Research Position Search App

Requirements Specifications

CTRL_Z

Jared Lustig, Musa Husseini,Michael Stawowski

Course: CptS 322 - Software Engineering Principles I

Instructor: Sakire Arslan Ay

TABLE OF CONTENTS

Document Revision History

Rev x <10/11/2021> <comment> (Rev 1.0 2021-10-13 Initial version)

I. Introduction

I.1. Document Purpose

The purpose of this document is to develop an application that will allow students and faculty to be better informed of potential research opportunities at Washington State University. Furthermore, our team will discuss the different requirements of the application, as well as an overall description of how the app will look and perform. This document will also go over who the intended customer, user, and stakeholders are, and how they stand to benefit from this application. Our audience is directed towards faculty members who are looking for a better way to advertise their research positions, and open up opportunities to qualified students who are not enrolled in their courses. Included will be several use cases to depict how users will be able to use the application, as well as a list of features that could make its way into the final product. Finally sketch ups of what the interface will be added towards the end.

I.2. Product Scope

The product will consist of two main parts, a student view and a faculty view. The product being provided for the student is a platform where they can look for research opportunities and apply for positions. The faculty platform will provide a software where they can make job postings and find qualified candidates.

The student view will allow students to create an account that includes information about their skills, relevant coursework, and experience. Once completed, students will have the ability to search for open research positions and apply to any they are interested in. Submitting an application will direct them to another page that will show their application status. The primary goal for the student view is to provide an efficient and easy way for them to find research opportunities.

The faculty view will also have an account page where they can input basic contact information. Once the account is set up they will be able to create posts for positions they would like filled. The goal of this page is to allow them to post an advertisement for a position that includes all relevant information for anyone interested. For each post made, the faculty member who created it will be able view students who have submitted an application. The faculty member will be able to see student qualifications that they input into their account, (i.e GPA, relevant courses, skills, etc.).
After reading a student application the faculty member can then decide to move on with their application status to “Approved for interview” or “Not Interested”. Furthermore, after an interview the faculty can change the status to “Hired” or “Not Hired”. Lastly a faculty member should be able to delete any posts they have created. The main goal for the faculty page is to create an efficient way for faculty members to post research opportunities and find qualified students to fill the role.

I.3. Document Overview

The second section of our document will provide definitions for our requirements specifications; this section will be broken down into three parts. The first part will be describing who our customers, users and stakeholders are. It will go in depth of how each potential customer, user, or stakeholder will use our software. The next part will include use cases for the project and in this section the use cases will be defined and then followed by a use case table. This part will also include a swim-lane diagram to illustrate the message flow and activities. The third part of this section will cover non-functional requirements. This will include descriptions of constraints on certain requirements such as response time, scalability, security, robustness. This part will also ensure all the non-functional requirements are verifiable.

The third section of our document will provide visual representations of our user interface. It will include our vision of what each page should look like through sketches, flowcharts, and UML tables.

The fourth Section is our reference page where our citations will be posted.

II. Requirements Specification

II.1. Customer, Users, and Stakeholders

The app our team will be developing will encompass that of students and faculty members of Washington State University. As of now the problem is there is not an easy way for faculty to post job opportunities or research or for students to find these opportunities. Usually faculty will advertise in their own classes about research opportunities or a school counselor will inform the students. Our goal with our system design is to provide an easy to use interface that allows our user to find what they are looking for. Students interested in a field of research, will now be able to contact different faculty members in their particular field, and apply for a research position from a faculty member's posting. Faculty members will have the ability to create postings for qualified undergraduate students, as well as select candidates for whom they would like to interview for the position. Potential customers of this application would include Washington State University and/or other Universities interested in the opportunity of filling research positions for students and faculty. Stakeholders that can benefit from this application would be companies and / or faculty members looking to have more research positions filled by qualified individuals. Other potential Users of the application could include individuals in pursuing research, as well as other researchers interested in finding candidates who are qualified.

