SW Engineering CSC 648/848 Fall 2022

Team 2: Binary Brains

The Hiring Guru

Farhan Haider (Team Lead/Backend Lead): shaider1@sfsu.edu
Kenny Leong (Frontend Lead/GitHub Master): kleong2@mail.sfsu.edu
Mamadou Bah (Frontend): mbah1@mail.sfsu.edu
Khushi Khanna (Backend) kkhanna@sfsu.edu
Eric Leow (Backend): elow@mail.sfsu.edu

https://hiring-guru.us-west-1.elasticbeanstalk.com/

Dec 15th, 2022

Row	Document Version	Review Date	Description
1	Version 1	12/1/2022	Milestone 5 was assigned, Document was first created with sections to be filled out
2	Version 2	12/8/2022	Backend work and Front work has been assigned with all team members knowing their tasks
2	Version 3	12/14-12/15 /2022	Final moments of both the document and the application. Wrapping everything up to turn it all in.

Table of Contents

1 - Product Summary……..…….……..3

2 - Milestone Documents M1-M4……..…..…....7

3 - Screenshots of key DB table…….….….…8

4 - Screenshots of Task Management System……..…….....11

5 - Team Contribution 12

Product Summary

Product name: Hiring Guru

All Major Committed Functional Requirements

Authorized User

All Committed Functions
A User is able to logout of the website anytime. Additionally, the User is able to log out from anywhere on the website.
A User is able to create a company. To create a company, the User needs to provide company name, company website,company email,industry type of company,a phone number of User, an email of the User, company size, and company type.
A User is able to create a new role. To create a role, the User needs to provide the role title, role expectations, and role benefits.
A User is able to update an existing role. To make this work, a User has to pick an existing role and update its properties.
A User should be able to see the list of all roles.For each role, they are able to see the role title, expectations of the role, and the benefits of the role benefits.
A User should be able to remove a role from the system. To do so, they have to go to the role page, and click on the section called actions, and from there the user is able to select delete which will allow the user to delete the role.
A User is able to create a new job. To create a job, the User needs to provide the job title, job workplace/location, job employment type, job location, and job description.
A User is able to close a job, if they don’t wish to leave the job listing on the website.
A User should be able to see the list of all job.For each role, they are able to see the job title, job workplace/location, job employment type, job location, and job description, and change any or all of the requested job fields.
A User should be able to remove a job from the system. To do so, they have to go to the job page, and click on the section called actions, and from there the user is able to select remove which will allow the user to delete the role.
A User should be able to see the list of all jobs.For each job, they are able to see job title, job workplace/location, job employment type, job location, and job description,
A User should be able to add a referral for a specific position. For a user to be able to do this, the user has to include their first name, last name, user email, and a description of their chosen

candidate, and the applicant's resume.
A User is able to update an existing application. To make this work, a User has to pick an existing application and update its properties.
A User should be able to remove an application from the system. To do so, they have to go to the application page, and click on the section called actions, and from there the user is able to select remove which will allow the user to delete the application .
A User should be able to create the hiring process for a Role. For this, the User should be able to add different stages that are needed for the role hiring process.
A User should be able to add a new stage to the hiring process for a Role. They are able to add a new stage and provide a description of the new stage added.
AUser should be able to see the hiring process for a role. The hiring process shows the different stages and a description of each of the stages.
A User should be able to move a stage backward in the hiring process. For a User to do this, they have to go to the recruitment process page and find the stage they wish to move backward and click on the arrow that allows them to move the stage backwards.
A User should be able to move a stage forward in the hiring process. For a User to do this, they have to go to the recruitment process page and find the stage they wish to move backward and click on the arrow that allows them to move the stage forward.
A User should be shown a confirmation before deleting a stage in the hiring process. This ensures that the action done by the User is done as expected, and a stage isn't deleted by accident.
A User should be able to move from page to page by navigating through the various links in the side. They are able to go from one page to another easily.
A User should be able to search for jobs using a query field
A User should be able to search for a job. A User should start off by selecting a job type(Full Time, Part Time, or All), and entering a keyword to match the job they are looking for.

General User

Requirement Statement
A General user should be able to sign up directly for an individual account. The General user would be prompted to select an email address and a password to be able to login.
A General User should be able to sign up using a Google account.
A General User should be able to login using a Google account. The General user should select

an existing google account and the matching password.
A General User should be able to login using Username and Password. As long as the General User enters an existing and valid username and password the user should be able to login and view the website.
A General user should be able to recover his/her account using email. If an account that matches the provided email exists, an email will be sent to the user that would allow them to change their password.
A General User should be able to go back to the home screen anytime by clicking on the HiringGuru logo.
A General User should be able to see the HiringGuru about page in the page footer.
A General User should be able to see all of the product features in the home page, and furthermore should be able to navigate through the entire website.
Any General User should be able to search for a job. A General User should start off by selecting a job type(Full Time, Part Time, or All), and entering a keyword to match the job they are looking for.
Any General User should be able to apply for a job. To do this, the General User needs to provide their name, phone number, email, and resume.

Additional Features

All the code changes are done by creating feature branches and pull requests
Master branch is protected against direct commits
Atleast one approval is required for merging a pull request
Backend deployment is done using continuous deployment automatically after every commit to master branch
Frontend deployment is done using continuous deployment automatically after every commit to master branch
Docker containers are used for enhanced scalability and isolation
Frontend application is server over SSL
Backend application is served over SSL

What is Unique about our Product

Can you think of a software application that allows organizations to not only create companies, add employees, and manage their teams, but also have the ability to post jobs, view various candidates that applied, manipulate their hiring process with ease, and have an easy way to contact them? No? Well not anymore. With the Hiring Guru application, you get the combined features of managing your entire company or a designated team, while also being able to view and oversee the candidates that apply to the jobs you post throughout their application process. With just a few buttons, anyone in your company can create any job that is desired and have the wonderful option of posting it to multiple job searching platforms such as LinkedIn. There, any candidate will have the option to apply and enter the hiring process, where the employee can easily view all candidates who apply and what stage they are at. It's all on one application and all made for ease. No longer will you have to use different applications for hiring people, such as posting your job on LinkedIn, and then going into a shared Excel Sheet and manually typing in the candidates you receive and updating it based on their status through the process. The Hiring Guru will do all that for you and more all on the application: Want to search for a specific job? We got that! Want to view the hiring process for every job posted? Already done! Want to remove, edit, and submit a job referral on the same page?? Hiring Guru has got you covered! And keep in mind, all of this is for some sweet deals and payment plans to help even the smallest of startup companies hire the best employees for their companies. That sounds like an awesome deal to me! No one is doing it like Hiring Guru is and you should definitely check out our site below!

URL:

https://hiring-guru.us-west-1.elasticbeanstalk.com/

Milestone Documents M1-M4

Documents are Merged with PDF

Screenshots of key DB Tables

- Company Table

Data Output Messages Notifications

	id [PK] integer	description character varying (255)
1	553	Twitter, Inc. is an American social media company based in San Francisco, California. The company opera
2	134	Meta Platforms, Inc., doing business as Meta and formerly named Facebook, Inc., and TheFacebook, Inc.,
3	264	Nike, Inc. is an American multinational corporation that is engaged in the design, development, manufactu
4	155	"Amazon Ads is one of the fastest-growing businesses at Amazon. Our ambition is to reinvent advertising,
5	160	Sephora is a French multinational retailer of personal care and beauty products.
6	164	Creative Circle is a recruiting and consulting services company. We specialize in digital marketing and cre
7	170	Google Cloud Platform, offered by Google, is a suite of cloud computing services that runs on the same in
8	173	Meta Platforms, Inc., doing business as Meta and formerly named Facebook, Inc., and TheFacebook, Inc.,
9	177	Uber, based in San Francisco, provides mobility as a service, ride-hailing allowing users to book a car and

- Employee Table

File Object Tools Help

Dashboard Properties SQL Statistics Dependencies Dependents postgres/postgres@HiringGuru*

postgres/postgres@HiringGuru

No limit

Data Output Messages Notifications

	id [PK] bigint	auth0_id character varying (255)	created_at character varying (255)	designation character varying (255)	roles character varying (255)	company_id integer	user_id integer
1	229	[null]	2022-11-30 08:37:31.2...	Binary Brains	Team Lead	177	228
2	294	[null]	2022-11-30 22:33:47.9...	Backend Engineer	Hygiene Technician	177	293
3	457	[null]	2022-12-06 21:47:12.1...	Backend Engineer	Roles	177	456
4	652	[null]	2022-12-15 11:27:00.9...	Frontend Engineer	Test role	177	651

- Role Table

File Object Tools Help

Dashboard Properties SQL Statistics Dependencies Dependents postgres/postgres@HiringGuru*

postgres/postgres@HiringGuru

No limit

Data Output Messages Notifications

	id [PK] integer	benefits character varying (255)
1	161	Include enhanced insurance coverage, flexible work hours, 24x7 well-being support, sessions on physical and mental wellness, resources for financial support and special allowance and reimburseme
2	174	Our holistic well-being programs include benefits to help boost your physical health, build resilience to manage mental health and stay healthy in all areas of life.s
3	380	Beyond competitive pay, you can receive incentive awards for your performance. Other great perks include 401(k) match, stock purchase plan, paid maternity and parental leave, PTO, multiple health p
4	156	Beyond competitive pay, you can receive incentive awards for your performance. Other great perks include 401(k) match, stock purchase plan, paid maternity and parental leave, PTO, multiple health p
5	165	At Creative Circle you will receive medical, disability, and life insurance; retirement benefits; paid time off; and fringe benefits.
6	171	Paid time off, including vacation, bereavement, jury duty, sick leave, parental leave, disability, holidays, and global reset/wellbeing days.Hybrid work model two work from home days each week for mo
7	558	Beyond competitive pay, you can receive incentive awards for your performance. Other great perks include 401(k) match, stock purchase plan, paid maternity and parental leave, PTO, multiple health p
8	226	Offers partners the choice of multiple coverage levels for medical, dental and vision plans, as well as Health Care and Dependent Care reimbursement accounts, life insurance, disability and accident
9	587	You will get a 10 percent discount on regularly priced general merchandise and fresh produce when you use your Associate Discount Card at any Wstore in the United States.You can also get the sam
10	477	Beyond competitive pay, you can receive incentive awards for your performance. Other great perks include 401(k) match, stock purchase plan, paid maternity and parental leave, PTO, multiple health p
11	544	Beyond competitive pay, you can receive incentive awards for your performance. Other great perks include 401(k) match, stock purchase plan, paid maternity and parental leave, PTO, multiple health p

- Job Table

Data Output Messages Notifications

	id [PK] integer	description character varying (255)	location character varying (255)	title character varying (255)	type character varying (255)	role_id integer
1	494	Sephora is looking for a Senior Accountant for their Sunnyvale off...	Sunnyvale,California	Senior Accountant	FULL_TIME	161
2	495	Sephora is looking for a Senior Accountant for their Sunnyvale off...	Sunnyvale,California	Senior Accountant	PART_TIME	161
3	497	Amazon of looking for a Software Engineer for its Seattle office.	Seattle,WA	Software Engineer	PART_TIME	477
4	498	Amazon of looking for a Software Engineer for its Seattle office.	Seattle,WA	Software Engineer	FULL_TIME	477
5	500	We are looking for someone knowledgeable in the frontend	Seattle,WA	Web Developer-HTML	PART_TIME	165
6	549	Uber is looking for a driver for their Atlanta office.	Atlanta,GA	Driver	FULL_TIME	380

- Job Application Table

File Object Tools Help

Dashboard Properties SQL Statistics Dependencies Dependents postgres/postgres@HiringGuru*

postgres/postgres@HiringGuru

No limit

Data Output Messages Notifications

	id [PK] integer	applicant_email character varying (255)	applicant_name character varying (255)	applicant_profile_link character varying (255)	applicant_resume character varying (255)	cover_letter character varying (255)	phone character varying (255)	submitted_at character varying (255)	job_id integer
1	499	khannakhushi09@gma...	Khushi Khanna	KhushiKhanna.linkedin	KhushiKhanna	KhushiKhanna-CoverLe...	650-454-4969	12/10/22	49
2	501	kleong@gmail.com	Kenny Leong	KennyLeong.linkedin	KennyLeongResume	KennyLeong-CoverLetter	123-457-3999	12/10/22	50
3	502	mbah@gmail.com	Mamadou Bah	MamadouBah.linkedin	Mamadou_Bah_Resume	MamadouBah-CoverLe...	480-456-2344	12/10/22	50
4	614	khannakhushi09@gma...	Khushi Khanna	KhushiKhanna.linkedin	KhushiKhanna	KhushiKhanna-CoverLe...	650-454-4969	12/10/22	49
5	685	eleow@gmail.com	Eric Leow	EricLeow.linkedin	EricLeow_Resume	EricLeow-CoverLetter	925-726-3456	12/15/22	54
6	686	fhaider@gmail.com	Farhan Haider	FarhanHaider.linkedin	FarhanHaider_Resume	FarhanHaider-CoverLet...	925-445-7888	12/15/22	54

- Hiring Process Stage Table

ls Help

Dashboard Properties SQL Statistics Dependencies Dependents postgres/postgres@HiringGuru*

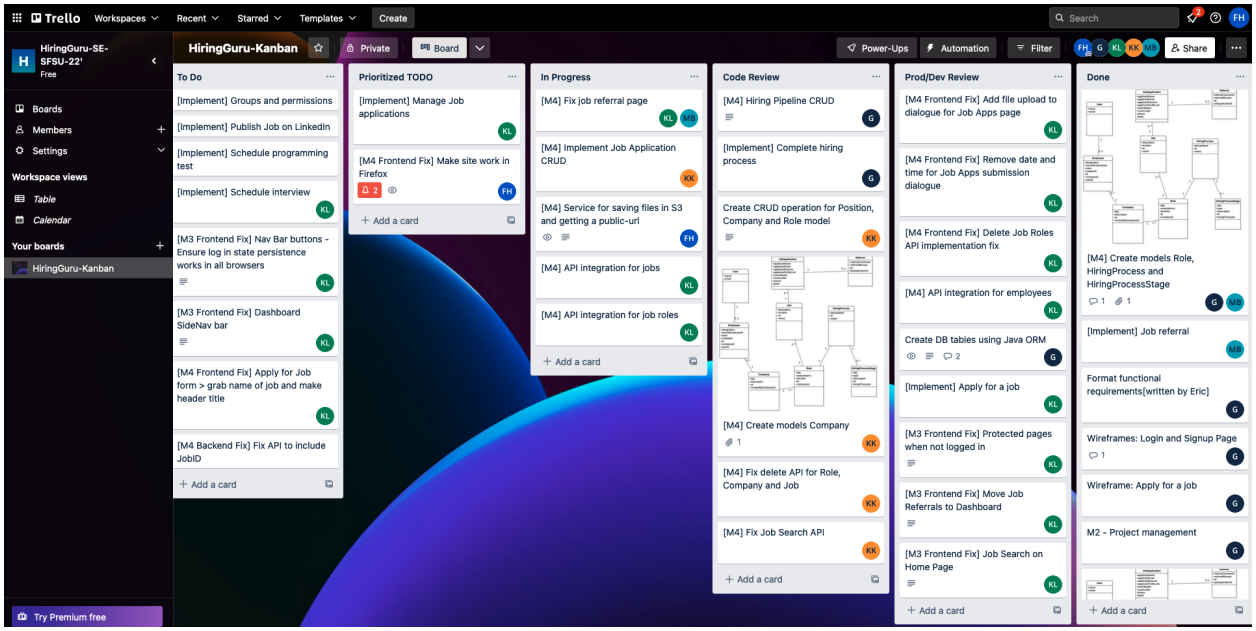
postgres/postgres@HiringGuru

No limit

Data Output Messages Notifications

	id [PK] bigint	description character varying (255)	index integer
1	687	The interview will be with a Software Engineer. It should cover basic concepts like OOP, DS and Problem Solving.	7
2	688	Coding tests are typically programming challenges that help technical recruiters test programming skills and competencies of a developer or progra...	8
3	664	Coding tests are typically programming challenges that help technical recruiters test programming skills and competencies of a developer or progra...	3

Screenshots of Task Management System(Trello)



Team Contributions

M5 Contributions

Farhan Haider

- Worked on some of the Hiring Process CRUD functionality
- Helped the backend team resolve issues with the backend
- Helped Kenny and Mamadou in debugging frontend
- Organized team meetings
- Reviewed the committed work and other documentation work
- Added post analysis section
- Fixed build issues on the backend

Kenny Leong

- Fixed bugs/addressed frontend feedback from M4
    - Fixes for Job Roles and Manage Employees frontend API implementations
    - Apply for Job form
    - General UI and UX improvements
- Job Roles frontend API implementation
- Hiring Pipeline frontend API implementation
- Worked with Mamadou on Job Applications frontend API implementation

Khushi Khanna

- Job Application CRUD
- Hiring Process Delete and Patch functionality
- M5 DB Screenshots

Mamadou Bah

- Job Application Implementation
- Aided with smaller Frontend bugs
- Helped with overall document

Eric Leow

- Non Functional Requirements: Remove unnecessary entries
- Hiring Pipeline CRUD
- StoryBoard: Redo Apply for Job
- StoryBoard: Redo Apply Referral for Job
- Testing Hiring Process

Overall Semester Contribution Scores

➔ Kenny Leong - 8

- ➔ Mamadou Bah - 6
- ➔ Eric Leow - 5
- ➔ Khushi Khanna - 7

Post Analysis

Based on my experience working on this project, I would keep the following points in mind or would do differently if I work on a similar project again.

1. As a team lead, instead of focusing entirely on your own contribution, allocate significant time to planning and overseeing work from other team members.
2. Have a thorough discussion with each team member about their experience in their chosen role. Special attention should be given to the experience survey filled at the start of the semester.
3. Evaluate the team not just on their hard work but also on their contribution to the project since that's what eventually matters to the team.
4. Meet at least twice a week with the team, if the team faces issues working in isolation have longer sessions with everyone on call.
5. When using services like Auth0, keep in mind that they require your frontend and backend to be running on SSL. Configuring SSL for servers might not be trivial. ZeroSSL can be used to get free SSL certificates.
6. When provisioning the deployment infrastructure, keep an eye on the costs. For example, with AWS if you are running separate instances for frontend and backend, your EC2 fee-tier limit with run out and you will be charged for extra use.
7. After every milestone, do a retrospective meeting to discuss what went well and what didn't.

SW Engineering CSC 648/848 Fall 2022

The Hiring Guru

Team 2: Binary Brains
Farhan Haider (Team Lead/Backend Lead): shaider1@sfsu.edu
Kenny Leong (Frontend Lead/GitHub Master): kleong2@mail.sfsu.edu
Mamadou Bah (Frontend): mbah1@mail.sfsu.edu
Khushi Khanna (Backend) kkhanna@sfsu.edu
Eric Leow (Frontend): elow@mail.sfsu.edu

Milestone #4

Dec 1st, 2022

Row	Document Version	Review Date	Description
1	Version 1	11/14/2022	Milestone 4 was assigned, Document was first created with sections to be filled out
2	Version 2	11/16/2022	Sections were slowly being filled out with various work to be done on the application
3	Version 3	11/21/2022	Continue to work on the application while the document is being completed.
4	Version 4	11/28/2022	Final few meetings to discuss last things needed for the application as well what is left for the document
5	Version 5	11/30-12/1	Wrapping up the document for submission and anything left for backend work

Table of Contents

1 - Product Summary……..……..……..3

2 - Usability Test Plan……..…..…....6

3 - Quality Assurance Test Plan……..……..…14

4 - Code Review ……..…....21

5 - Self-Check for Best Practices for Security 44

6 - Adherence to original Non-Functional Requirement specs.....48

7 - Team Contribution 52

Product Summary

Product name: Hiring Guru

All Major Committed Functional Requirements

Authorized User

All Committed Functions
A User is able to logout of the website anytime. Additionally, the User is able to log out from anywhere on the website.
A User is able to create a company. To create a company, the User needs to provide company name, company website,company email,industry type of company,a phone number of User, an email of the User, company size, and company type.
A User is able to create a new role. To create a role, the User needs to provide the role title, role expectations, and role benefits.
A User is able to update an existing role. To make this work, a User has to pick an existing role and update its properties.
A User should be able to see the list of all roles.For each role, they are able to see the role title, expectations of the role, and the benefits of the role benefits.
A User should be able to remove a role from the system. To do so, they have to go to the role page, and click on the section called actions, and from there the user is able to select delete which will allow the user to delete the role.
A User is able to create a new job. To create a job, the User needs to provide the job title, job workplace/location, job employment type, job location, and job description.
A User is able to close a job, if they don’t wish to leave the job listing on the website.
A User should be able to see the list of all job.For each role, they are able to see the job title, job workplace/location, job employment type, job location, and job description, and change any or all of the requested job fields.
A User should be able to remove a job from the system. To do so, they have to go to the job page, and click on the section called actions, and from there the user is able to select remove which will allow the user to delete the role.
A User should be able to see the list of all jobs.For each job, they are able to see job title, job workplace/location, job employment type, job location, and job description,
A User should be able to add a referral for a specific position. For a user to be able to do this, the user has to include their first name, last name, user email, and a description of their chosen

candidate, and the applicant's resume.
A User is able to update an existing application. To make this work, a User has to pick an existing application and update its properties.
A User should be able to remove an application from the system. To do so, they have to go to the application page, and click on the section called actions, and from there the user is able to select remove which will allow the user to delete the application .
A User should be able to create the hiring process for a Role. For this, the User should be able to add different stages that are needed for the role hiring process.
A User should be able to add a new stage to the hiring process for a Role. They are able to add a new stage and provide a description of the new stage added.
AUser should be able to see the hiring process for a role. The hiring process shows the different stages and a description of each of the stages.
A User should be able to move a stage backward in the hiring process. For a User to do this, they have to go to the recruitment process page and find the stage they wish to move backward and click on the arrow that allows them to move the stage backwards.
A User should be able to move a stage forward in the hiring process. For a User to do this, they have to go to the recruitment process page and find the stage they wish to move backward and click on the arrow that allows them to move the stage forward.
A User should be shown a confirmation before deleting a stage in the hiring process. This ensures that the action done by the User is done as expected, and a stage isn't deleted by accident.
A User should be able to move from page to page by navigating through the various links in the side. They are able to go from one page to another easily.
A User should be able to search for jobs using a query field
A User should be able to search for a job. A User should start off by selecting a job type(Full Time, Part Time, or All), and entering a keyword to match the job they are looking for.

General User

Requirement Statement
A General user should be able to sign up directly for an individual account. The General user would be prompted to select an email address and a password to be able to login.
A General User should be able to sign up using a Google account.
A General User should be able to login using a Google account. The General user should select

an existing google account and the matching password.
A General User should be able to login using Username and Password. As long as the General User enters an existing and valid username and password the user should be able to login and view the website.
A General user should be able to recover his/her account using email. If an account that matches the provided email exists, an email will be sent to the user that would allow them to change their password.
A General User should be able to go back to the home screen anytime by clicking on the HiringGuru logo.
A General User should be able to see the HiringGuru about page in the page footer.
A General User should be able to see all of the product features in the home page, and furthermore should be able to navigate through the entire website.
Any General User should be able to search for a job. A General User should start off by selecting a job type(Full Time, Part Time, or All), and entering a keyword to match the job they are looking for.
Any General User should be able to apply for a job. To do this, the General User needs to provide their name, phone number, email, and resume.

What is Unique about our Product

Can you think of a software application that allows organizations to not only create companies, add employees, and manage their teams, but also have the ability to post jobs, view various candidates that applied, manipulate their hiring process with ease, and have an easy way to contact them? No? Well not anymore. With the Hiring Guru application, you get the combined features of managing your entire company or a designated team, while also being able to view and oversee the candidates that apply to the jobs you post throughout their application process. With just a few buttons, anyone in your company can create any job that is desired and have the wonderful option of posting it to multiple job searching platforms such as LinkedIn. There, any candidate will have the option to apply and enter the hiring process, where the employee can easily view all candidates who apply and what stage they are at. It's all on one application and all made for ease. No longer will you have to use different applications for hiring people, such as posting your job on LinkedIn, and then going into a shared Excel Sheet and manually typing in the candidates you receive and updating it based on their status through the process. The Hiring Guru will do all that for you and more all on the application: Want to search for a specific job? We got that! Want to view the hiring process for every job posted? Already done! Want to

remove, edit, and submit a job referral on the same page?? Hiring Guru has got you covered! And keep in mind, all of this is for some sweet deals and payment plans to help even the smallest of startup companies hire the best employees for their companies. That sounds like an awesome deal to me! No one is doing it like Hiring Guru is and you should definitely check out our site below!

URL:

https://hiring-guru.us-west-1.elasticbeanstalk.com/

Usability Test Plan

Tests

1 - Add a new stage to the recruitment process

Test Objectives

In a company, the hiring process for a role can involve multiple stages including interviews and technical assessments. One of the main features of the HiringGuru app is that it allows users to design the hiring process for a particular role in the company. The purpose of this test is to understand how easy it is to add a new stage to the hiring process for a role, one of the most basic steps in the process of designing the hiring process. The usability of this feature is critical to the success of the feature because all other features related to designing the hiring process require the creation of the hiring process stages.

More specifically, the test has the following objectives

1. Check whether it's easier for the user to navigate the Recruitment Process Page
2. Check whether the user can easily locate the button for adding a new stage
3. Check whether the form for adding a new stage is easy to fill
4. Check whether the form validation errors are being communicated properly
5. Check how efficient the current user experience is
6. Measure the overall user satisfaction with the feature

Test Description

System setup

Since the application is accessible through the browser, the only setup required is a GUI system with a supported browser i.e. active version of Firefox or Chrome.

Starting point

This usability test starts when the user is on the Hiring Guru dashboard and wants to add a new stage to the hiring process for a particular role. Here’re the steps needed to reach the starting point for this test

1. Go to the Hiring Guru URL
2. On the Landing Page, click on the Login button
3. Enter test credentials

Intended Users

The intended users for this feature are the HR executives in the company because it’s their job to administer the recruitment process.

URL of the system

https://hiring-guru.us-west-1.elasticbeanstalk.com/

2 - Change position of a recruitment stage(Mamadou)

Test Objectives

There are multiple stages in a hiring process that can vary in a number of ways depending on the job. A more technical-focused job may have numerous coding assessments compared to a non-technical job that will be having a lot more behavioral interviews. With these various recruitment stages, Hiring Guru gives any company the ability to shift the different positions of the recruitment stage to accommodate for the skills they want to see from their candidate. An example would be shifting a technical interview before a behavioral interview for certain candidates, so their coding skills can be assessed even before hiring consideration. Features such as this are important for companies to make every recruitment process unique to every job posted.

More specifically, the test has the following objectives

Check whether it's easy to find the buttons to move a recruitment stage
Check whether the functions that move the recruitment stages are understandable
Check how effectively the features respond to the user's actions
Check how efficiently the current user experience is
Measure the overall user satisfaction with the feature

Test Description

System setup

Since the application is accessible through the browser, the only setup required is a GUI system with a supported browser i.e. active version of Firefox or Chrome.

Starting point

This usability test starts when the user is on the Hiring Guru dashboard and wants to move a recruitment stage to a different position within the entire recruitment process.

1. Go to the Hiring Guru URL
2. On the Landing Page, click on the Login button
3. Enter test credentials
4. Click on *Hiring Pipeline* under *Recruitment* in Dashboard
5. Use the given features

Intended Users

The intended users for this feature are the HR executives in the company because it's their job to administer the recruitment process.

URL of the system

https://hiring-guru.us-west-1.elasticbeanstalk.com/

3 - Delete a recruitment stage(Mamadou)

Test Objectives

Now as time goes on, certain features within a company may need some modifications of sorts or just need a sense of general updating. This can apply to the recruitment process, where some stages within the process are no longer needed and should be removed entirely. The Hiring already has a feature like that implemented within the Hiring Pipeline. Not only can certain stages be created or moved, but they can also be entirely deleted if they are no longer needed. A good example of this can be a company no longer finding a need for certain technical roles to have a certain amount of behavioral tests. These companies have the ability to remove any or all the behavioral tests stages from their specific recruitment process of choice to then focus on the other stages of the process. These features continue to add more flexibility for the company for their recruitment process.

More specifically, the test has the following objectives

1. Check whether it's easy to find the button to remove a recruitment stage
2. Check whether the functions that remove the recruitment stages are understandable
3. Check how effectively the features respond to the users actions
4. Check how efficient the current user experience is
5. Measure the overall user satisfaction with the feature

Test Description

System setup

Since the application is accessible through the browser, the only setup required is a GUI system with a supported browser i.e. active version of Firefox or Chrome.

Starting point

This usability test starts when the user is on the Hiring Guru dashboard and wants to move a recruitment stage to a different position within the entire recruitment process.

1. Go to the Hiring Guru URL

2. On the Landing Page, click on the Login button
3. Enter test credentials
4. Click on *Hiring Pipeline* under *Recruitment* in Dashboard
5. Use the given features

Intended Users

The intended users for this feature are the HR executives in the company because it's their job to administer the recruitment process.

URL of the system

https://hiring-guru.us-west-1.elasticbeanstalk.com/

4 - See applications for a job

Test Objectives

When a company posts a job that candidates can apply for, they should have the ability to view all job application submissions. The objective of this test is to prove just that for the Hiring Guru. Once a candidate submits a job application, it's very important for employers to be able to see all of the applications for a job.Seeing all of the applicants, allows the employers to make wise decisions about selecting a candidate. The Hiring Guru application easily allows users to view all job applications submitted for any job posted.

More specifically, the test has the following objectives

1. Check to see if all job applications for the job are shown to the employer
2. Check to see if the employer is able to Reject/Deny applicant
3. Check to see if employer can view selected job application
4. Check how efficient the current user experience is
5. Measure the overall user satisfaction with the feature

Test Description

System setup

Since the application is accessible through the browser, the only setup required is a GUI system with a supported browser i.e. active version of Firefox or Chrome.

Starting point

This usability test starts when the user is on the Hiring Guru dashboard and wants to move to viewing all job applications

1. Go to the Hiring Guru URL
2. On the Landing Page, click on the Login button
3. Go to the Dashboard
4. Click on Job Applications

5 - Add referral for a job

Test Objectives

When employers network with other candidates, they should have the ability to submit referrals to certain candidates that they feel would excel at the job. The Hiring Guru makes this process swift and easy for the user, allowing them to submit a referral for any job posted. This is extremely important,because it allows for an Employer to select an applicant for a job referral. To create a job referral, the employer has to list the first and last name of the applicant, applicant's email, reasoning for choosing the applicant, and the applicant's resume.

More specifically, the test has the following objectives

1. Check to see if an Employer is able to fill in all of the requested fields
2. Make sure that the Job referral can't be submitted unless all of the requested fields are filled
3. Check to see if Employer can easily find job referral section
4. Check how efficient the current user experience is
5. Measure the overall user satisfaction with the feature

Test Description

System setup

Since the application is accessible through the browser, the only setup required is a GUI system with a supported browser i.e. active version of Firefox or Chrome.

Starting point

This usability test starts when the user is on the Hiring Guru dashboard and wants to move to adding a job referral.

1. Go to the Hiring Guru URL
2. On the Landing Page, click on the Login button
3. Go to the Dashboard
4. Go to Job Referral and fill out all of the fields

Test Results

Effectiveness

Testcase	Completion	Errors	Comments
Add a new stage to the recruitment process	PASS	None	N/A
Change position of a recruitment stage	PASS	None	Very efficient, users like the one click
Delete a recruitment stage	PASS	None	N/A
See applications for a job	PASS	None	N/A
Add referral for a job	PASS	None	Job Referral had slowest time to use, might look into in the future

Efficiency

Usecase	Benchmark - Average time to complete	Benchmark - Average number of clicks	Benchmark - Average screen navigations
Add a new stage to the recruitment process	Within 30 Second, including time to create stage	Approximately 8 Clicks	No screen navigation, all on the same page
Change position of a recruitment stage	A few seconds	Approximately 3 clicks	No screen navigation, all on the same page
Delete a recruitment stage	Within 10 seconds	Approximately 2 clicks	No screen navigation, all on the same page
See applications for a job	Within 30 seconds	Approximately 5 clicks	No screen navigation, all on the same page
Add referral for a job	Under a minute	Approximately 10 clicks	2 screen navigation

User Satisfaction

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It was easy to navigate to the recruitment process page					
It was clear where I had to click to add a new recruitment step stage					
The sequence of stages in the recruitment process was easy to understand					
It was easy to add a new stage					
I think I would use the system frequently					
It was easy to change the sequence of recruitment stages					
I found the system unnecessarily complex					
I thought the system was easy to use					
I think I would need the support of a technical person					

to use this software					
I found these functions were well integrated					
There were too many inconsistencies in the system					
I would be easy for most people to use this system					
I felt very confident using this system					
It was clear where I had to navigate to see all jobs					
It was easy to add a job referral					

QA Test Plan

Tests

1- The backend server shall support TLS 1.1, 1.2 and 1.3

Test Objectives

The objective of this test is to make sure that our backend server supports all variations of TLS for the application. TLS represents the current standard for communication privacy so having our backend server supporting these features is crucial to the quality of our application

Hardware and Software Setup

There is no specific Hardware and Software Setup for this test. Simply submit a report through this link(https://www.ssllabs.com/ssltest) and view the results from the report to view with the application that supports TLS.

Features to be Tested

1. Hiring Guru server supports TLS 1.1
2. Hiring Guru server supports TLS 1.2
3. Hiring Guru server supports TLS 1.3

QA Test Plan

Title	Description	Test Input	Expected Output	PASS or FAIL
TLS Test 1	Test to see if the backend server supports TLS 1.1	Submit requested link to https://www.ssllabs.com/ssltest for report	TLS 1.1 is supported and shown in reports	PASS
TLS Test 2	Test to see if the backend server supports TLS 1.3	Submit requested link to https://www.ssllabs.com/ssltest for report	TLS 1.2 is supported and shown in reports	PASS
TLS Test 3	Test to see if the backend server supports TLS 1.3	Submit requested link to https://www.ssllabs.com/ssltest for report	TLS 1.3 is supported and shown in reports	FAIL

2- The application shall be usable in all supported browsers

Test Objectives

The primary objective of this test is to make sure our application is supported in all major browsers such as Chrome, Firefox, or Edge for example. We understand that our users can utilize many different browsers for their personal preferences, so the purpose of this test it to make sure the application is fully functional throughout every browser and no one browser is lacking in quality.

Hardware and Software Setup

There is no specific Hardware and Software Setup. The only setup necessary is to have access to all testing browsers and to utilize the application (https://hiring-guru.us-west-1.elasticbeanstalk.com/) in these respected browsers.

Features to be Tested

1. All functions are still operational in every browser

2. All features are in their designated alignment in every browser
3. Every application paged is aligned well to various screen sizes

QA Test Plan

Title	Description	Test Input	Expected Output	PASS or FAIL
Chrome Test	Test to see if application is functional in Chrome	Open the Hiring Guru in Chrome	Hiring Guru fully operational	PASS
Firefox Test	Test to see if application is functional in Firefox	Open the Hiring Guru in Firefox	Hiring Guru fully operational	PASS
Microsoft Edge Test	Test to see if application is functional in Microsoft Edge	Open the Hiring Guru in Microsoft Edge	Hiring Guru fully operational	PASS

3- The application shall be usable on all supported devices

Test Objectives

The primary objective of this test is to make sure our application is supported in all major devices such as Android, IOS, or Microsoft for example. We understand that our users own many different smartphone devices for their personal preferences, so the purpose of this test is to make sure the application is fully functional throughout every device and no one device is lacking in quality. Another major objective within this test is to make sure our application is able to adjust to any given screen size, so that the application still looks neat and presentable on any device.

Hardware and Software Setup

There is no specific Hardware and Software Setup. The only setup necessary is to have access to all testing devices and to utilize the application (https://hiring-guru.us-west-1.elasticbeanstalk.com/) in these respected devices.

Features to be Tested

1. Application Page is adjustable to various screen sizes
2. All features all still functional on various devices
3. Hiring Guru si fully functional on every major smartphone

QA Test Plan

Title	Description	Test Input	Expected Output	PASS or FAIL
IOS Test	Test to see if application is functional on an IOS device	Open the Hiring Guru on an Iphone	Hiring Guru fully operational	FAIL
Samsung Test	Test to see if application is functional on a Samsung Device	Open the Hiring Guru on a Samsung	Hiring Guru fully operational	PASS
Google Test	Test to see if application is functional on a Google Device	Open the Hiring Guru on a Google Pixel	Hiring Guru fully operational	PASS

4- All team member pull requests shall be approved by the team lead

Test Objectives

All team members pull requests(PR for short) must be approved by the team lead before merging with the master branch. The purpose of this test is to make sure all team members' PR's are operational and functional before they are merged in with the designated prototype. This allows for not only an extra set of eyes to look at the work done in the PR, but prevents any major bugs from appearing on the master branch. By having the person who is not only the most experienced team member but understands the project best, reviews every PR, we only improve with every PR pushed to the master branch and do not run any risks with massive failures.

