Client: Joanne Chung

Instructor: Achala Vinnakota

Group Members:

* Kent Pawson
* Kane Imler
* Amir Soltani
* Anthony Hosemann
* Armaan Singh Klair

System Document

**Level Ground Recruitment**

April 23, 2020

Contents

[Preface 4](#_Toc70006456)

[Purpose of Document 4](#_Toc70006457)

[Documentation Standards 4](#_Toc70006458)

[System Overview 4](#_Toc70006459)

[System Purpose 4](#_Toc70006460)

[Clients 4](#_Toc70006461)

[End-Users 4](#_Toc70006462)

[Main Functions of the System 4](#_Toc70006463)

[User Requirements 5](#_Toc70006464)

[Business Overview and Objectives 5](#_Toc70006465)

[Project Overview 5](#_Toc70006466)

[Statement of the Problem 5](#_Toc70006467)

[System Requirements 6](#_Toc70006468)

[Use Case Diagram 6](#_Toc70006469)

[Use Case Descriptions and Scenarios 9](#_Toc70006470)

[Non-Functional Requirements 14](#_Toc70006471)

[System Interface Requirements 14](#_Toc70006472)

[Usability Requirements 14](#_Toc70006473)

[Problem Domain Class Diagram 15](#_Toc70006474)

[System Design 15](#_Toc70006475)

[Layered Architecture 15](#_Toc70006476)

[Hardware Architecture 20](#_Toc70006477)

[Hardware Platform 20](#_Toc70006478)

[Software Platform 20](#_Toc70006479)

[Interaction Model 20](#_Toc70006480)

[Persistence Model 24](#_Toc70006481)

[Class Diagram 26](#_Toc70006482)

[Interaction Sequence Diagrams 31](#_Toc70006483)

[State chart Diagram 32](#_Toc70006484)

[Testing 32](#_Toc70006485)

[Test Plans 32](#_Toc70006486)

[System Administration and Maintenance 36](#_Toc70006487)

[Security 36](#_Toc70006488)

[Operation 36](#_Toc70006489)

[Backup and Restore 36](#_Toc70006490)

[Data Archival 36](#_Toc70006491)

[System Maintenance 36](#_Toc70006492)

[Deficiency List 36](#_Toc70006493)

[Requirements Not Met 36](#_Toc70006494)

[Non-Functioning Components 37](#_Toc70006495)

[Appendix 38](#_Toc70006496)

[Program Listings 38](#_Toc70006497)

[Glossary 38](#_Toc70006498)

[Index 39](#_Toc70006499)

[Figure 1: Candidate Use Case 6](https://mysait.sharepoint.com/sites/TheGenerousIndividualsCMPS-303Team/Shared%20Documents/General/Final%20Document%20Submission/Level%20Ground%20Recruitment%20System%20Document.docx#_Toc70006266)

[Figure 2: Business Client Use Case 7](https://mysait.sharepoint.com/sites/TheGenerousIndividualsCMPS-303Team/Shared%20Documents/General/Final%20Document%20Submission/Level%20Ground%20Recruitment%20System%20Document.docx#_Toc70006267)

[Figure 3: System Admin Use Case 8](https://mysait.sharepoint.com/sites/TheGenerousIndividualsCMPS-303Team/Shared%20Documents/General/Final%20Document%20Submission/Level%20Ground%20Recruitment%20System%20Document.docx#_Toc70006268)

[Figure 4: Problem Domain Class Diagram 15](#_Toc70006269)

[Figure 5: Data Access Class Diagram 15](#_Toc70006270)

[Figure 6: Filters Class Diagram 16](#_Toc70006271)

[Figure 7: Problem Domain Class Diagram 16](#_Toc70006272)

[Figure 8: Services Class Diagram 17](#_Toc70006273)

[Figure 9: Servlets Class Diagram 17](#_Toc70006274)

[Figure 10: Strategies Class Diagram 18](#_Toc70006275)

[Figure 11: Util Class Diagram 18](#_Toc70006276)

[Figure 12: Validation Class Diagram 19](#_Toc70006277)

[Figure 13: Landing Page 20](#_Toc70006278)

[Figure 14: Sign-up/Log-in Page 21](#_Toc70006279)

[Figure 15: Candidate Profile Page 21](#_Toc70006280)

[Figure 16: Candidate Edit Profile Page 22](#_Toc70006281)

[Figure 17: Candidate Job Posting Search Page 22](#_Toc70006282)

[Figure 18: Business Client Profile Page 22](#_Toc70006283)

[Figure 19: Business Client Edit Profile Page 23](#_Toc70006284)

[Figure 20: Business Client Job Posting Page 23](#_Toc70006285)

[Figure 21: System Admin Profile Page 24](#_Toc70006286)

[Figure 22: System Admin Business View Page 24](#_Toc70006287)

[Figure 23: Conceptual ERD 25](#_Toc70006288)

[Figure 24: Physical ERD 25](#_Toc70006289)

[Figure 25:Data Access Package 26](#_Toc70006290)

[Figure 26: Problem Domain Package 26](#_Toc70006291)

[Figure 27: Services Package 27](#_Toc70006292)

[Figure 28: Servlets Package 28](#_Toc70006293)

[Figure 29: Strategies Package 29](#_Toc70006294)

[Figure 30: Utility Package 29](#_Toc70006295)

[Figure 31:Validation Package 30](#_Toc70006296)

[Figure 32: Create Profile Interaction Sequence Diagram 31](#_Toc70006297)

[Figure 33: Create Job Posting Interaction Sequence Diagram 31](#_Toc70006298)

[Figure 34: State Chart Diagram 32](#_Toc70006299)

[Figure 35: Glossary Table 39](#_Toc70006300)

[Figure 36: Index Table 39](#_Toc70006301)

# Preface

## Purpose of Document

This is a system document for a system for a managing recruitment company and recruited personnel data and information. This system does not require the recruitment company to meet with the recruited person. The company can decide whether to admit the person through the data stored by the recruited person, which is more objective and fairer, and the scope and efficiency of recruitment can be improved through the database. This document describes the scope, objectives and system environment. Then, this document demonstrates the function of the project through various diagram modeling. Through these diagrams, you can easily and clearly understand how the functions of this project are realized. For example, the Use Case Diagram shows the relationship between the three different roles of employees, employers and system administrators and the system. This document is intended to demonstrate the design and implementation of the target system in an object-oriented language.

## Documentation Standards

Header: Times New Roman (body), 18, (Bold)

Sub-header: Times New Roman (body), 14, (Bold)

Body: Times New Roman (body), 12

# System Overview

## System Purpose

The purpose of this system is to reduce bias in the hiring process. Business that uses this system will reduce bias in their hiring process and job seeker using this system will know that a business first impression of them will be based on education, work history and skill and not on age, race, or gender.

## Clients

Joanne Chung

## End-Users

This system is designed with corporate business in mind so those businesses will be the ones that use this system another end user would be any job seekers looking to work in a corporate setting. The last end user would be the system Admins who would keep the system running by adding new skill and roles and selecting candidates for job postings.