Some basic functional requirements our system should accept as inputs are entering account information to make an account, logging into the account, creating a job post, and a search for relevant posts. For creating an account the system should prompt the user to enter in text fields and make a submission request. Logging into the account will consist of the user being prompted to enter in their login information and a request will be made to the database to post their information. Creating a job post will call upon our database to save the post and update the posts page with the post. The user will be able to sort the posts by relevance to what they are looking for. Outputs we should expect from our system will be displaying job postings, account information, and application statuses. The displaying job posts will be computed by calling upon our database to access all posted jobs and display them on the given page. To access account information the request to the database must verify that it will only return information to the signed in user, this is a major security task we must verify. Application statuses will be displayed by the database accessing the selected posts information and the related applications to it. All of these inputs and outputs should be computed by calling upon the database to access previously filled information.

II.2. Use Cases

The actors involved when using our software will be our regular users, faculty and students, and our administrators.
List of Use Cases:

Both Student and Faculty:

- Create Account
- Login

Student:

- View research positions
- Apply to research positions
- Withdraw applications
- Advanced Search

Faculty:

- Create research positions
- View list of applicants
- Update application status

Name	Create Account
Users	Faculty and Students
Rationale	Each user needs a unique account for themselves and depending on the account this determines whether the student or faculty view is displayed. These accounts allow users to save personal information and view applications/applicants depending on if the user is a student or faculty member.
Triggers	User selects the button labeled “Create Account”.
Preconditions	The webpage detects no user is signed in
Actions	1. User selects create account 2. The software redirects them to a registration page where they fill in information relevant to them 3. Once the information is filled in they select create account and are registered
Alternative paths	The user may realize they already have an account with our system, in this case they may return to the main page and be prompted to enter their existing information to login.
Postconditions	The user is registered within our database and can login.
Acceptance Tests	Making sure that the user receives a verification email and verifies the password is correct during the account creation process.
Iteration	**1**

Use case # 2

Name	Login
Users	Student and Faculty
Rationale	The users should be able to access their accounts to apply to more jobs, check on applications, or post more job opportunities
Triggers	User completes the login form.
Preconditions	There should be no users already logged in.
Actions	1. The user enters their username 2. The user enters their password 3. The user selects the login button
Alternative paths	If the user does not have an account they can select a button that will direct them to the create account page.
Postconditions	The user is able to view the given page depending on what kind of user they are.
Acceptance Tests	Ensure all login in fields are filled and that username and password match
Iteration	**1**

Use case # 3

Name	View research positions
Users	Student
Rationale	The user should be able to view all job listings so they can pick and choose what they want to apply to.
Triggers	Right after the user logs in this will be the first page they see.
Preconditions	There must be a logged in user.
Actions	After the user is logged in they can scroll and click on what interests them.
Alternative paths	The User could click the button labeled “View Positions”.
Postconditions	The page displays all jobs
Acceptance Tests	After the student is logged in they should be able to see all job listings and select what they want. After the faculty is logged in they should make a post and see if it appears on the display page.
Iteration	**2**

Use case # 4

Name	Apply to research positions
Users	Student
Rationale	If the student finds a job that interests them, they should be able to send in their application for the given job.

Triggers	When the user selects the apply button.
Preconditions	The user must be logged in and have their personal information filled in.
Actions	1. User selects apply button 2. User fills in information fields 3. User selects submit button to send their application
Alternative paths	The user can find the faculty members information and personally send them an email to apply for the position.
Postconditions	The user is informed the application has been submitted
Acceptance Tests	The user can view a page that lists all their current applications The user will be given a flash message saying something along the lines of "Submission sent".
Iteration	**1**

Use case # 5

Name	Withdraw Applications
Users	Student
Rationale	The user should be able to withdraw an application if they change their mind.
Triggers	User selects the withdrawal button.
Preconditions	The user must already have a pending application.
Actions	1. The user selects the withdraw button. 2. They are prompted with a verification to withdraw. 3. User confirms the verification and withdraws the application.
Alternative paths	N/A
Postconditions	The user will no longer see the pending application for the one they withdrew from.
Acceptance Tests	The user can no longer see the pending application
Iteration	**2**