Hardware and Software Setup

There is no specific Hardware and Software Setup. The only setup necessary is to have access to all Github Repositories (https://github.com/sfsu-joseo/csc648-848-05-sw-engineering-fall-2022-02) and view the pull requests section that has all the team members PRs.

Features to be Tested

1. Every PR cannot be merged by PR owner
2. All PR’s are approved by team lead before merging
3. Any PR not approved is given comments for correction

QA Test Plan

Title	Description	Test Input	Expected Output	PASS or FAIL
Khushi PR Test	Test to see if Khushi's PR have either approval by team lead or comments for correction	Open Github and view PR section	Either approved PR's or commented PRs for future approval	PASS
Kenny PR Test	Test to see if Kenny's PR have either approval by team lead or comments for correction	Open Github and view PR section	Either approved PR's or commented PRs for future approval	PASS
Mamadou PR Test	Test to see if Mamadou's PR have either approval by team lead or comments for correction	Open Github and view PR section	Either approved PR's or commented PRs for future approval	PASS

5- Application shall not be prone to SQL injection attacks

Test Objectives

The purpose of this objective is to make sure our database server and backend server is well protected and users with malicious intent do not have access to this information. When users create their companies on the Hiring Application and post job applications for candidates to apply to, an abundance of information is being put into the database server. This can range from addresses, to emails, to phone numbers, to even payment information. All of this needs to be protected from SQL injection attacks for not only the safety of the application but the safety of its users.

Hardware and Software Setup

There is no specific Hardware or Software Setup for this test. Certain tests just need to be run on either the application (https://hiring-guru.us-west-1.elasticbeanstalk.com/) or the database query for the tes,

Features to be Tested

1. Check whether basic users have access to restricted information
2. Check how functional is our backend server

3. Check how the application responds to unexpected SQL inputs.

QA Test Plan

Title	Description	Test Input	Expected Output	PASS or FAIL
SQL Injection Test 1	To see if the search field will fetch all records	Entering * in a search field	Should not fetch all records	PASS
SQL Injection Test 2	Writing a delete query in the search field	Entering a delete query	Does not Delete Records	PASS
SQL Injection Test 3	Writing a delete query in the post field	Entering a delete query	Does not Delete Records	PASS

Code Review

Selected Code Styling Practices

1. Use PascalCase class names
2. Use all-caps SNAKE_CASE for constants
3. Two line spaces before module-level functions and classes
4. 4-space Tabs
5. Max line character length of 120
6. Name variables, functions, constants, and classes must be easy to understand
7. Large functions must be broken down into smaller ones
8. Instead of duplicating a piece multiple times, define it once and re-use
9. Follow the best practices for Spring Boot and React

10. Separate modules for logically separate features

Code Reviews

File1 - ManageJobs.js

Description

This is the code for one of the most crucial pages on the frontend side. It contains a number of private React components that are used internally by this page. The only component that is exported is the page component itself. As obvious by the name, this page is for managing jobs for a particular role in the company. It has the functionality to create, update and delete a job as well as see a list of all jobs.

Code Review

Review by Team 03 - Alexander Bjeldanes

This is my code review from another team's code. Without much context from what is happening in other files, this is my initial assessment/review of the "JobEditDialog" function in the ManageJobs file.

- Good to have an error check
- Good placeholder information to guide the user
    - Maybe change the "job title" and "workplace" placeholder, but it's not that important
- Clear division of each section of the function
    - Each section has a designated and unique purpose
- Everything else looks good and is well planned out

```
function JobEditDialog(props) {
    return (
        <Dialog
            show={props.show}
            title={props.title}
            actions={props.actions}
        >
            <div>
                <div className={"recruitment-step-errors"}>
                    {
                        props.errors.map((error, index) => {
                            return (
                                <div key={`create-rect-step-error-${index}`}
className="alert alert-danger" role="alert">
                                    {error}
```

```
                                </div>
                            )
                        })
                    }
                </div>
                <div className="mb-3">
                    <label htmlFor="recruitmentStageNameInput"
className="form-label">
                        Job Title
                    </label>
                    <input className="form-control"
id="recruitmentStageNameInput"
                        placeholder="Enter job title"
                        value={props.name}
                        onChange={props.onNameChange}
                    />
                </div>
                <div className="mb-3">
                    <label htmlFor="recruitmentStageNameInput"
className="form-label">
                        Workplace
                    </label>
                    <input className="form-control"
id="recruitmentStageNameInput"
                        placeholder="San Francisco, CA"
                        value={props.workplace}
                        onChange={props.onWorkplaceChange}
                    />
                </div>
                <div className="mb-3">
                    <label htmlFor="recruitmentStageNameInput"
className="form-label">
                        Employment Type
                    </label>
                    <input className="form-control"
id="recruitmentStageNameInput"
                        placeholder="Full-Time, Part-Time, Internship"
                        value={props.employment}
                        onChange={props.onEmploymentChange}
```

```
                    />
                </div>
                <div className="mb-3">
                    <label htmlFor="recruitmentStageNameInput"
className="form-label">
                        Location
                    </label>
                    <input className="form-control"
id="recruitmentStageNameInput"
                        placeholder="San Francisco, CA"
                        value={props.location}
                        onChange={props.onLocationChange}
                    />
                </div>
                <div className="mb-3">
                    <label htmlFor="recruitmentStageDescriptionInput"
className="form-label">
                        Description
                    </label>
                    <textarea className="form-control"
id="recruitmentStageDescriptionInput"
                        rows="5" placeholder="Enter description"
                        value={props.description}
                        onChange={props.onDescriptionChange}
                    >
                    </textarea>
                </div>
            </div>
        </Dialog>
    )
}
```

File2 - RecruitmentProcess.js

Description

This is another file from our frontend application. It contains code for the components related to managing the hiring process for a particular role. Specifically, this page has the functionality to add new stages to the hiring process, modify the sequence of stages and update individual stages. This is one of the core components and unique features of our application.

Code Review

Review by Team 03 - Seth Pavlicek

Overall, this file has good indentation and consistency. It is missing a header and it needs more comments to describe what is happening in each method, component, or constant. One thing that should be considered is to not exceed a line length. There are many lines that are over 100 characters long. However, conforming to this rule may break some of the indentation consistencies. There are more comments below in the code.

This naming scheme is a little confusing. Is the ui supposed to display ‘Software Engineer’ or is it for ‘Software Engineer’;

```
const jobPositions = [
    {
        ui: 'Software Engineer',
        server: 'SOFTWARE_ENGINEER'
    },
    {
        ui: 'CEO',
        server: 'CEO'
    },
]

const RecruitmentStepType = {
    Interview: {
        ui: 'Interview',
        server: 'INTERVIEW'
    },
    ProgrammingTest: {
        ui: 'Programming Test',
        server: 'PROGRAMMING_TEST'
    },
}
```

The naming scheme for your recruitment pipelines is fine. Try to include comments. This file is massive and without some serious digging, I have no clue where these constants are being used and why.

```
const RecruitmentPipelineSoftwareEngineer = [
    {
        type: RecruitmentStepType.Interview,
        title: "First Technical Interview",
        detail: "First interview should be with a Software Engineer. It should cover basic" +
            " concepts like OOP, DS and Problem Solving."
    },
    {
        type: RecruitmentStepType.Interview,
        title: "Second Technical Interview",
        detail: "First interview should be with a Software Engineer. It should cover
advanced" +
            " aspects like Algorithms, Design and Networks."
    },
```

```
    {
        type: RecruitmentStepType.ProgrammingTest,
        title: "HackerRank Test",
        detail: "First interview should be with a Software Engineer. It should cover all" +
            " the technical aspects like OOP, DB, DS, Algorithms and Networks."
    },
    {
        type: RecruitmentStepType.Interview,
        title: "HR Interview",
        detail: "HR Interview should cover discussion about personality and salary package."
    },
]

const RecruitmentPipelineCEO = [
    {
        type: RecruitmentStepType.Interview,
        title: "Interview Evaluation",
        detail: "Interview with the founder to discuss past experience and assess personality
traits."
    },
    {
        type: RecruitmentStepType.Interview,
        title: "HR Interview",
        detail: "HR Interview should cover discussion about personality and salary package."
    },
]

const RecruitmentPipelinePositionMap = {
    SOFTWARE_ENGINEER: RecruitmentPipelineSoftwareEngineer,
    CEO: RecruitmentPipelineCEO
}
```

Good comment at the top describing what this is. You may want to include a comment above “props.errors.map” and “Object.keys(RecruitmentStepType).map” and describe what it is supposed to do.

```
function RecruitmentStepDialog(props) {
    // dialog component for adding a new step to the recruitment process
    return (
        <Dialog
            show={props.show}
            title={"Add a new stage to the recruitment process"}
```

```
            actions={props.actions}
        >
            <div>
                <div className={"recruitment-step-errors"}>
                    {
                        props.errors.map((error, index) => {
                            return (
                                <div key={`create-rect-step-error-${index}`} className="alert
                                 alert-danger" position="alert">
                                    {error}
                                </div>
                            )
                        })
                    }
                </div>
                <div className="mb-3">
                    <label htmlFor="recruitmentStageTitleInput" className="form-label">
                        Title
                    </label>
                    <input className="form-control" id="recruitmentStageTitleInput"
                        placeholder="Enter title ..."
                        value={props.stepTitle}
                        onChange={props.onTitleChange}
                    />
                </div>
                <div className="mb-3">
                    <label htmlFor="recruitmentStageDescriptionInput" className="form-label">
                        Description
                    </label>
                    <textarea className="form-control" id="recruitmentStageDescriptionInput"
                        rows="5" placeholder="Enter description ..."
                        value={props.stepDescription}
                        onChange={props.onDescriptionChange}
                    >
                    </textarea>
                </div>
                <select className="form-select" aria-label="Default select example"
                    value={
                        props.selectedStageType ||
```

```
                        "Select the type of this recruitment stage"
                    }
                    onChange={props.onStageTypeChange}
                >
                    <option disabled>Select the type of this recruitment stage</option>
                    {
                        Object.keys(RecruitmentStepType).map((k) => {
                            return <option
                                key={`recruitment-stage-type-select-${k}`}
                                value={k}
                            >{RecruitmentStepType[k].ui}</option>
                        })
                    }
                </select>
            </div>
        </Dialog>
    )
}
```

Other than the main component, I have no clue what these constants are keeping track of.

```
function RecruitmentProcess() {
    // main recruitment process page
    const appContext = useContext(ApplicationContext);
    const [recruitmentProcessState, setRecruitmentProcessState] = useState({
        selectedjobPosition: jobPositions.All,
        recruitmentProcess: undefined
    })
    const [createRecruitmentStepFormState, setCreateRecruitmentStepFormState] = useState({
        showCreationDialog: false,
        showModificationDialog: false,
        title: "",
        description: "",
        type: undefined,
        errors: [],

    })
    const [modifyRecruitmentStepFormState, setModifyRecruitmentStepFormState] = useState({
        showModificationDialog: false,
        title: "",
        description: "",
```

```
        type: undefined,
        errors: [],
        index: undefined
    })

    const moveStepUp = (index) => {
        // move the step identified by the index one step upward
        let process = recruitmentProcessState.recruitmentProcess
        const saveState = process[index - 1]
        process[index - 1] = process[index]
        process[index] = saveState
        setRecruitmentProcessState({
            ...recruitmentProcessState,
            recruitmentProcess: process
        })
    }
    const moveStepDown = (index) => {
        // move the step identified by the index one step downward
        let process = recruitmentProcessState.recruitmentProcess
        const saveState = process[index + 1]
        process[index + 1] = process[index]
        process[index] = saveState
        setRecruitmentProcessState({
            ...recruitmentProcessState,
            ...recruitmentProcessState,
            recruitmentProcess: process
        })
    }

    const modifyRecruitmentStep = () => {
        // update a recruitment step
        let newSteps = []
        for (let i = 0; i < recruitmentProcessState.recruitmentProcess.length; i++) {
            if (i === modifyRecruitmentStepFormState.index) {
                newSteps.push({
                    type: RecruitmentStepType[modifyRecruitmentStepFormState.type],
                    title: modifyRecruitmentStepFormState.title,
                    detail: modifyRecruitmentStepFormState.description
                })
```

```
            }
            else {
                newSteps.push(recruitmentProcessState.recruitmentProcess[i])
            }
        }
        setModifyRecruitmentStepFormState({
            ...modifyRecruitmentStepFormState,
            showModificationDialog: false
        })
        setRecruitmentProcessState({
            ...recruitmentProcessState,
            recruitmentProcess: newSteps
        })
    }

    const createRecruitmentStep = () => {
        // validate recruitment step info and create a new step
        let errors = []
        if (!createRecruitmentStepFormState.title ||
createRecruitmentStepFormState.title.length === 0) {
            errors.push("Please provide a title for this stage")
        }
        if (!createRecruitmentStepFormState.description ||
createRecruitmentStepFormState.description.length === 0) {
            errors.push("Please provide a description for this stage")
        }
        if (!createRecruitmentStepFormState.type ||
createRecruitmentStepFormState.type.length === 0) {
            errors.push("Please select the type of this stage")
        }
        if (errors.length > 0) {
            setCreateRecruitmentStepFormState({
                ...createRecruitmentStepFormState,
                showCreationDialog: true,
                errors: errors,
            })
        }
        else {
            setCreateRecruitmentStepFormState({
```

```
            ...createRecruitmentStepFormState,
            showCreationDialog: false
        })
        setRecruitmentProcessState({
            ...recruitmentProcessState,
            recruitmentProcess: [
                ...recruitmentProcessState.recruitmentProcess,
                {
                    type: RecruitmentStepType[createRecruitmentStepFormState.type],
                    title: createRecruitmentStepFormState.title,
                    detail: createRecruitmentStepFormState.description
                }
            ]
        })
    }
}

const removeRecruitmentStep = (index) => {
    // delete recruitment step identified by the index
    let newSteps = []
    for (let i = 0; i < recruitmentProcessState.recruitmentProcess.length; i++) {
        i !== index && newSteps.push(recruitmentProcessState.recruitmentProcess[i])
    }
    appContext.closeDialog()
    setRecruitmentProcessState({
        ...recruitmentProcessState,
        recruitmentProcess: newSteps
    })
}
```

I think your naming scheme is doing a good job at describing what is being displayed in this page but it may still need a few comments.

```
return (
    <div>
        <RecruitmentStepDialog
            show={createRecruitmentStepFormState.showCreationDialog}
            title={"Add a new stage to the recruitment process"}
            actions={[
                {
                    title: "Close",
```

```
                handler: () => {
                    setCreateRecruitmentStepFormState({
                        ...createRecruitmentStepFormState,
                        showCreationDialog: false
                    })
                },
                variant: "secondary"
            },
            {
                title: "Create Step",
                handler: createRecruitmentStep,
                variant: "primary"
            }
        ]}
        errors={createRecruitmentStepFormState.errors}
        onTitleChange={(e) => {
            setCreateRecruitmentStepFormState({
                ...createRecruitmentStepFormState,
                title: e.target.value
            })
        }}
        onDescriptionChange={(e) => {
            setCreateRecruitmentStepFormState({
                ...createRecruitmentStepFormState,
                description: e.target.value
            })
        }}
        onStageTypeChange={(e) => {
            setCreateRecruitmentStepFormState({
                ...createRecruitmentStepFormState,
                type: e.target.value
            })
        }}
        selectedStageType={createRecruitmentStepFormState.type}
        stepTitle={createRecruitmentStepFormState.title}
        stepDescription={createRecruitmentStepFormState.description}
    />
    <RecruitmentStepDialog
        show={modifyRecruitmentStepFormState.showModificationDialog}
```

```
title={"Add a new stage to the recruitment process"}
actions={[
    {
        title: "Close",
        handler: () => {
            setModifyRecruitmentStepFormState({
                ...modifyRecruitmentStepFormState,
                showModificationDialog: false
            })
        },
        variant: "secondary"
    },
    {
        title: "Modify stage",
        handler: modifyRecruitmentStep,
        variant: "primary"
    }
]}
errors={modifyRecruitmentStepFormState.errors}
onTitleChange={(e) => {
    setModifyRecruitmentStepFormState({
        ...modifyRecruitmentStepFormState,
        title: e.target.value
    })
}}
onDescriptionChange={(e) => {
    setModifyRecruitmentStepFormState({
        ...modifyRecruitmentStepFormState,
        description: e.target.value
    })
}}
onStageTypeChange={(e) => {
    setModifyRecruitmentStepFormState({
        ...modifyRecruitmentStepFormState,
        type: e.target.value
    })
}}
selectedStageType={modifyRecruitmentStepFormState.type}
stepTitle={modifyRecruitmentStepFormState.title}
```

```
                stepDescription={modifyRecruitmentStepFormState.description}
            />
            <div className={"page-container"}>
                <Breadcrumb>
                    <Breadcrumb.Item href="/dashboard/home">Dashboard</Breadcrumb.Item>
                    <Breadcrumb.Item active>Recruitment: Hiring Pipeline</Breadcrumb.Item>
                </Breadcrumb>
                <h1>Design Hiring Pipeline</h1>
                <div className="position-selection-control">
                    <div className={"position-selection-header"}>
                        <h5>Select a position to configure its Recruitment Process:</h5>
                    </div>
                    <div className={"position-selection-dropdown"}>
                        <div className="input-group input-group-sm">
                            <Dropdown className={"input-group-text"}>
                                <Dropdown.Toggle id="dropdown-basic">
                                    <Filter /> Select Job Position
                                </Dropdown.Toggle>
                                <Dropdown.Menu>
                                    {
                                        jobPositions.map((jobPosition) => {
                                            return (
                                                <Dropdown.Item key={jobPosition.ui}
onClick={(e) => {
                                                    setRecruitmentProcessState({
                                                        ...recruitmentProcessState,
                                                        recruitmentProcess:
RecruitmentPipelinePositionMap[jobPosition.server],
                                                        selectedjobPosition: jobPosition
                                                    })
                                                }}
active={recruitmentProcessState.selectedjobPosition === jobPosition}>
                                                    {jobPosition.ui}
                                                </Dropdown.Item>
                                            )
                                        })
                                    }
                                </Dropdown.Menu>
                            </Dropdown>
```

```
                        </div>
                    </div>
                </div>
                {
                    recruitmentProcessState.recruitmentProcess !== undefined &&
                    <div className={"recruitment-pipeline-detail-container"}>
                        <div className={"recruitment-pipeline-controls
container-vcenter-hright"}>
                            <Button
                                variant="primary"
                                onClick={() => {
                                    setCreateRecruitmentStepFormState({
                                        ...createRecruitmentStepFormState,
                                        showCreationDialog: true
                                    })
                                }}
                            >
                                Add a new stage
                            </Button>
                        </div>
                        <div className={"recruitment-pipeline-detail"}>
                            {
                                recruitmentProcessState.recruitmentProcess.map((step, index)
=> {
                                    const lenSteps =
recruitmentProcessState.recruitmentProcess.length
                                    return (
                                        <div key={`index-${step.title}`}
className={"recruitment-pipeline-step row"}>
                                            <div className={"step-number-container
container-all-center col-1"}>
                                                <Button disabled={true}
className={"step-number btn btn-circle btn-sm"}
                                                        variant="primary">{index +
1}</Button>
                                            </div>
                                            <div className={"step-detail-container col-9"}>
                                                <div className={"step-detail"}>
```

```
                                                    <div className={'step-title
h5'}>{step.title}</div>
                                                    <div
className={'step-type'}>{step.type.ui}</div>
                                                    <div
className={'step-description'}>{step.detail}</div>
                                                </div>
                                            </div>
                                            <div className={"step-actions-container
container-all-center col-2"}>
                                                <div className={"step-actions"}>
                                                    <div className={"icon-container"}>
                                                        <OverlayTrigger
                                                            placement="bottom"
                                                            overlay={
                                                                <Tooltip>Update
stage</Tooltip>
                                                            }
                                                        >
                                                            <button type="button"
                                                                    className="btn btn-circle
btn-sm btn-success"
                                                                onClick={() => {
                                                                    let stepType
                                                                    Object.keys(

RecruitmentStepType).map((k) => {
                                                                        if
(RecruitmentStepType[k].ui === step.type.ui) {
                                                                            stepType = k
                                                                        }
                                                                    })

setModifyRecruitmentStepFormState({

...modifyRecruitmentStepFormState,
                                                                        title: step.title,
                                                                        description:
step.detail,
```

```
                                                                                type: stepType,

showModificationDialog: true,
                                                                                index: index
                                                                            })
                                                                        }}
                                                                    >
                                                                        <PencilFill></PencilFill>
                                                                    </button>
                                                                </OverlayTrigger>
                                                            </div>
                                                            <div className={"icon-container"}>
                                                                <OverlayTrigger
                                                                    placement="bottom"
                                                                    overlay={
                                                                        <Tooltip>Delete
stage</Tooltip>
                                                                    }
                                                                >
                                                                    <button type="button"
                                                                            className="btn btn-circle
btn-sm btn-danger"
                                                                        onClick={() =>
appContext.openDialog(
                                                                            "Are you sure?",
                                                                            [
                                                                                {
                                                                                    title: "Close",
                                                                                    handler:
appContext.closeDialog,
                                                                                    variant:
"secondary"
                                                                                },
                                                                                {
                                                                                    title: "Remove
stage",
                                                                                    handler: () =>
removeRecruitmentStep(index),
```

```
                                                                                variant:
"primary"
                                                                                    }
                                                                            ],
                                                                            "Once deleted, this can't
be undone. " +
                                                                            "Are you sure you want to
proceed?"
                                                                        )}
                                                                    >
                                                                        <Trash3Fill></Trash3Fill>
                                                                    </button>
                                                                </OverlayTrigger>
                                                            </div>
                                                            {
                                                                index !== 0 && <div
className={"icon-container"}>
                                                                    <OverlayTrigger
                                                                        placement="bottom"
                                                                        overlay={
                                                                            <Tooltip>Move stage one
step up</Tooltip>
                                                                        }
                                                                    >
                                                                        <button type="button"
onClick={() => moveStepUp(index)}
                                                                                className="btn
btn-circle btn-sm btn-warning">
                                                                            <ArrowBarUp></ArrowBarUp>
                                                                        </button>
                                                                    </OverlayTrigger>
                                                                </div>
                                                            }
                                                            {
                                                                index !== lenSteps - 1 && <div
className={"icon-container"}>
                                                                    <OverlayTrigger
                                                                        placement="bottom"
                                                                        overlay={
```

```
                                                                                <Tooltip>Move stage one
step down</Tooltip>
                                                                                }
                                                                            >
                                                                                <button type="button"
onClick={
                                                                                    () =>
moveStepDown(index)}
                                                                                        className="btn
btn-circle btn-sm btn-warning">

<ArrowBarDown></ArrowBarDown>
                                                                                </button>
                                                                            </OverlayTrigger>
                                                                        </div>
                                                                    }
                                                                </div>
                                                            </div>
                                                            {
                                                                index !== lenSteps - 1 && <div
className={"step-detail-footer col-12"}>
                                                                    <div
className={"connector-arrow-container container-all-center"}>
                                                                        <ArrowDown></ArrowDown>
                                                                    </div>
                                                                </div>
                                                            }
                                                        </div>
                                                    )
                                                })
                                            }
                                        </div>
                                    </div>
                                }
                            </div>
                        </div>
    );
}
```

```
export default RecruitmentProcess;
```

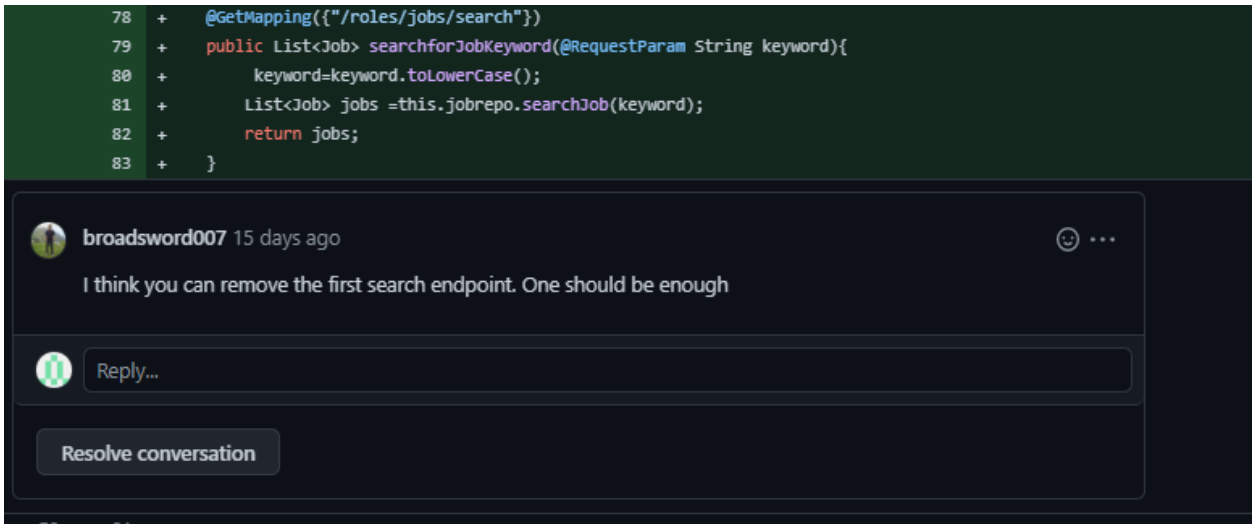
Internal Code Reviews

Description

The following snapshots are examples of the internal code review process within our team. Whenever a PR is created, our team lead spends time reviewing the files to make sure the code is satisfactory and able to be merged in the master branch.

Code Review Snapshots

1.

2. 

3.

```
16 + )
17 + public class hiring_process {
```

broadsword007 8 days ago

Please follow the naming conventions we are using across our codebase

Reply...

Resolve conversation

4.

...backend/hiring_guru_be/src/main/java/com/hiringguru/hiring_guru_be/models/HiringProcess.java

```
14 + @Table(
15 +         name = "hiring_process"
16 + )
17 + public class hiring_process {
```

broadsword007 8 days ago

Please follow the naming conventions we are using across our codebase

Reply...

Resolve conversation

5.

...nd/hiring_guru_be/src/main/java/com/hiringguru/hiring_guru_be/models/HiringProcessStage.java

Comment on lines +45 to +52

```
45 +     public Role(String title, String expectations, String benefits, Company company) {
46 +         this.id = this.id;
47 +         this.title = title;
48 +         this.type = type;
49 +         this.description = description;
50 +         this.hiring_process = hiring_process;
51 +
52 +     }
```

broadsword007 8 days ago

What is the purpose of adding this function?

Reply...

Resolve conversation

6.

Self-check on Best Practices for Security

Following are the major assets that we are protecting

1. Database
2. AWS account and Infrastructure
3. Code
4. Documentation
5. In-flight data

Following is the detail about the security of these items

Database

Our DB is provisioned in AWS. We have tried to ensure maximum security for our database as described below

1. We are using the latest version of Postgres to make sure we are using the stable and most secure build
2. All of our AWS users have read-only permission on the database. They cannot change critical settings on the database
3. We have enforced a max provision limit on the size of the DB. The DB size cannot increase by 20GB. So even if someone gets access to our DB the limit of damage is limited.
4. We are planning to limit DB access to our production server only.

AWS Account and Infrastructure

We are using AWS for all of our infrastructures. The security of our AWS account is very important. We have taken the following steps

1. Enabled two-factor authentication on the root AWS account
2. Limited the usage of the root account.
3. Created IAM users with minimum permission required, primarily with read-only access

4. Using AWS security groups as a line of defense
5. Using AWS-provided application load balancer to mitigate network attacks.

Code

Code is one of the most critical assets and we have taken steps to ensure it's secure

1. The code repository is private
2. Only the team members have access to repository
3. The master branch is protected against direct commits.
4. All PRs must get approved by the Team lead to get merged
5. Continuous integration pipeline that makes sure that the code builds correctly

Documentation

We are using Google for saving our documents. All of our Google docs are shared only amongst the team members over email. At the end of a milestone, the documents are converted to PDFs and submitted. A version of the documents is also kept in the code repo which is also secured.

In-flight data

We understand the importance of keeping the inflight data secure.

1. All of our inflight data is encrypted.
2. We provide support for TLS 1.1, 1.2, and 1.3 with strong encryption algorithms
3. Our front end is server over SSL
4. The backend is also server over SSL
5. We support strong encryption algorithms

The security report for our servers is attached below

Certificate #1: RSA 2048 bits (SHA384withRSA)

Server Key and Certificate #1

Subject	hiring-guru-backend-prod.us-west-1.elasticbeanstalk.com Fingerprint SHA256: 0216f5c846e1e9fa9d12affbc1dcc58a2238ed1df6ffa6509d8f835521a0ca50 Pin SHA256: Ilg2iXA2TbFzLG3U/ysrWwz5A6X+ByUmgnGUzuWn3Ls=
Common names	hiring-guru-backend-prod.us-west-1.elasticbeanstalk.com
Alternative names	hiring-guru-backend-prod.us-west-1.elasticbeanstalk.com
Serial Number	00b380926a474a8058dabef3f90294581c
Valid from	Thu, 10 Nov 2022 00:00:00 UTC
Valid until	Wed, 08 Feb 2023 23:59:59 UTC (expires in 2 months and 7 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	ZeroSSL RSA Domain Secure Site CA AIA: http://zerossl.crt.sectigo.com/ZeroSSLRSADomainSecureSiteCA.crt
Signature algorithm	SHA384withRSA
Extended Validation	No
Certificate Transparency	**Yes (certificate)**
OCSP Must Staple	No
Revocation information	OCSP OCSP: http://zerossl.ocsp.sectigo.com
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	**Yes** **Mozilla Apple Android Java Windows**

Configuration

Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites

 # TLS 1.2 (suites in server-preferred order)

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH secp256r1 (eq. 3072 bits RSA) FS		128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH secp256r1 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH secp256r1 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH secp256r1 (eq. 3072 bits RSA) FS		256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH secp256r1 (eq. 3072 bits RSA) FS	WEAK	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH secp256r1 (eq. 3072 bits RSA) FS	WEAK	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)	WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c)	WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	WEAK	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)	WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d)	WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)	WEAK	256

Protocol Details

DROWN	No, server keys and hostname not seen elsewhere with SSLv2 (1) For a better understanding of this test, please read this longer explanation (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Not mitigated server-side (more info) TLS 1.0: 0xc013
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Zombie POODLE	No (more info) TLS 1.2 : 0xc027
GOLDENDOODLE	No (more info) TLS 1.2 : 0xc027
OpenSSL 0-Length	No (more info) TLS 1.2 : 0xc027
Sleeping POODLE	No (more info) TLS 1.2 : 0xc027
Downgrade attack prevention	Yes, TLS_FALLBACK_SCSV supported (more info)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
Ticketbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
ROBOT (vulnerability)	No (more info)
Forward Secrecy	With modern browsers (more info)
ALPN	Yes h2 http/1.1
NPN	Yes h2 http/1.1
Session resumption (caching)	Yes
Session resumption (tickets)	Yes

OCSP stapling	No
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No (more info)
Public Key Pinning Report-Only	No
Public Key Pinning (Static)	No (more info)
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No, DHE suites not supported
DH public server param (Ys) reuse	No, DHE suites not supported
ECDH public server param reuse	No
Supported Named Groups	secp256r1, secp521r1, brainpoolP512r1, brainpoolP384r1, secp384r1, brainpoolP256r1, secp256k1, sect571r1, sect571k1, sect409k1, sect409r1, sect283k1, sect283r1 (server preferred order)
SSL 2 handshake compatibility	Yes

Self-check: Adherence to Original Non-functional Specs

1. The application shall have efficient response time **DONE**
2. The application shall have an easy to use platform **DONE**
3. The application shall have sufficient space for all data **DONE**
4. The application should have an option for extra space **DONE**
5. General User shall be able to create an account **DONE**
6. General User should be able to mark required regions in account registration **ON TRACK**
7. General User shall be able to add required information to account **DONE**
8. General User shall be able to add a resume to account **DONE**
9. General User shall be able to able to have other accounts linked to this account **ON TRACK**
10. General User should be able to able to save jobs to a folder **ON TRACK**
11. General User shall be able to able to search for jobs **DONE**
12. General User should be able to able to sort jobs by date applied **ON TRACK**
13. General User shall be able to able to view necessary information underneath job **DONE**
14. General User shall be able to able to add necessary information underneath job **DONE**
15. The application should have an easy to follow design **DONE**
16. The application should have a unique and Colorful Design **DONE**
17. The application should have no unnecessary information on the page **DONE**
18. The application shall have adequate Connection to Wi-Fi **DONE**
19. The application shall have see Wi-Fi connection status **DONE**
20. General User shall be able to see visibility status with other companies **ON TRACK**
21. General User shall be able to able to customize profile picture for account **DONE**
22. The application should have a career history to be added to account **DONE**
23. General User should be able to an academic history to be added to account **ON TRACK**
24. General User should be able to any achievements to be added to account **ON TRACK**
25. General User shall be able to able to add custom bio to account **DONE**
26. The application shall have strong security **DONE**
27. General User should be able to able to sign in using 2 Factor Authentication **DONE**

28. General User should be able to answer security questions in case of account lockout **ON TRACK**
29. General User shall be able to hide password when typing **DONE**
30. General User shall be able to view password when typing **DONE**
31. General User shall be able to update password **DONE**
32. General User shall be able to choose username that's not taken yet **DONE**
33. General User shall be able to choose email that's not taken yet **DONE**
34. General User should be able to choose a secure password with specific security requirements **DONE**
35. General User shall be able to define ideal location preferences **DONE**
36. General User shall be able to define ideal job preferences **DONE**
37. General User shall be able to recover lost account **DONE**
38. General User shall be able to notify users of any updates/changes **ON TRACK**
39. General User should be able to deliver reminders to users **ON TRACK**
40. General User should be able to customize reminder time **ON TRACK**
41. General User should be able to customize reminder date **ON TRACK**
42. General User shall be able to see all future schedules **ON TRACK**
43. General User shall be able to contact users easily **DONE**
44. General User should be able to make your account private or public **ON TRACK**
45. General User shall be able to see past completed interviews **ON TRACK**
46. General User should be able to see previous emails sent **ON TRACK**
47. General User should be able to see points of contact underneath jobs (if applicable) **DONE**
48. General User should be able to see the location of job **DONE**
49. General User shall be able to easily see what the job is requiring **DONE**
50. General User should be able to be informed if this job is a good match **DONE**
51. General User should be able to be informed if the job is a bad match **DONE**
52. General User shall be able to able to assign an evaluation score to a candidate from an interview **DONE**
53. General User should be able to know the evaluation score of an interview **ON TRACK**
54. General User should be able to know the name and job title of your interviewer **DONE**
55. General User shall be able to have the link sent to them for interview **ON TRACK**
56. General User shall be able to see what jobs they have previously viewed **ON TRACK**

57. General User shall be able to see what jobs they have already applied for **ON TRACK**
58. The application shall be should to able to indicate job preferences **DONE**
59. The application shall be should to have potential jobs sent to them daily **DONE**
60. The application shall be should to have potential jobs sent to them weekly **DONE**
61. The application shall be should to have no preferences sent at all **ON TRACK**
62. General User shall be able to request updates for job postings of a specific company **DONE**
63. General User shall be able to login easily into account **DONE**
64. General User shall be able to log out easily of account **DONE**
65. General User shall be able to option to have account sign in remembered **DONE**
66. General User shall be able to able to delete account **DONE**
67. General User shall be able to able to deactivate account **DONE**
68. General User should be able to able to receive notifications for scheduled interviews **ON TRACK**
69. General User should be able to customize notifications for scheduled interviews **ON TRACK**
70. General User should be able to choose between light and dark theme for application **ON TRACK**
71. General User shall be able to cancel interviews **ON TRACK**
72. General User should be able to request different interview times **ON TRACK**
73. General User should be able to see who will be attending the Interviews **ON TRACK**
74. General User should be able to RSVP for an Interview **ON TRACK**
75. General User shall be able to deny an Interview denial **ON TRACK**
76. General User shall be able to remind attendees for Interview 15 minutes prior **ON TRACK**
77. General User shall be able to gives users options to change reminder time **ON TRACK**
78. General User should be able to add description within scheduled interview **ON TRACK**
79. General User should be able to request to create any non-interview meetings for questions **ON TRACK**
80. Authorized User shall be able to to see which exact candidate is applying **DONE**
81. Authorized User shall be able to to view resume attached to candidate profile **DONE**
82. Authorized User should be able to contact candidate freely **DONE**