## Main Functions of the System

* Create accounts for Business Clients and Candidates.
* Business Clients create Job Postings.
* Candidates create a Profile.
* Candidates search through job postings and apply to job postings.
* Business Clients select candidates for interview.
* System Admins select Candidates for job postings.

# User Requirements

## Business Overview and Objectives

The purpose of the system is to reduce bias in the hiring process by removing age, race, and gender and choose a candidate for the job based on their skills and work experience.

## Project Overview

## Statement of the Problem

Create a system that user will be able to input their information into and then hide certain parts of the information when viewed by a prospective employer so that they make a judgement about that user based on their skills and work experience and not on the users age, race, or gender.

# System Requirements

## Use Case Diagram

Candidate:

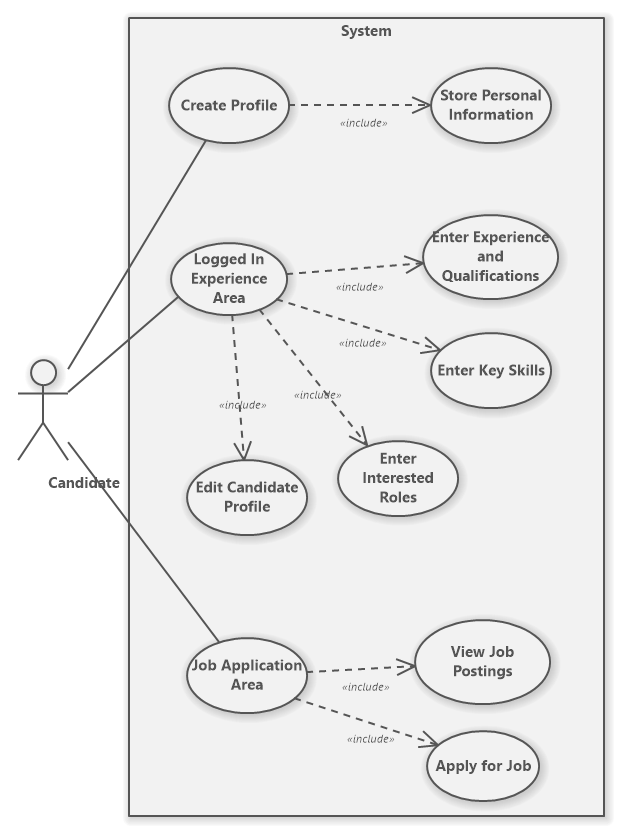


Figure 1: Candidate Use Case

Business Client:

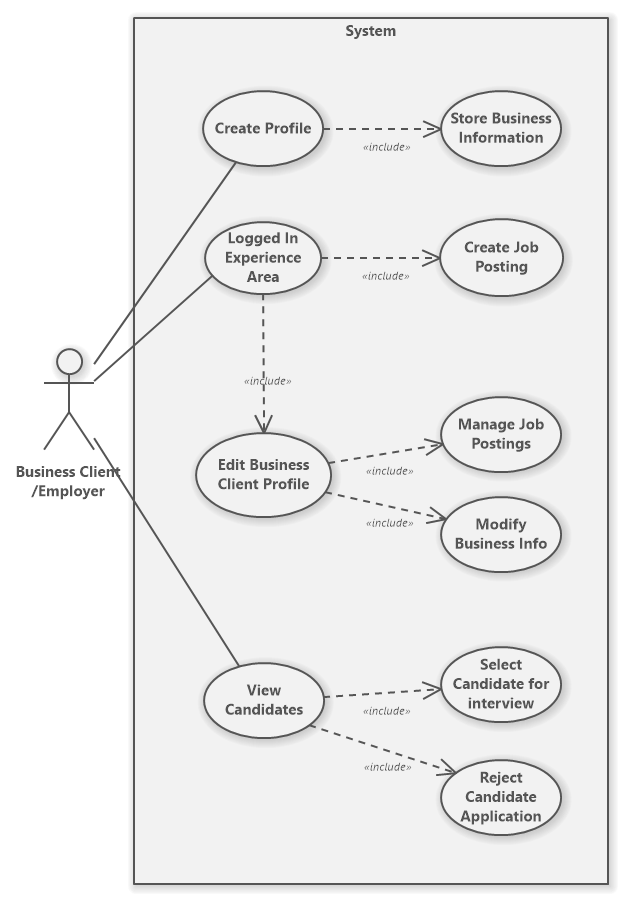


Figure 2: Business Client Use Case

Advisor:

Figure 3: System Admin Use Case

## Use Case Descriptions and Scenarios

**CANDIDATE:**

Candidate Use Case 1: Create Profile

* Click on the Sign-Up button and the system will redirect the user to the registration page.
* Select Candidate and input the appropriate information.
* Click on Sign Up button within registration form.
* The system will validate the registration information. If valid, the system will display a success message and a Candidate account will be created. If invalid, the system will display the appropriate error message and no account will be created.

Candidate Use Case 2: Enter Experience and Qualifications

* Select Candidate and login to the application with a valid username and password.
* To add an education, click on the green icon beside Education. A popup form will appear.
  + Input the appropriate data into the popup form.
  + Click Add to add the education.
  + The system will validate the education information. If valid, the system will display a success message and the education will be added to the Candidate’s account. If invalid, the system will display the appropriate error message and no education will be added.
* To add work history, click on the green icon beside Work History. A popup form will appear.
  + Input the appropriate data into the popup form.
  + Click Add to add the work history.
  + The system will validate the work history information. If valid, the system will display a success message and the work history will be added to the Candidate’s account. If invalid, the system will display the appropriate error message and no work history will be added.

Candidate Use Case 3: Enter Skills

* Select Candidate and login to the application with a valid username and password.
* To add a skill, click on the green icon beside Skills. A popup form will appear.
* Input the appropriate data into the popup form.
* Click Add to add the skill.
* The system will validate the skill information. If valid, the system will display a success message and the skill will be added to the Candidate’s account. If invalid, the system will display the appropriate error message and no skill will be added.

Candidate Use Case 4: Enter Interested Roles

* Select Candidate and login to the application with a valid username and password.
* To add a role, click on the green icon beside Roles. A popup form will appear.
* Input the appropriate data into the popup form.
* Click Add to add the role.
* The system will validate the role information. If valid, the system will display a success message and the role will be added to the Candidate’s account. If invalid, the system will display the appropriate error message and no role will be added.

Candidate Use Case 5: Edit Candidate Profile

* Select Candidate and login to the application with a valid username and password.
* Click on the edit profile icon beside Info. The Candidate will be redirected to their edit page.
* To edit any of the Candidate’s personal information, select the Personal Information tab (selected by default) and input new data into the appropriate field(s)
* To edit any of the Candidate’s work history, select the Work History tab and input new data into the appropriate field(s)
* To edit any of the Candidate’s education, select the Education tab and input new data into the appropriate field(s)
* To edit any of the Candidate’s skills, select the Skills tab and input new data into the appropriate field(s)
* To edit any of the Candidate’s roles, select the Roles tab and input new data into the appropriate field(s)
* After editing any of the data in the Candidate’s profile, press Save Changes button, to save the changes. The system will validate the profile information. If valid, the system will display a success message and the Candidate’s profile will be updated. If invalid, the system will display the appropriate error message and the Candidate’s profile will not be updated.