Use case # 6

Name	Search For Relevance
Users	Student
Rationale	The user should be able to do filter searches in order to find jobs that would match their interests faster.
Triggers	User types or selects advanced search features
Preconditions	The user must be logged in and there must be jobs posted
Actions	1. User typed in advanced search keys. 2. The page sorts the post by most relevant depending on the advanced search.
Alternative paths	The user can just scroll through the page looking for their specific keywords

Postconditions	The page is filtered by relevance on what the user searched by
Acceptance Tests	Type in a key word that will match a post and see if the post filters to the top
Iteration	**3**

Use case # 7

Name	Create research positions
Users	Faculty
Rationale	The faculty should be able to make job postings so students can view them.
Triggers	User selects the create position button.
Preconditions	The user logged in should be a faculty member
Actions	1. User selects create job button 2. The user inputs a title and description. 3. User posts the job
Alternative paths	The user could potentially edit an existing research post then resubmit it
Postconditions	The user should be able to see their job on the jobs listing page
Acceptance Tests	The user should be able to see their specific job on the listings page
Iteration	**1**

Use case # 8

Name	View list of applicants
Users	Faculty
Rationale	The user can view students who have applied for their job and view their application.
Triggers	Users select their job posting and view the applicants.
Preconditions	A job must already be posted
Actions	1. The User can view any job they have posted. 2. They select the button labeled “view applicants”. 3. The user is provided with a list of applicants.
Alternative paths	N/A
Postconditions	The user can view each application which includes basic student information.
Acceptance Tests	Make sure when a student applies the faculty side is updated with the student's application.
Iteration	**2**

Use case # 9

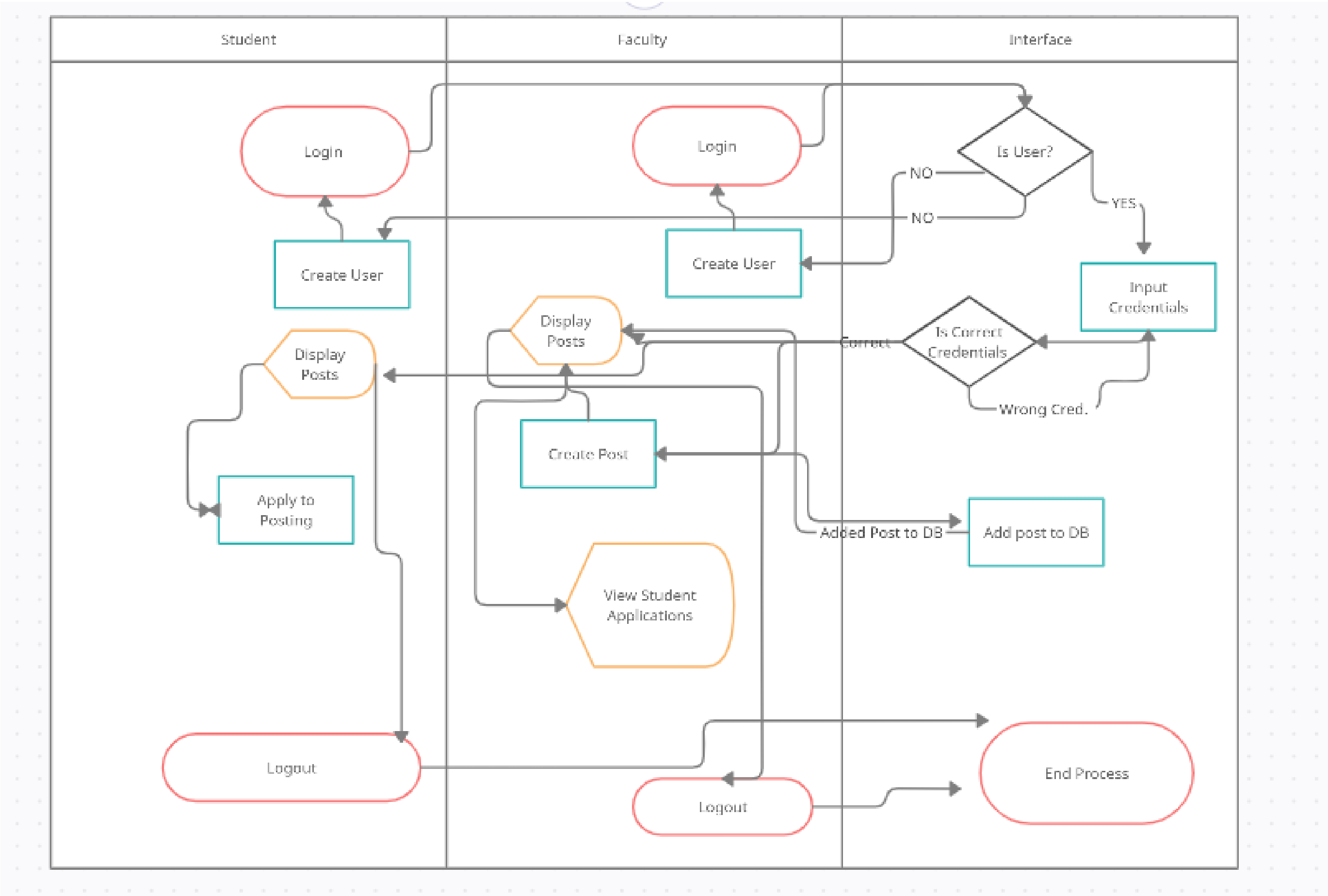
Name	View applicants information
Users	Faculty

