - Fall 2021

Term Project

I. Project Description

Though many students falsely assume that only professors or graduate students are involved in research, there are many research opportunities at WSU EECS for undergraduate students. Usually, faculty present the research opportunities in their classes and students who are interested in the project directly contact the faculty. However, if the faculty is not teaching a lower-level course, they can't easily reach out to sophomore and junior cohorts. There is a need for an online platform that will allow the faculty advertise their research positions and connect them with the qualified undergraduate students.

You will build a web application where:

- students can enter their contact information, completed coursework, research interests, and prior research experience,
- faculty can advertise research opportunities for undergraduate students,
- students can apply for research positions,
- faculty can select the candidates that they would like to interview with for the position.

The application should have a "student page" and a "faculty page".

On the student page, a student user can:

1. Create a student account and enter the profile information:
 - a. set the account username and password (username should be the WSU email)
 - b. Enter contact information (name, last name, WSU ID, email, phone)
 - c. Enter additional information (major, cumulative GPA, expected graduation date, etc.)
 - d. Enter the technical elective courses they completed and the grades they received.
 - e. Select the research topics they are interested in.
 - f. Select the programming languages they have experience with.
 - g. Describe their prior research experience if there is any.
2. Login with username and password.
3. View the open research positions.
 - Your app should list all open research positions.
 - In addition, it should identify the research positions that match the student's "research interests" and list them separately under the "Recommended Research Positions". You can implement a simple recommendation algorithm to find the matching positions. For example: *If the student's research interests include "Machine Learning", the positions in that field should be recommended to the student.*

4. For each research position, various information should be displayed:
 - a. Research project title,
 - b. A brief description of the project goals and objectives
 - c. Start date and end date
 - d. Required time commitment (e.g., 10 hours per week)
 - e. Research field(s) (e.g., “Machine Learning, High Performance Computing, etc.)
 - f. A brief description of the required qualifications
 - g. Faculty’s name and contact information
5. Apply for research positions. A student can apply for more than one research position. For each position they apply to,
 - They should submit a brief statement describing why they are interested in this research topic and what they hope to gain by participating in that project.
 - They should provide the name and email of one faculty who can provide them a reference for the position.
6. View the research positions they already applied to and check the statuses of their applications.
 - a. When the application is submitted, its status will appear as “*Pending*”.
 - b. After a faculty accept this application, the status should be updated as “*Approved for Interview*”. Student should reach out to faculty to schedule an appointment for the interview.
 - c. After the interview, the faculty should update the status as either “*Hired*” or “*Not hired*”. Once updated, the changed status should be displayed on the student page.
7. Withdraw their pending applications.
 - If the student is no longer interested in a research position, they can withdraw their application.

On the faculty page, a faculty user can:

1. Create a faculty account and enter profile information:
 - Set the account username and password (user name should be the WSU email)
 - Enter contact information (name, last name, WSU ID, email, phone)
2. Login with username and password
3. Create undergraduate research positions. Faculty should enter the details of the position and qualifications needed, i.e.,:
 - a. Research project title,
 - b. A brief description of the project goals and objectives
 - c. Start date and end date

- d. Required time commitment (e.g., 10 hours per week)
- e. Research field(s) (e.g., “Machine Learning, High Performance Computing, etc.)
- f. A brief description of the required qualifications.

Of course, a faculty can create more than one position.

4. See the list of the students who applied for their positions.
 - If a student was “*approved for interview*” or was “*hired*” for another position, those information should also be displayed. A faculty should be aware of the other offers students get.
5. View the qualifications of each student, i.e.,
 - a. their GPAs,
 - b. the technical elective courses they have taken,
 - c. the research topics they are interested in,
 - d. the programming languages they have experience with, and
 - e. prior research experience.
6. The faculty can approve the application of one or more students and the status of those applications should be updated as “*Approved for Interview*”.
7. After interviewing with the student, the faculty can update the status of applications as either “*Hired*” or “*Not hired*”
8. The faculty may delete the existing research positions. Once deleted, all applications to that position should be updated as “Position is not available”.

II. Project Timeline:

i. Requirements document: (6%)

- Once we form the teams, you will have to start right away getting the team working together and collecting the requirements. In this stage you only need to elaborate the details of the requirements for the complete project.
- You must start using GitHub Issues (Issues , Labels and Milestones) to manage your plan.
- Since there are only couple weeks after we form the teams until the full Requirements and Specification Doc is due, it is important for the team to start meeting right away, because it takes time to figure out what the team will be building.
- The requirements document should be pushed to GitHub repo before the deadline.

ii. Iterations: The main part of the project will be done in several two-week iterations. In each iteration you will have to work on several tasks:

– Iteration 1: (8%)

– First draft of your Design Document (5%)

- Write a short progress report at the end of your design document where you briefly summarize the project progress.