Candidate Use Case 6: Job Application Area

* Select Candidate and login to the application with a valid username and password.
* Specify the search criteria.
  + Choose to search by either Title or Location.
  + Apply filters for the job type and salary range.
* Click on the magnifying glass icon to search.
* The system will check the database for any jobs matching the search criteria. If a match was found, the system will display “Found Match” and below an accordion menu will contain the job(s) information.
* To apply for a job posting, expand the accordion menu and press Apply Now. If the Candidate has not yet applied for this job posting, the system will display a success message and the Candidate will be added as an applicant to the job posting. Else if the candidate already has applied for the job, the system will display an error message and the Candidate will not be reinserted into the job posting.

**BUSINESS CLIENT:**

Client Use Case 1: Create Profile

* Click on the Sign-Up button and the system will redirect the user to the registration page.
* Select Business Client and input the appropriate information.
* Click on Sign Up button within registration form.
* The system will validate the registration information. If valid, the system will display a success message and a Business Client account will be created. If invalid, the system will display the appropriate error message and no account will be created.

Client Use Case 2: Create Job Posting

* Select Business and login to the application with a valid username and password.
* To add a job posting, click on the green icon beside Postings. A popup form will appear.
* Input the appropriate data into the popup form.
* Click Add to add the job posting.
* The system will validate the job posting information. If valid, the system will display a success message and the job posting will be created. If invalid, the system will display the appropriate error message and no job posting will be created.

Client Use Case 3: Edit Profile

* Select Business and login to the application with a valid username and password.
* Click on the edit profile icon beside Info. The Business Client will be redirected to their edit page.
* To edit any of the Business Client’s business information, select the Business Information tab (selected by default) and input new data into the appropriate field(s)
* To edit any of the Business Client’s job postings, select the Job Postings tab and input new data into the appropriate field(s)
* After editing any of the data in the Business Client’s profile, press Save Changes button, to save the changes. The system will validate the profile information. If valid, the system will display a success message and the Business Client’s profile will be updated. If invalid, the system will display the appropriate error message and the Business Client’s profile will not be updated.

Client Use Case 3: View Candidates

* Select Business and login to the application with a valid username and password.
* If the logged in Business Client does not have any job postings, they must first create one.
* Under the Postings section, click on View Candidates for a specific job posting. The system will redirect the user to that specific job posting.
* Any candidates that have been approved by the Advisor will appear here. By expanding the accordion menu, the Business Client can choose to either reject or select the Candidate for interview.
* To select Candidate for interview, press select for interview.
  + The system will send an email to the Candidate letting them know they’ve been selected for interview.
* To reject the Candidate, press reject.

**SYSTEM ADMIN/ADVISOR:**

Advisor Use Case 1: Manage Database

* Select Advisor and login to the application with a valid username and password.
* To create a new advisor select Add new advisor. A popup form will appear.
  + Input the appropriate registration information for the advisor into the form.
  + Click Add to create the new advisor account.
  + The system will validate the registration information. If valid, the system will display a success message and the new advisor account will be created. If invalid, the system will display the appropriate error message and the advisor account will not be created.
* To create a new skill select Add new skill.
  + Input the skill description into the form.
  + Click Add to add the new skill.
  + The system will validate the skill description and check if the skill currently exists. If the skill does not exist, the system will display a success message and the new skill will be persisted in the database. If the skill does exist, the system will display an error message saying, “skill already exists.”
* To create a new role select Add new role.
  + Input the role description into the form.
  + Click Add to add the new role.
  + The system will validate the role description and check if the role currently exists. If the role does not exist, the system will display a success message and the new role will be persisted in the database. If the role does exist, the system will display an error message saying, “role already exists.”

Advisor Use Case 2: Manage Assigned Businesses

* Select Advisor and login to the application with a valid username and password.
* The system automatically assigns job postings and their associated business client with the advisors in our application.
* If this advisor has an assigned business, it will be displayed under the Assigned Businesses section.
* To view business information, expand the accordion menu for the specific business client.
* To view a list of job postings for a specific business, press List job postings. The system will redirect the user to the job postings page for the advisor.
* Job postings assigned to the advisor will be listed under the Job Posts section.
* To see job posting information, expand the accordion menu for that specific job posting.
* To view job applicants, click on Applicants button. The system will redirect the user to the candidate view page for the advisor.
* Candidates that have applied to the specified job posting will be listed under the Candidates section.
* To view Candidate information, expand the accordion menu for that specific Candidate.
* To reject a Candidate, Press Reject. The system will display a success message saying, “Candidate Rejected”.
* To approve a Candidate, Press Approve. The system will display a success message saying, “Candidate Approved”

## 

## Non-Functional Requirements

The System administrators/advisors are required to vet all information for legitimacy within our system and ensure no scams are occurring. Additionally, the System administrators/advisors are required to oversee any fees that take place on our system.

## System Interface Requirements

###### The project is required to be a web application so that anybody can access it through the internet.

## Usability Requirements

Our client never explicitly stated usability requirements. However, any web application requires UI elements that are common across different websites so that the application is more user friendly in terms of the user’s preexisting mental model.

## Problem Domain Class Diagram

Diagram, schematic

Description automatically generated

Figure 4: Problem Domain Class Diagram

# System Design

## Diagram Description automatically generatedLayered Architecture

Figure 5: Data Access Class Diagram

Text

Description automatically generated with medium confidence

Figure 6: Filters Class Diagram

Diagram, schematic

Description automatically generated

Figure 7: Problem Domain Class Diagram

A picture containing text

Description automatically generated

Figure 8: Services Class Diagram

Table, Word

Description automatically generated

Figure 9: Servlets Class Diagram

Diagram, schematic

Description automatically generated

Figure 10: Strategies Class Diagram

Graphical user interface, text, application, Word

Description automatically generated

Figure 11: Util Class Diagram

Diagram

Description automatically generated with low confidence

Figure 12: Validation Class Diagram

## Hardware Architecture

Our application requires a server to store our database as well as another server to host our application. Currently we are using Heroku’s server to host our application and JawsDB’s server to keep our database. We are currently using TCP/IP to connect from our devices to the Heroku application. The Heroku application then connects to the JawsDB database when needed.

## Hardware Platform

Processor: Intel Core i5-8350U CPU at 1.70GHz

RAM: 16 GB

System Type: 64-bit Operating System

## Software Platform

The software platform that was used to develop this project was Apache NetBeans. MySQL was used for the databases. For the deployment we used Heroku and JawsDB.