83. Authorized User shall be able to schedule meeting with candidate freely **DONE**
84. Authorized User shall be able to score candidate accordingly **DONE**
85. Authorized User shall be able to see score/interview notes freely **DONE**
86. Authorized User shall be able to see available contact information for candidate **DONE**
87. Authorized User should be able to customize profile based on employment within company **DONE**
88. Authorized User should be able to be verifiable for better perks **DONE**
89. Authorized User should be able to see other verified accounts **DONE**
90. General User should be able to follow other users on the site **DONE**
91. General User should be able to see other users who are on the site **DONE**
92. Authorized User shall be able to see who you follow and what jobs they have applied for **DONE**
93. The Application should be able to recommend usernames and passwords for registering users **ON TRACK**
94. The Application should be able to recommend passwords for users non-registering users **ON TRACK**
95. The Application should be able to recommend usernames for users non-registering users **ON TRACK**
96. The Application should be able to have sufficient space for all users on platform **DONE**
97. The Application should be able to have a limit for resume size **DONE**
98. The Application should be able to have a preferences in how resume should be submitted **DONE**
99. The Application should be able to ask user if they are a robot **DONE**
100. The Application shall be able to deactivate user if they are a robot **DONE**

Team Contributions

Contributions

Farhan Haider

- Implemented the Create, Read, Update, Delete functionality for Employee
- Helped Kenny fix the manage jobs page
- Did internal code reviews - reviewed PRs from team members
- Did code review for Team 3
- Worked on Self-Check on Best practices for Security
- Created the basic skeleton for Usability tests and completed the first usability test
- Finalized the non-functional requirements for QA test
- Helped Khushi and Eric with the backend work
- Organized team meetings and planned the overall work

Kenny Leong

- Frontend implementation of backend APIs
    - Employees
    - Jobs
    - Job Roles
- UI Bug Fixes/UI improvements

Khushi Khanna

- Worked on the Backend
    - Created Company,Role, Job Models and CRUDs
    - Added a Job Search API
- Helped with the document
    - Sections 1&2

Mamadou Bah

- Worked on the backend
    - Created Role,HiringProcess, and HiringProcessStage Models
    - Helped with Job Application CRUD
- Helped with the document
    - Section 1-3
    - Organization and formatting

Eric Leow

- Backend
    - ➢ Helped Create models Role, HiringProcess and HiringProcessStage
    - ➢ Hiring Pipeline CRUD
    - ➢ Created models Job, JobApplication and Referral

        - Tested Job, Role and Company API. To see if any fixes were required.
- Wireframe
    - Login and Signup Page
    - Apply for a job
    - Job Referral

Scores

- Kenny Leong - 8/10
- Mamadou Bah - 6/10
- Eric Leow - 6.5/10
- Khushi Khanna - 6.5/10

SW Engineering CSC 648/848 Fall 2022

The Hiring Guru

Team 2: Binary Brains
Farhan Haider (Team Lead/Backend Lead): shaider1@sfsu.edu
Kenny Leong (Frontend Lead/GitHub Master): kleong2@mail.sfsu.edu
Mamadou Bah (Frontend): mbah1@mail.sfsu.edu
Khushi Khanna (Backend) kkhanna@sfsu.edu
Eric Leow (Backend): elow@mail.sfsu.edu

Milestone #3

November 9th, 2022

Row	Document Version	Review Date	Description
1	Version 1	10/21/2022	Milestone 3 was assigned, met up and discussed how we want to complete the Milestone. Made assignments to be completed prior to class on Thurs
2	Version 2	10/25/2022	Completed data definitions, partially prioritized the functional requirements
3	Version 3	11/02/2022	Completed EER and UML diagrams
4	Version 4	11/07/2022	Added wireframes

5	Version 5	11/09/2022	Finalized document for submission

Table of Contents

1 - Data Definitions…….……4

2 - Prioritized Functional Requirements V3……..…….....8

3 - Wireframes Based on your Mockups/Storyboards……16

4 - High Level Database Architecture and Organization V2……..…....17

5 - High Level UML Diagrams V2 19

6 - Detailed list of Contributions.....…….....20

1 - Data Definitions

HR Manager

An HR Manager does the following:

- An HR manager shall be able to be the main actor and the most frequent user of this application, responsible for managing the complete hiring process.
- A HR Manager shall be able to schedule the various steps of the application process
- A HR Manager shall be able to reach out to Candidates throughout the application process
- A HR Manager shall be able to move the Candidate along the recruitment pipeline given positive feedback from Interviewer
- A HR Manager shall be able to be able to engage with communication channels
- A HR Manager shall be able to view hiring insights to find the most suitable Candidate
- A HR Manager shall be responsible for ensuring the onboarding process is completed by the Candidate

Candidate

A Candidate does the following:

- The Candidate is a prominent user within the application, responsible for going through the entire Recruitment Pipeline Process from start to finish
    - Involves being aware of open positions and requirements of each position
    - Involves completing each step of the recruitment process on time
    - Involves remaining in communication during the recruitment process with various other users

Executive

An Executive does the following:

- The Executive entity is an entity that has the same influence as the HR Manager but is one above the HR Manager.
- The Executive shall be able to do all of the tasks that the HR Manager can, such as:
    - An Executive is able/responsible for scheduling the various steps of the application process
    - An Executive is able to reach out to Candidates throughout the application process
    - An Executive moves the Candidate along the recruitment pipeline given positive feedback from Interviewer
    - An Executive is able to be able to engage with communication channels

    - An Executive is able to view hiring insights to find the most suitable Candidate
    - An Executive is responsible for ensuring the onboarding process is completed by the Candidate
- An Executive includes creating referral programs and overseeing the entire recruitment process and managing the progress of the other users within the recruitment pipeline

Tech Recruiter/Marketing

A Tech Recruiter does the following:

- A Tech Recruiter, or the Marketing Team, is the main user that starts the entire Recruitment Pipeline Process within the Hiring Guru, responsible for hiring the Candidate that will undergo the process.
- A Tech Recruiter shall be able to promote the job on various platforms and figure out what is the best platform and marketing strategy to target the appropriate group of Candidates
- A Tech Recruiter shall be responsible for presenting suitable Candidates needed for the job description

Interviewer

An Interviewer does the following:

- An Interviewer is a user who is an employee within the desired company by the general user and interviews them to see if they qualify for the specific role.
- An Interviewer shall be responsible for scoring each of the Candidate's based on the scoresheet provided by the company
- As an Interviewer, it's crucial to understand the skills and experience that are necessary for the job description

Job Application

A Job Application does the following:

- A Job Application shall be created by an Employee
- A Job Application shall be filled out by a Candidate
- A Job Application shall contain profile information of many candidates
- A Job Application shall be able to be modified by an Employee

Employee

An Employee does the following:

- An Employee shall be able to view candidates in the Recruitment Pipelines Process

- An Employee shall be able to interview Candidates
- An employee shall be able to schedule interview of candidates
- An employee shall be able to setup Permissions and Groups
- An authorized employee shall be able to shortlist Applications
- An authorized employee setup Recruitment Pipelines Process
- An authorized employee shall be creating many Job positions
- An authorized employee shall be able to create many Job roles

General User

A General User can do the following:

- A General User is a user who is only exploring the application and has the most basic features accessible to them, most often an unauthenticated user.

Group

A Group can do the following:

- A group can have many employees
- A group can be assigned multiple permissions

Permission

A Permission can do the following:

- A Permission can be assigned to multiple Groups

2 - Prioritized Functional Requirements V2

P1

Authorized User

Requirement ID	Requirement Statement
FRAUTH001	Authorized user shall be able to log out anytime
FRAUTH002	Authorized user shall be able to create an organization
FRAUTH004	Authorized user shall be able to invite employees to the organization
FRAUTH008	Authorized Users shall be able to create a new role (e.g., QA Engineer, Software Engineer, etc.)
FRAUTH009	Authorized User shall be able to update a role
FRAUTH010	Authorized User should be able to see the list of all roles
FRAUTH011	Authorized User shall be able to remove a role from the system
FRAUTH012	Authorized User shall be able to open a new position for a particular role
FRAUTH013	Authorized User shall be able to close a listed position
FRAUTH014	Authorized User shall be able to see all the applications for a specific position
FRAUTH015	Authorized User shall be able to add referral for a specific position
FRAUTH016	Authorized User shall be able to see all the referrals for a specific position
FRAUTH017	Authorized User shall be able to drop an application
FRAUTH018	Authorized User shall be able to design the hiring process for a position
FRAUTH019	Authorized User shall be able to update the hiring process for a position
FRAUTH020	Authorized User shall be able to see the hiring process for a position
FRAUTH003	Authorized User shall be able to move a stage backward in the hiring process
FRAUTH005	Authorized User shall be able to move a stage forward in the hiring process
FRAUTH006	Authorized User shall be shown a confirmation before deleting a step

FRAUTH007	Authorized User shall be able to easily move to another page using a side navigation
FRAUTH021	Authorized User shall be able to search for job roles using a query field
FRAUTH054	Authorized User shall be able to write a job description

General User

Requirement ID	Requirement Statement
FRGENR000	General user shall be able to sign up directly for an individual account
FRGENR001	General user shall be able to signup using Google account
FRGENR003	General user shall be able to login using Google account
FRGENR005	General user shall be able to login using Username and Password
FRGENR006	General user shall be able recover his/her account using email
FRGENR007	General User shall be able to go back to the home screen anytime by clicking on the HiringGuru logo
FRGENR009	General User shall be able to see the HiringGuru about page in the page footer
FRGENR010	General User shall be able to see the product features in the home page
FRGENR011	General User shall be able to search for jobs on the home page
FRGENR012	General User shall be able to apply for a job

P2

Authorized User

Requirement ID	Requirement Statement
FRAUTH062	System should automatically add an accepted application to the hiring process
FRAUTH022	Authorized User shall be able to see a list of all the accepted applications
FRAUTH023	Authorized User shall be able to remove a Candidate from the hiring process

FRAUTH024	Authorized User shall be able to add an Employee
FRAUTH025	Authorized User shall be able to view a list of all Employees
FRAUTH026	Authorized User shall be able to remove an Employee from the company
FRAUTH027	Authorized User shall be able to schedule an interview of the Candidate
FRAUTH028	Authorized User shall be able to send a calendar invite to the Interviewers and the Candidate
FRAUTH029	Authorized User shall be able to see the schedule detail of an interview of the candidate
FRAUTH030	Authorized User shall be able to see the detail of an interview of the candidate
FRAUTH031	Authorized User shall be able to schedule a programming test of the candidate
FRAUTH032	Authorized User shall be able to update the time and date of a programming test
FRAUTH033	Authorized User shall be able to add the feedback/decision of a programming test
FRAUTH034	Authorized User shall be able to attempt a programming test
FRAUTH035	Authorized User should be able to create a job post on LinkedIn
FRAUTH036	Authorized User should be able to create a new advertising campaign on LinkedIn
FRAUTH037	Authorized User should be able to analyze LinkedIn profiles using Artificial Intelligence
FRAUTH038	Authorized User shall be able to send a message to the matching profiles to see if they are interested
FRAUTH039	Authorized User shall be able to add profiles of interest to the recruitment pipeline
FRAUTH030	Authorized User shall be able to like a listed position
FRAUTH041	Authorized User shall be able to update his profile

FRAUTH042	Authorized User shall be able to search for job by the posted time
FRAUTH043	Authorized User shall be able to post the job advertisement on LinkedIn
FRAUTH044	Authorized User shall be able to schedule a programming test for the candidate
FRAUTH045	Authorized User shall be able to schedule an interview of the candidate
FRAUTH046	The system shall send email to the candidate once programming test is scheduled
FRAUTH047	The system shall send email to the candidate once an interview is scheduled
FRAUTH048	Authorized User shall be able to send acceptance email to the candidate
FRAUTH049	Authorized User shall be able to send rejection email to the candidate
FRAUTH050	Authorized User shall be able to send a background check must to a candidate, which must be passed before being able to accept a job offer
FRAUTH051	Authorized User shall be able to review each applicant’s resume and experience prior to interviewing
FRAUTH052	Authorized User shall be able to determine how to score candidate's responses based off of company’s expectations
FRAUTH052	Authorized User shall be able to answer Candidate's questions appropriately
FRAUTH053	Authorized User shall be able to administer situational and personality tests to narrow the list of Candidates
FRAUTH055	Authorized User shall be able to ensure that the employment paperwork is correct and complete
FRAUTH056	Authorized User shall be able to ensure that the application should be easy to complete
FRAUTH057	Authorized User shall be able to schedule a phone screen
FRAUTH058	Authorized User shall be able to schedule a skills test

FRAUTH059	Authorized User shall be able to ensure appropriate questions are on the skills test related to the job requirement
FRAUTH060	Authorized User shall be able to schedule a new time for the skills test if Candidate is unable to complete it within the provided time frame
FRAUTH061	Authorized User shall be able to allow for convenient and quick channels of communication for the candidate to ask questions throughout the interview process
FRAUTH119	Authorized User shall be able to assign role to candidate
FRAUTH120	The application should be able to provide job-specific programs for each candidate
FRAUTH121	The application should be able to grant access to technical tools needed by the candidate
FRAUTH122	The application should be able to have a paid or unpaid boarding process
FRAUTH123	The application should be able to allocate mentor for Candidate
FRAUTH124	The application should be able to create a schedule for the Candidate
FRAUTH125	The application should be able to organize the workspace
FRAUTH126	Introduce the candidate to technologies they will need to learn how to use
FRAUTH127	Authorized User shall be able to prepare orientation with other Candidates
FRAUTH128	The application shall have an administration tools for the interviewer
FRAUTH129	Authorized User shall be able to Take in any comments or complaints that the candidate might have
FRAUTH130	Authorized User shall be able to Organize meetings between the employer and the candidates
FRAUTH131	Manage any health benefits that should be given to the employee

FRAUTH132	Authorized User shall be able to request Employment background screening
FRAUTH133	Authorized User shall be able to Send offer letter to candidate
FRAUTH134	Authorized User shall be able to Email reminders of the next steps in onboarding

General User

Requirement ID	Requirement Statement
FRGENR013	General user shall be able to book a product demo
FRGENR014	General user shall be enable or disable cookies
FRGENR015	General user shall be able to chat with the organization bot
FRGENR002	General user shall be able to signup using a Microsoft account
FRGENR016	General user shall be able to login using Microsoft account

P3

Authorized User

Requirement ID	Requirement Statement
FRAUTH094	System should automatically remove the Candidate from the hiring process if a step is failed
FRAUTH095	Authorized User shall be able to view the detail of an Employee
FRAUTH096	Authorized User shall be able to update the detail of an Employee
FRAUTH097	Authorized User shall be able to allow an interviewer to set their availability time
FRAUTH098	Authorized User shall be able request Candidate availability for an interview
FRAUTH099	System should automatically schedule an interview of the Candidate

FRAUTH100	System should automatically schedule a programming test if it is the current step
FRAUTH101	Authorized User should be able to create a job post on Stackoverflow
FRAUTH102	Authorized User should be able to create a new advertising campaign on YouTube
FRAUTH103	Authorized User should be able to create a new advertising campaign on StackOverflow
FRAUTH104	Authorized User should be able to calculate the budget and allocate money for specific advertising campaigns
FRAUTH105	Authorized User shall be able to calculate budget to see how many open positions are possible to create
FRAUTH106	Authorized User should be able to calculate the budget to allocate for training interviewers on how to conduct an interview
FRAUTH107	Authorized User shall be able to talk about the day-to-day duties and inform the candidate about the work culture
FRAUTH108	Authorized User shall be able to identify if the hire is an external or internal applicant
FRAUTH109	Authorized User shall be able to determine if an internal hire- minimal advertising is needed
FRAUTH110	Authorized User shall be able to design a recruitment pipeline for individual candidates
FRAUTH111	Authorized User shall be able to provide specific feedback as to why they didn’t pass that round
FRAUTH135	Authorized User shall be able to prepare new Candidate for training
FRAUTH136	Authorized User shall be able to create a new onboarding program
FRAUTH137	The application should be able to have a calculate candidate hours
FRAUTH138	The application should be able to have a decrease wage

FRAUTH139	The application should be able to have a give raise
FRAUTH140	The application should be able to have a calculate trust level
FRAUTH141	The application should be able to have a survey candidate
FRAUTH142	The application should be able to have a find jobs for the candidate within company
FRAUTH143	The application should be able to have a create presentations
FRAUTH144	The application should be able to host meetings
FRAUTH145	The application should be able to have a way to communicate with other data items
FRAUTH146	Authorized User shall be able to create tasks for candidate
FRAUTH147	The application should be able to have a deadline setter
FRAUTH148	Authorized User shall be able to set goals for the candidate to complete
FRAUTH149	Authorized User shall be able to find candidate function
FRAUTH140	Authorized User shall be able to pause candidate function
FRAUTH141	Authorized User shall be able to remove candidate function
FRAUTH142	The application should be able to have memory allocation for Candidate information
FRAUTH143	The application should be able to organize team dynamics
FRAUTH144	The application should be able to organize team dynamics
FRAUTH145	The application should be able to organize where and how they will be working
FRAUTH146	Authorized User shall be able to do a performance review scheduling
FRAUTH147	Manage space for Candidates
FRAUTH148	Authorized User shall be able to display information needed by the candidate

General User

Requirement ID	Requirement Statement
FRGENR016	General user shall be able to see product pricing
FRGENR017	General User shall be able to see statistics of a job post

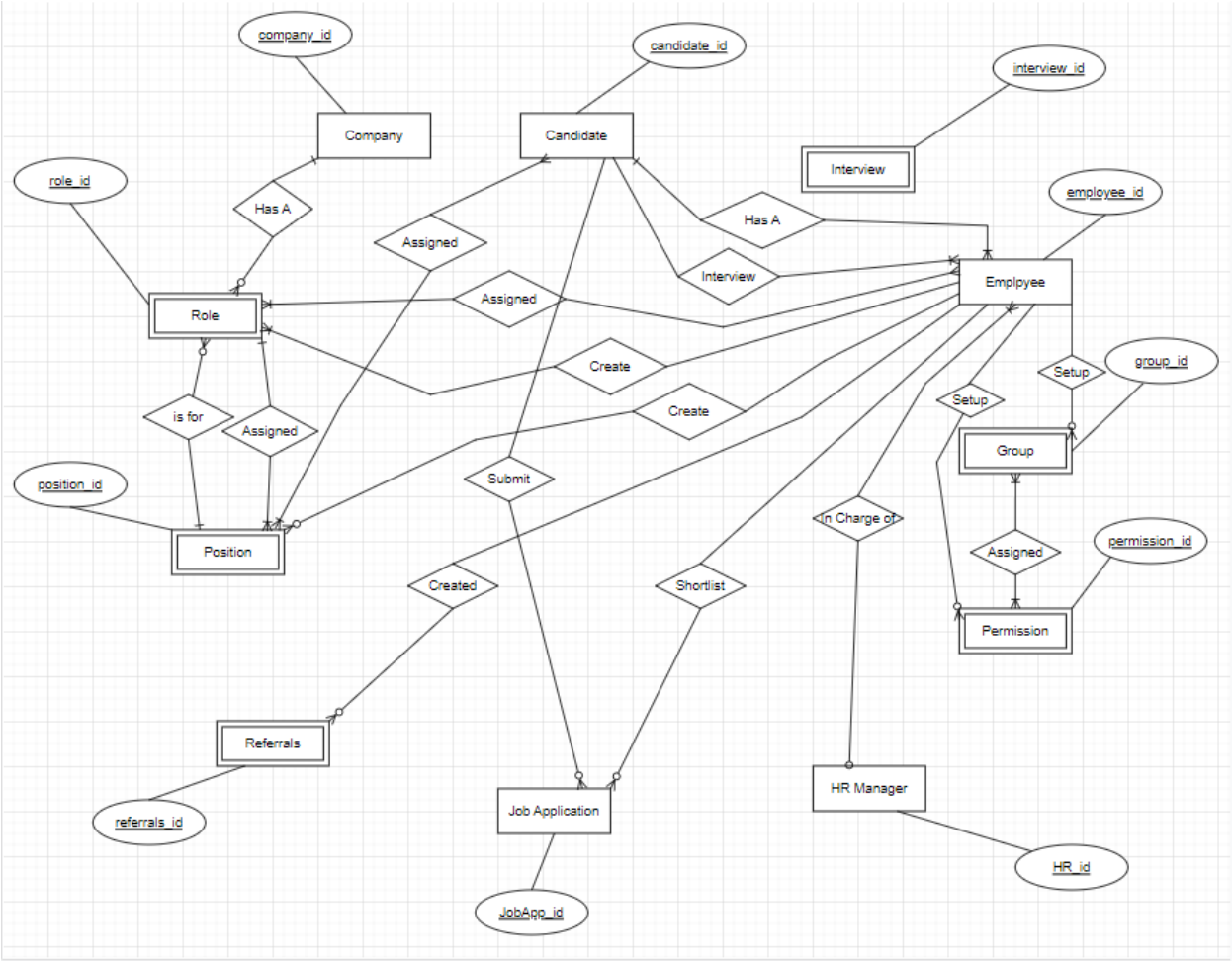
3 - Wireframes based on your Mockups/Storyboards

Please use the following Figma Link to go through the wireframes.

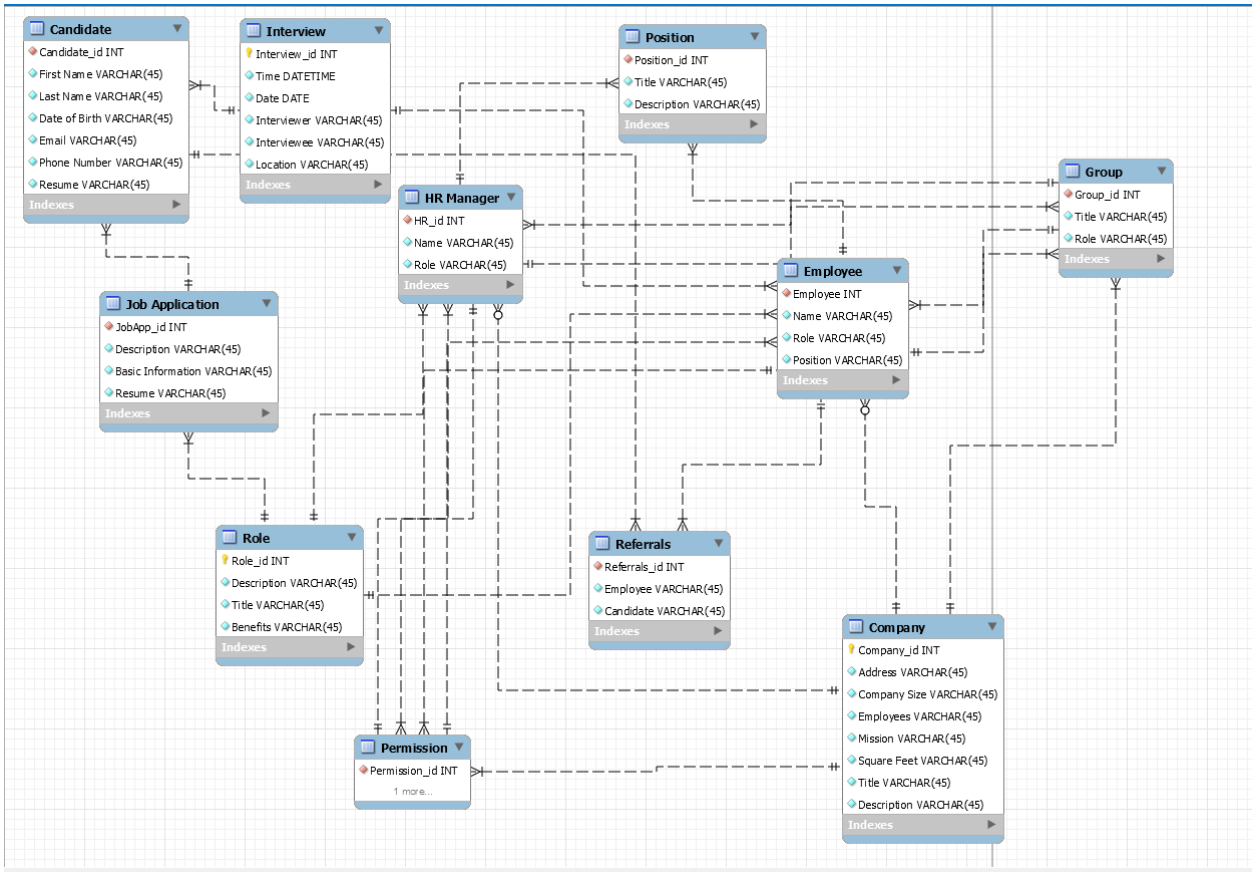
https://www.figma.com/file/rHSzFAi6KRzUgaVS3QQsUq/Hiring-Guru-M3-Wireframe-Diagrams?node-id=0%3A1

4 - High Level Database Architecture and Organization

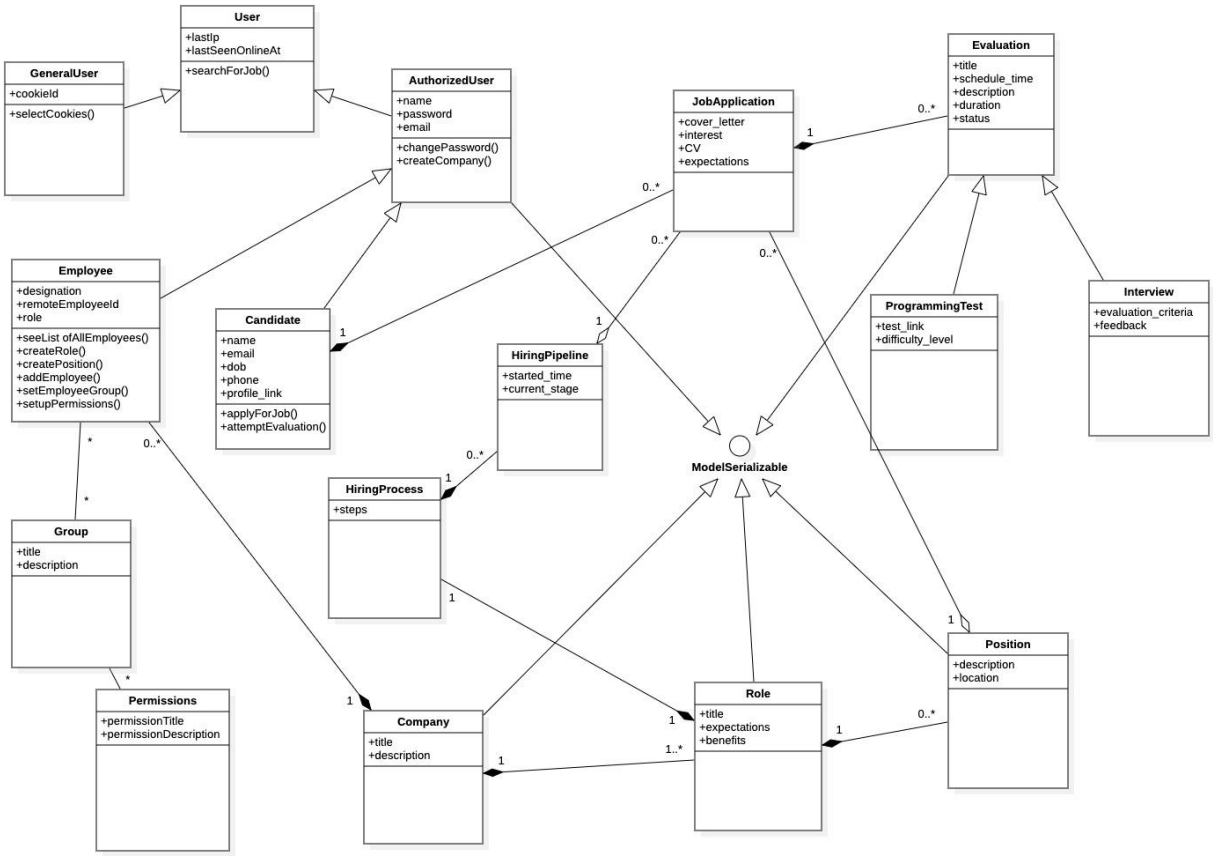
Entity Relationship Diagram (ERD)



Enhanced Entity Relationship (EER)



5 - High Level UML Diagram V2



6 - Detailed list of Contributions

- Contributions:
  - Kenny Leong
    - Dashboard
    - Manage applications
    - Apply for job
    - Wireframe: Home/Landing page, Pages interfacing with Dashboard
    - Fixed M2 feedback on Vertical Prototype
  - Mamadou Bah
    - Job referral
    - Create job
    - Company creation
    - ERR
  - Khushi Khanna
    - Wireframes: Dashboard, Apply for a Job, Recruitment Process, Manage Job Applications
    - Fixed M2 feedback on High Level APIs and Algorithm
    - CRUD Model for Company, Role, and Job
  - Eric Leow
    - Wireframes: Login and Signup Page
    - Wireframe: Job Roles
    - Wireframe: Create Job/Position
    - Wireframe: Job Referral
  - Farhan Haider
    - Setup authentication
    - Setup recruitment process
    - Manage roles
    - Helped other team members in their work
    - Planned and organized the work
    - Worked on M2 feedback
- Score (Out of 10 - 10 being ideal)
  - Kenny Leong - 8
  - Mamadou Bah - 6
  - Khushi Khanna - 6

- Eric Leow - 4

SW Engineering CSC 648/848 Fall 2022

The Hiring Guru

Team 2: Binary Brains
Farhan Haider (Team Lead/Backend Lead): shaider1@sfsu.edu
Kenny Leong (Frontend Lead/GitHub Master): kleong2@mail.sfsu.edu
Mamadou Bah (Frontend): mbah1@mail.sfsu.edu
Khushi Khanna (Backend) kkhanna@sfsu.edu
Eric Leow (Frontend): elow@mail.sfsu.edu

Milestone #2

October 20th, 2022

Row	Document Version	Review Date	Description
1	Version 1	9/22/2022	Milestone 2 was assigned, Document was first created with sections to be filled out
2	Version 2	9/26/2022	Went over Milestone 1 feedback and assigned work to revise it to gain filled credit.
3	Version 3	9/28/2022	Went through Milestone 2 together and assigned less high maintenance tasks to different team members to have done.
4	Version 4	10/6/2022	Received Constructive Feedback from the professor and planned out how we were going to further improve our progress
5	Version 5	10/8/2022	Met up for a quick meeting to work on Trello tasks given to us

6	Version 6	10/8/2022	Had tasks assigned to be due 10/14/2022 for the home page, involving both Frontend and Backend Team
7	Version 7	10/15/2022	Individually met with team lead for designated updates and future assignments to complete Milestone.
8	Version 8	10/17/2022	Made sure all team members had completed prior assignments and split up any necessary work that was left for the document
9	Version 9 Final Version	10/19/2022	Had our final M2 Meeting with a check-in and final touches for M2. Spend the evening wrapping everything up
10	Version 10 - V2 Updates	11/09/2022	Updates for M3V2

Table of Contents

1 - Data Definitions V3…….……4

2 - Prioritized Functional Requirements V3…….....7

3 - Use Cases and Storyboard……..…….……18

4 - High Level Database Architecture and Organization…….....33

5 - High Level APIs and Main Algorithm.....37

6 - High Level UML Diagrams 39

7 - High Level Application Network and Deployment Diagrams.....40

8 - Identifying *Actual* Key Risks.....42

9 - Project Management.....…….....44

10 - Detailed list of Contributions.....…….....45

1 - Data Definitions

HR Manager

An HR Manager does the following:

- An HR manager shall be able to be the main actor and the most frequent user of this application, responsible for managing the complete hiring process.
- A HR Manager shall be able to schedule the various steps of the application process
- A HR Manager shall be able to reach out to Candidates throughout the application process
- A HR Manager shall be able to move the Candidate along the recruitment pipeline given positive feedback from Interviewer
- A HR Manager shall be able to be able to engage with communication channels
- A HR Manager shall be able to view hiring insights to find the most suitable Candidate
- A HR Manager shall be responsible for ensuring the onboarding process is completed by the Candidate

Candidate

A Candidate does the following:

- The Candidate is a prominent user within the application, responsible for going through the entire Recruitment Pipeline Process from start to finish
    - Involves being aware of open positions and requirements of each position
    - Involves completing each step of the recruitment process on time
    - Involves remaining in communication during the recruitment process with various other users

Executive

An Executive does the following:

- The Executive entity is an entity that has the same influence as the HR Manager but is one above the HR Manager.
- The Executive shall be able to do all of the tasks that the HR Manager can, such as:
    - An Executive is able/responsible for scheduling the various steps of the application process
    - An Executive is able to reach out to Candidates throughout the application process
    - An Executive moves the Candidate along the recruitment pipeline given positive feedback from Interviewer
    - An Executive is able to be able to engage with communication channels

    - An Executive is able to view hiring insights to find the most suitable Candidate
    - An Executive is responsible for ensuring the onboarding process is completed by the Candidate
- An Executive includes creating referral programs and overseeing the entire recruitment process and managing the progress of the other users within the recruitment pipeline

Tech Recruiter/Marketing

A Tech Recruiter does the following:

- A Tech Recruiter, or the Marketing Team, is the main user that starts the entire Recruitment Pipeline Process within the Hiring Guru, responsible for hiring the Candidate that will undergo the process.
- A Tech Recruiter shall be able to promote the job on various platforms and figure out what is the best platform and marketing strategy to target the appropriate group of Candidates
- A Tech Recruiter shall be responsible for presenting suitable Candidates needed for the job description

Interviewer

An Interviewer does the following:

- An Interviewer is a user who is an employee within the desired company by the general user and interviews them to see if they qualify for the specific role.
- An Interviewer shall be responsible for scoring each of the Candidate's based on the scoresheet provided by the company
- As an Interviewer, it's crucial to understand the skills and experience that are necessary for the job description

Job Application

A Job Application does the following:

- A Job Application shall be created by an Employee
- A Job Application shall be filled out by a Candidate
- A Job Application shall contain profile information of many candidates
- A Job Application shall be able to be modified by an Employee

Employee

An Employee does the following:

- An Employee shall be able to view candidates in the Recruitment Pipelines Process

- An Employee shall be able to interview Candidates
- An employee shall be able to schedule interview of candidates
- An employee shall be able to setup Permissions and Groups
- An authorized employee shall be able to shortlist Applications
- An authorized employee setup Recruitment Pipelines Process
- An authorized employee shall be creating many Job positions
- An authorized employee shall be able to create many Job roles

General User

A General User can do the following:

- A General User is a user who is only exploring the application and has the most basic features accessible to them, most often an unauthenticated user.