- A completion of a running version of your code with a partial set of features.
- The design document and the code should be pushed to GitHub repo before the iteration deadline. (3%)
- **Iteration 2: (8%)**
 - Update your Design Document to reflect your current accomplishments and your test plans for iteration 2 (3%)
 - Write a short progress report at the end of your design document where you briefly summarize the project progress and tests performed.
 - A completion of a running version of your code with additional set of features. (5%)
 - The revised design document and all code should be pushed to GitHub repo before the iteration deadline.
- **Iteration 3: (15%)**
 - A final version of your code. All code should be uploaded to GitHub repo before the iteration deadline. (4%)
 - At the end of the semester each team will demo their project to the faculty and the course TA. (8%)
 - At the end of the semester each team will make a 3 min video demo of their application. (0.5%)
 - Testing support and tests for the implemented features. This iteration should include unit, integration, and functional tests. You should also include some automated unit tests. All automated test code should be uploaded to GitHub repo before the iteration deadline. (2.5%)
- **Project management: (3%)**
 - Your “project management” score will be based on:
 - How effectively you used **GitHub**? (We will use GitHub issues to maintain the project backlog and milestones for sprint planning)
 - Did all members have frequent commits throughout the semester?
 - Has the team members communicated effectively using **Microsoft Teams**? (We will use Microsoft Teams for communication among team members and document sharing.)

III. Project Deliverables:

1. Requirements Specifications:

There is no standard for requirements or specifications documents and in fact many organizations blend aspects of requirements, and specification in a single document. We will use a simple template for requirements and specification and document requirements as UML use cases. Inevitably in preparing this document you will describe of how the system should interact with the outside world.

Please see the “Requirements Specifications” document template that will be posted on Canvas.

2. Design Document:

This will be a living document. For the first iteration you will fill in the document with the design details as you can see them before the first iteration. In subsequent iterations you will expand this document.

A template for the “Design Document” will be available on Canvas.

3. Progress Report:

For iterations 1 and 2 you will write a short progress report covering the topics specified below. This report will be included at the end of your design document.

- **Main Difficulties:** What were the main difficulties so far? You should consider both technical and organization issues.
- **Features:** What features were completed? Were there any features you did not implement as planned? Are you pushing some features to later iterations, and if so, why?
- **Tests:** What tests did you prepare for this iteration, and what are they covering? What features are you not testing yet? Did you use any test frameworks?

4. Your Code and Tests:

In each iteration you need to commit a running version of your code to the iteration branch (i.e., iteration1, iteration2, or iteration3) in your repo. All code should be uploaded to GitHub before the iteration deadline. There should be a link to your GitHub iteration branch in your progress report. In iteration3, you should also commit your test code.

5. Presentation:

At the end of the semester, each team will make a short video and we will make these videos available to the whole class. The goal is to allow you learn from the experience of all groups. The videos will be graded based on the course staff evaluation of how well prepared and useful the presentation is, along with the peer feedback submissions.

The presentation should be about **3 minutes** long and should have 2 parts:

- An **overview of your project**, perhaps with a very short demo of the current state.
- Some of the **lessons you have learned**. Please do not be afraid to say what went wrong; being frank and drawing educational conclusions from your experience is what we are looking for.

IV. Submission Instructions:

All project documents including “Requirements Specifications” and “Design Document” (both versions) should be committed to your team’s GitHub repo.

All code and test case implementations should also be committed to your GitHub repo.

Your GitHub repo should have the following branch structure :

- main:
 - Final version of your code (after iteration 3);
 - (under “tests” directory) all your test code;
 - (under documents directory) Requirements Specifications document; final version of your Design Document.
- iterationX:
 - All the code you produced in iteration X;
 - Iteration1 and Iteration 2 only - (under documents directory) Version X of your Design Document;
 - Iteration3 only - (under “tests” directory) your test code for iteration 3.

Reading:

<https://about.gitlab.com/blog/2018/03/05/gitlab-for-agile-software-development/>
<https://robinpowered.com/blog/best-practice-system-for-organizing-and-tagging-github-issues/>
<https://about.gitlab.com/devops-tools/jira-vs-gitlab.html>

Appendix:

Iteration-1 use-cases

Iteration-2 use-cases

Iteration-3 use-cases

TBA