Rationale	The user can view an individual student applicant and see in detail their information to see if they are fit for an interview.
Triggers	Users select their job posting and view the applicants.
Preconditions	A job must already be posted and the user must view the list of applicants page.
Actions	1. The user selects a specific student 2. The user reads the given information about the student
Alternative paths	N/A
Postconditions	The user can view each application and determine if they would be a good fit for the position.
Acceptance Tests	When the user selects a specific student they are able to view the right information and nothing private.
Iteration	**2**

Use case # 10

Name	Update Application Status
Users	Faculty
Rationale	The user can determine if the applicant is fit for the job and either accepted for an interview, hired, or rejected.
Triggers	The faculty member decides to make a decision on an application.
Preconditions	There must be existing applications.
Actions	1. The User selects an applicant. 2. The User then selects if the applicant is accepted for an interview,hired or rejected 3. The application status is then updated.
Alternative paths	N/A
Postconditions	The application status becomes updated to what the user chose.
Acceptance Tests	When the status field is changed check the student page and see if it updated their application status on their end also.
Iteration	**2**

Swim Lane Diagram:



II.3. Non-Functional Requirements

1. Reliability : Our system should be able to handle many requests at once.

2. Availability : Our system will be available 24/7

3. Scalability : System will adapt to a variety of different screen sizes.

4. Multiple Sign in Methods (Security) : Users will be able to sign in through google's authentication, or they may sign in through the web application if they do not want to use google.

5. Recovery from failure : The system should redirect the user to the main page in the case of a rendered template failure.

6. Response Time: The system will respond to triggers in around 100ms so the user experience isn't hindered.

7. Throughput: The system will be able to process 50gb of data over about a one minute period

8. Robustness: The system will be able to cope with unexpected errors this means that the system will have limited downtime

9. Security: The system will implement basic web security standards that will prevent an exploit like a cross site scripting attack.