Group

A Group can do the following:

- A group can have many employees
- A group can be assigned multiple permissions

Permission

A Permission can do the following:

- A Permission can be assigned to multiple Groups

2 - Prioritized Functional Requirements

P1

Authorized User

Requirement ID	Requirement Statement
FRAUTH001	Authorized user shall be able to log out anytime
FRAUTH002	Authorized user shall be able to create an organization
FRAUTH003	Authorized user shall be able to modify organization detail
FRAUTH004	Authorized user shall be able to invite employees to the organization
FRAUTH005	Authorized user shall be able to create permission groups
FRAUTH006	Authorized user shall be able to add employees to user groups
FRAUTH007	Authorized user shall be able to remove employees from user groups
FRAUTH008	Authorized Users shall be able to create a new role (e.g., QA Engineer, Software Engineer, etc.)
FRAUTH009	Authorized User shall be able to update a role
FRAUTH010	Authorized User should be able to see the list of all roles
FRAUTH011	Authorized User shall be able to remove a role from the system
FRAUTH012	Authorized User shall be able to open a new position for a particular role
FRAUTH013	Authorized User shall be able to close a listed position
FRAUTH014	Authorized User shall be able to see all the applications for a specific position
FRAUTH015	Authorized User shall be able to add referral for a specific position
FRAUTH016	Authorized User shall be able to see all the referrals for a specific position
FRAUTH017	Authorized User shall be able to drop an application
FRAUTH018	Authorized User shall be able to design the hiring process for a position
FRAUTH019	Authorized User shall be able to update the hiring process for a position

FRAUTH020	Authorized User shall be able to see the hiring process for a position
FRAUTH021	Authorized User shall be able to accept an application for a specific position
FRAUTH022	Authorized User shall be able to see a list of all the accepted applications
FRAUTH023	Authorized User shall be able to remove a Candidate from the hiring process
FRAUTH024	Authorized User shall be able to add an Employee
FRAUTH025	Authorized User shall be able to view a list of all Employees
FRAUTH026	Authorized User shall be able to remove an Employee from the company
FRAUTH027	Authorized User shall be able to schedule an interview of the Candidate
FRAUTH028	Authorized User shall be able to send a calendar invite to the Interviewers and the Candidate
FRAUTH029	Authorized User shall be able to see the schedule detail of an interview of the candidate
FRAUTH030	Authorized User shall be able to see the detail of an interview of the candidate
FRAUTH031	Authorized User shall be able to schedule a programming test of the candidate
FRAUTH032	Authorized User shall be able to update the time and date of a programming test
FRAUTH033	Authorized User shall be able to add the feedback/decision of a programming test
FRAUTH034	Authorized User shall be able to attempt a programming test
FRAUTH035	Authorized User should be able to create a job post on LinkedIn
FRAUTH036	Authorized User should be able to create a new advertising campaign on LinkedIn
FRAUTH037	Authorized User should be able to analyze LinkedIn profiles using Artificial Intelligence

FRAUTH038	Authorized User shall be able to send a message to the matching profiles to see if they are interested
FRAUTH039	Authorized User shall be able to add profiles of interest to the recruitment pipeline
FRAUTH030	Authorized User shall be able to like a listed position
FRAUTH041	Authorized User shall be able to update his profile
FRAUTH042	Authorized User shall be able to search for job by the posted time
FRAUTH043	Authorized User shall be able to post the job advertisement on LinkedIn
FRAUTH044	Authorized User shall be able to schedule a programming test for the candidate
FRAUTH045	Authorized User shall be able to schedule an interview of the candidate
FRAUTH046	The system shall send email to the candidate once programming test is scheduled
FRAUTH047	The system shall send email to the candidate once an interview is scheduled
FRAUTH048	Authorized User shall be able to send acceptance email to the candidate
FRAUTH049	Authorized User shall be able to send rejection email to the candidate
FRAUTH050	Authorized User shall be able to send a background check must to a candidate, which must be passed before being able to accept a job offer
FRAUTH051	Authorized User shall be able to review each applicant’s resume and experience prior to interviewing
FRAUTH052	Authorized User shall be able to determine how to score candidate's responses based off of company’s expectations
FRAUTH052	Authorized User shall be able to answer Candidate's questions appropriately
FRAUTH053	Authorized User shall be able to administer situational and personality tests to narrow the list of Candidates
FRAUTH054	Authorized User shall be able to write a job description

FRAUTH055	Authorized User shall be able to ensure that the employment paperwork is correct and complete
FRAUTH056	Authorized User shall be able to ensure that the application should be easy to complete
FRAUTH057	Authorized User shall be able to schedule a phone screen
FRAUTH058	Authorized User shall be able to schedule a skills test
FRAUTH059	Authorized User shall be able to ensure appropriate questions are on the skills test related to the job requirement
FRAUTH060	Authorized User shall be able to schedule a new time for the skills test if Candidate is unable to complete it within the provided time frame
FRAUTH061	Authorized User shall be able to allow for convenient and quick channels of communication for the candidate to ask questions throughout the interview process

General User

Requirement ID	Requirement Statement
FRGENR000	General user shall be able to sign up directly for an individual account
FRGENR001	General user shall be able to signup using Google account
FRGENR002	General user shall be able to signup using a Microsoft account
FRGENR003	General user shall be able to login using Google account
FRGENR004	General user shall be able to login using Microsoft account
FRGENR005	General user shall be able to login using Username and Password
FRGENR006	General user shall be able recover his/her account using email
FRGENR007	General User shall be able to go back to the home screen anytime by clicking on the HiringGuru logo
FRGENR008	Authorized User shall be able to like a listed position
FRGENR009	General User shall be able to see the HiringGuru about page in the page footer

FRGENR010	General User shall be able to see the product features in the home page
FRGENR011	General User shall be able to search for jobs on the home page
FRGENR012	General User shall be able to apply for a job

P2

Authorized User

Requirement ID	Requirement Statement
FRAUTH062	System should automatically add an accepted application to the hiring process
FRAUTH063	System should automatically move an accepted application to the next step when a step is successful
FRAUTH064	System shall automatically notify the Candidate about the success/failure in a step
FRAUTH065	Authorized User shall be able to mark an Employee as an Interviewer
FRAUTH066	Authorized User shall be able to see a list of all Interviewers
FRAUTH067	Authorized User shall be able to remove an Employee from the Interviewer status
FRAUTH068	System should automatically send a calendar invite to the Interviewers and the Candidate
FRAUTH069	Authorized User shall be able to update the details of an interview - date, interviewers etc.
FRAUTH070	Authorized User shall be able to see the results of the programming test of a Candidate
FRAUTH071	Authorized User should be able to create a job post on Handshake
FRAUTH072	Authorized User should be able to update job posts created using the System
FRAUTH073	Authorized User should be able to pause an advertising campaign

FRAUTH074	Authorized User should be able to see statistics of an advertising campaign
FRAUTH075	Authorized User should be able to scrape profiles from LinkedIn given a set of keywords
FRAUTH076	Authorized User shall be able to save the list of matching profiles
FRAUTH077	Authorized User shall be able to ensure the Interviewer understands what skills and experience the job require
FRAUTH078	Authorized User shall be able to ensure the Interviewer has the proper skills to take the interview
FRAUTH079	Authorized User shall be able to create relevant application questions
FRAUTH080	Authorized User shall be able to identify the hiring need
FRAUTH081	Authorized User shall be able to determine if an external hire- go through usual advertising channels
FRAUTH082	Authorized User shall be able to perform a reference check
FRAUTH083	Authorized User shall be able to create a checklist of all of the paperwork related to employment the Candidate needs to fill out after accepting an offer
FRAUTH084	Authorized User shall be able to give Candidate ample time to complete the employment paperwork
FRAUTH085	Authorized User shall be able to, if no candidates meet the hiring criteria, decide whether to restart the hiring process
FRAUTH086	Authorized User shall be able to ask the candidate about their salary expectations
FRAUTH087	Authorized User shall be able to, if they wish to restart the hiring process, decide how to adjust the hiring criteria to yield more candidates
FRAUTH088	Authorized User shall be able to, if they wish to restart the hiring process, decide how to advertise the position to more favorable candidates

FRAUTH089	Authorized User shall be able to, if the candidate is ready to accept the job offer, decide which elements of the offer letter are negotiable
FRAUTH090	Authorized User shall be able to ensure that Artificial Intelligence is able to parse through the candidate's resume to fill out fields in the application
FRAUTH091	Authorized User shall be able to give a certain time frame for the candidate to accept the offer letter
FRAUTH092	Authorized User shall be able to create convenient methods for applicants to be able to withdraw from the application process at any time
FRAUTH093	Authorized User shall be able to provide detailed feedback on what they liked about the candidate's performance but also reservations they might have
FRAUTH112	Authorized User shall be able to create activities for the Candidate to do
FRAUTH113	The application should be able to have a total cost calculator
FRAUTH114	The application should be able to have a digital document process
FRAUTH115	The application should be able to have a workload manager
FRAUTH116	The application should be able to have a candidate portal
FRAUTH117	Screening function for making sure the candidate is cleared
FRAUTH118	Authorized User shall be able to create a database for tracking candidate’s progress
FRAUTH119	Authorized User shall be able to assign role to candidate
FRAUTH120	The application should be able to provide job-specific programs for each candidate
FRAUTH121	The application should be able to grant access to technical tools needed by the candidate
FRAUTH122	The application should be able to have a paid or unpaid boarding process
FRAUTH123	The application should be able to allocate mentor for Candidate
FRAUTH124	The application should be able to create a schedule for the Candidate

FRAUTH125	The application should be able to organize the workspace
FRAUTH126	Introduce the candidate to technologies they will need to learn how to use
FRAUTH127	Authorized User shall be able to prepare orientation with other Candidates
FRAUTH128	The application shall have an administration tools for the interviewer
FRAUTH129	Authorized User shall be able to Take in any comments or complaints that the candidate might have
FRAUTH130	Authorized User shall be able to Organize meetings between the employer and the candidates
FRAUTH131	Manage any health benefits that should be given to the employee
FRAUTH132	Authorized User shall be able to request Employment background screening
FRAUTH133	Authorized User shall be able to Send offer letter to candidate
FRAUTH134	Authorized User shall be able to Email reminders of the next steps in onboarding

General User

Requirement ID	Requirement Statement
FRGENR013	General user shall be able to book a product demo
FRGENR014	General user shall be enable or disable cookies
FRGENR015	General user shall be able to chat with the organization bot

P3

Authorized User

Requirement ID	Requirement Statement

FRAUTH094	System should automatically remove the Candidate from the hiring process if a step is failed
FRAUTH095	Authorized User shall be able to view the detail of an Employee
FRAUTH096	Authorized User shall be able to update the detail of an Employee
FRAUTH097	Authorized User shall be able to allow an interviewer to set their availability time
FRAUTH098	Authorized User shall be able request Candidate availability for an interview
FRAUTH099	System should automatically schedule an interview of the Candidate
FRAUTH100	System should automatically schedule a programming test if it is the current step
FRAUTH101	Authorized User should be able to create a job post on Stackoverflow
FRAUTH102	Authorized User should be able to create a new advertising campaign on YouTube
FRAUTH103	Authorized User should be able to create a new advertising campaign on StackOverflow
FRAUTH104	Authorized User should be able to calculate the budget and allocate money for specific advertising campaigns
FRAUTH105	Authorized User shall be able to calculate budget to see how many open positions are possible to create
FRAUTH106	Authorized User should be able to calculate the budget to allocate for training interviewers on how to conduct an interview
FRAUTH107	Authorized User shall be able to talk about the day-to-day duties and inform the candidate about the work culture
FRAUTH108	Authorized User shall be able to identify if the hire is an external or internal applicant
FRAUTH109	Authorized User shall be able to determine if an internal hire- minimal advertising is needed

FRAUTH110	Authorized User shall be able to design a recruitment pipeline for individual candidates
FRAUTH111	Authorized User shall be able to provide specific feedback as to why they didn't pass that round
FRAUTH135	Authorized User shall be able to prepare new Candidate for training
FRAUTH136	Authorized User shall be able to create a new onboarding program
FRAUTH137	The application should be able to have a calculate candidate hours
FRAUTH138	The application should be able to have a decrease wage
FRAUTH139	The application should be able to have a give raise
FRAUTH140	The application should be able to have a calculate trust level
FRAUTH141	The application should be able to have a survey candidate
FRAUTH142	The application should be able to have a find jobs for the candidate within company
FRAUTH143	The application should be able to have a create presentations
FRAUTH144	The application should be able to host meetings
FRAUTH145	The application should be able to have a way to communicate with other data items
FRAUTH146	Authorized User shall be able to create tasks for candidate
FRAUTH147	The application should be able to have a deadline setter
FRAUTH148	Authorized User shall be able to set goals for the candidate to complete
FRAUTH149	Authorized User shall be able to find candidate function
FRAUTH140	Authorized User shall be able to pause candidate function
FRAUTH141	Authorized User shall be able to remove candidate function
FRAUTH142	The application should be able to have memory allocation for Candidate information

FRAUTH143	The application should be able to organize team dynamics
FRAUTH144	The application should be able to organize team dynamics
FRAUTH145	The application should be able to organize where and how they will be working
FRAUTH146	Authorized User shall be able to do a performance review scheduling
FRAUTH147	Manage space for Candidates
FRAUTH148	Authorized User shall be able to display information needed by the candidate

General User

Requirement ID	Requirement Statement
FRGENR016	General user shall be able to see product pricing
FRGENR017	General User shall be able to see statistics of a job post

3 - Use Cases and Storyboards with UI Mockups

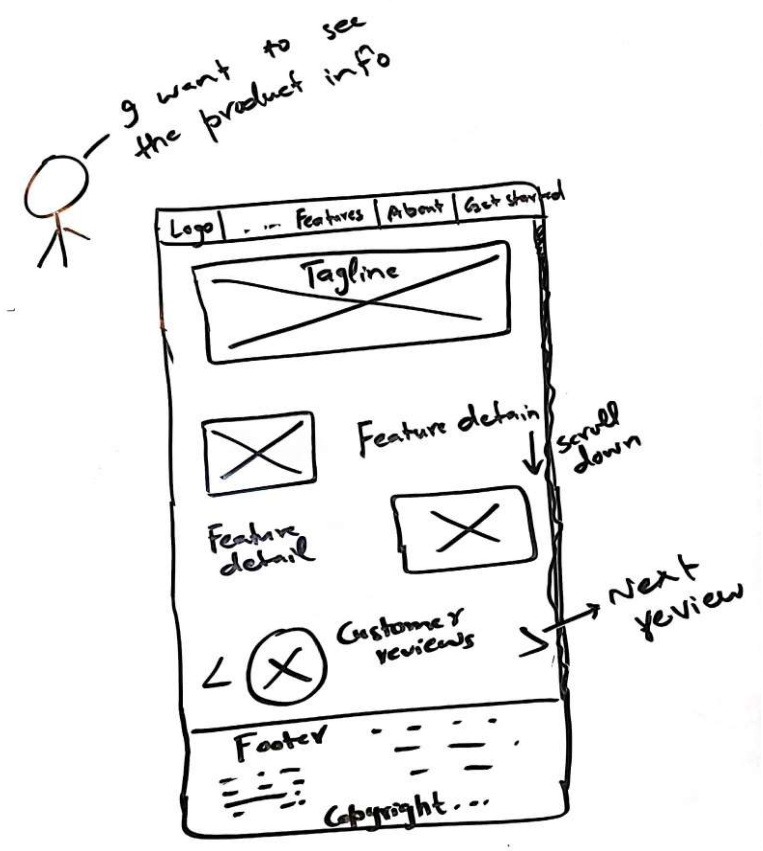
This section provides high-level mockups/storyboards for prioritized use cases.

Use case - See product information

Description

This use case starts when a general user wants to see our product's information which primarily includes the tagline, features, and customer reviews or testimonials. To archive this, the user goes to the *Hiring Guru* landing page, which is the index page of the website. The tagline and logo are visible at the top. The user scrolls down to see various product features which are associated with assistive snapshots of the UI. The user scrolls until he reaches the bottom, where he can see the customer testimonials and reviews. The page footer contains information about copyright and other quick links.

Storyboard

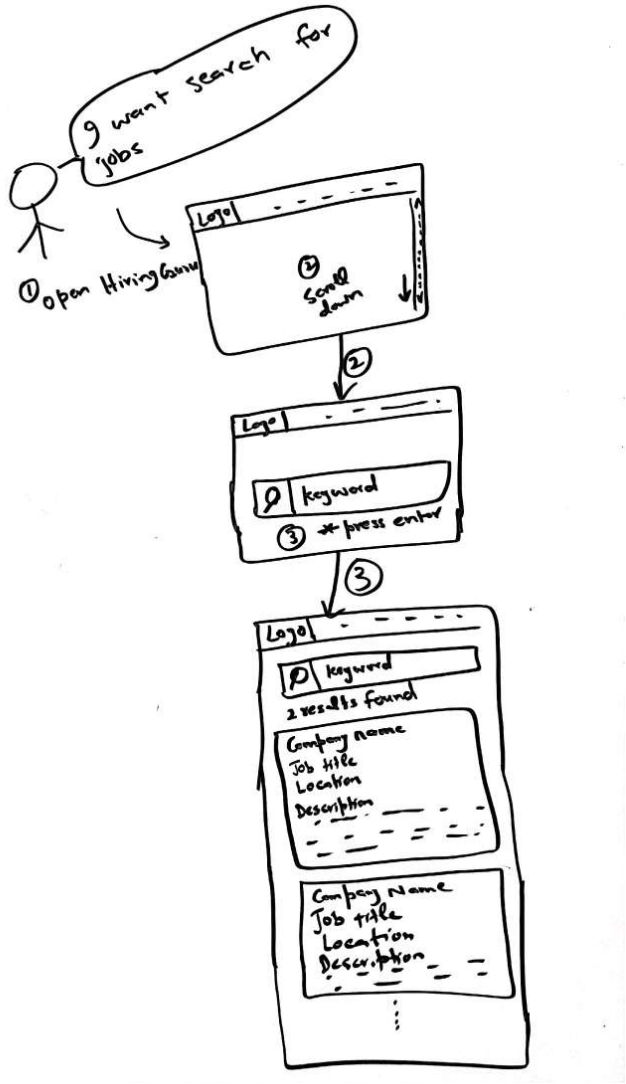


Use Case - Search for jobs

Description

This use case starts when a general user wants to search for jobs listed by companies operating with *Hiring Guru*. To achieve this, the user goes to the *Hiring Guru* main page, which is also the landing page of the application. User scrolls down to the section that displays a search field. The user then types his query into that field and presses the enter button. The system loads a spinner until the search results are being fetched. During this time, the user sees a spinner. Once results are fetched, the user can view a list of jobs along with the company title, job location, and job description.

Storyboard

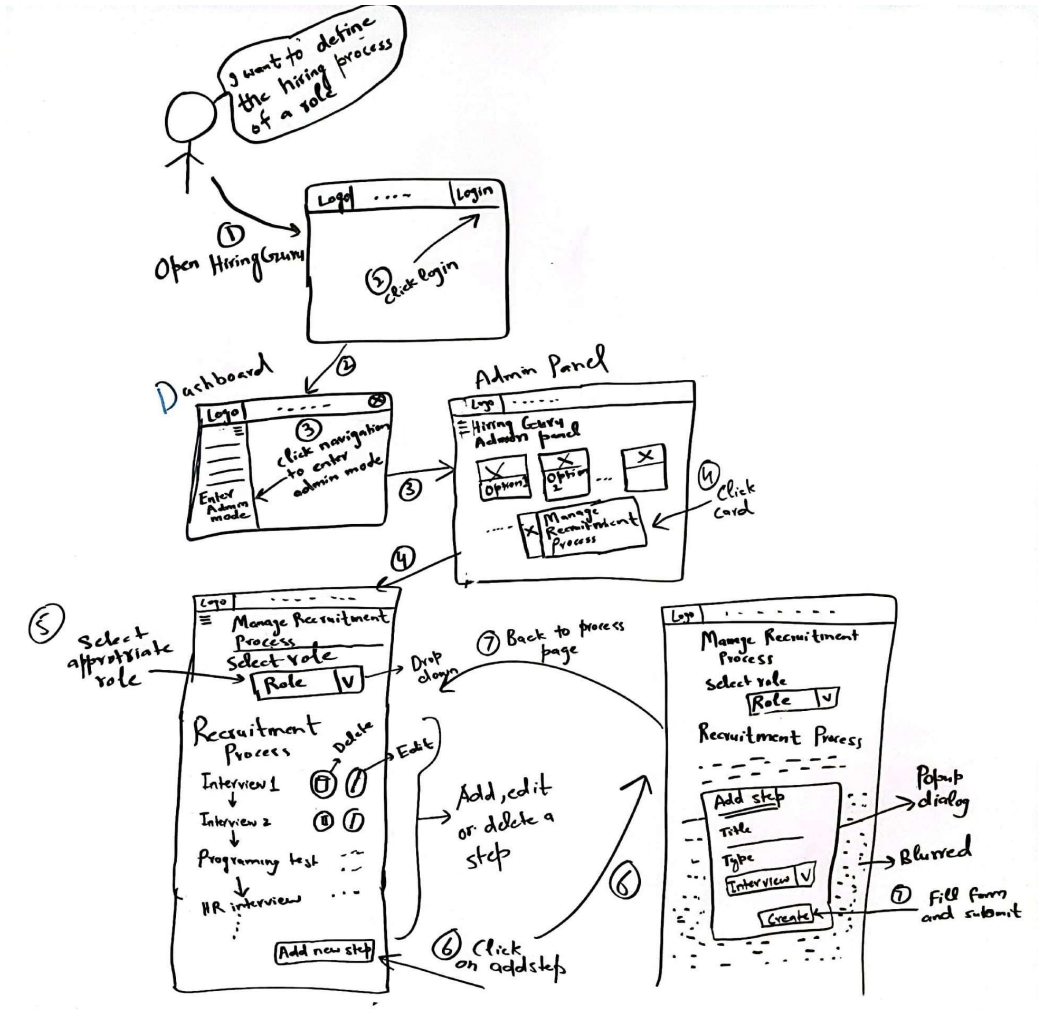


Use Case - Design hiring process for job role

Description

This use case describes the scenario when an Admin user wants to design the hiring process for a particular role (e.g., Software engineer, Graphic Designer). For this, the user first logs in to the system from the landing page with his admin credentials. Then from the side navigation, he goes to the administrative dashboard, which shows various high-level operations like designing the hiring process for a role. Clicking on this card takes the user to the Design Hiring Process page. The user selects a role for which he needs to design the process. The system displays the existing process if present and provides UI for adding new steps, removing an existing step, or updating a step.

Storyboard

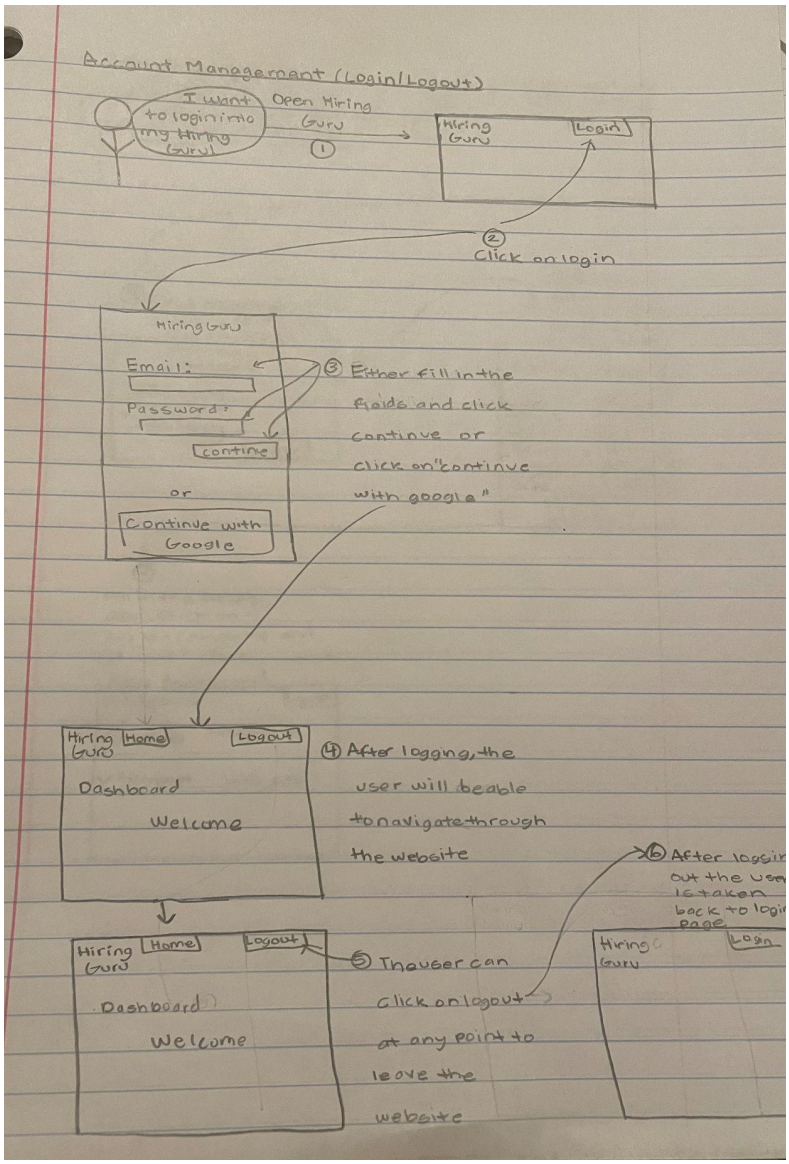


Use Case - Account management

Description

Within this use case, any specific user will be able to do multiple things with their general account for the Hiring Guru. This includes the signup, login, logout, and recover account flow. Upon Hiring Guru, you will be given the prompt to Login or Create an Account. When Creating an Account, the user will need to fill out the required information to have that account created. They can also recover an older account if need be as well. All these various features will be used for users when creating an account.

Storyboard

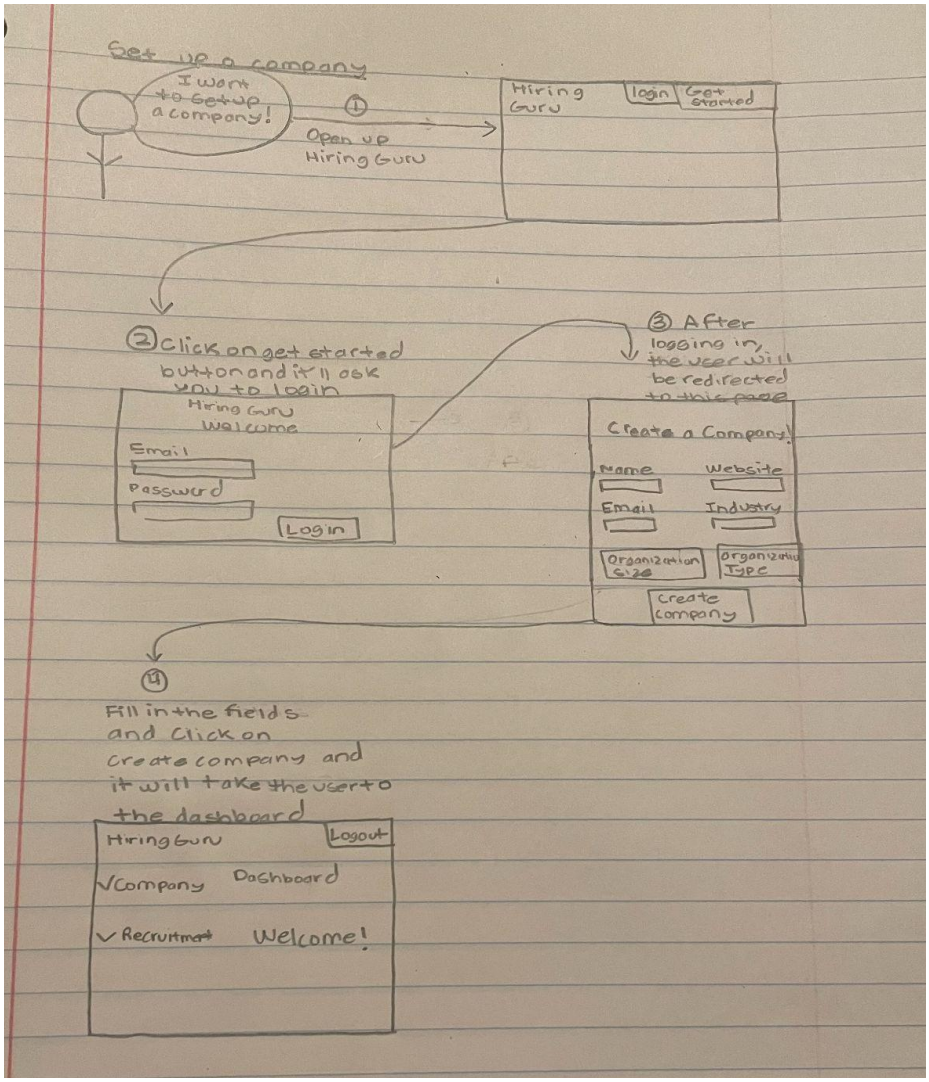


Use Case - Set up a Company

Description

This use case involves creating a new Team or Company within *Hiring Guru*. The idea is an individual will start with an account but will be able to create a Team from their account. This allows the user to set up an organization and allow new or existing members who have an account on the *Hiring Guru* platform to join it as employees. The user will be walked through the options for creating a new Team, and once created, will be able to view the newly-created Team members and their settings.

Storyboard

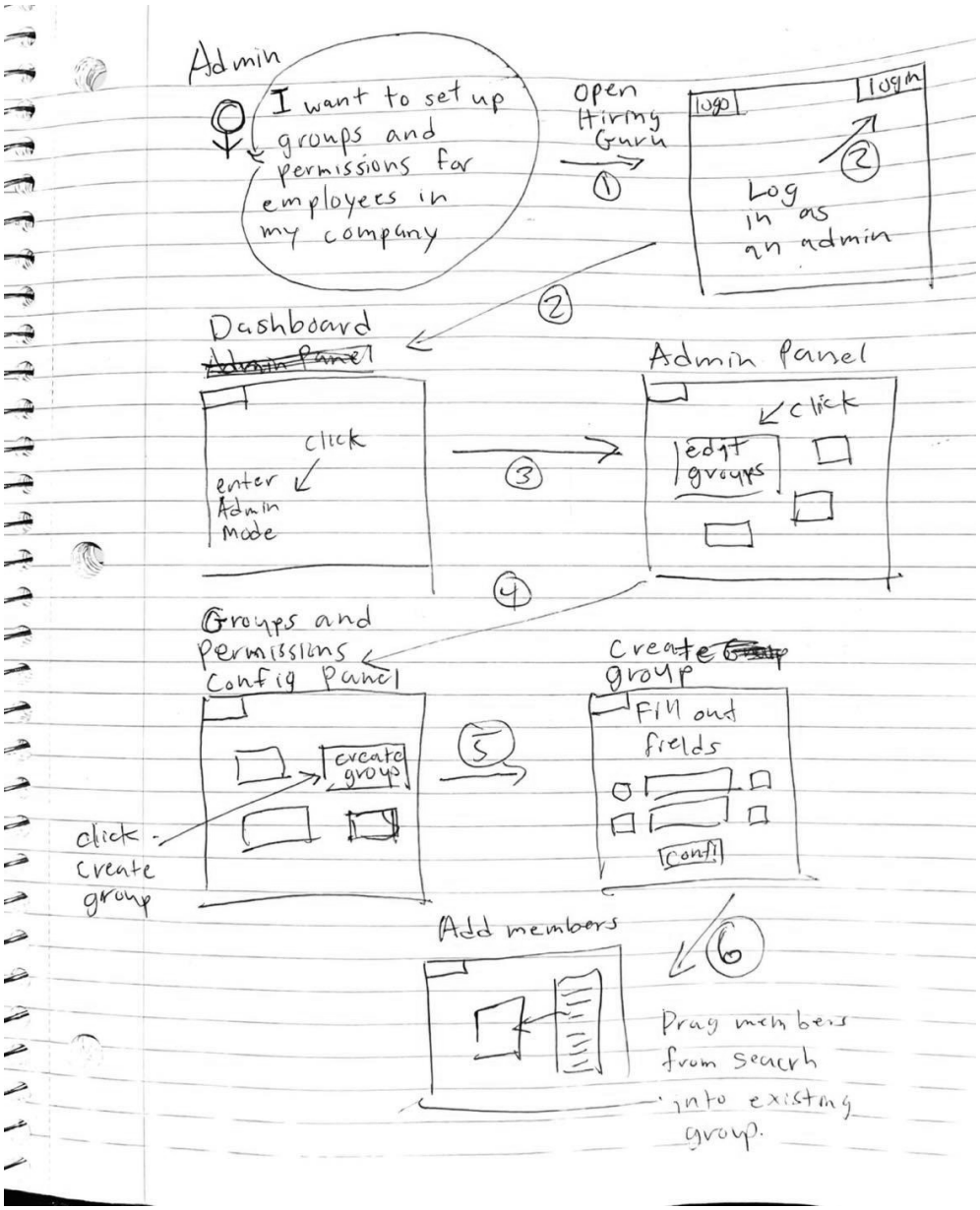


Use Case - Set up user groups and permissions

Description

This is related to the “Set up a company” use case (it comes after that). A company has many employees, and they have various permissions etc. Any authorized user will find a need to create a specific group or specific permissions for certain needs. need to be met. This feature will allow any authorized user to fulfill these needs by swiftly creating a team with the features on the *Hiring Guru*. Some examples will be creating a description for said groups and permissions as well as adding any members that are to be added for these groups.

Storyboard



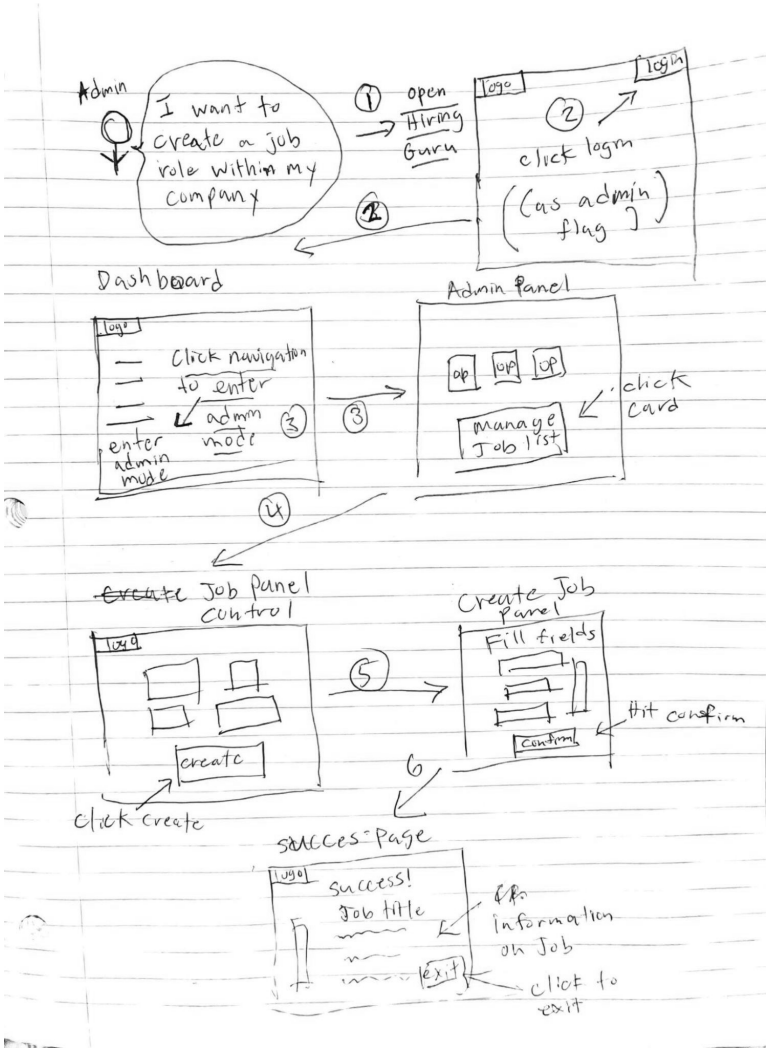
Use Case - Create Job Roles

Description

This use case is about creating a role for the overall company using the Hiring Guru Application. Authorized employees will only be able to do these features where they can create a role and fill out all the necessary requirements for that role. Once the authorized user has completed all the needed sections and filled out the job requirements, then they will be able to submit the job post,

which will then be posted on the Hiring Guru application for Candidates and other general users to apply for.