## Interaction Model

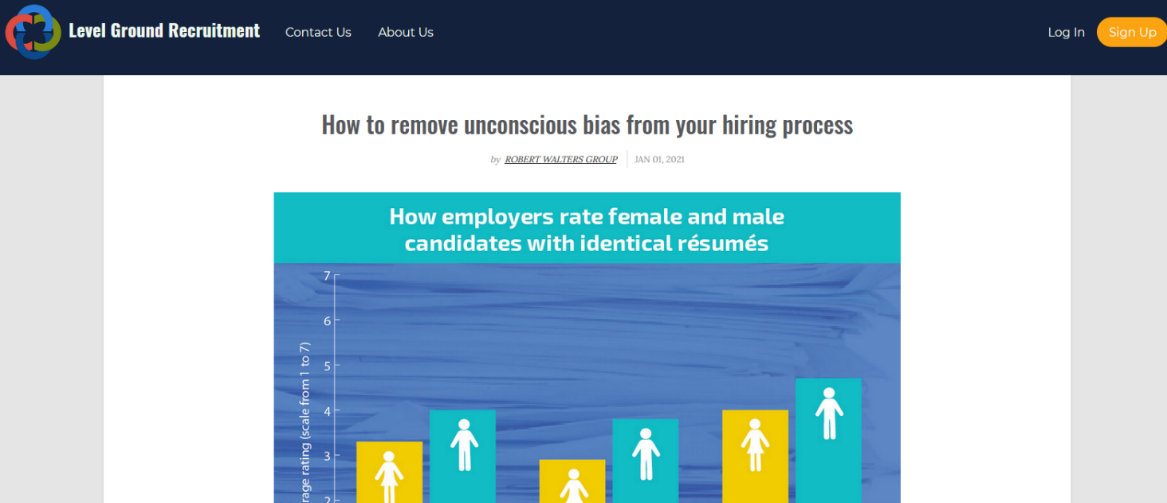


Figure 13: Landing Page

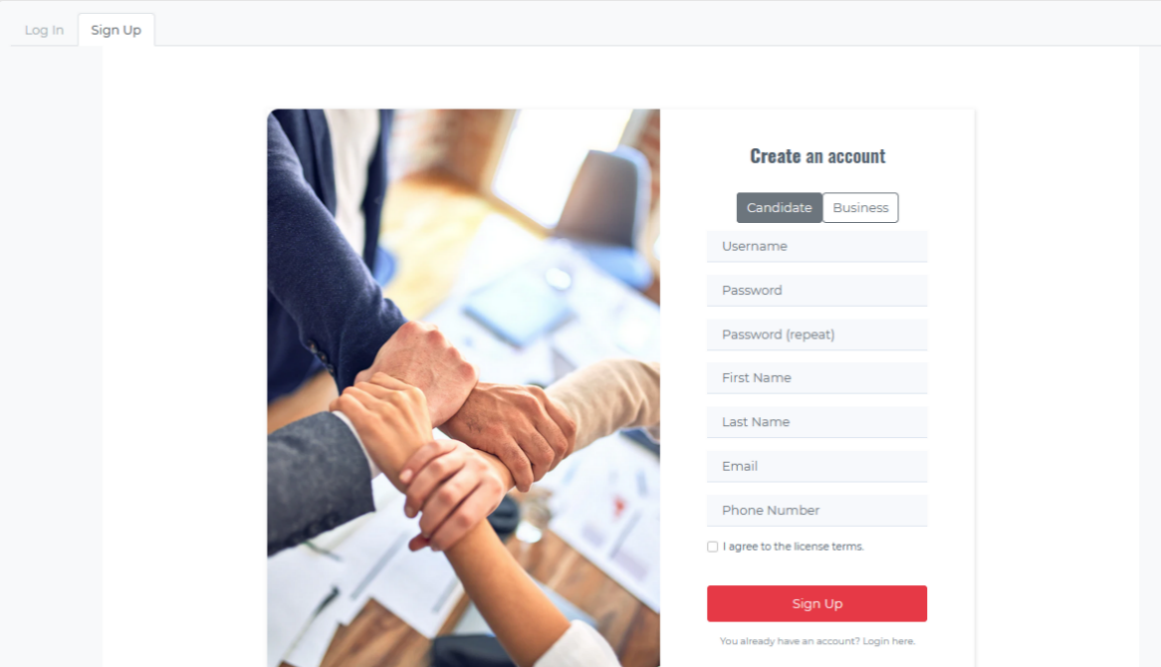


Figure 14: Sign-up/Log-in Page

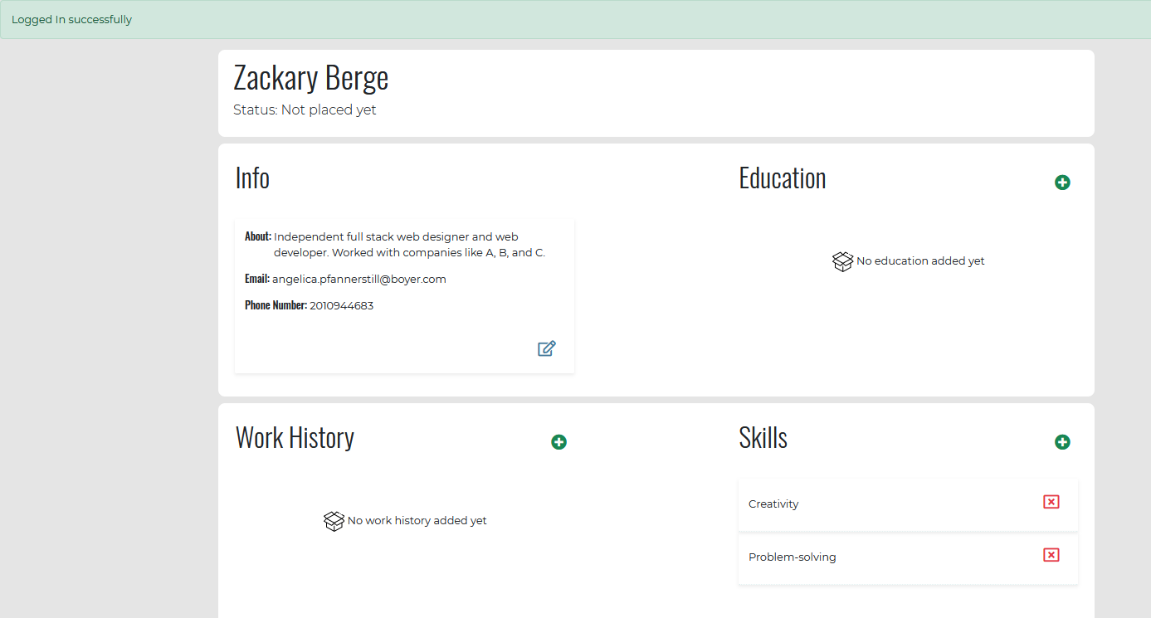


Figure 15: Candidate Profile Page

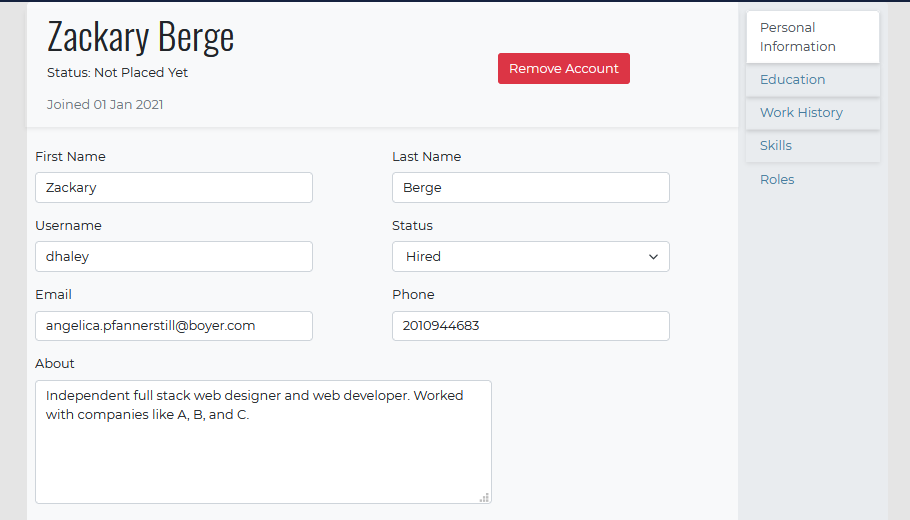


Figure 16: Candidate Edit Profile Page

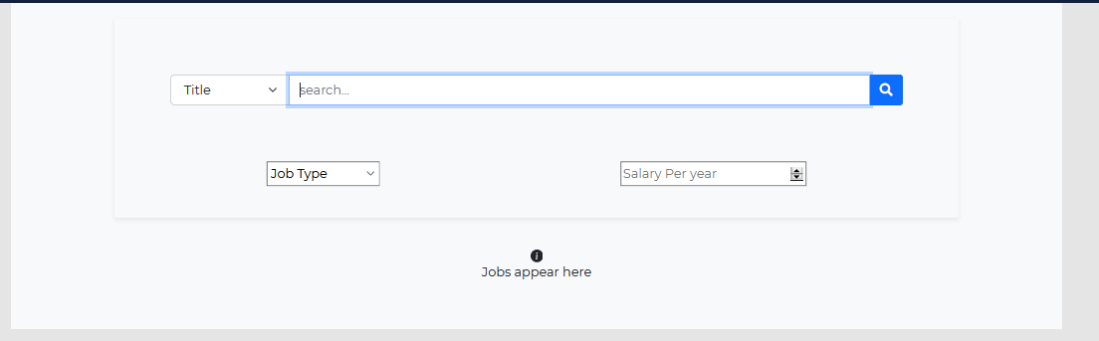


Figure 17: Candidate Job Posting Search Page



Figure 18: Business Client Profile Page

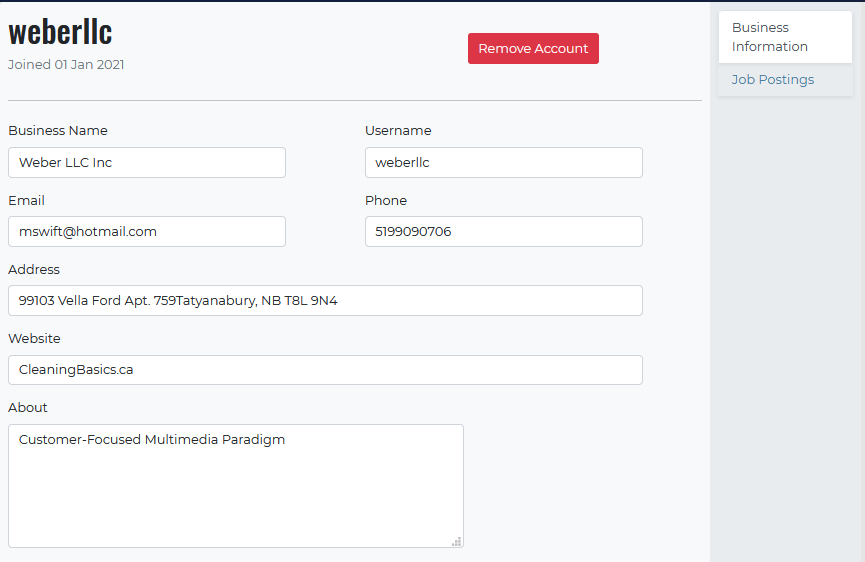