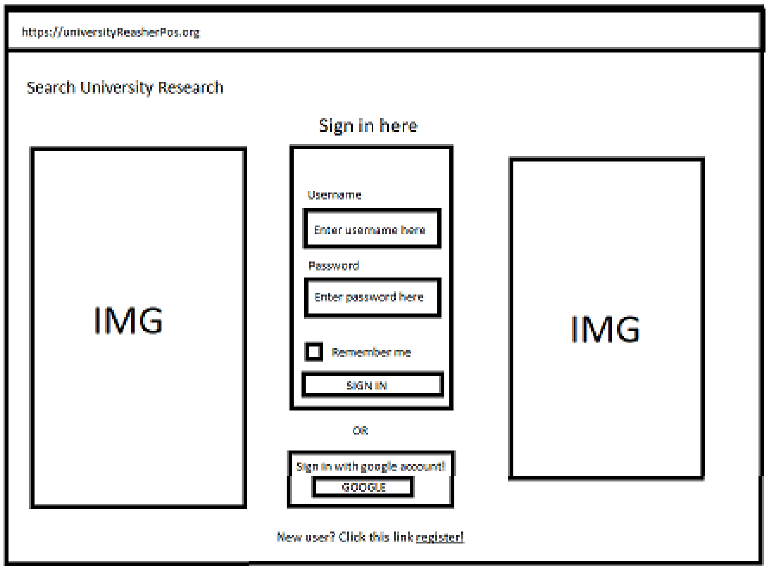
10. Allowances for reusability: We will make sure to approach the project with a structured and modular programming approach.

11. Allowance for maintainability and enhancement: When designing the software we will have maintainability in mind so any potential issues or changes made won't cause substantial downtime for the user.

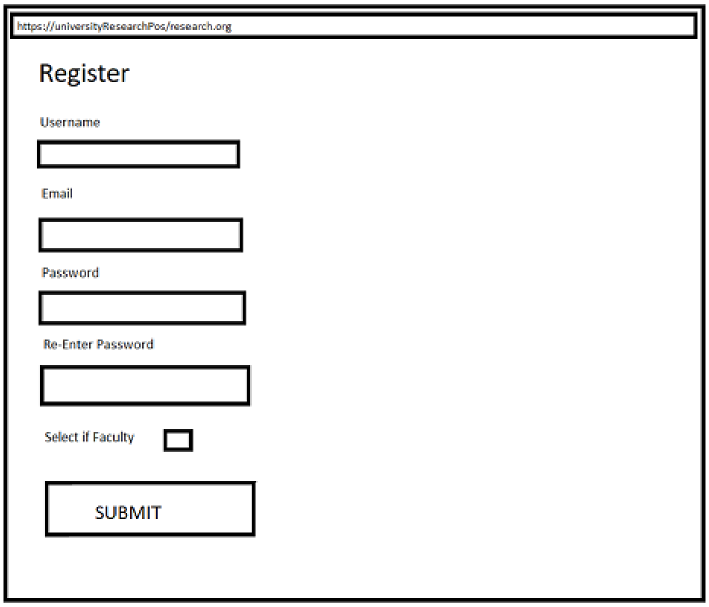
III. User Interface

Here you should have sketches or mockups for the main parts of the interface.

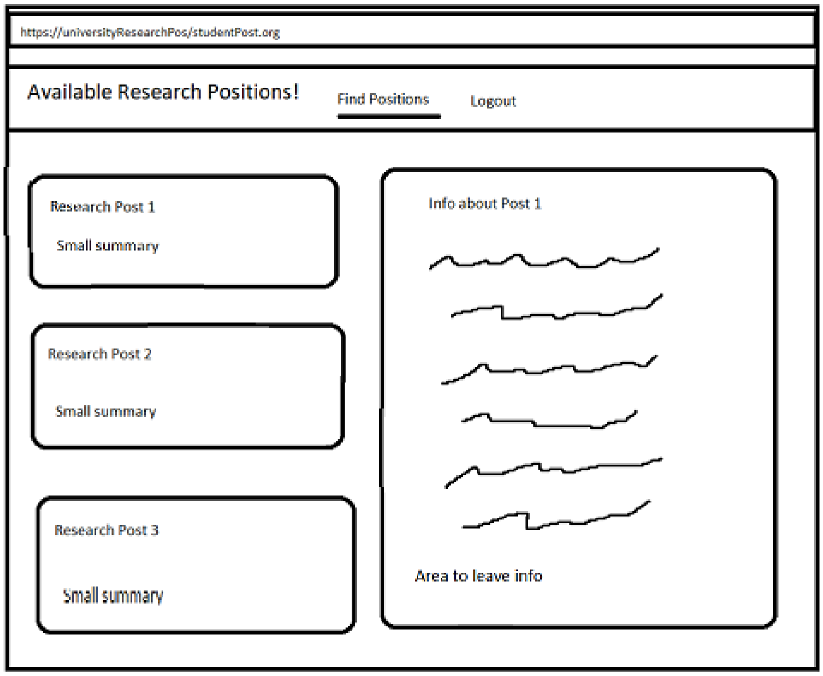
Login Screen:


https://universityReasherPos.org
Search University Research
Sign in here
IMG
Username
Enter username here
Password
Enter password here
□ Remember me
SIGN IN
OR
Sign in with google account!
GOOGLE
New user? Click this link register!
IMG

Register

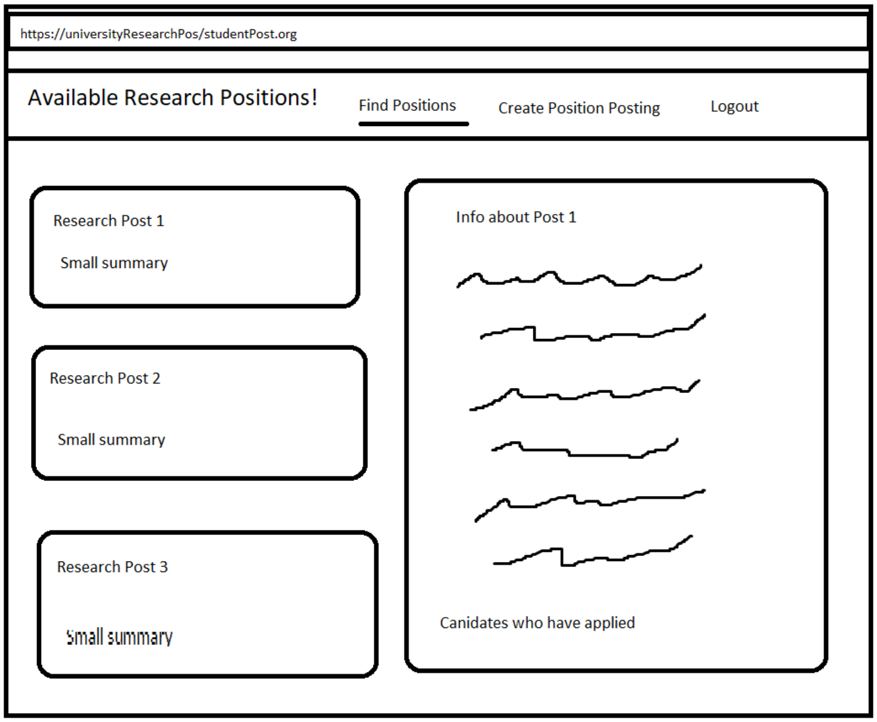


Student Side:



1. Indeed-position-inspiration

Faculty Side:



1. indeed-position-inspiration

IV. References

1. “Indeed-Position-Inspiration.” *HCaptcha Solve Page*, https://www.indeed.com/jobs?q=software+engineer+internship&l=Pullman%2C+WA&vjk=4fc4c8d4a1b3e252.
2. Ay, Sakire Arslan. “Project-Requirements-Doc.” *Requirements Position Search App - Project Description*, WSU, https://wsu.instructure.com/courses/1486805/files/81626945?wrap=1.
3. “Uses Cases” Usability, https://www.usability.gov/how-to-and-tools/methods/use-cases.html
4. “Swim Lane Diagram” Smartdraw, https://www.smartdraw.com/swim-lane-diagram/
5. “What are Non Functional Requirements” PERFORCE, https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples
6. “WSU Login”, WSU, https://login.wsu.edu/