Storyboard



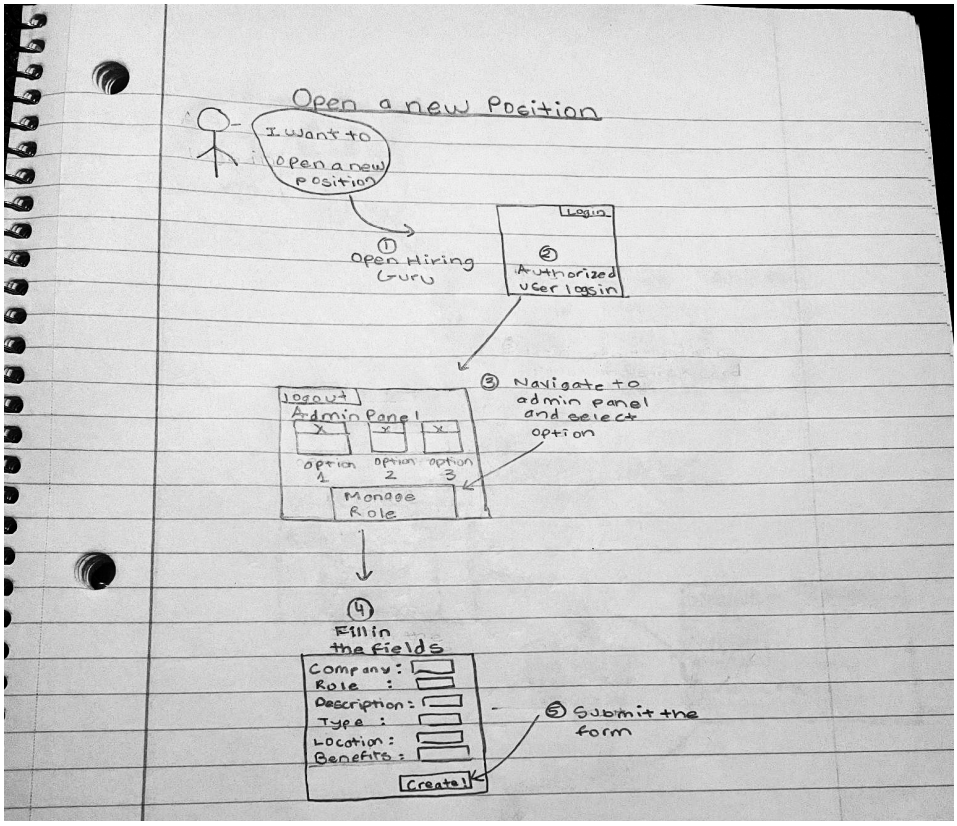
Use Case - Open a new position/job

Description

Whenever a company needs a new position, the Hiring Guru application will have a feature for them to do just that in a very simple and convenient way. Once again, this feature is for authorized users only, where they can log in and select the option to create a few positions. There they can fill out the needed requirements for that position as well as what it entails such as

benefits, salary expectations, location, job description, and whatever else is needed. After they complete that, they will then have the option to post that to the Hiring Guru site where other users can apply for said position.

Storyboard

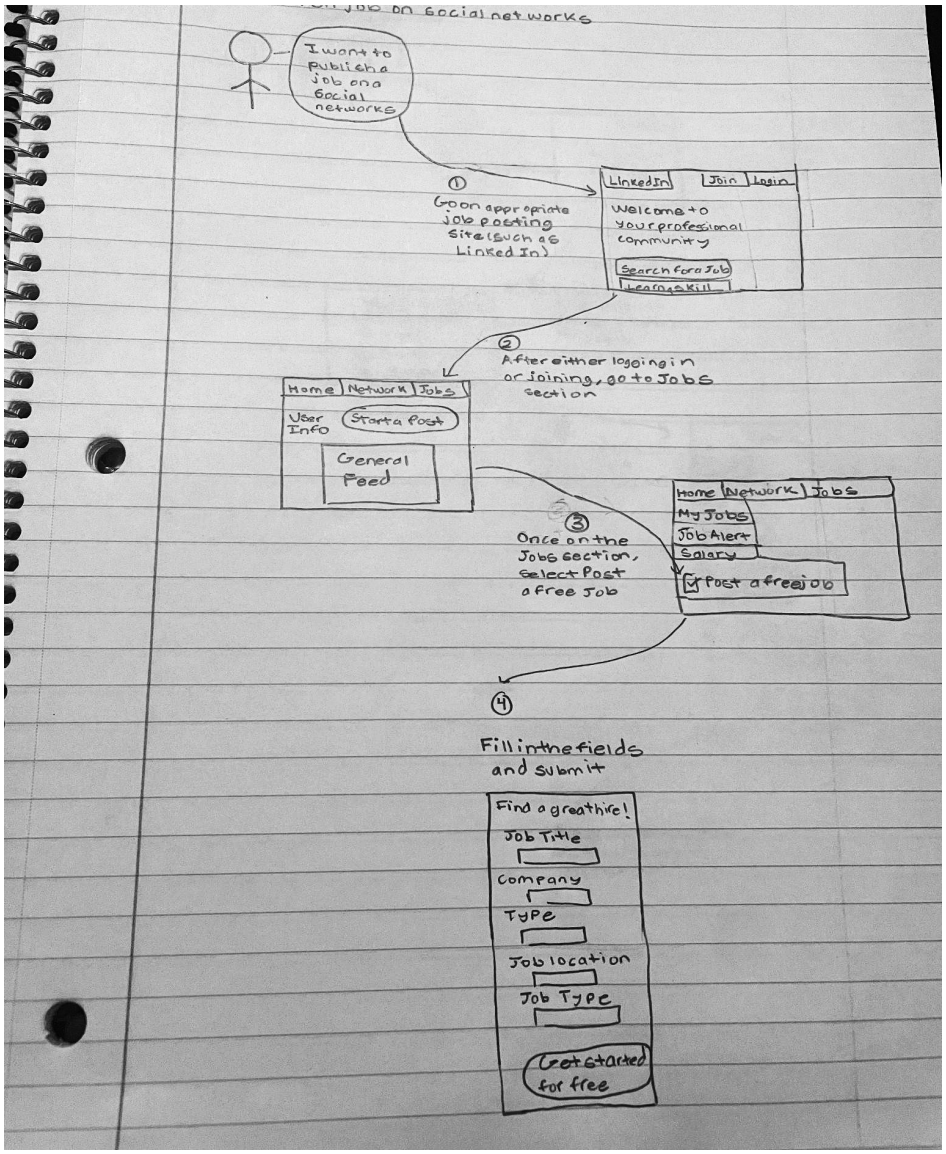


Use Case - Publish job on social networks

Description

Once a job is created, the next thing is to publish it on a social network. This will allow any users from across the Internet to ideally find the job opening and apply for it. After a specific job position is posted, there will be a prompt to promote this job on certain social media platforms. These platforms can include LinkedIn, Twitter, Handshake, Indeed, Instagram, and so much more. In addition, a job link can be generated and copied so any user can share this among other individuals for their own usage.

Storyboard

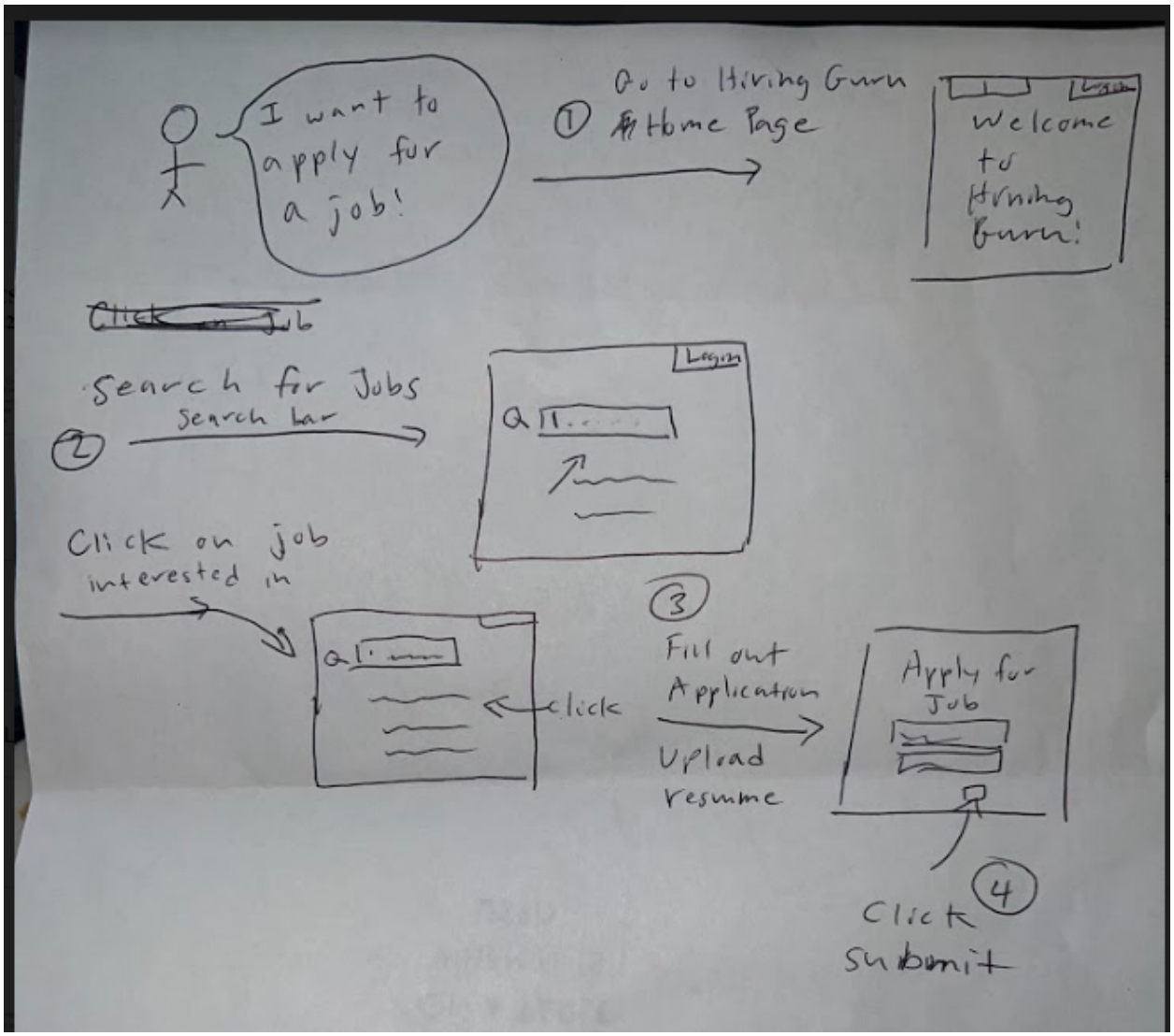


Use Case - Apply for Job

Description

With this use case, a general user is able to apply for a job in the Hiring Guru application. After the user logins, they are able to search for a job role that matches the description of what they want to apply for. An example would be a software engineer. A user is able to see all of the roles that match the description of what they entered while searching for a job, and a user is able to apply for multiple jobs. After selecting a job title, the user is taken to a form in which they are provided many required fields to fill out before they are able to submit the form.

Storyboard

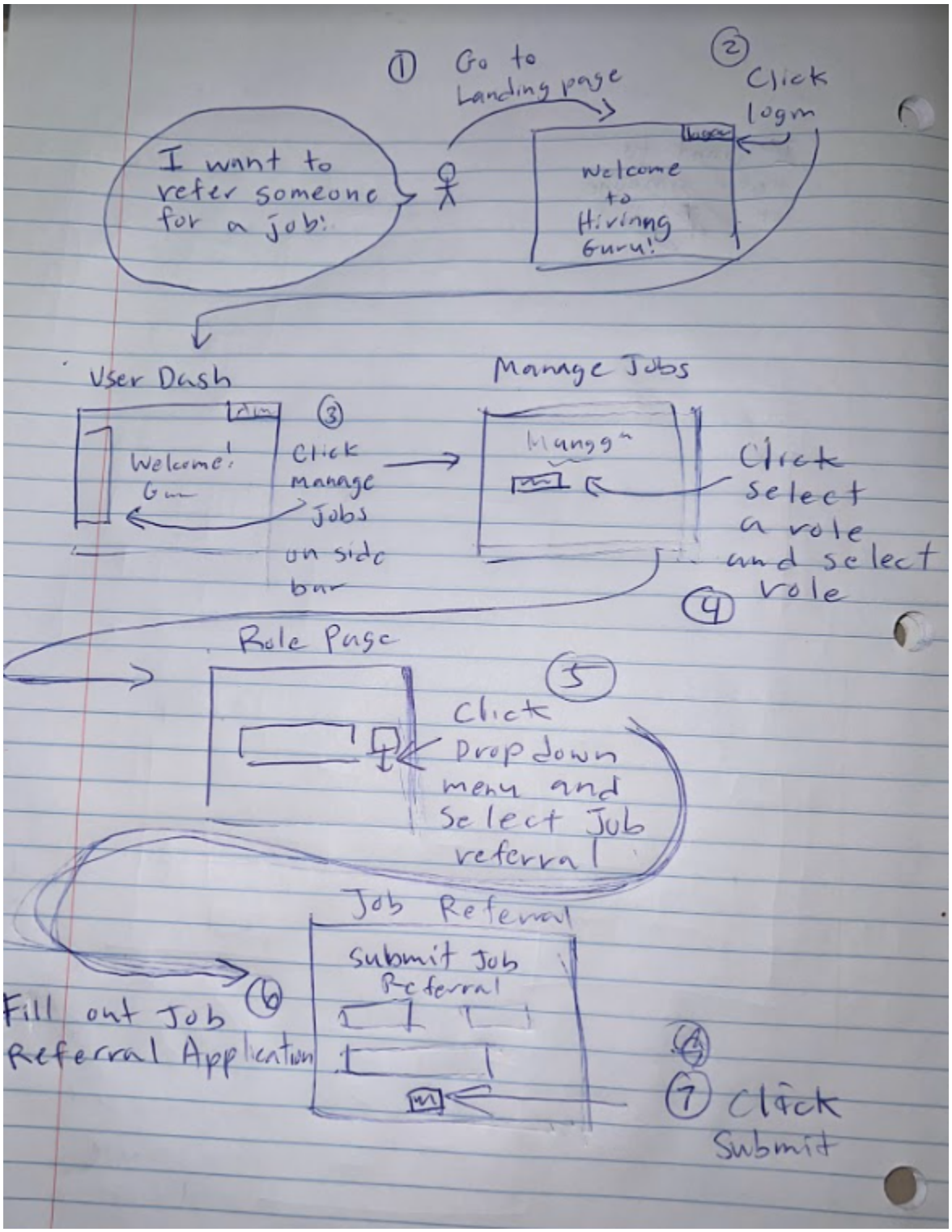


Use Case - Add Referral for Job

Description

Within this use case, an Employee is able to add a referral for a specific job by navigating to the manage jobs section of the dashboard. They can refer any candidate or user that they deem to be fit for the role by filling out a job referral application. Here, they can add general information about the user as well as a resume for any other employees to view as well. Once that is completed, they will review all the written information and submit the referral.

Storyboard

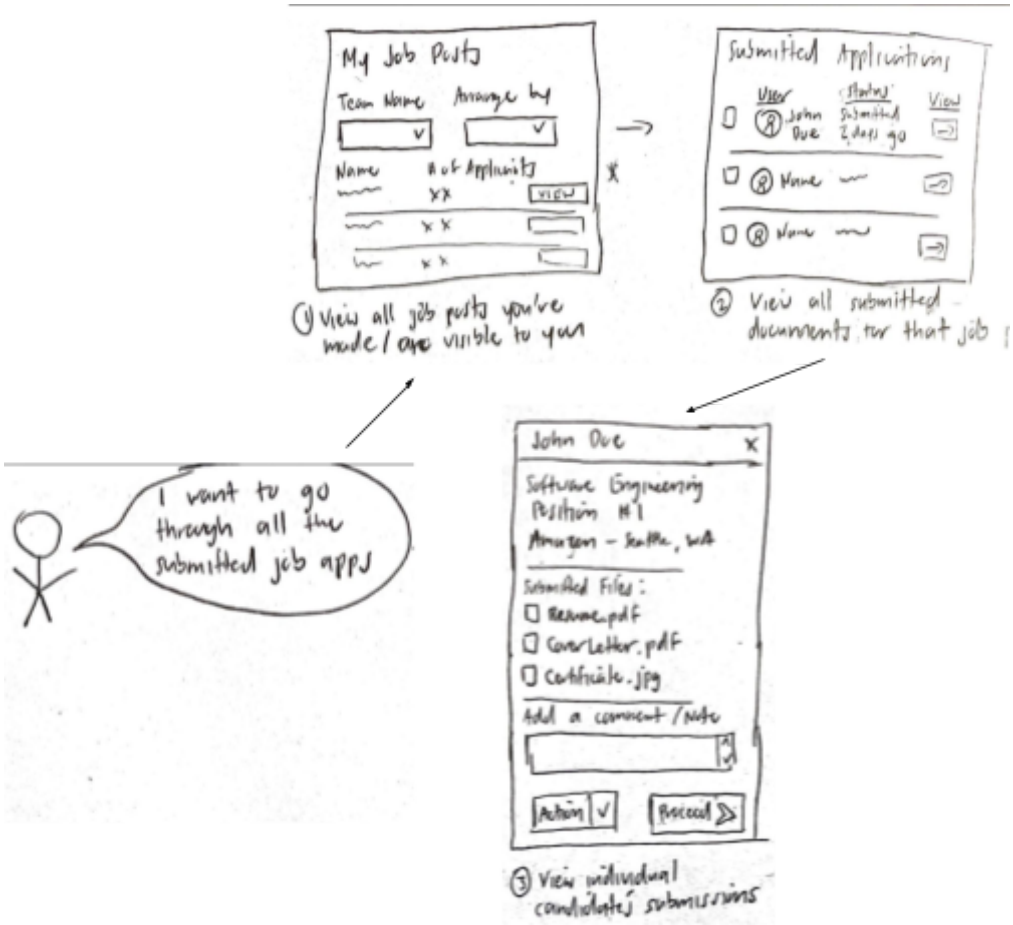

① Go to Landing page
② Click login
I want to refer someone for a job!
Welcome to Hiring Guru!
User Dash
Welcome!
③ Click manage Jobs on side bar
Manage Jobs
Click select a role and select role
④
Role Page
⑤ Click Drop down menu and Select Job referral
Job Referral
Submit Job Referral
⑥ Fill out Job Referral Application
⑦ Click Submit

Use Case - Manage job applications

Description

This use case accounts for the user going through submitted job posts for a position where they can accept (send forward through the hiring process) or reject (throw out) applications. The user is able to view all candidates who have made a submission for a chosen position and see a detailed view of a particular candidate's submissions. Within the detailed view, the user can get a closer look at the candidate's submitted files as well as add comments/notes for their personal reference. Once the user has made their decision, the user can choose to contact or accept or reject the candidate.

Storyboard

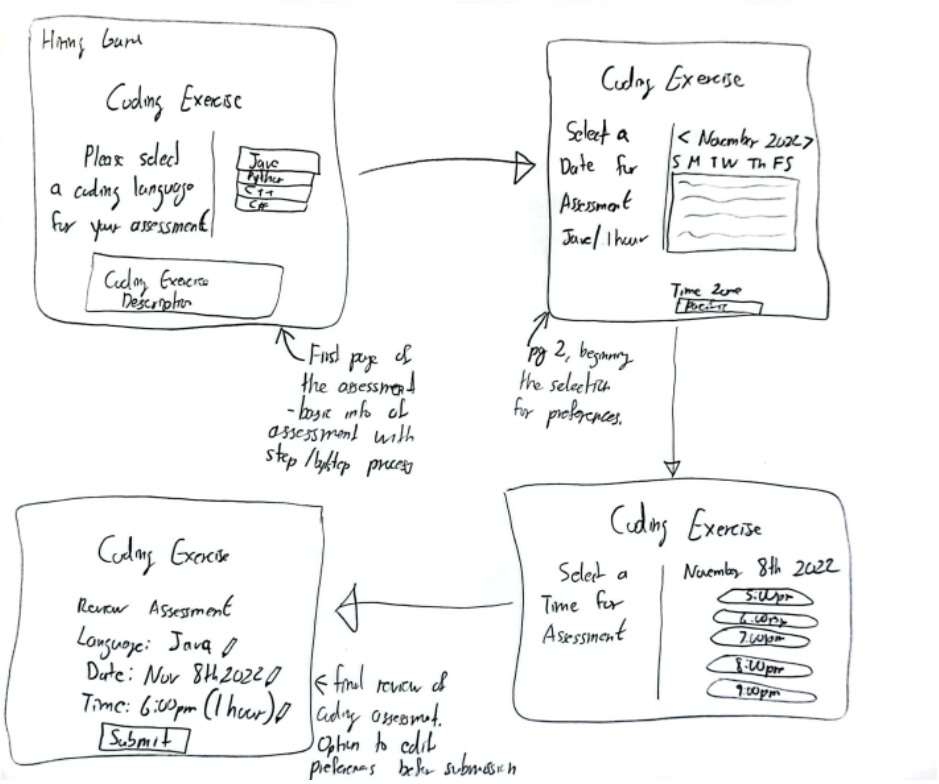


Use Case - Schedule Programming Assessment

Description

When a candidate is in the programming test stage, the system will show an option to schedule a programming test. This will probably include setting a date and time for the test and a test variant (a company usually has multiple variants of programming tests). The system will then use HackerRank API to schedule a test of the candidate (the candidate will receive an email from HackerRank).

Storyboard

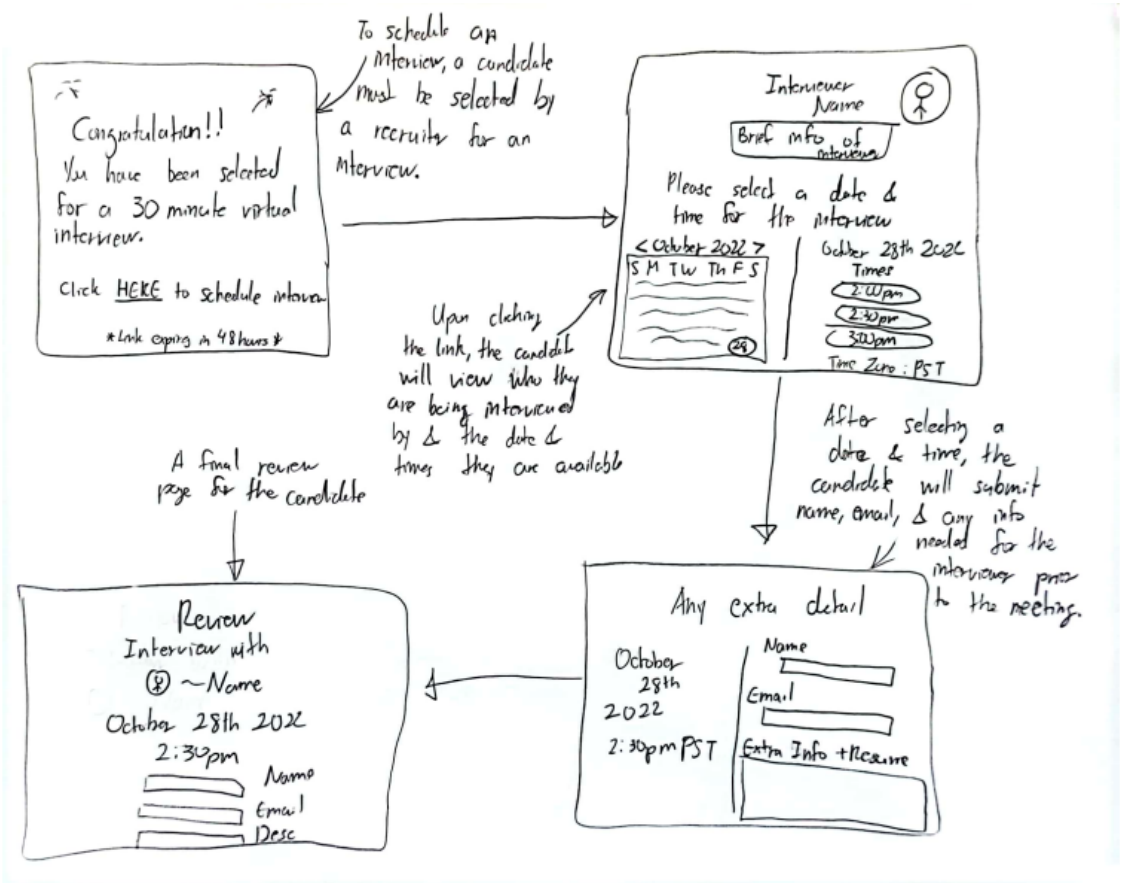


Use Case - Schedule an interview

Description

Once a Candidate is selected for an interview, they will be given a link which will take them to a section of the site where they can view their upcoming interview information. This information will include the individual who will be interviewing them, as well as what their role is within the company. After that, they can select a date and time as well as add any information they want to be known prior to the interview. Upon submission, this will then go back to the Interviewer, who will be notified of the date and time chosen for the interview.

Storyboard

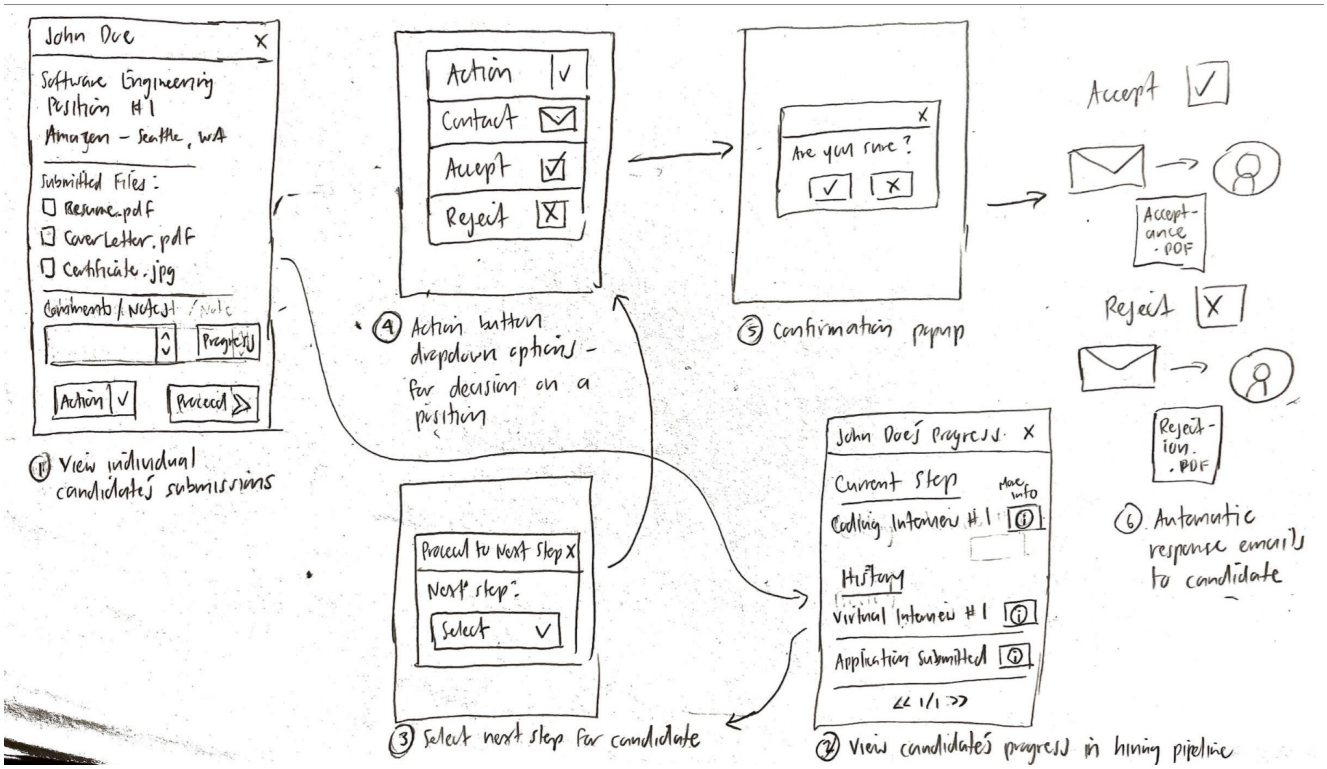


Use Case - Complete Hiring Process

Description

At the last stage of the hiring process, the system shows options to complete the hiring process. Specifically, a button to either reject the application or place a job offer. Rejection will include writing an email, the system will send the email itself. For acceptance, the system will allow uploading a PDF file with a job offer and will email that to the candidate.

Storyboard



4 - High-level database architecture and organization

- Company (Strong Entity)
    - Database Requirements
        - The company shall have at least one employee
        - The company shall be created by one user
    - Entities and Attributes
        - Company_id: key, alphanumeric, multi-value, composite
            - Address
            - Company Size
            - Employees
            - Mission
            - Square Feet
            - Title
            - Description

- Role (Weak Entity)
    - Database Requirements
        - A Job Role shall be assigned to many employees
        - A Job Role shall be created by an Employee
        - A role shall have many positions opened
    - Entities and Attributes
        - Role_id: key, alphanumeric, multi-value, composite
            - Description
            - Title
            - Benefits

- Position (Weak Entity)
    - Database Requirements
        - A Job Position shall be opened for a Role
        - A Job Position shall be created by the authorized Employee
    - Entities and Attributes
        - Position_id: key, alphanumeric, derived

- Candidate(Strong Entity)
    - Database Requirements

        - A Candidate shall be able to view at zero to many job positions
        - A Candidate shall be able to be interviewed by an Employee
        - A Candidate shall be send Applications for multiple job Positions
    - Entities and Attributes
        - Candidate_id: key, alphanumeric, derived
        - Information: alphanumeric, composite
            - First name
            - Last name
            - Date of Birth
            - Email
            - Phone Number
            - Resume

- Employee (Strong Entity)
    - Database Requirements
        - An Employee shall be able to view candidates in the Recruitment Pipelines Process
        - An Employee shall be able to interview Candidates
        - An employee shall be able to schedule interview of candidates
        - An employee shall be able to setup Permissions and Groups
        - An authorized employee shall be able to shortlist Applications
        - An authorized employee setup Recruitment Pipelines Process
        - An authorized employee shall be creating many Job positions
        - An authorized employee shall be able to create many Job roles
    - Entities and Attributes
        - Employee_id: key, alphanumeric, composite
            - Name
            - Role
            - Position
        - Employee number: numeric, derived

- JobApplication (Strong Entity)
    - Database Requirements
        - A job application shall be opened by a candidate for a specific Position
        - A job application shall contain profile information of many candidates
    - Entities and Attributes
        - JobApp_id: key, alphanumeric, derived

- Interview (Weak Entity)
    - Database Requirements
        - An Interview shall be able conducted by at least one Employees
        - An Interview shall have at least one Candidate
    - Entities and Attributes
        - Interview_id: key, alphanumeric, composite
            - Time
            - Date
            - Interviewer
            - Interviewee
            - Location

- Group (Weak Entity)
    - Database Requirements
        - A group can have many employees
        - A group can be assigned multiple permissions
    - Entities and Attributes
        - group_id: key, alphanumeric, derived

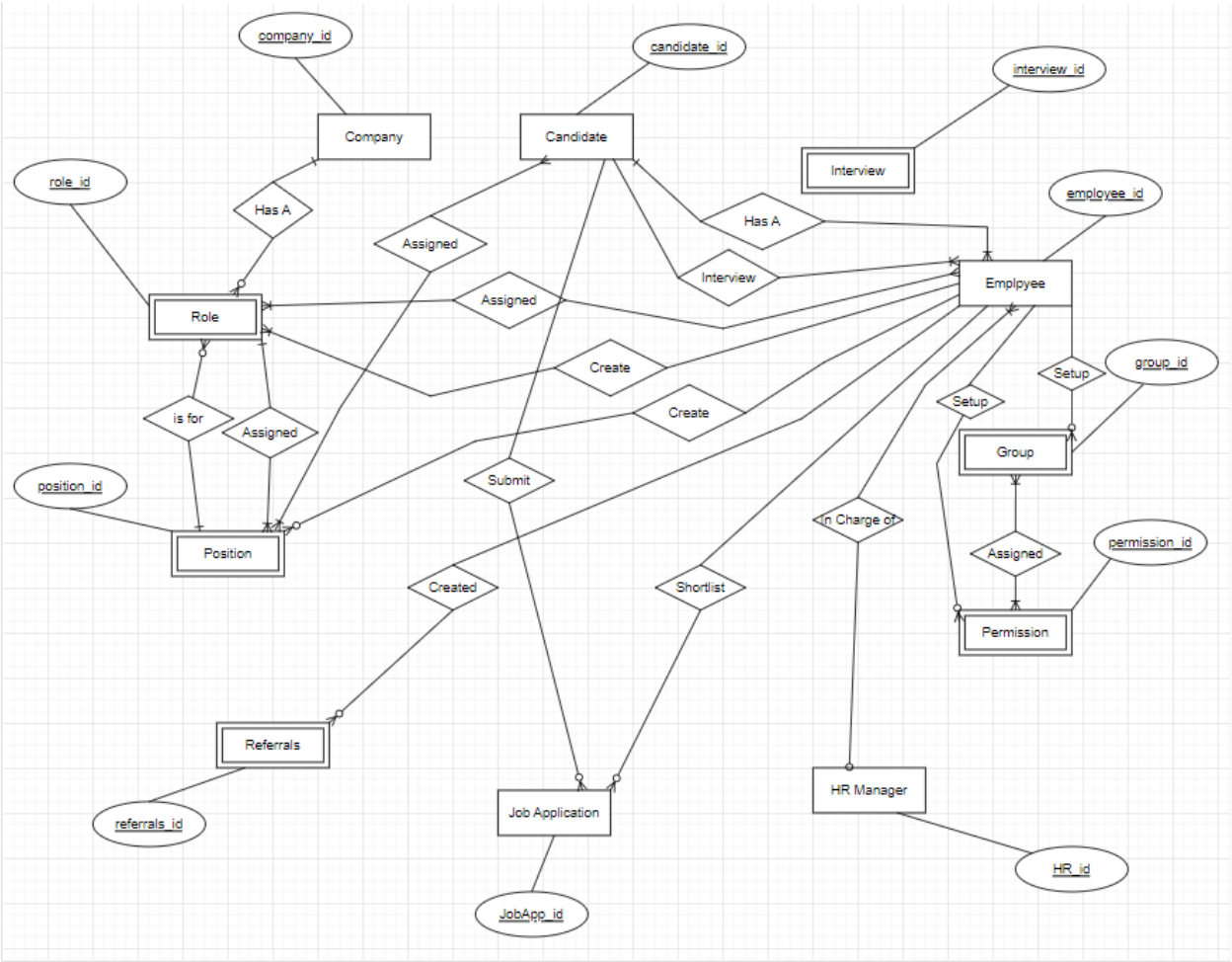
- Permission (Weak Entity)
    - Database Requirements
        - A Permission can be assigned to multiple Groups
    - Entities and Attributes
        - Permission_id: key, alphanumeric, derived

- Referrals (Weak Entity)
    - Database Requirements
        - A Referral shall be created by at least Employees
        - A Referral shall be used for one Candidates
    - Entities and Attributes
        - Referral_id: key, alphanumeric, derived

- HR Manager (Strong Entity)
    - Database Requirements
        - A HR Manager shall be creating many Job positions

        - A HR Manager shall be creating many Job roles
        - A HR Manager shall be one Authorized user
        - A HR Manager shall be in charge of many employees
    - Entities and Attributes
        - HR_id: key, alphanumeric, Composite
            - Role
            - Position
            - Title

Entity Relationship Diagram (ERD)



5 - High Level APIs and Main Algorithms

APIs

The following are the REST resources and their associated APIs:

Company

1. Create a new Company
    - This API creates a new company with the requested parameters : title and description. Both title and description are requested in the Company constructor, and once provided are able to create a new Company.
2. Get the detail of a Company
    - This API gets a certain company once a company id is provided. If a company that matches the id exists, a company gets returned.
3. Get a list of all Company
    - This API gets all companies that exist within the database. Within the API response, the company title, company description, and company ID is returned.
4. Update the details of a Company
    - This API updates an existing company. For this API, an existing company id is requested as a parameter as well as the new company description and title. After this API is sent, the database updates the details of the company of the provided id.
5. Delete a Company
    - This API deletes a company from the database. For this API, a company id is needed to delete the company.

Role

1. Create a new Role
    - This API creates a new role. For the path of this API, a company id is requested in the path, and title,expectations, and benefits are requested as parameters. Title, expectations, benefits, and a Company variable are requested in the Role constructor, and once all are provided a new Role is created.
2. Get the detail of a Role
    - This API gets a certain role once a company id is provided.For the path of this API, a company id is requested in the path. If an existing company id and a corresponding role match the path variable, then a role gets returned, which includes the corresponding company information.
3. Get a list of all Role

    - This API gets all roles that exist within the database. Within the API response, the role title, role expectations,role benefits and role ID is returned.
4. Update the details of a Role
    - This API updates an existing role. For this API, a company id is requested in the path. Additionally, a new role title, expectations, and benefits is requested as a parameter. After the API is run, the role details are updated on the corresponding company id.
5. Delete a Role
    - This API deletes a role from the database. For this API, a company id is needed to delete the role and company details.