Figure 19: Business Client Edit Profile Page

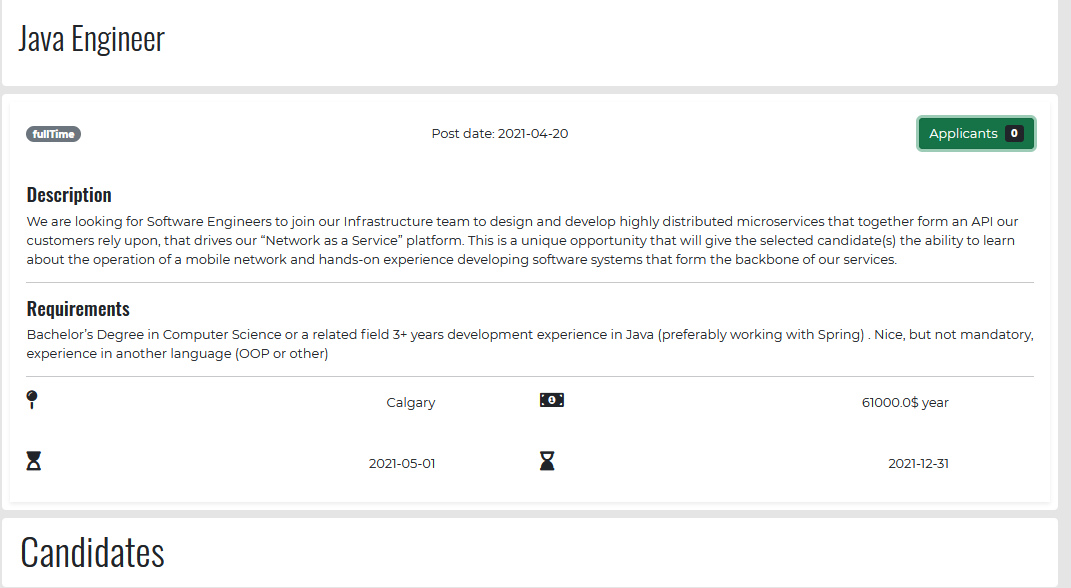


Figure 20: Business Client Job Posting Page



Figure 21: System Admin Profile Page

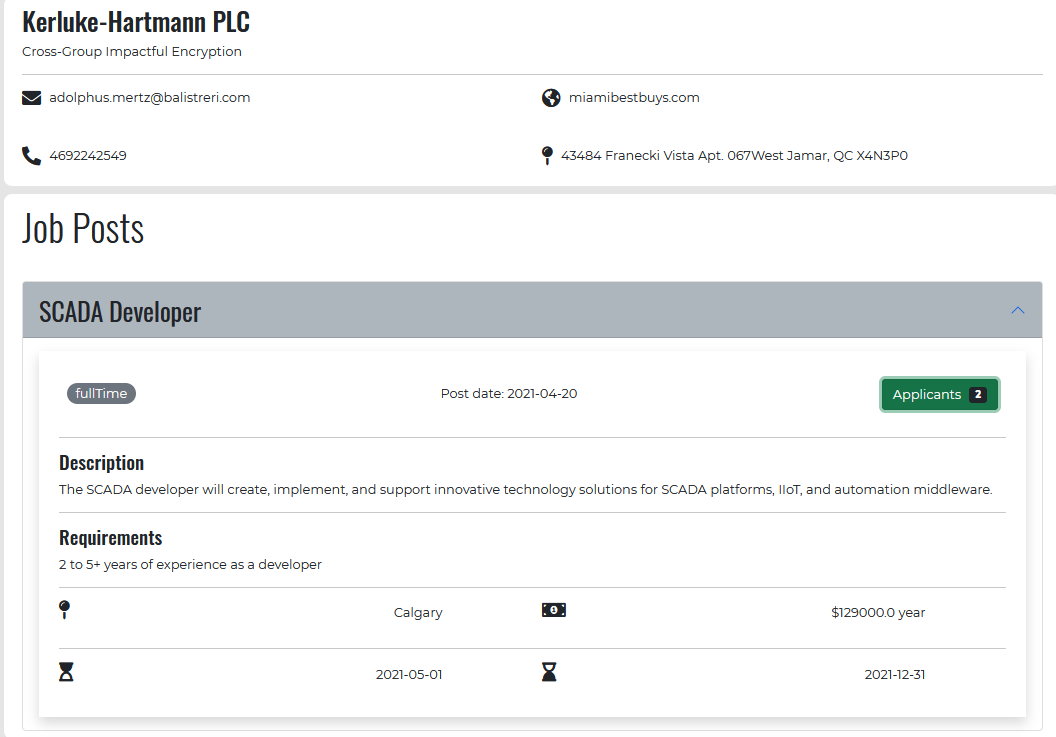


Figure 22: System Admin Business View Page

## Persistence Model

In our application we use JPA to persist java objects onto our database. These objects are defined in the problem domain package and represent rows of data in our database. The database can handle up to 2GB of data.

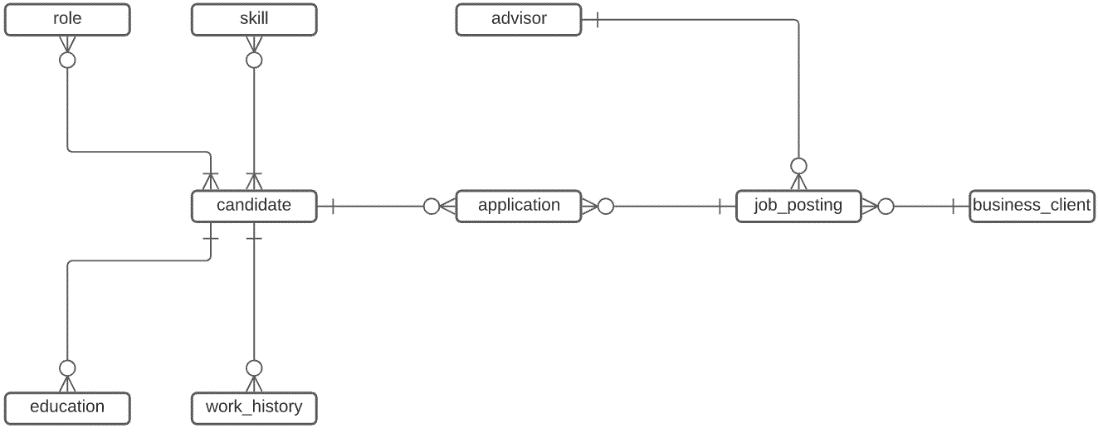


Figure 23: Conceptual ERD

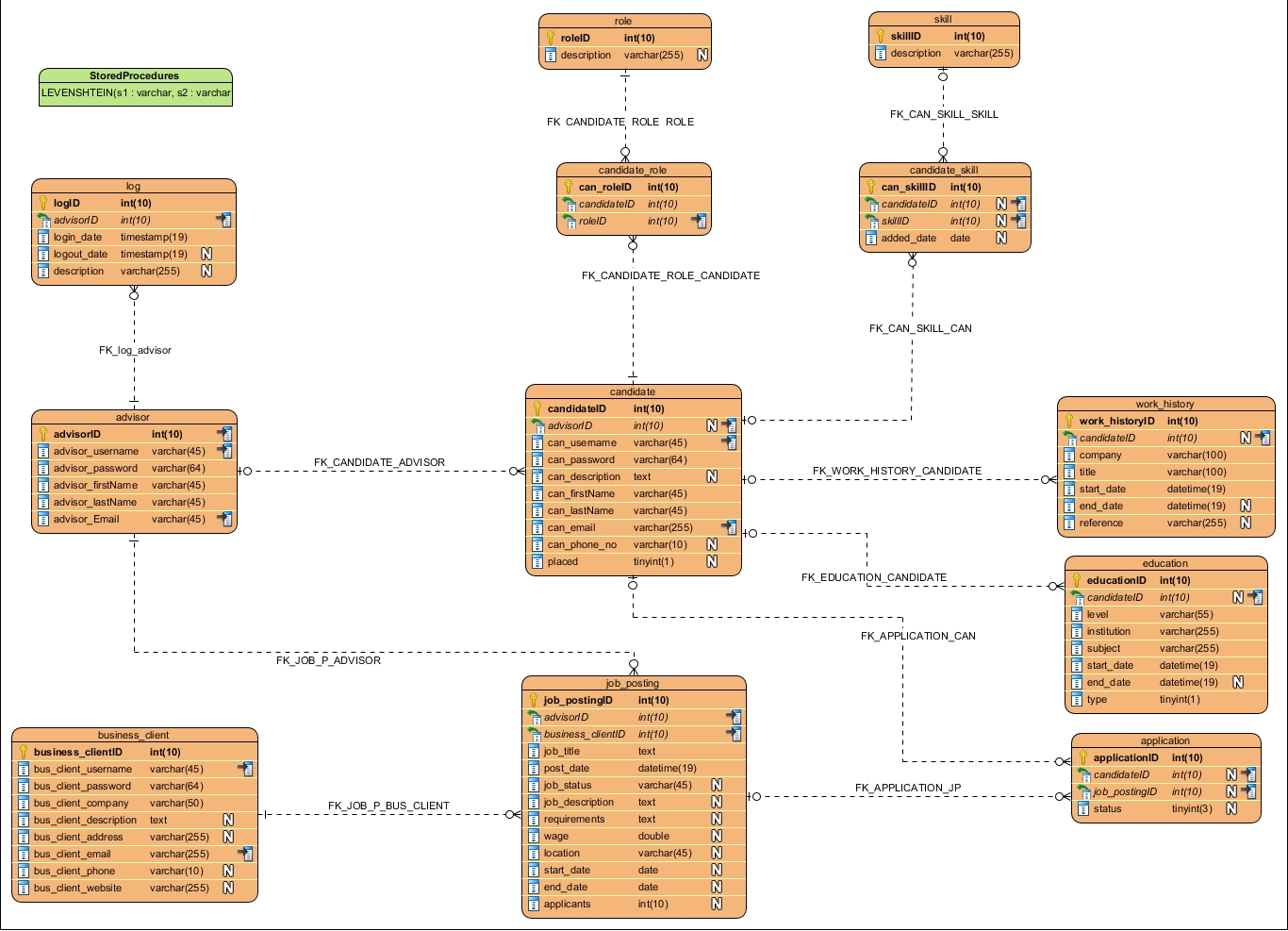


Figure 24: Physical ERD

## Class Diagram

Graphical user interface, text, application

Description automatically generated

Figure 25:Data Access Package

Diagram

Description automatically generated

Figure 26: Problem Domain Package

Table

Description automatically generated with medium confidence

Figure 27: Services Package

A picture containing graphical user interface

Description automatically generated

Figure 28: Servlets Package

Timeline

Description automatically generated

Figure 29: Strategies Package

Graphical user interface, text, application, Word

Description automatically generated

Figure 30: Utility Package

Graphical user interface, application

Description automatically generated

Figure 31:Validation Package

## Interaction Sequence Diagrams

Diagram

Description automatically generated with medium confidence

Figure 32: Create Profile Interaction Sequence Diagram

A picture containing table

Description automatically generated

Figure 33: Create Job Posting Interaction Sequence Diagram

## State chart Diagram

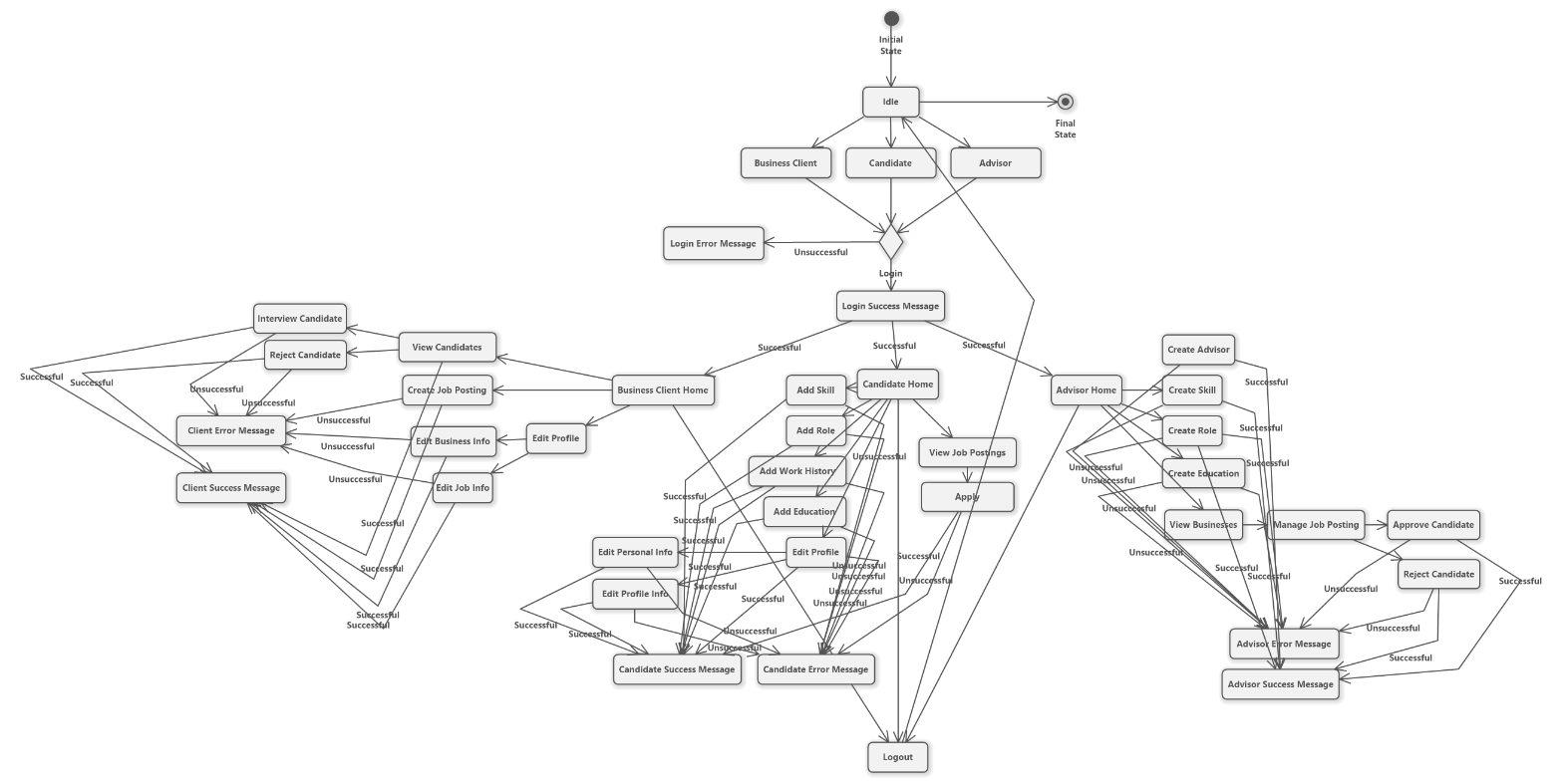


Figure 34: State Chart Diagram

# Testing

## Test Plans

The testing process for our project can is segmented into two categories, Components, and Integration. Component testing will include single methods both in the front-end and the back end, while integration will include scenarios as a system user.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Component / Integration Point Description | Component / Integration | Test Value (Input) | Expected Output | Successful / Unsuccessful |
| 1 | logo/title | Component | mouse click | redirect to home | Successful |
| 2 | contact us | Component | mouse click | open contact us form | Successful |
| 3 | about us | Component | mouse click | open about us form | Successful |
| 4 | login/sign up | Component | mouse click | redirect to login/signup page | Successful |
| 5 | sending an email through contact from | Integration | submitting a form through contact us from | user's default email client will open | Successful |
| 5 | login as buttons | Component | trying to enter with different type of users | filter users based on their user type and restrict access. | Successful |
| 6 | login process | Integration | user tries to login as different types of users | form submit ion takes the user to their relevant page | Successful |
| 7 | signup types | Component | clicking on different user types during sign up process | input fields get updated accordingly after clicking on each user type | Successful |
| 8 | signup process | Integration | users tries to sign up as different types of users | successful signup progress and redirected to the relevant page | Successful |
| 9 | candidate profile page popups and buttons | Component | mouse click | opens the relevant popup | Successful |
| 10 | data pulled from the database | Component | education, work, skill, role section | each section shows the relevant information | Successful |
| 11 | deleting items | Component | clicking on the delete icon | removes the relevant item | Successful |
| 11 | creating new items | Integration | user opens popup and create a new item | successfully created item show in the relevant section | Successful |
| 12 | edit profile | Component | mouse click | redirect to the edit profile page | Successful |
| 13 | remove account | Integration | user tries to remove s/he account | account will be removed and redirected to the home page | Successful |
| 14 | editing profile items | Integration | user tries to modify their profile info and items | successfully edit item and new value show up in the correct spot | Successful |
| 15 | searching for jobs | Integration | user tries to search for a job based on title | relevant result will be shown on the bottom of the page | Successful |