Position

1. Create a new Position
    - For this API, the title and experience necessary for the position are requested as parameters. The title and experience necessary are requested in the Referral constructor and once both are provided a new Position is created.
2. Get the detail of a Position
    - For this API, a position id is required in the path of this API. If a position with that id exists, then the position is returned.
3. Get a list of all Position
    - This API gets all of the positions that exist within the database. The response returns the title and experience necessary for the position.
4. Update the details of a Position
    - This API updates the details of existing positions. For this API, a position id is requested in the path. Additionally, the title and experience necessary are requested as parameters.
5. Delete a Position
    - This API deletes a position that exists within the database. For this API, the position id is required in the path of the API.
6. Search all open positions using a query string
    - This API requires a parameter that provides a keyword that the query string parses through and finds positions that match the keyword. The API returns all positions that match the keyword.

Referrals

1. Create a new Referral
    - For this API, the first name of the candidate, the last name of the candidate, the email of the candidate, and the resume are requested as parameters. All of the requested parameters are requested in the Referral constructor.
2. Get the detail of a Referral

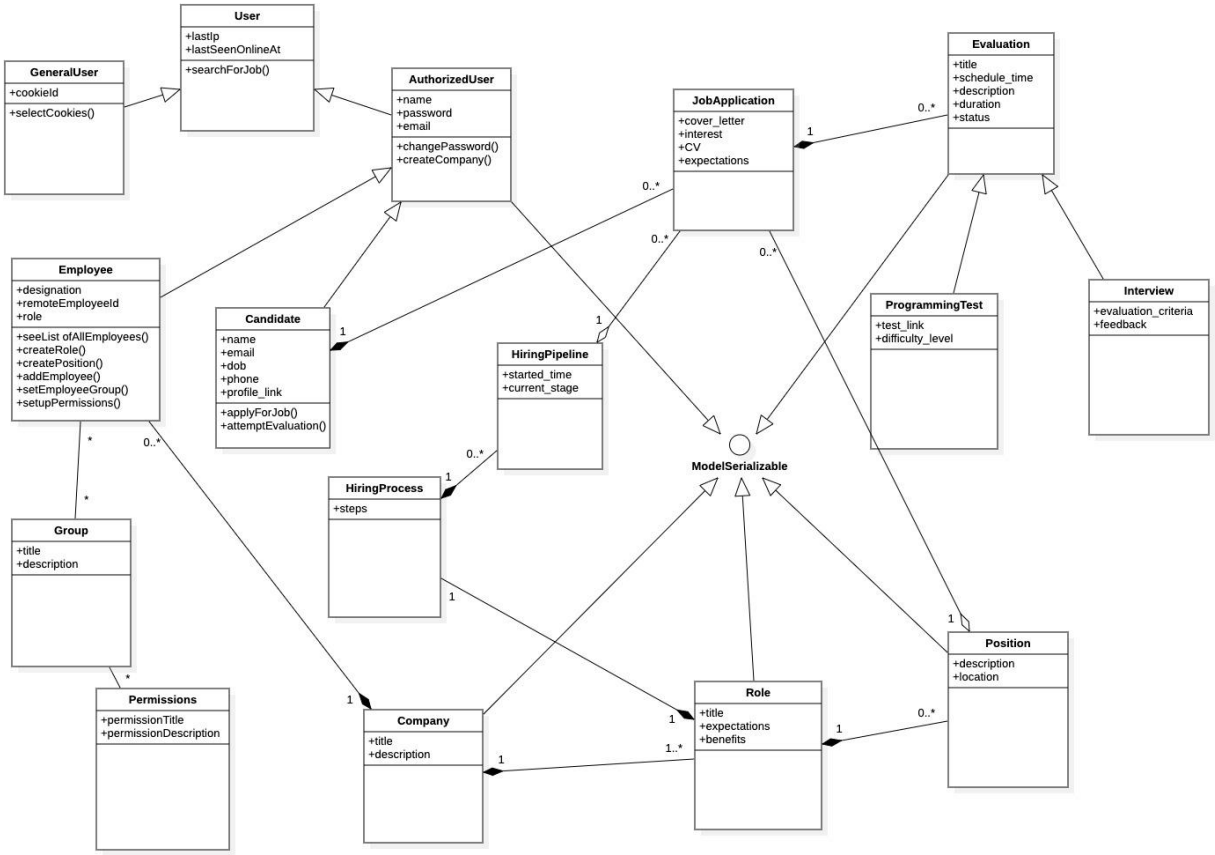
    - For this API, a referral id is required in the path of this API. If a referral with that id exists, then the referral is returned.
3. Get a list of all Referral
    - This API gets all of the referrals that exist within the database. The response returns the first name of the candidate, the last name of the candidate, the email of the candidate, and the resume file name.
4. Update the details of a Referral
    - This API updates the details of existing referral. For this API, a referral id is requested in the path. Additionally, the first name of the candidate, the last name of the candidate, the email of the candidate, and the resume are requested as parameters.
5. Delete a Referral
    - This API deletes a referral from the database. The API takes the referral id as the path variable.

Job Application

1. Create a new Application
    - This API creates a new job. For the path of this API, a role id is requested in the path, and title,location, description, and job type are requested as parameters. Title,location, description, job type, and a Role variable are requested in the Job constructor, and once all are provided a new Job is created.
2. Get the detail of an Application
    - This API gets a certain job once a role id is provided.For the path of this API, a role id is requested in the path. If an existing role id and a corresponding job match the path variable, then a job gets returned, which includes the corresponding role and company information.
3. Get a list of all Applications
    - This API gets all jobs that exist within the database. Within the API response, the job title, job location, job description, job type and job ID is returned. Additionally, the corresponding role and company information is returned as well.
4. Update the details of an Application
    - This API updates an existing job. For this API, a role id is requested in the path. Additionally, a new title,location, description, and job type is requested as a parameter. After the API is run, the job details are updated on the corresponding role id.
5. Delete an Application
    - This API deletes a job from the database. For this API, a role id is needed to delete the job, role, and company details.

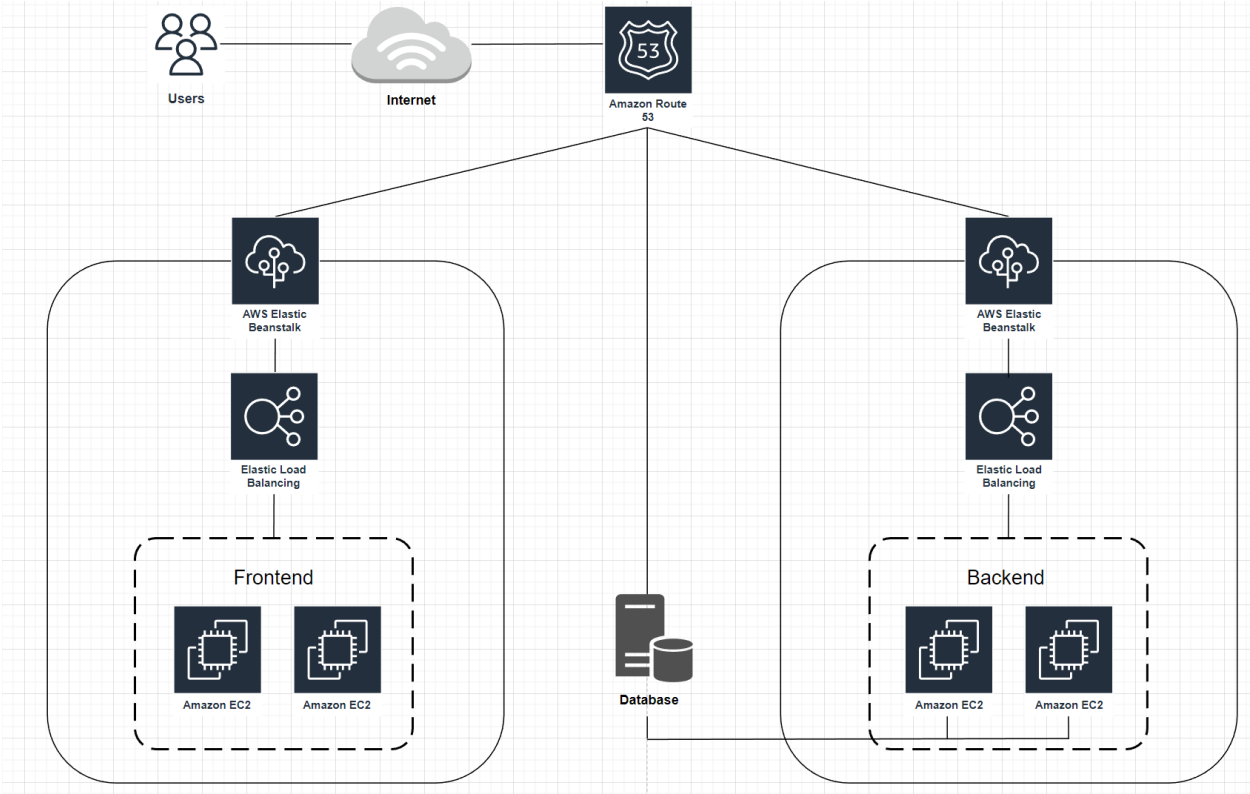
6 - High-Level UML Diagrams

System class diagram

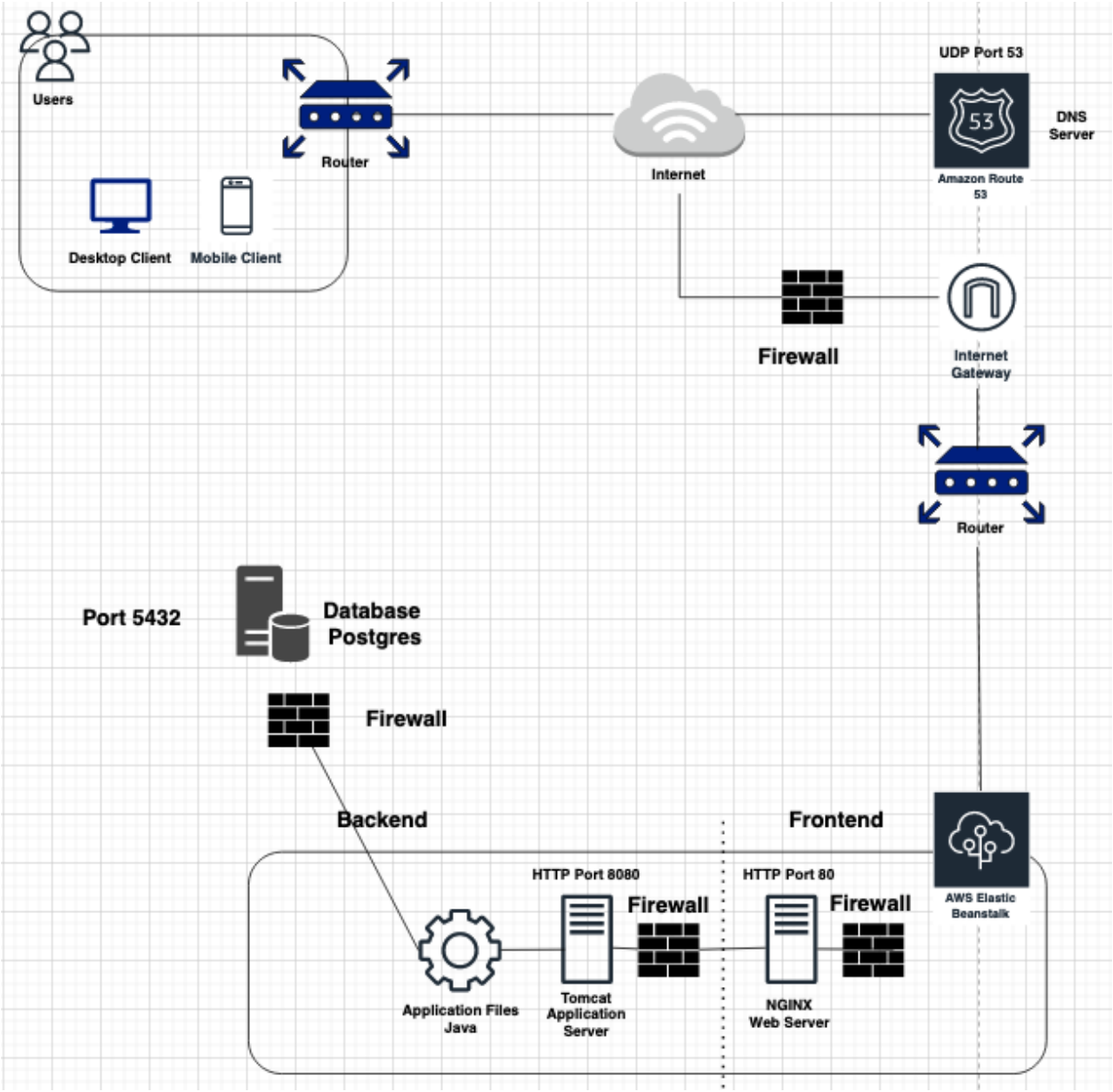


7 - High-Level Application Network and Deployment Diagrams

AWS Elastic Beanstalk Deployment Diagram



Network Diagram



8 - Identify actual key risks for your project at this time

1. **Course Conflicts**
    - Risk: Various high-course assignments and conflicts interfere with the time needed for this project. Many of us are taking high-level courses in addition to this class, such as Intro to Database, UNIX Programming, and even some Grad

School work, to name a few. Every class requires a strong level of commitment and assigns rigorous work to be completed in a timely manner. All this work can become overbearing at times and conflict with the work needed to be done within the Software Engineering Course.

- Solution: Our solution for this project risk is to further improve our time management skills. In the real world, many different aspects of life are going to come into play when it comes to Software Engineering, and this course is a strong example of a real-life scenario. We have to navigate the different course workloads ourselves and make sure we get all the necessary requirements of the milestone done while not compromising other course work

2. Scheduling Conflicts

- Risk: All team members have a number of other commitments and different schedules (commute time, off and on-campus jobs, other class times), which all conflict and make having everyone present in the meeting rather difficult. Our main meeting time is 7:30 PM every Mon and Wed, and even then, that is right when some team members finish meetings, finish class time, and right when they get home. It gives us very little time to decompress and relax at all, which can greatly show within the milestone quality.
- Solution: To solve our scheduling conflicts, we did our very best to find a general time that works for everyone. In addition to that, we also made sure we established a strong base of communication with one another. The reason for this is that when certain things come up, that may conflict with a team member's ability to attend the meeting, we make sure to have detailed meeting notes for them to understand exactly what they missed so we are not wasting any precious time catching them up.

3. Knowledge Gap

- Risk: All the team members have different experiences with software engineering as a whole. Some came from community colleges, some had industry experience, and some only have SFSU experience. It is difficult to find common ground in the technologies and project where everyone feels comfortable in what they are working on
- Solution: The way we have decided to deal with this problem is by making sure all group members have a strong sense of patience and understanding for one another. We all must acknowledge that everyone has different academic experiences, and some may be more beneficial in this project than others. That is why we make sure to do our own side homework to ensure all team members understand the technologies we will be using and for those who already understand the coding language to help teach others, so we are all skilled enough to complete the project.

4. Lack of Team Member

- Risk: At the beginning of the semester, we had a full team of 6 members ready to tackle this entire project. Unfortunately, one of our team members at the time decided to move groups to aid a smaller team in the project. While we hold no resentment towards him and fully understand the reason for their team switch, it now puts the load of the project onto only five of us. This makes the workload rather intense for all of us to complete, so we have to work extra hard to compensate for that.
- Solution: Due to an uneven classroom size, some teams have a small number of individuals compared to others. We once had a full team ready to tackle this project. However, they decided to move to another group needing a member because their skills and experiences better fit their project. This now leaves us with 5 members rather than 6. This put an extra set of pressure on us, but we did our very best to work as a team to do the work. We understand that even though we are down a member, if we put in that extra effort, then we will be able to compensate for the missing team member and get this project done.

9 - Project management

Trello is the main tool we are currently using to keep a unified progression on the project. In addition, we agreed to meet twice a week, on Wednesdays and Mondays, in order to go over what is expected of each team member and update each other on what is going on. We try to

keep the meetings short, around twenty minutes, as all of our schedules are pretty busy. We also use the Trello board for retrospection every week.

Posting our daily status in discord is a requirement that has been made by the team lead. This means that each team member must post every day what they have done, what they will do the next day and any questions or concerns about the project. This really keeps us up to date with each other and helps ensure that we stay connected as a team.

Another important part of our team process is evaluating how we are doing every few weeks. We have anonymous concern meetings where we anonymously post things that we think we are doing well, any things that we could do better at. We then go over them all and discuss how we can improve.

10 - Detailed list of contributions

- Contributions:
    - Kenny Leong

        - Worked on the Landing page - frontend. Completed NavBar, CustomerReview, Tagline, and Footer components
        - Did 3 mockups/storyboards
        - Worked on Application and Network architecture diagrams
        - Helped plan the frontend work
    - Mamadou Bah
        - Worked on the Landing page - frontend. Completed ProductFeature component
        - Did 3 mockups/storyboards
        - Worked on “Identify actual key risks …” section
        - Worked on the high-level database architecture section
        - Formatting and improving the document
    - Khushi Khanna
        - Did 3 mockups/storyboards
        - Worked on Data Definitions section
        - Worked on High-Level APIs section
        - Worked on backend APIs
        - Helped to prioritize the functional requirements
    - Eric Leow
        - Did 3 mockups/storyboards
        - Tried to setup the ORM
        - Worked on the Project management section
        - Helped to prioritize the functional requirements
    - Farhan Haider
        - Did 3 mockups/storyboards
        - Worked on the UML diagram
        - Prioritized functional requirements
        - Completed the JobSearch component on the frontend
        - Planned the overall work
        - Helped other team members with their work
    - Score (Out of 10 - 10 being ideal)
        - Kenny Leong - 9
        - Mamadou Bah - 8

- Khushi Khanna - 8
- Eric Leow - 6

SW Engineering CSC 648/848 Fall 2022

The Hiring Guru

Team 2
Farhan Haider (Team Lead/Backend Lead): shaider1@sfsu.edu
Kenny Leong (Frontend Lead/GitHub Master): kleong2@mail.sfsu.edu
Mamadou Bah (Frontend): mbah1@mail.sfsu.edu
Khushi Khanna (Backend) kkhanna@sfsu.edu
Eric Leow (Frontend): elow@mail.sfsu.edu

Milestone #1

September 21st, 2022

Row	Document Version	Review Date	Description
1	Version 1	9/10/2022	Document was first created, sections were written down and organized for later usage.
2	Version 2	9/13/2022	Began filling out the document. Rough drafting use cases, requirements, data entities, and such.
3	Version 3	9/16/2022	Evenly split up work on what needed to be done. Made a deadline by Monday (9/19) to have use cases, data entities, and requirements
4	Version 4	9/19/2022	Had a lot of information but was very unorganized. Spent time organizing information and also adding use case diagrams. Began work on Competitive Analysis and finishing Requirements
5	Final	9/21/2022	Wrapping up any missing work and helping each

	Version		other in organizing work, formatting, editing any unsatisfactory work, and finalizing last details for this milestone.

Executive Summary

Hiring is not a simple process. It's a hectic job that takes days. From finding the right candidate for your organization and job requirements to managing the whole hiring process consumes countless hours and effort. That's because a lot of this processing is manual. Automation is the silver bullet. Many solutions are out there in the market which automate this process to a certain extent. Hiring Guru aims to reduce this effort to the bare minimum. From providing help finding the right candidate to designing and managing the whole hiring pipeline, Hiring Guru will be there to help you.

Hiring Guru will help you find suitable candidates by helping you run advertising campaigns and receive job applications. With every application received, Hiring Guru will provide you with a match score with the help of its intelligent algorithms. It will automatically add the candidates scoring above a certain threshold set by you to the hiring process. It will allow you to design the Hiring Process for different positions and will automatically follow that process for every promising candidate. You won't need to worry about communicating with the candidate, interviewers, etc., since Hiring Guru will do all of these for you. You will be served with hiring insights. Lastly, Hiring Guru would love to have you intervene anytime to see the status or modify any part of the process.

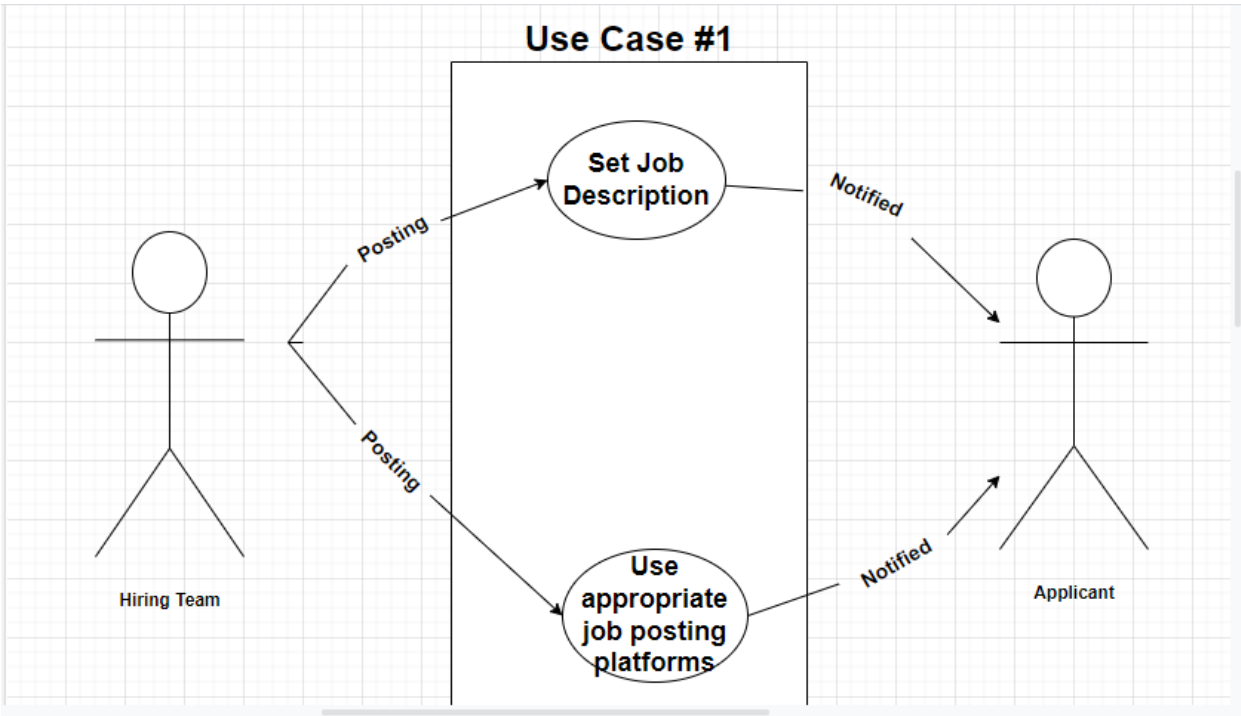
Main Use Cases

Use Case - 1

Use Case Title	Job Posting
Actors	Marketing Team

Description	The marketing team is responsible for posting the job on the appropriate websites and ensuring that the job is well advertised for. Essentially, possible candidates should be sure of general information surrounding the job posting, such as when the job closes and the general requirements surrounding the job. It's important for the marketing team to advertise the relevant information about the job so that it reaches more suitable candidates. Ex. This use case begins when a marketing campaign is launched. Specific third party recruiting accounts are given to the program, it finds all the positions in the company that have not yet been filled, and it makes posts on each recruiting site for the list of positions. Resumes are then passed on to the recruiter for further processing.

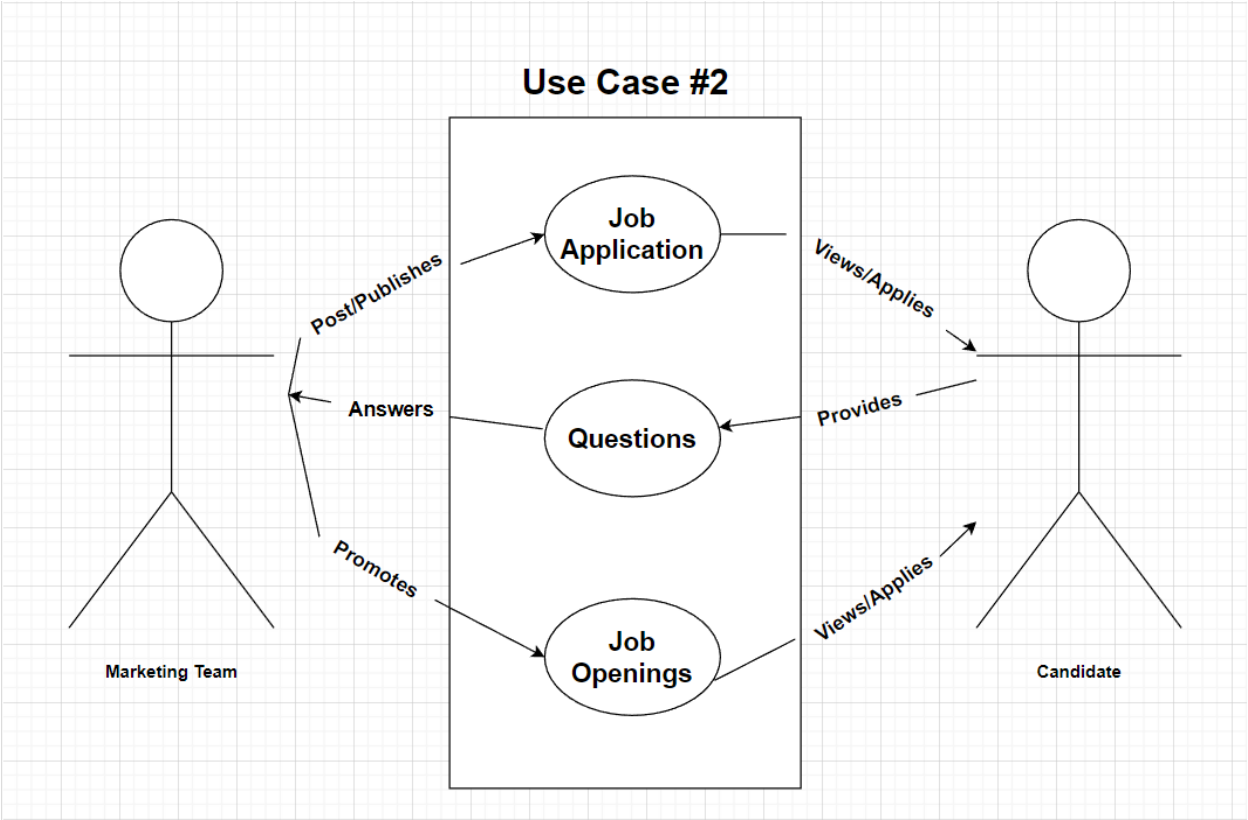
Diagram:



Use Case - 2

Use Case Title	Job Advertising
Actors	Marketing Team, Executive, Recruiter
Description	Within the Use Case of Job Advertising, the main purpose here is to advertise not only certain jobs but the Hiring Application on other media platforms. The Executive Director can involve both the Marketing Team and other Recruiters to promote open positions as well as promote the application used within the process. The marketing team can promote job openings through many means, such as social media, other job hiring sites, and through partnerships with organizations and universities. All these available job postings will then send the user to our Hiring Guru application, where they enter the Recruitment Pipeline Process. The Recruiter is also able to do the same job as the Marketing team, except their job is much more personable. Instead of posting entire jobs available, they will individually message talented individuals who they feel will succeed in a particular position.

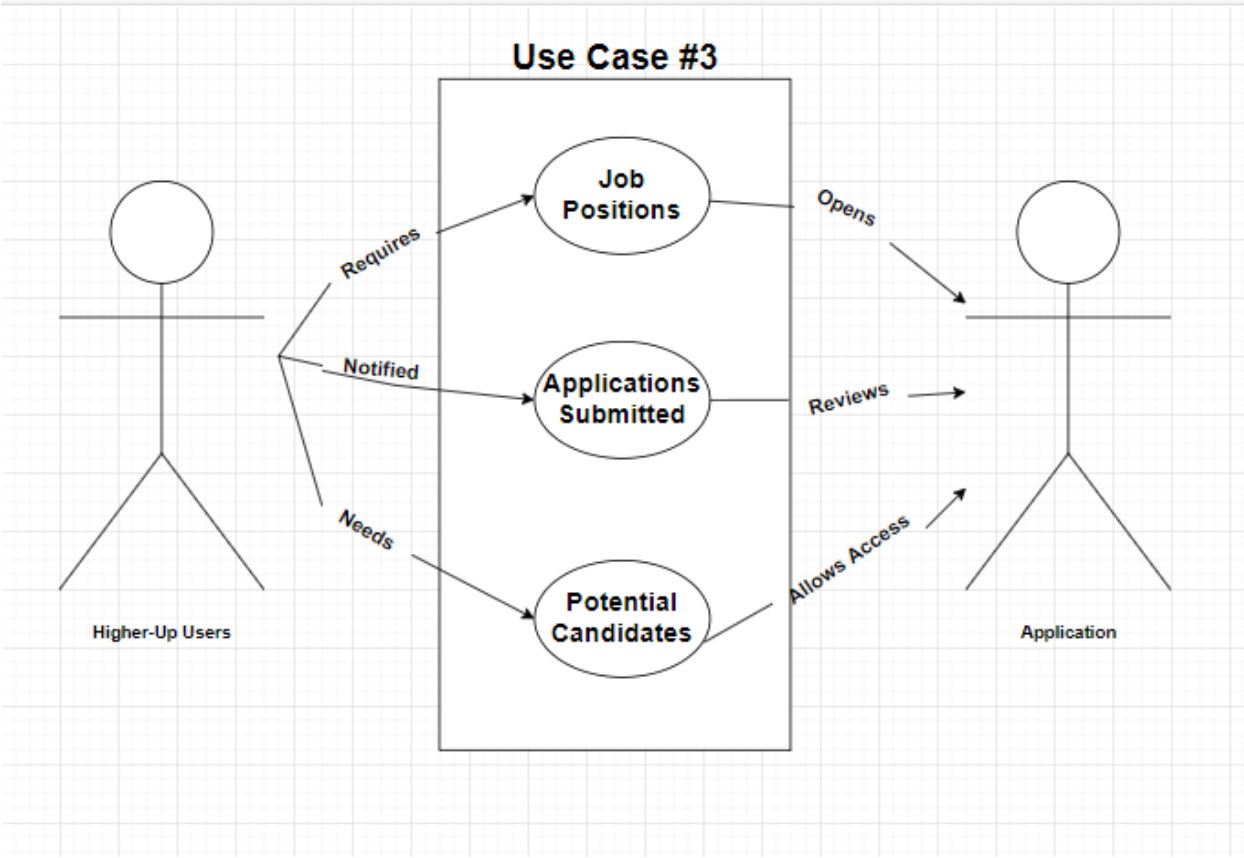
Diagram:



Use Case - 3

Use Case Title	Open a Position
Actors	HR Manager, Executive
Description	This use case begins when an HR Manager or a company Executive decides that the company needs to hire a new employee for some specific position. They will use the Hiring Guru application to open whatever role they need and include all the necessary and preferred requirements for that position.

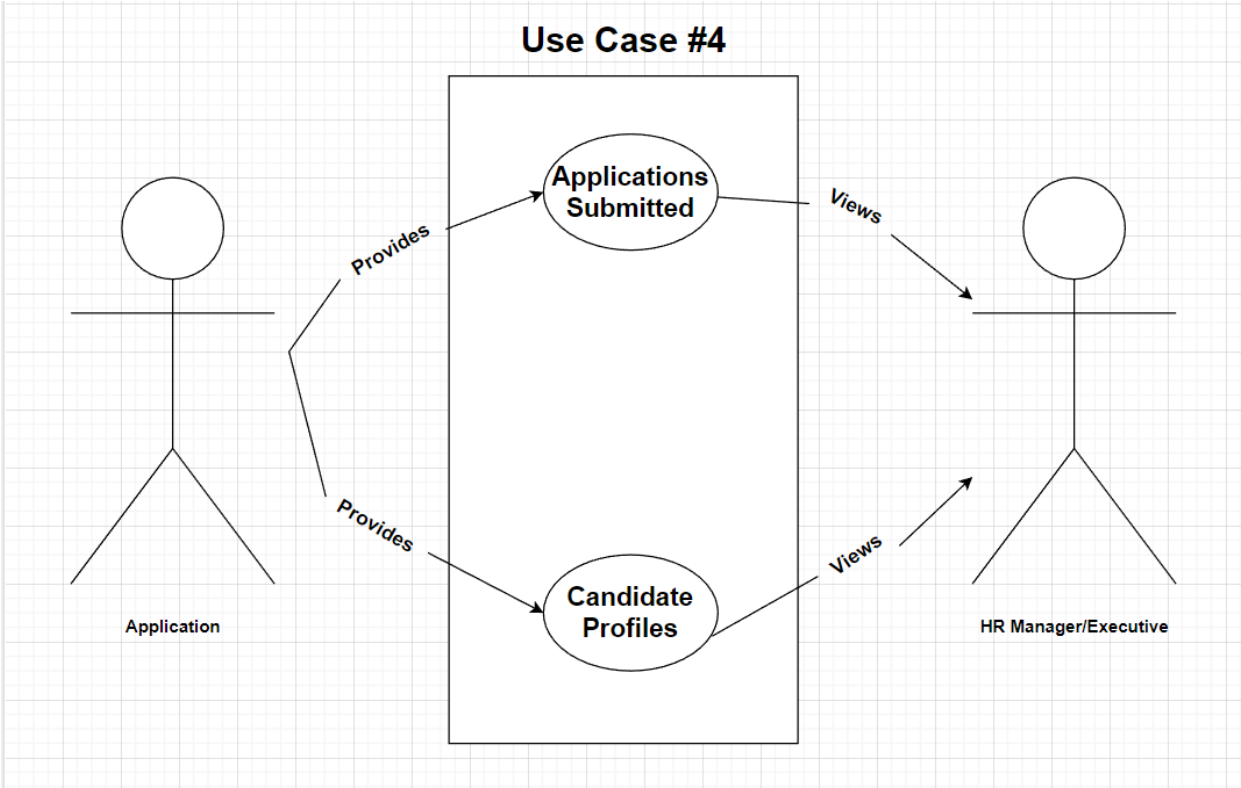
Diagram:



Use Case - 4

Use Case Title	See Applications for a Position
Actors	HR Manager, Executive
Description	The HR Manager and Executives want to view all the submitted applications for a job position so they can decide which Candidate moves onto the next stage of the Hiring Pipeline. The Users will be able to access the submitted applications and preview the documents within the program at a glance. For a more in-depth look, they are able to download the documents onto their device to open using the program of their choice on their devices.

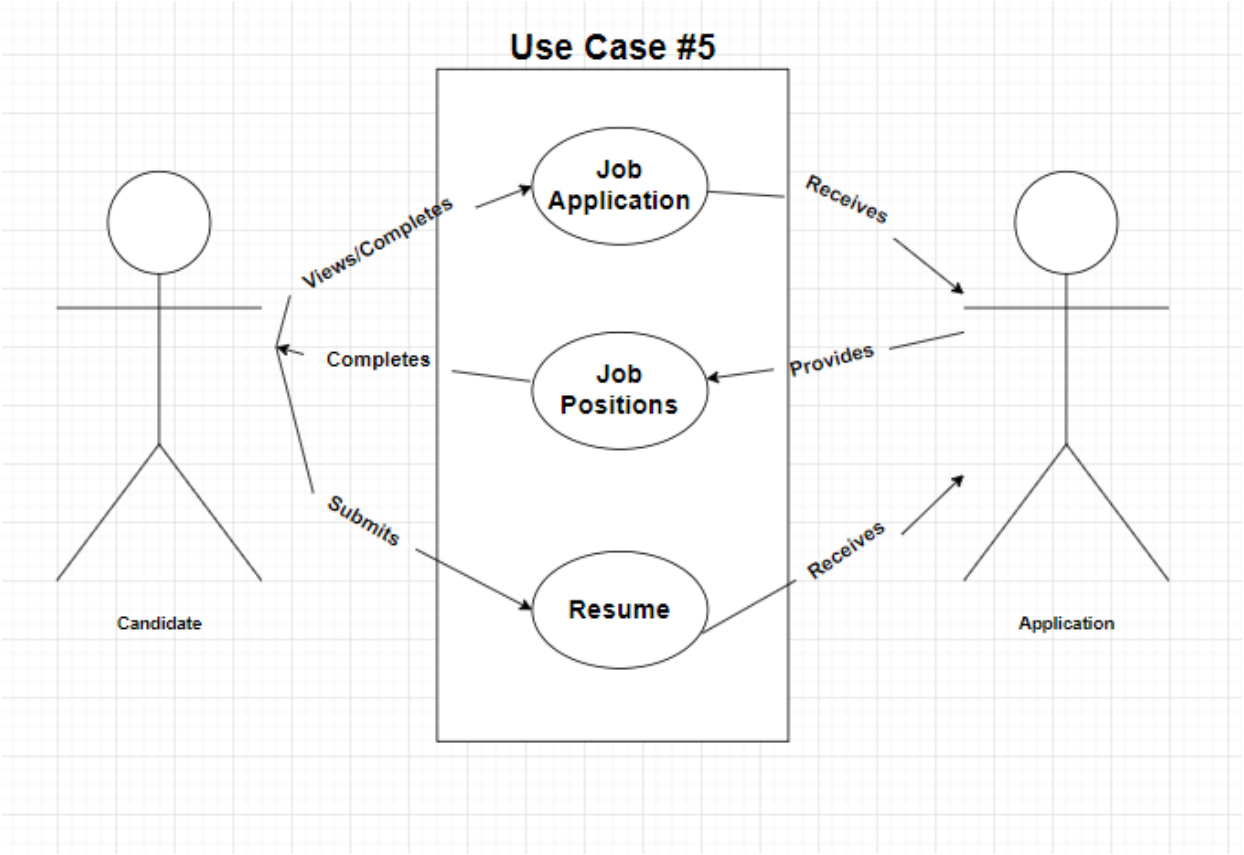
Diagram:



Use Case - 5

Use Case Title	Seeing and Applying for a Position
Actors	Candidate
Description	This use case details the ability of your average general user, specifically a candidate's ability to apply for a position at a specific company. Within this ability, candidates are able to search for a certain career of their preference based on their own skill sets and expertise and see what job positions are available to them that have been opened up by another user. The candidate, and only the candidate, have the ability to apply for whatever position that they desire, to then be put into the Recruitment Pipeline Process.