| 16 | searching for jobs | Integration | user tries to search for a job based on location | relevant result will be shown on the bottom of the page | Successful |
| 17 | searching for jobs | Integration | using filters to narrow down the search result | relevant result will be shown on the bottom of the page | unsuccessful |
| 17 | applying for jobs | Integration | user tries to apply for a job | successfully applies for the job | Successful |
| 18 | add new advisor | Integration | advisor tires to add a new advisor to the system | new advisor will be added to the database | Successful |
| 19 | add new skill | Integration | advisor tires to add a new skill to the system | new skill will be added to the database | Successful |
| 20 | add new role | Integration | advisor tires to add a new role to the system | new role will be added to the database | Successful |
| 21 | viewing relevant businesses | Component | assigned business section | correct businesses show up | Successful |
| 22 | list job posting | Component | mouse click | redirect to the job posting view page | Successful |
| 23 | job posts | Component | job posting section | correct postings are being shown | Successful |
| 23 | applicants | Component | mouse click | direct to the candidate view page | Successful |
| 24 | candidate view page | Component | business, job post, and candidate section | correct info being shown in each section | Successful |
| 25 | approve candidate | Integration | advisor approves a candidate | successfully approve the relevant candidate | Successful |
| 26 | reject candidates | Integration | reject the candidate | successfully removes the candidate | Successful |
| 27 | business page | Component | info and posting sections | correct info being shown in each section | Successful |
| 28 | creating new job posting | Integration | business tires to create a new job posting | successfully creates the new job posting | Successful |
| 29 | removing a job posting | Integration | business tires to remove a job posting | successfully removes the job posting | Successful |
| 29 | view candidate | Component | user tires to view candidates | successfully directed to the candidate view page | Successful |
| 30 | reject candidates | Integration | business tires to reject the candidate | successfully removes the candidate form the list | Successful |
| 31 | approve candidate | Integration | business tires to setup interview | successfully opens email client | Successful |

Table 1: Test Plan

# System Administration and Maintenance

## Security

Although security was not the first consideration of our project, we still implemented some general security measurements that are necessary for any project.

* Input validation on the front-end and back-end to prevent exploits like SQL injection.
* Using filter classes to restrict user's access to pages and made a modular authentication process.
* Forms are processed using the POST method to restrict unexpected parameters, which will help to harden the application against CRSF and XSS attacks.

## Operation

The system requires JDK version 8.0, Tomcat version 9.0, MySQL version 8.0 to operate. the project uses Java EE libraries as well as JPA, JSTL, and MySQL connector libraries to operate. The application is created and developed using NetBeans IDE and can be run on Windows, Linux, and Mac environment.

## Backup and Restore

We plan to do a MySQL dump every midnight, since applicants could potentially apply at any time. We also plan on saving the archived logs. Whenever database breaks, we would restore the backup.

## Data Archival

We plan to store the exported data on a separate secondary storage device.

## System Maintenance

In order to maintain a user-friendly application, UI elements and front-end design changes are likely to require changes. A recommended technique is conducting user surveys to collect valuable information from our users.