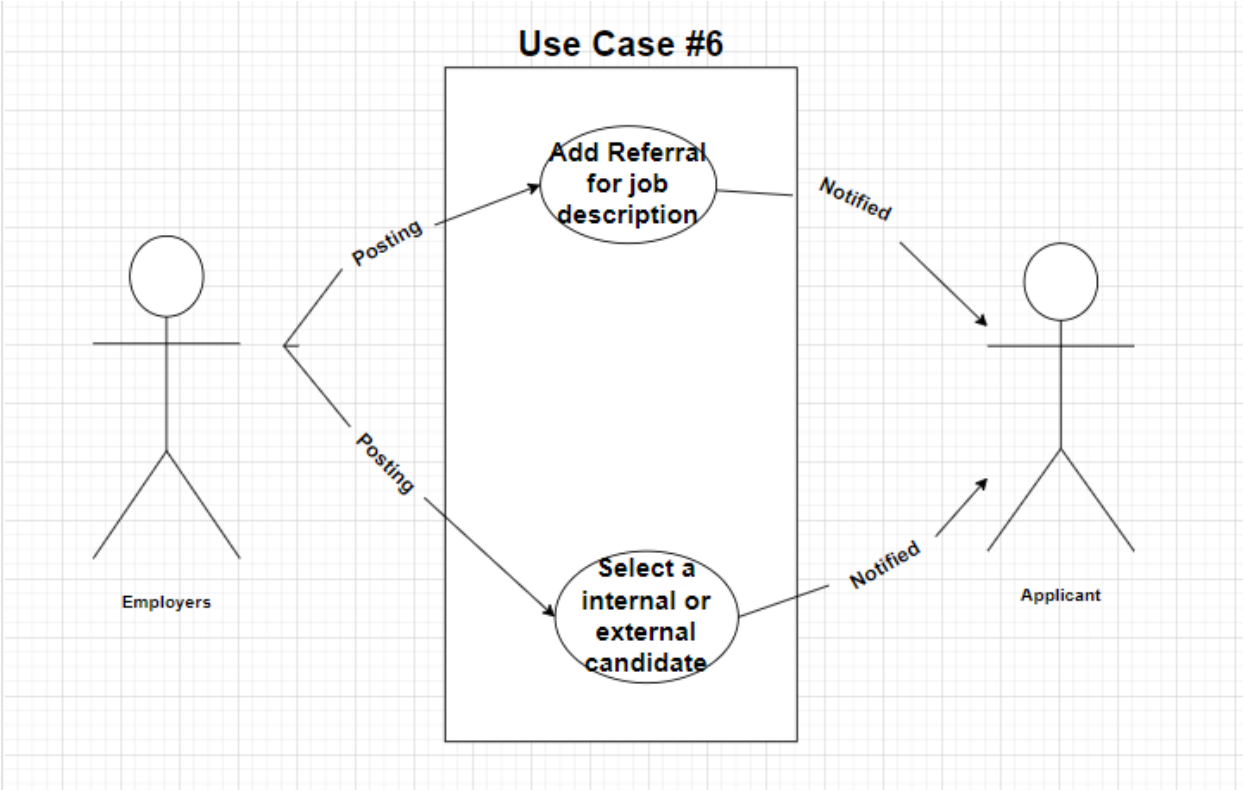
Diagram:



Use Case - 6

Use Case Title	Add Referral for position
Actors	Employers
Description	This allows for Employers to be able to grant a user a referral. The process involves

Diagram:

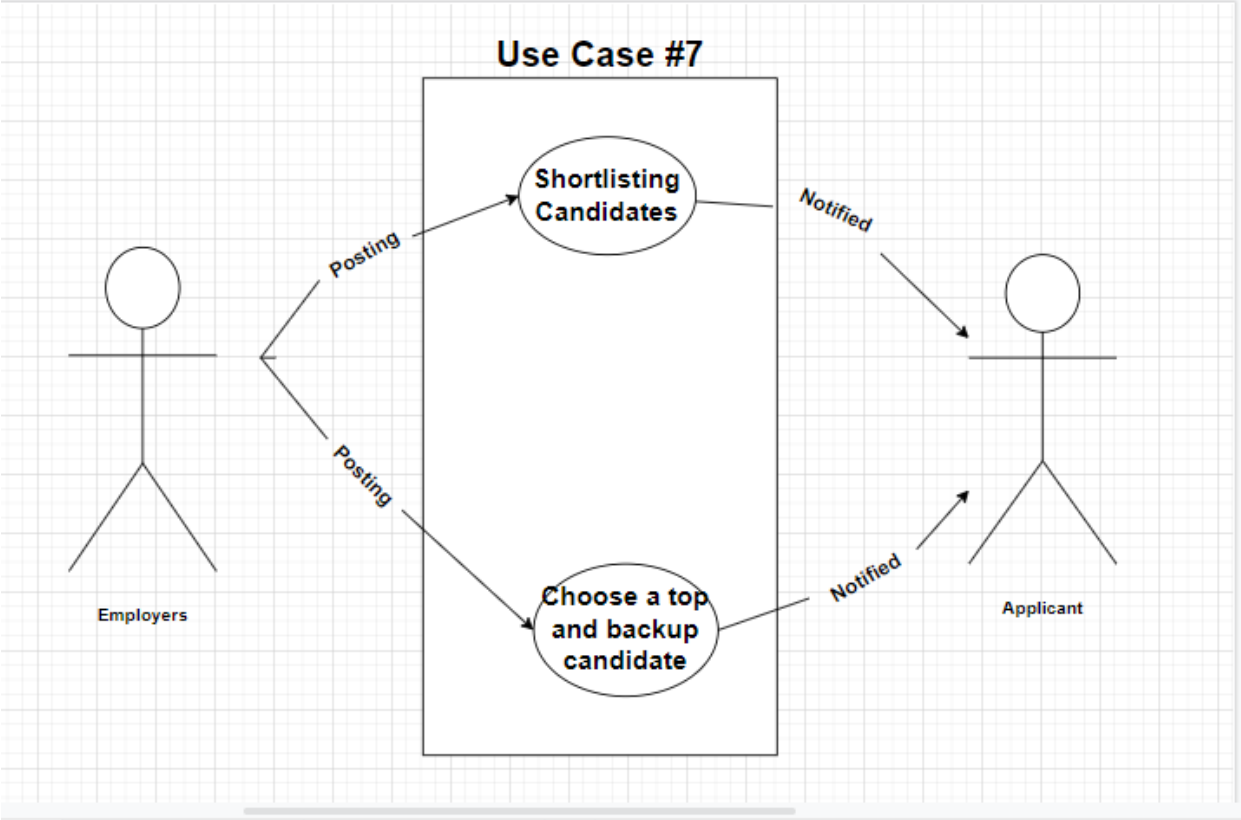


Use Case - 7

Use Case Title	Shortlisting candidates
Actors	HR Manager
Description	HR Managers rank candidates based off of the feedback and assessment provided by the interviewers. HR managers must come up with a shortlist criteria, in which they determine which candidates to move forward with given that some have the same feedback score from interviewers. Lastly, HR Managers should create a scorecard and crossmatch resumes against the scorecard to ensure the most suitable candidate is shortlisted. This allows for the pool of candidates to be more selective. Furthermore, shortlisting candidates will enable employers to narrow down the long list of all candidates, making it easier for them to choose a top

	and a backup candidate.

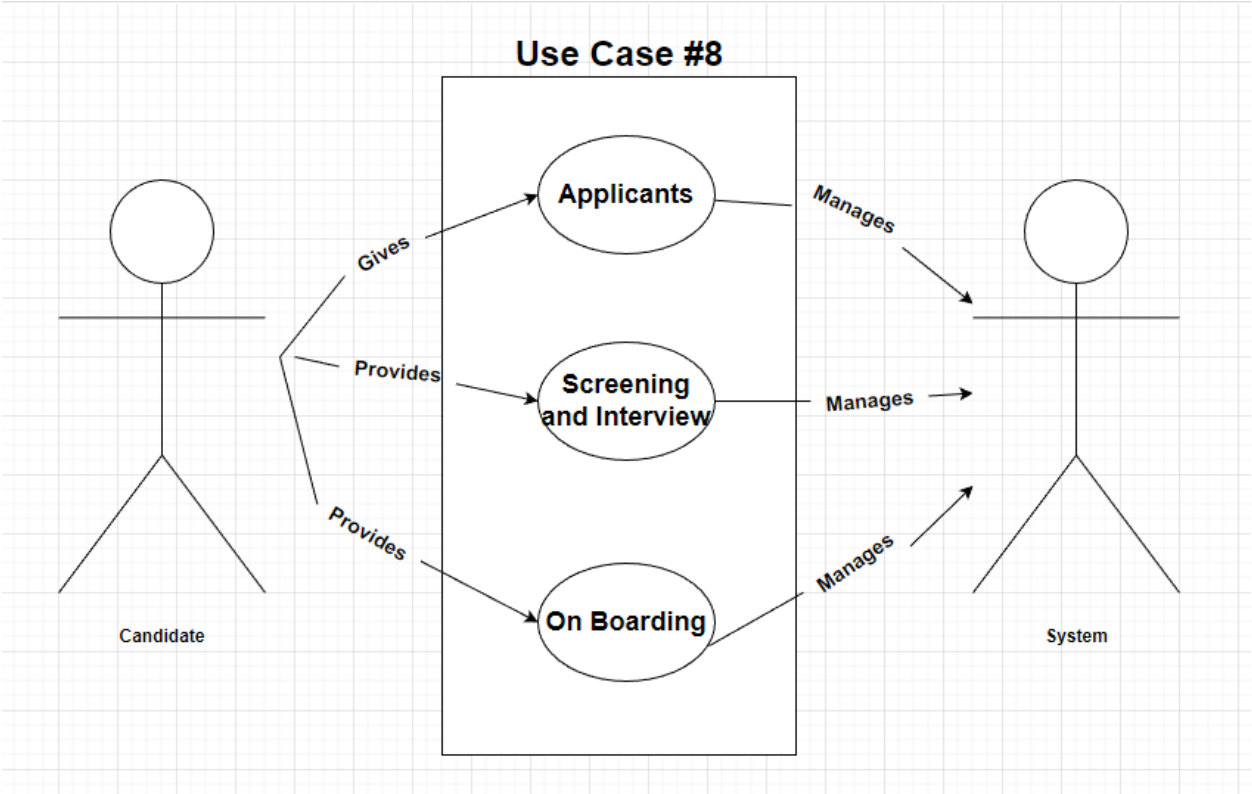
Diagram:



Use Case - 8

Use Case Title	Automatically execute the pipeline
Actors	System, Candidate
Description	The development pipeline is a series of commands that automatically executes in a row to test, build, or deploy a software product

Diagram:

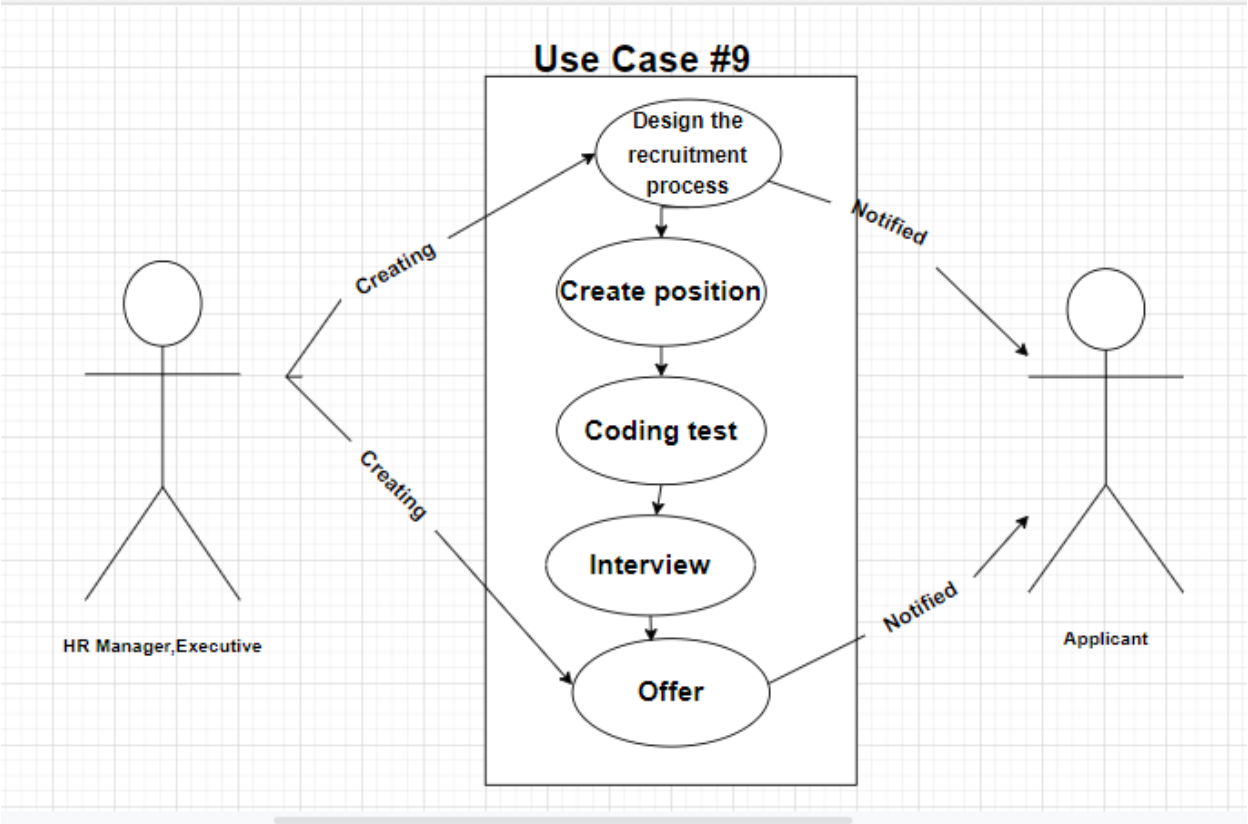


Use Case - 9

Use Case Title	Design Recruitment Process
Actors	HR Manager, Executive
Description	With this, the HR Manager and Executive decide on how to create the recruitment process. They start off by creating a job position with a specific job description to match their company's needs. Secondly, they create a coding test that tests the candidates on the specific skills needed for the job. Next, there are a series of interview rounds in which the interviewer tests the Candidate's technical and behavioral skills and comes up with a way of scoring the Candidate's answers. If the Candidate fails any of the interview rounds, then a rejection email must be sent to them. If the candidate passes all of the rounds, then an offer is presented to them, and the candidate must accept or

	decline the offer.

Diagram:

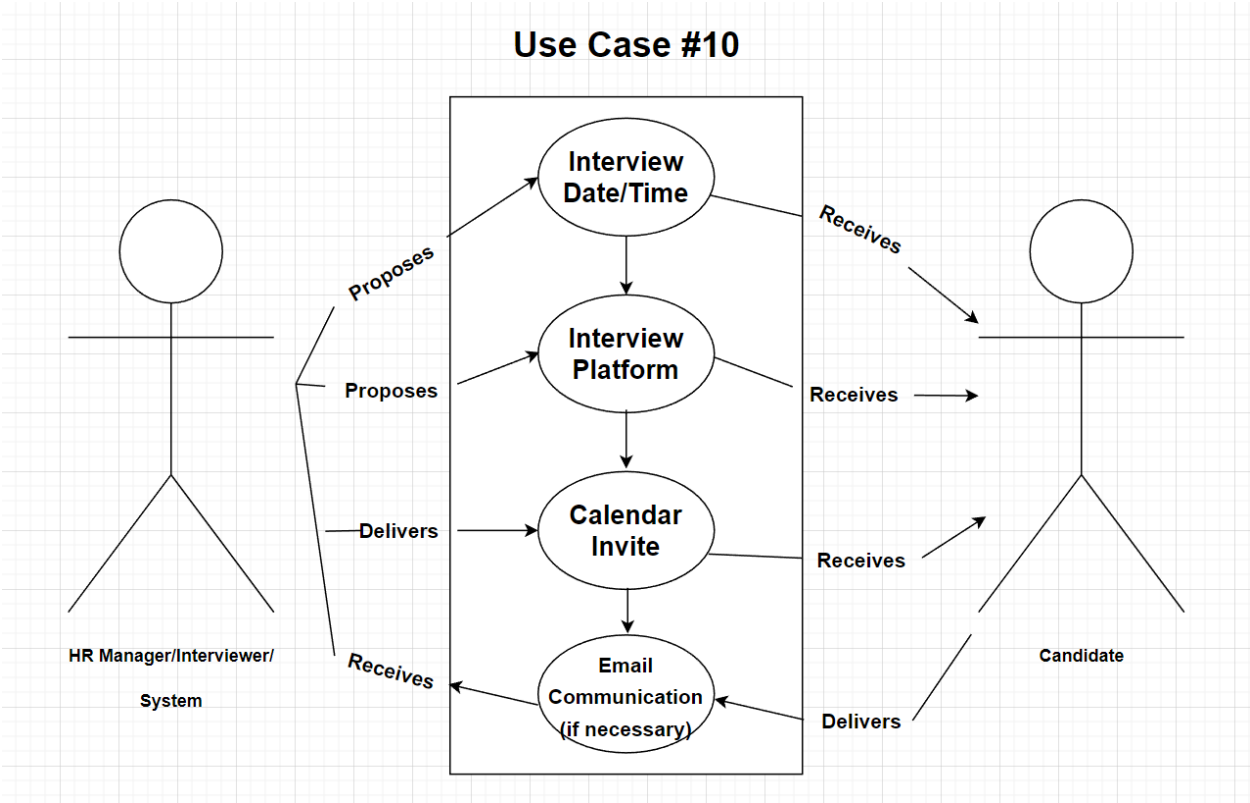


Use Case - 10

Use Case Title	Schedule an Interview
Actors	HR Manager, Interviewer, Candidate, System
Description	Employees at a company like a Candidate after viewing their submitted application and wish to pursue the next step of the Hiring Pipeline, an interview. Company employees such as HR Managers and Interviewers have the ability to schedule an interview with a Candidate(s) within the program. To schedule, the Users can choose options like the date, time, and platform for the interview to be conducted on. The Candidate is able to accept or reject a scheduled

	interview and, if rejected, communicate via email to reschedule. To keep the Users informed, the program also notifies all parties involved as a calendar event, in which an advance reminder notification date and time can be customized.

Diagram:

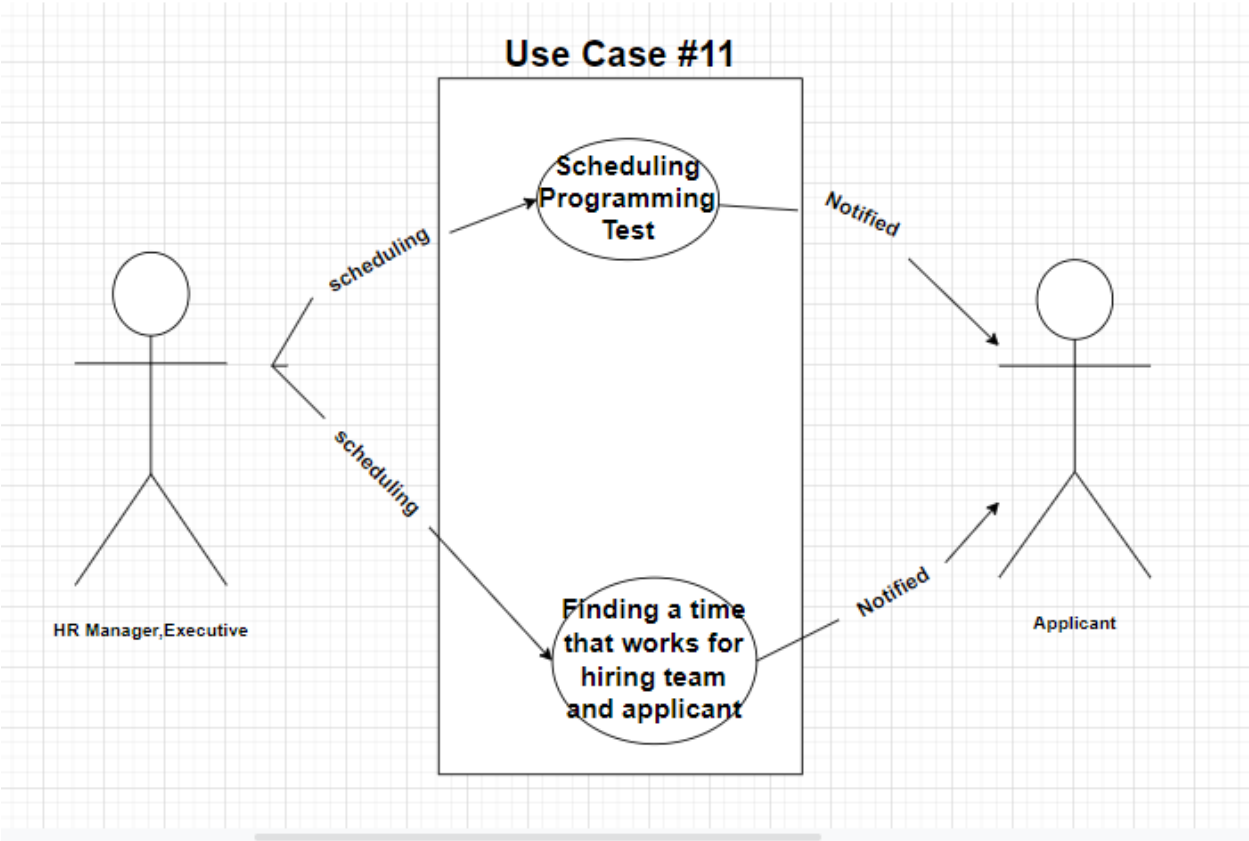


Use Case - 11

Use Case Title	Schedule a Programming Test
Actors	HR Manager, Executive
Description	This use case allows for either an HR manager or an executive to schedule a programming test for a candidate for a specific position. It involves having an HR Manager or Executive reach out to a candidate and give them a time frame in which they must complete the test, and if the candidate is unable to

	find a time to complete the test in the given time frame, the HR Manager and Executive schedule a new programming time.

Diagram:

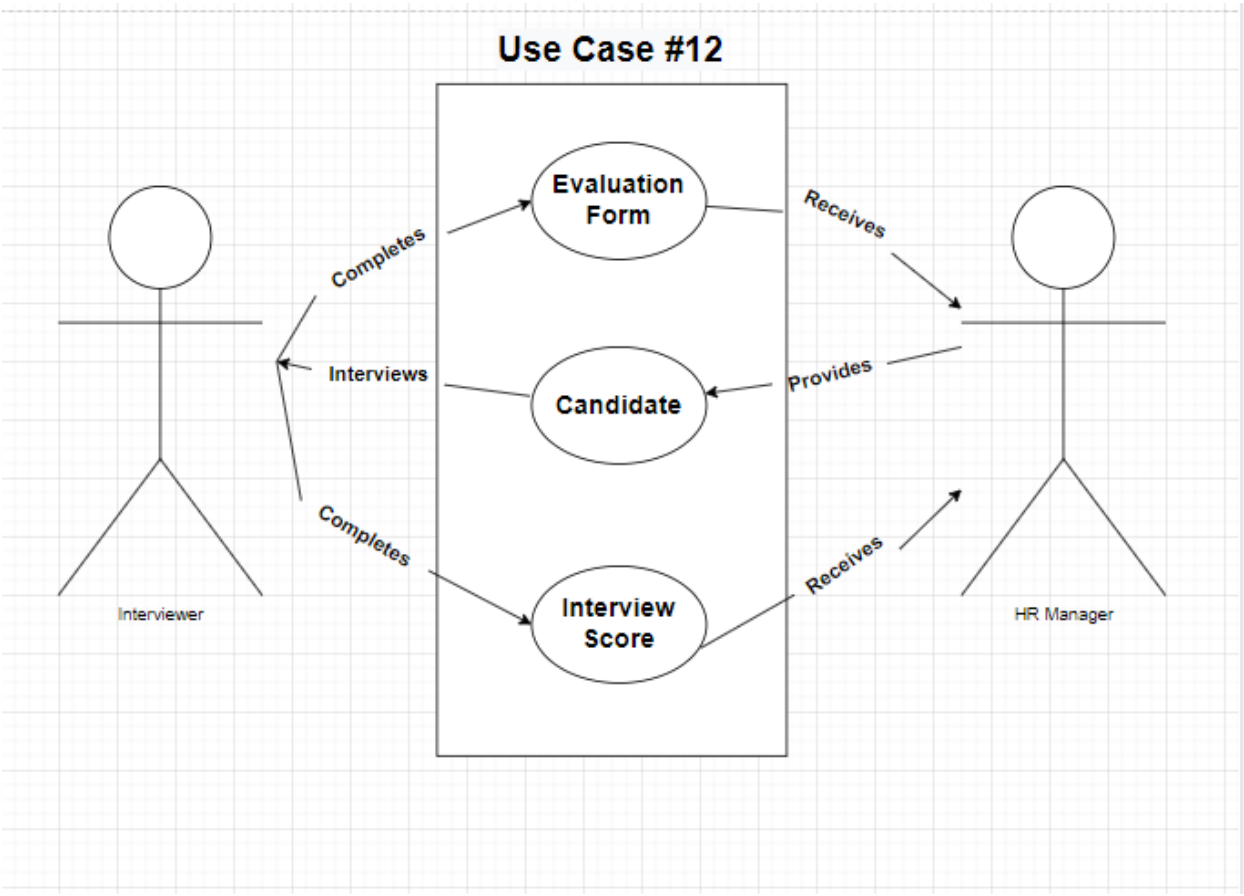


Use Case - 12

Use Case Title	Interviewer Evaluation
Actors	Interviewer, HR Manager, Executive
Description	This use case is describing the Interview Evaluation that only the Interview is capable of doing. To further describe this use case, a process that occurs within the Recruitment Pipeline is an Interview. An Interview is done between an Interviewer and a Candidate. Throughout the Interview and even after the Interview, the Interviewer will be taking a series of notes based

	on the actions and responses that the Candidate gives that may benefit them or harm their chance of getting the position. Once all these notes have been written in their entirety, the Interviewer then gives the Candidate an Interviewer Evaluation. This is essentially a score for the Candidate that will help others, such as the HR Manager and/or the Executive, determine if this candidate is capable of moving on to the next step within the Recruitment Pipeline Process.

Diagram:

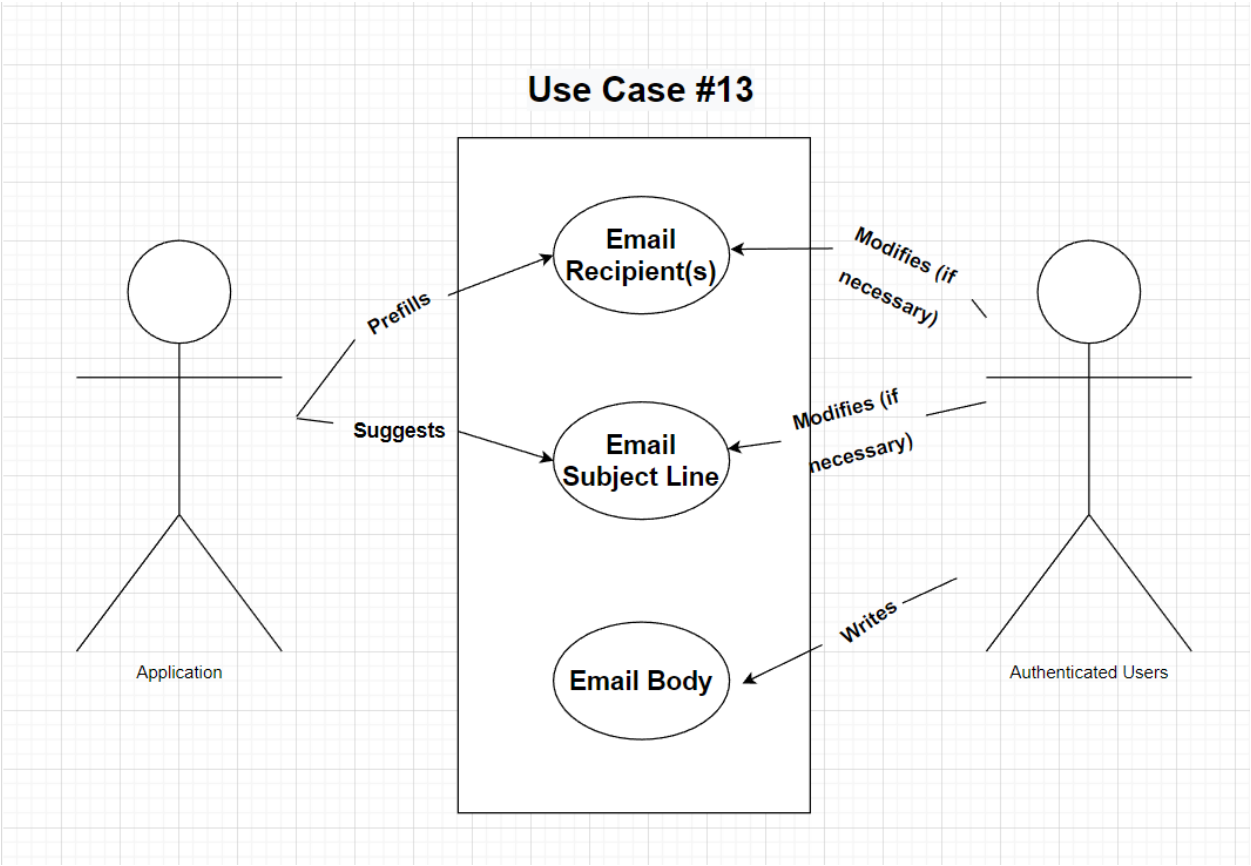


Use Case - 13

Use Case Title	Send Communication Emails

Actors	Candidate, Interviewer, HR Manager, Executive, Employer
Description	This case gives all Authenticated Users (Candidate, Interviewer, HR Manager, Executive, Employer) the ability to send prefilled emails for easy communication with each other. The Users shall be able to click on a button that opens the default email client on their device, with the email address of the recipient(s) prefilled and a suggested subject line.

Diagram:

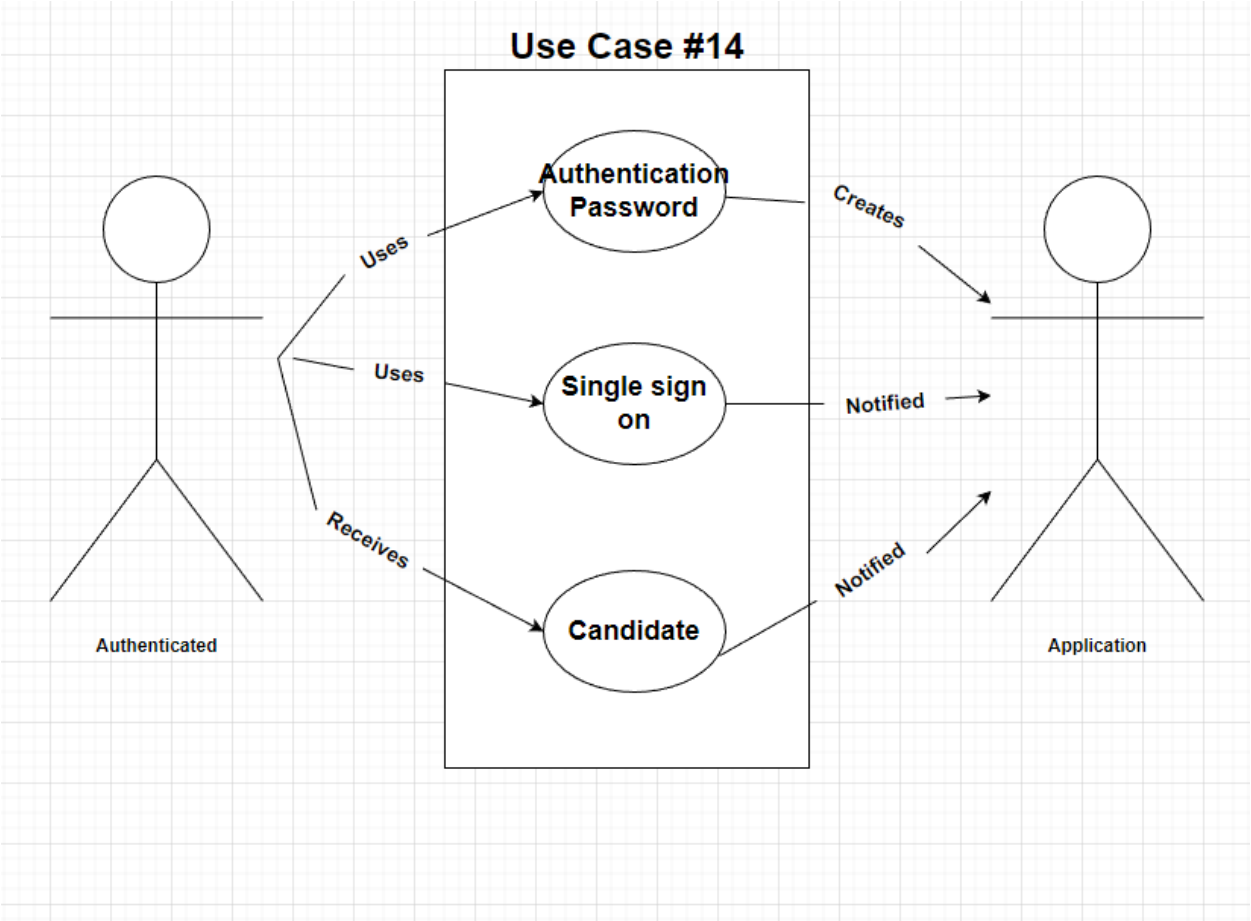


Use Case - 14

Use Case Title	Authentication

Actors	Authenticated Users (Candidate, Interviewer, HR Manager, Executive, Employer)
Description	This use case allows authenticated users to manage emails. Authenticated users are allowed to log into a portal. This use case starts when a user enters the site. If the user logs in and is authenticated then they will be directed to a portal that has access to more information and tools than just a general user. The user wants to log in. If the user is not yet registered then there is a button that they can click that will redirect them to a registration form. Once registered the user can then log in. They will start off as a general user with only general user abilities and access. The account can be given flags by an admin such as Candidate, Interviewer, HR Manager, Executive, or Employer which will give them access to specific resources and abilities.