Additionally, as our system grows bigger, the system administrators/advisors may require extra functionality on their dashboard to better support and manage job postings and their associated business client. A technique to maintain this section, is conducting feedback from the system advisors and determine what features need to be added.

# Deficiency List

## Requirements Not Met

Not Applicable. All the requirements are met.

## Non-Functioning Components

The filters are not working as they should on the Job Postings search page.

# Appendix

## Program Listings

The Javadoc for our application is located within the “dist” folder of our final project folder.

* Go to the project’s root directory.
* Select dist directory. In here you’ll see javadoc folder and a .war file for our project.
* Select javadoc
* Within the javadoc folder, click on index.html. This will open the javadoc for our project.

# Glossary

|  |  |
| --- | --- |
| Term | Description |
| Backup and Restore | Strategies to make copies of application data. |
| Business Client | A business that is looking to employ candidates. |
| Candidate | A person seeking to be employed. |
| Class Diagram | A diagram that describes the structure of the system’s classes, their attributes, operations, and relationship among objects. |
| Client | A person who initiates the project and whom the project is being completed for. |
| Data Dictionary | A set of information describing the content, format and structure, and information about data models and elements. |
| Deficiency List | A list of work that needs repair or incomplete work in a project. |
| End-User | A person who uses the final product. Our system is designed for these users. |
| Hardware Architecture | The physical architecture of a system. |
| Hardware Platform | Hardware required for the platform. |
| Javadoc | Documentation for Java code. |
| Layered Architecture | The different layers at which data is sent to for processing. |
| Maintainability requirements | Requirements to keep the system maintained over time. |
| Non-Functional Requirements | The requirements that are performed outside of the project scope. |
| Persistence Model | A model that models the storage structure of a system. |
| Problem Domain | The targeted area that a projects |
| Project Scope | List of goals, deliverables, tasks, costs and deadlines. |
| Software Platform | Platform the software is designed to run on. |
| State chart Diagram | A diagram representing the states and transitions within an application. |
| System Administrator/Advisor | An admin that facilitates the hiring process. |
| System Interface Requirements | Requirements, if necessary, of a system to communicate with other computer systems. |
| System Maintenance | Sections of a system that will likely need reworking in the future. |
| Test Plan | A detailed step by step plan to ensure the component(s) is working as expected. |
| Usability Requirements | Requirements of a system to match system to users, and tasks users will perform. |
| Use Case Description | A detailed, step by step scenario of each action contained in a use case. |
| Use Case Diagram | A diagram that provides actions for a specific user of the system. |

Figure 35: Glossary Table

# Index

|  |  |
| --- | --- |
| Topic | Pages |
| Business client | 7,11,22,23 |
| Candidate | 6,9,10,21,22 |
| Advisor | 8,12,13,24 |
| Job Posting | 4, 5, 11, 12, 36, 37 |
| Profile | 4, 9, 10, 11 |
| Class diagram | 25, 36, 32 |
| State | 5, 14 |
| Hardware | 9, 10, 11, 12, 13, 14 |
| Software | 20 |
| Model | 14, 20, 24, 25 |
| Admin | 4, 5, 12, 14 |
| Create | 4,5, 9, 11, 12 |
| Role | 4, 9, 10, 12, 33, 34 |
| Skill | 4, 5, 9, 10, 12, 33 |
| Requirement | 4, 14, 37 |
| Error | 9, 10, 11, 12 |
| Sequence | 33 |

Figure 36: Index Table