Diagram:

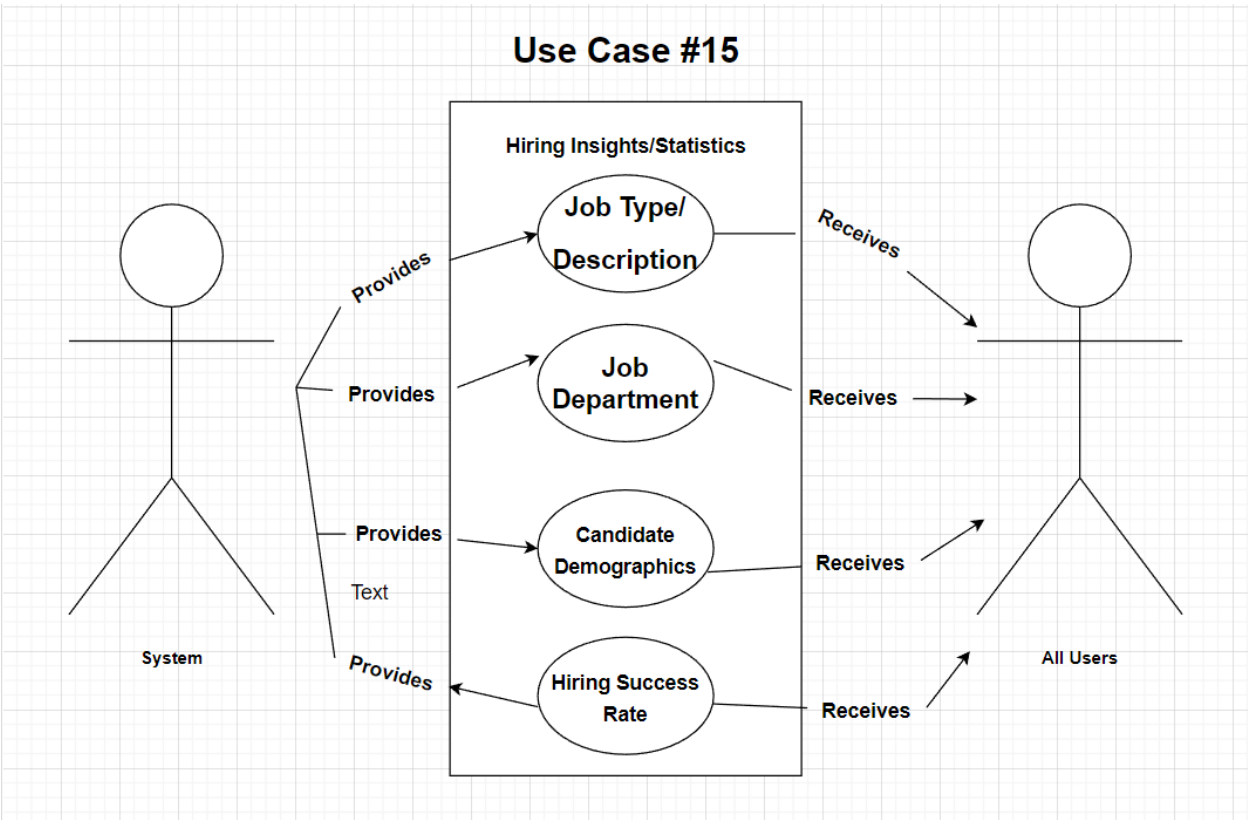


Use Case - 15

Use Case Title	Hiring Insights
Actors	System, Candidate, Interviewer, HR Manager, Executive, Employer
Description	This use case allows for Candidates, Interviewers, HR Managers, Executives, and Employers to view hiring insights in the System from Candidates who have previously and are currently going through the Recruitment Hiring Pipeline. Users can view statistics such as job type, department, Candidate demographics, location, and hiring success rate for a given company/position. Employees at a company and hiring staff can use these metrics to see how well

	they are doing, while Candidates can gauge their chances of getting the position or not. Ex. A candidate wants to check the hiring success rate of the entry level UX designers. Basic Path: 1. The candidate logs into the portal 2.

Diagram:

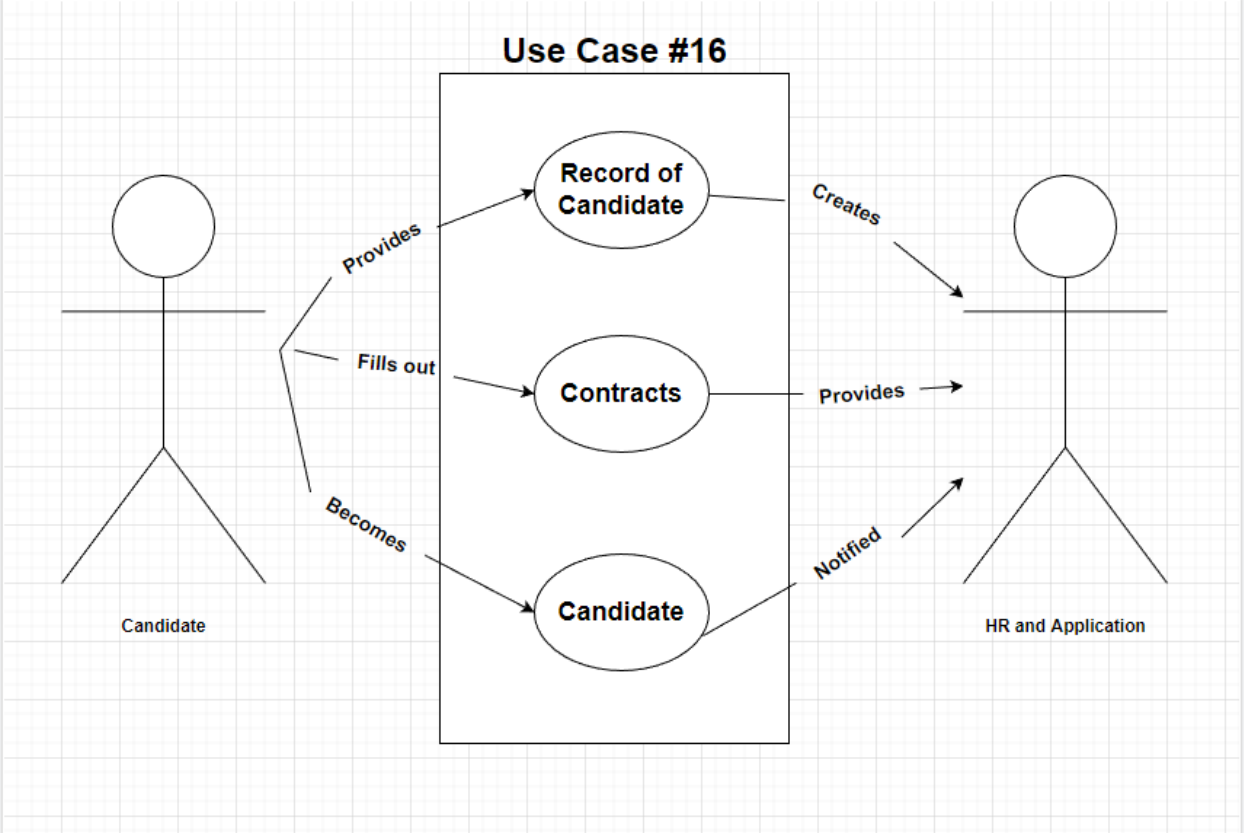


Use Case - 16 (Eric)

Use Case Title	Candidate Onboarding
Actors	Candidate, HR Manager, Application

Description	This use case begins when a general user is given the candidate flag. Any user that holds the candidate flag will be given specific permissions and an onboarding program will commence geared towards transitioning the candidate to assume a role in the company. The candidate will be evaluated by the HR Manager and the Application.

Diagram:

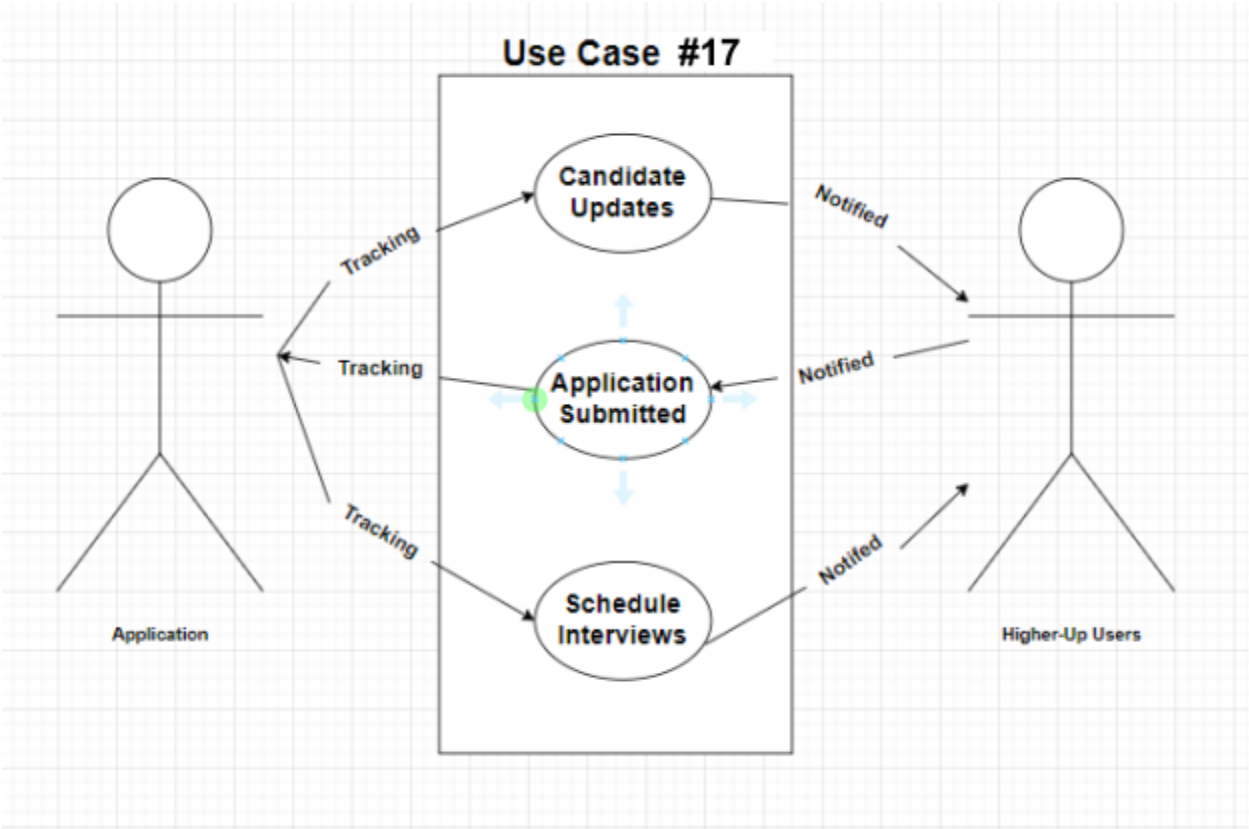


Use Case - 17 (Mamadou)

Use Case Title	Task Tracking
Actors	HR Manager, Executive, Tech Recruiter, Interviewer,
Description	The main purpose of this use case is for all employees who have special access to the Hiring Guru and are able to see tasks needed to be done for

	every candidate that is going through the Recruitment Process. These individuals have perks the candidates do not have which allows for them to view all the needed tasks to be completed.

Diagram:



List of Main Data items and entities

HR Manager

An HR Manager does the following:

- An HR manager is the main actor and the most frequent user of this application, responsible for managing the complete hiring process.

Candidate

A Candidate does the following:

- The Candidate is a prominent user within the application, responsible for going through the entire Recruitment Pipeline Process from start to finish

Executive

An Executive does the following:

- The Executive entity is an entity that has the same influence as the HR Manager but is one above the HR Manager.

Tech Recruiter/Marketing

A Tech Recruiter does the following:

- A Tech Recruiter, or the Marketing Team, is the main user that starts the entire Recruitment Pipeline Process within the Hiring Guru, responsible for hiring the Candidate that will undergo the process.

Interviewer

An Interviewer does the following:

- An Interviewer is a user who is an employee within the desired company by the general user and interviews them to see if they qualify for the specific role.

General User

A General User can do the following:

- A General User is a user who is only exploring the application and has the most basic features accessible to them, most often an unauthenticated user.

Initial List of Functional Requirements

Authorized User Requirements

Requirement ID	Requirement Statement
FRAUTH001	Authorized user shall be able to log out anytime
FRAUTH002	Authorized user shall be able to create an organization
FRAUTH003	Authorized user shall be able to modify organization detail
FRAUTH004	Authorized user shall be able to invite employees to the organization
FRAUTH005	Authorized user shall be able to create permission groups
FRAUTH006	Authorized user shall be able to add employees to user groups
FRAUTH007	Authorized user shall be able to remove employees from user groups
FRAUTH008	Authorized Users shall be able to create a new role (e.g., QA Engineer, Software Engineer, etc.)
FRAUTH009	Authorized User shall be able to update a role
FRAUTH010	Authorized User should be able to see the list of all roles
FRAUTH011	Authorized User shall be able to remove a role from the system
FRAUTH012	Authorized User shall be able to open a new position for a particular role
FRAUTH013	Authorized User shall be able to close a listed position
FRAUTH014	Authorized User shall be able to see all the applications for a specific position
FRAUTH015	Authorized User shall be able to add referral for a specific position
FRAUTH016	Authorized User shall be able to see all the referrals for a specific position
FRAUTH017	Authorized User shall be able to drop an application
FRAUTH018	Authorized User shall be able to design the hiring process for a

	position
FRAUTH019	Authorized User shall be able to update the hiring process for a position
FRAUTH020	Authorized User shall be able to see the hiring process for a position
FRAUTH021	Authorized User shall be able to accept an application for a specific position
FRAUTH022	Authorized User shall be able to see a list of all the accepted applications
FRAUTH023	Authorized User shall be able to remove a Candidate from the hiring process
FRAUTH024	Authorized User shall be able to add an Employee
FRAUTH025	Authorized User shall be able to view a list of all Employees
FRAUTH026	Authorized User shall be able to remove an Employee from the company
FRAUTH027	Authorized User shall be able to schedule an interview of the Candidate
FRAUTH028	Authorized User shall be able to send a calendar invite to the Interviewers and the Candidate
FRAUTH029	Authorized User shall be able to see the schedule detail of an interview of the candidate
FRAUTH030	Authorized User shall be able to see the detail of an interview of the candidate
FRAUTH031	Authorized User shall be able to schedule a programming test of the candidate
FRAUTH032	Authorized User shall be able to update the time and date of a programming test
FRAUTH033	Authorized User shall be able to add the feedback/decision of a programming test
FRAUTH034	Authorized User shall be able to attempt a programming test

FRAUTH035	Authorized User should be able to create a job post on LinkedIn
FRAUTH036	Authorized User should be able to create a new advertising campaign on LinkedIn
FRAUTH037	Authorized User should be able to analyze LinkedIn profiles using Artificial Intelligence
FRAUTH038	Authorized User shall be able to send a message to the matching profiles to see if they are interested
FRAUTH039	Authorized User shall be able to add profiles of interest to the recruitment pipeline
FRAUTH030	Authorized User shall be able to like a listed position
FRAUTH041	Authorized User shall be able to update his profile
FRAUTH042	Authorized User shall be able to search for job by the posted time
FRAUTH043	Authorized User shall be able to post the job advertisement on LinkedIn
FRAUTH044	Authorized User shall be able to schedule a programming test for the candidate
FRAUTH045	Authorized User shall be able to schedule an interview of the candidate
FRAUTH046	The system shall send email to the candidate once programming test is scheduled
FRAUTH047	The system shall send email to the candidate once an interview is scheduled
FRAUTH048	Authorized User shall be able to send acceptance email to the candidate
FRAUTH049	Authorized User shall be able to send rejection email to the candidate
FRAUTH050	Authorized User shall be able to send a background check must to a candidate, which must be passed before being able to accept a job offer

FRAUTH051	Authorized User shall be able to review each applicant’s resume and experience prior to interviewing
FRAUTH052	Authorized User shall be able to determine how to score candidate's responses based off of company’s expectations
FRAUTH052	Authorized User shall be able to answer Candidate's questions appropriately
FRAUTH053	Authorized User shall be able to administer situational and personality tests to narrow the list of Candidates
FRAUTH054	Authorized User shall be able to write a job description
FRAUTH055	Authorized User shall be able to ensure that the employment paperwork is correct and complete
FRAUTH056	Authorized User shall be able to ensure that the application should be easy to complete
FRAUTH057	Authorized User shall be able to schedule a phone screen
FRAUTH058	Authorized User shall be able to schedule a skills test
FRAUTH059	Authorized User shall be able to ensure appropriate questions are on the skills test related to the job requirement
FRAUTH060	Authorized User shall be able to schedule a new time for the skills test if Candidate is unable to complete it within the provided time frame
FRAUTH061	Authorized User shall be able to allow for convenient and quick channels of communication for the candidate to ask questions throughout the interview process
FRAUTH062	System should automatically add an accepted application to the hiring process
FRAUTH063	System should automatically move an accepted application to the next step when a step is successful
FRAUTH064	System shall automatically notify the Candidate about the success/failure in a step
FRAUTH065	Authorized User shall be able to mark an Employee as an

	Interviewer
FRAUTH066	Authorized User shall be able to see a list of all Interviewers
FRAUTH067	Authorized User shall be able to remove an Employee from the Interviewer status
FRAUTH068	System should automatically send a calendar invite to the Interviewers and the Candidate
FRAUTH069	Authorized User shall be able to update the details of an interview - date, interviewers etc.
FRAUTH070	Authorized User shall be able to see the results of the programming test of a Candidate
FRAUTH071	Authorized User should be able to create a job post on Handshake
FRAUTH072	Authorized User should be able to update job posts created using the System
FRAUTH073	Authorized User should be able to pause an advertising campaign
FRAUTH074	Authorized User should be able to see statistics of an advertising campaign
FRAUTH075	Authorized User should be able to scrape profiles from LinkedIn given a set of keywords
FRAUTH076	Authorized User shall be able to save the list of matching profiles
FRAUTH077	Authorized User shall be able to ensure the Interviewer understands what skills and experience the job require
FRAUTH078	Authorized User shall be able to ensure the Interviewer has the proper skills to take the interview
FRAUTH079	Authorized User shall be able to create relevant application questions
FRAUTH080	Authorized User shall be able to identify the hiring need
FRAUTH081	Authorized User shall be able to determine if an external hire- go through usual advertising channels

FRAUTH082	Authorized User shall be able to perform a reference check
FRAUTH083	Authorized User shall be able to create a checklist of all of the paperwork related to employment the Candidate needs to fill out after accepting an offer
FRAUTH084	Authorized User shall be able to give Candidate ample time to complete the employment paperwork
FRAUTH085	Authorized User shall be able to, if no candidates meet the hiring criteria, decide whether to restart the hiring process
FRAUTH086	Authorized User shall be able to ask the candidate about their salary expectations
FRAUTH087	Authorized User shall be able to, if they wish to restart the hiring process, decide how to adjust the hiring criteria to yield more candidates
FRAUTH088	Authorized User shall be able to, if they wish to restart the hiring process, decide how to advertise the position to more favorable candidates
FRAUTH089	Authorized User shall be able to, if the candidate is ready to accept the job offer, decide which elements of the offer letter are negotiable
FRAUTH090	Authorized User shall be able to ensure that Artificial Intelligence is able to parse through the candidate's resume to fill out fields in the application
FRAUTH091	Authorized User shall be able to give a certain time frame for the candidate to accept the offer letter
FRAUTH092	Authorized User shall be able to create convenient methods for applicants to be able to withdraw from the application process at any time
FRAUTH093	Authorized User shall be able to provide detailed feedback on what they liked about the candidate's performance but also reservations they might have

FRAUTH112	Authorized User shall be able to create activities for the Candidate to do
FRAUTH113	The application should be able to have a total cost calculator
FRAUTH114	The application should be able to have a digital document process
FRAUTH115	The application should be able to have a workload manager
FRAUTH116	The application should be able to have a candidate portal
FRAUTH117	Screening function for making sure the candidate is cleared
FRAUTH118	Authorized User shall be able to create a database for tracking candidate's progress
FRAUTH119	Authorized User shall be able to assign role to candidate
FRAUTH120	The application should be able to provide job-specific programs for each candidate
FRAUTH121	The application should be able to grant access to technical tools needed by the candidate
FRAUTH122	The application should be able to have a paid or unpaid boarding process
FRAUTH123	The application should be able to allocate mentor for Candidate
FRAUTH124	The application should be able to create a schedule for the Candidate
FRAUTH125	The application should be able to organize the workspace
FRAUTH126	Introduce the candidate to technologies they will need to learn how to use
FRAUTH127	Authorized User shall be able to prepare orientation with other Candidates
FRAUTH128	The application shall have an administration tools for the interviewer

FRAUTH129	Authorized User shall be able to Take in any comments or complaints that the candidate might have
FRAUTH130	Authorized User shall be able to Organize meetings between the employer and the candidates
FRAUTH131	Manage any health benefits that should be given to the employee
FRAUTH132	Authorized User shall be able to request Employment background screening
FRAUTH133	Authorized User shall be able to Send offer letter to candidate
FRAUTH134	Authorized User shall be able to Email reminders of the next steps in onboarding
FRAUTH094	System should automatically remove the Candidate from the hiring process if a step is failed
FRAUTH095	Authorized User shall be able to view the detail of an Employee
FRAUTH096	Authorized User shall be able to update the detail of an Employee
FRAUTH097	Authorized User shall be able to allow an interviewer to set their availability time
FRAUTH098	Authorized User shall be able request Candidate availability for an interview
FRAUTH099	System should automatically schedule an interview of the Candidate
FRAUTH100	System should automatically schedule a programming test if it is the current step
FRAUTH101	Authorized User should be able to create a job post on Stackoverflow
FRAUTH102	Authorized User should be able to create a new advertising campaign on YouTube
FRAUTH103	Authorized User should be able to create a new advertising campaign on StackOverflow

FRAUTH104	Authorized User should be able to calculate the budget and allocate money for specific advertising campaigns
FRAUTH105	Authorized User shall be able to calculate budget to see how many open positions are possible to create
FRAUTH106	Authorized User should be able to calculate the budget to allocate for training interviewers on how to conduct an interview
FRAUTH107	Authorized User shall be able to talk about the day-to-day duties and inform the candidate about the work culture
FRAUTH108	Authorized User shall be able to identify if the hire is an external or internal applicant
FRAUTH109	Authorized User shall be able to determine if an internal hire-minimal advertising is needed
FRAUTH110	Authorized User shall be able to design a recruitment pipeline for individual candidates
FRAUTH111	Authorized User shall be able to provide specific feedback as to why they didn't pass that round
FRAUTH135	Authorized User shall be able to prepare new Candidate for training
FRAUTH136	Authorized User shall be able to create a new onboarding program
FRAUTH137	The application should be able to have a calculate candidate hours
FRAUTH138	The application should be able to have a decrease wage
FRAUTH139	The application should be able to have a give raise
FRAUTH140	The application should be able to have a calculate trust level
FRAUTH141	The application should be able to have a survey candidate
FRAUTH142	The application should be able to have a find jobs for the candidate within company
FRAUTH143	The application should be able to have a create presentations

FRAUTH144	The application should be able to host meetings
FRAUTH145	The application should be able to have a way to communicate with other data items
FRAUTH146	Authorized User shall be able to create tasks for candidate
FRAUTH147	The application should be able to have a deadline setter
FRAUTH148	Authorized User shall be able to set goals for the candidate to complete
FRAUTH149	Authorized User shall be able to find candidate function
FRAUTH140	Authorized User shall be able to pause candidate function
FRAUTH141	Authorized User shall be able to remove candidate function
FRAUTH142	The application should be able to have memory allocation for Candidate information
FRAUTH143	The application should be able to organize team dynamics
FRAUTH144	The application should be able to organize team dynamics
FRAUTH145	The application should be able to organize where and how they will be working
FRAUTH146	Authorized User shall be able to do a performance review scheduling
FRAUTH147	Manage space for Candidates
FRAUTH148	Authorized User shall be able to display information needed by the candidate

General User Requirements

Requirement ID	Requirement Statement
FRGENR000	General user shall be able to sign up directly for an individual account
FRGENR001	General user shall be able to signup using Google account

FRGENR002	General user shall be able to signup using a Microsoft account
FRGENR003	General user shall be able to login using Google account
FRGENR004	General user shall be able to login using Microsoft account
FRGENR005	General user shall be able to login using Username and Password
FRGENR006	General user shall be able recover his/her account using email
FRGENR007	General User shall be able to go back to the home screen anytime by clicking on the HiringGuru logo
FRGENR008	Authorized User shall be able to like a listed position
FRGENR009	General User shall be able to see the HiringGuru about page in the page footer
FRGENR010	General User shall be able to see the product features in the home page
FRGENR011	General User shall be able to search for jobs on the home page
FRGENR012	General User shall be able to apply for a job
FRGENR013	General user shall be able to book a product demo
FRGENR014	General user shall be enable or disable cookies
FRGENR015	General user shall be able to chat with the organization bot
FRGENR016	General user shall be able to see product pricing
FRGENR017	General User shall be able to see statistics of a job post

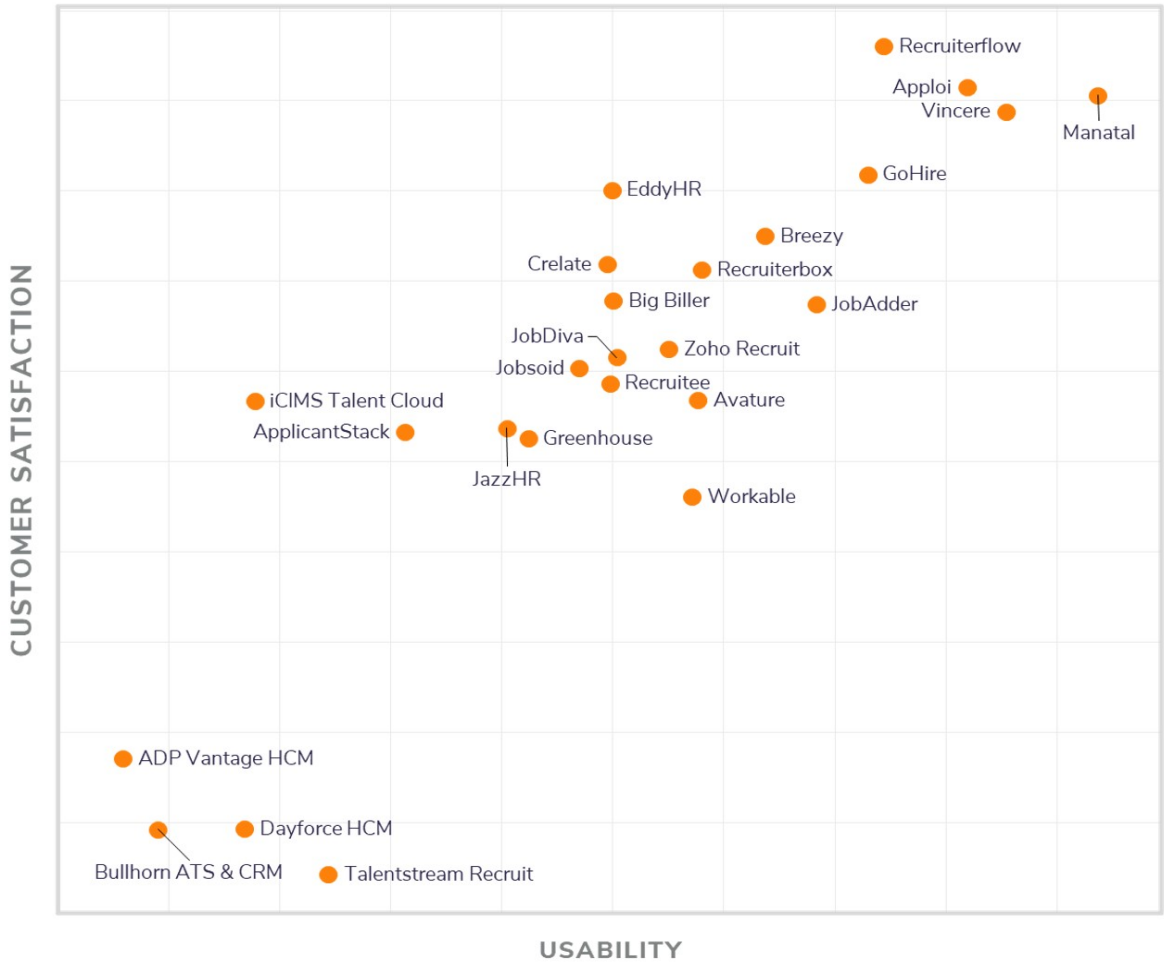
List of Non-Functional Requirements

1. The application shall have efficient response time
2. The application shall have an easy to use platform
3. The application shall have sufficient space for all data
4. General User shall be able to create an account
5. General User shall be able to add required information to account
6. General User shall be able to add a resume to account
7. General User shall be able to able to search for jobs
8. General User shall be able to able to view necessary information underneath job

9. General User shall be able to able to add necessary information underneath job
10. The application should have an easy to follow design
11. The application should have a unique and Colorful Design
12. The application should have no unnecessary information on the page
13. The application shall have strong security
14. General User shall be able to hide password when typing
15. General User shall be able to view password when typing
16. General User shall be able to choose username that's not taken yet
17. General User shall be able to choose email that's not taken yet
18. General User shall be able to recover lost account
19. General User should be able to see the location of job
20. General User shall be able to easily see what the job is requiring
21. General User shall be able to see what jobs they have previously viewed
22. General User shall be able to see what jobs they have already applied for
23. General User shall be able to login easily into account
24. General User shall be able to log out easily of account
25. General User shall be able to option to have account sign in remembered
26. Authorized User shall be able to to see which exact candidate is applying
27. Authorized User shall be able to to view resume attached to candidate profile
28. Authorized User should be able to contact candidate freely
29. Authorized User shall be able to schedule meeting with candidate freely
30. Authorized User should be able to have better perks than General User
31. The Application should be able to have sufficient space for all users on platform
32. The Application should be able to have a limit for resume size
33. The Application should be able to have a preferences in how resume should be submitted

Competitive Analysis

Current Market Space



(Source: Software Advice)

Analysis of Top Products

Comparison #1

Feature/Company	Manatal	BambooHR	Breezy
Strengths	-More feature rich and straightforward than other ATSs	-Convenient to use and easy to implement	-Solid reporting and analytics tools -High performance

	-Extremely user friendly -Highly accessible and simple to use -An abundant amount of features -Mobile-friendly -Tracking candidates is simple	-Aesthetically pleasing -Simple to navigate	analysis of all resumes sent -Simple to use interface -Simple to configure -Lots of templates for automating workflow
Weaknesses	-Doesn't work as well for resumes in languages other than English -Career Page is only available in English and Spanish -Is unable to do a boolean and an advanced search at the same time	-Some features, such as time tracking and reporting, are not robust - Missing mobile layout -Unable to select a dedicated IT admin -Not as a convenient option in comparison to other ATSs	-Reporting capabilities are nonexistent -Not attached to any API -Expensive compares to other ATSs -Lots of bugs and glitches -Integrations fail regularly and have to be redone
Pricing	-$15 for Professional Manatal application -$35 for Enterprise Manatal application	-Starts at $99/month for 12 employees and each additional employee is $8.25/month -To include payroll for the employees, it's $75/month for 10 employees and each additional employee is $4.25/month	-Bootstrap is free -Startup is $143/month -Growth is $249/month -Business is $399/month
Onboarding	Onboarding process is simple and convenient, and user interaction is convenient.	Onboarding process is convenient, however users still had some issues with certain features.	Onboarding process is favorable since the interface is very user friendly.

Social Media	Manatal is a popular ATS, and most of its users have little to no complaints about its functionality.	BambooHR seems to have a lot of mixed reviews since users are happy with certain functionalities, but this ATS isn't as robust as other ATS.	Breezy has a lot of negative reviews since it's missing the basic functionalities that other ATSs provide and is pricier in comparison to other ATSs.

Comparison #2

Feature/Company	**Recruiter Flow**	**Vincere**	**Apploi**
Strengths	accelerate your hiring processes and reduce busywork Job advertising, applications, recruitment process, hiring insights, task tracking, mobile app, candidate on boarding	Uses AI forecasting job advertising, applications, intelligent shortlisting, recruitment process, hiring insights, mobile app, brand web page	Automatic screening of applicants applications, recruitment process, hiring insights, task tracking, mobile app, branded web page
Weaknesses	Lacks job recommendations, referrals, intelligence shortlisting, fully automated process, offers and negotiation, branded web page, general hr activities	Lacks referrals, fully auto process, offer and negotiation, task tracking, candidate on-boarding, general hr activities	Only for healthcare employers and applicants Lacks job advertising, recommendations, referrals, intelligent shortlisting, fully automated process, offer negotiation, candidate on boarding, general hr

			activities
Pricing	$109 a month $90 a month annually	$39 a month	Cost not specified
Onboarding	Portal for candidates to interact with that tracks their progress	N/A	N/A
Social Media	Very popular on social media with little to no complaints by users	Competitive software that has mostly positive reviews	Often seen as a one top shop for posting open positions. Very positive reviews

Analysis of Product Features

Product	Job Advertising	Job Recommendations	Applications	Referrals	Intelligent shortlisting	Recruitment process	Fully-automated process	Offer and negotiation	Hiring insights	Task tracking	Mobile app	Candidate onboarding	Branded web page	General HR activities
Hiring Guru	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅	❌	✅	❌	❌
Manatal	✅	✅	✅	❌	✅	✅	❌	❌	✅	✅	✅	✅	✅	❌
BambooHR	✅	❌	✅	❌	❌	✅	❌	❌	✅	❌	✅	✅	✅	✅
Breezy	✅	❌	✅	❌	❌	✅	❌	✅	✅	✅	✅	✅	✅	✅
Recruiter Flow	✅	❌	✅	❌	❌	✅	❌	❌	✅	✅	✅	✅	❌	❌
Vincere	✅	❌	✅	❌	✅	✅	❌	❌	✅	❌	✅	❌	✅	❌
Apploi	❌	❌	✅	❌	❌	✅	❌	❌	✅	✅	✅	❌	✅	❌
Greenhouse	❌	❌	✅	❌	❌	✅	❌	❌	✅	❌	✅	❌	✅	✅

Go Hire	❌	❌	✅	❌	❌	✅	❌	❌	✅	❌	✅	❌	✅	❌

High-Level System Architecture and Technologies Used

Below is a list of the technologies the team has planned to use for this project:

Backend

- Spring Boot(Java) with Gradle
- IDE - Intellij IDEA

Frontend

- Framework - React | JavaScript
- IDE - Visual Studio Code

Continuous Integration

- Github for managing and organizing code and versioning
- Github Actions for Continuous Integration

Deployment

- Service - AWS Beanstalk with Docker Image
- EC2 Instance Type - Amazon Linux 2 - 1 GB RAM - 8 GB SSD EBS volume

Database

- DB - PostgreSQL 14.4

Third Party APIs

- Auth0 for Authentication
- HackerRank API for tests

Supported Browsers

- Chrome - not dead
- Firefox - not dead

Checklist

1. Team found a timeslot to meet outside of class
    a. Yes, the team is meeting every Monday and Wednesday
2. Github Master Chosen
    a. Yes, Kenneth is the Github Master
3. Team decided and agreed together on using the listed SW tools and deployment server
    a. Yes, the team is working on the approved stack. Both frontend and backend applications are deployed and CI/CD pipeline is set up.
4. Team ready and able to use the chosen back and front end framework and those who need to learn are working on learning and practicing
    a. Yes
5. Team lead ensured that all team members read the final M1 and agree/understand it before submission
    a. We didn't get a lot of time for revision. The team struggled with diagrams because this was the first time they were working on the UML diagrams.
6. Github organized as discussed in class
    a. Yes, all changes are being made using feature-branches and pull-requests. Master branch is protected against direct commits.

List of Team Contributions

- Contributions:
    - Kenny Leong
        - M1 editor: responsible for the overall formatting of the document
        - Worked on use cases 3-4
        - Non functional requirements
        - Formatting functional and non-functional requirements
    - Mamadou Bah:
        - Worked on use cases 3-4
        - Non functional requirements
        - Formatting functional and non-functional requirements

- Khushi Khanna:
    - Worked on use cases 3-4
    - Functional requirements ~50
    - Competitive analysis
- Eric Leow:
    - Worked on use cases 2
    - Functional requirements ~50
    - Competitive analysis
- Farhan Haider
    - Worked on setting up the codebase
    - Created frontend app
    - Setup the skeleton for about page
    - Setup backend app
    - Created the required resources on AWS
    - Setup CI/CD for frontend and backend
    - Worked on functional requirements(~50)
    - Worked on Competitive analysis(Current market space + Analysis of product features)
    - Lead the overall work on the M1 document