

CS 407

Spring 2018

Sprint 2 Planning Document

February 26, 2018

MakeItHire

Team 23

Avi Rakesh
Ben Weis
Nik Suprunov
Rajat Srivastava
Zack Fernandez

1 Sprint Overview

Overview

With Sprint 1 mostly focused on user-accounts, login and sign-up, with the the full-stack connection working, Sprint 2 will focus more on the the functionality of the web-app. Sprint 2 will mostly be focused on developing job and application class designs which will enable a basic interaction between the students and recruiters through job postings and applications.

In terms of the functional requirements, the overview goal of this sprint is to cover most of the requirements for students and recruiters. We will allow recruiters to create job posts and make them view-able by all students. We will also allow students to respond to those job posts and create an application. By the end of this sprint, we plan to have developed a working prototype of what our application is supposed to deliver.

In addition to the user-based focus of this sprint, we will also develop methods, in this sprint so that we as developers are able to step-in, override and remove companies which have signed up to be recruiters in our product.

Scrum Details:

Scrum Master: Rajat Srivastava

Scrum Meetings: MWF 4:30pm - 6pm

Risks and Challenges

Segregation of legitimate users from spam.

We intend to use confirmation on their email of users. Only allowing the correct user to be able to modify the job postings, and view the company details, and do other user specific defined working to keep it modular and not allowing a recruiter from company X to be able to modify job posting from company Y. This will be a challenge for primarily backend.

Making the UI for the admin dashboard and backend working for Admin

Admin has the most superior control over the system, and the dashboard will be complex with multiple confirmation request pop-ups, and their individual backend functionality. The backend and frontend needs to work seamlessly to make the Admin's dashboard functional.

Server-Database Data-Transfer Concerns

There will be a lot of different types of data being transferred from the server to the application which will furthermore be transferred to the companies' job applications. These transfers of data needs to be protected that only the correct/appropriate users can access that data. Inappropriate data access could result in a data breach situation.

Teams

Frontend: Ben Weis, Nik Suprunov, Zack Fernandez

Backend & Database: Avi Rakesh, Rajat Srivastava

Workload Summary Table

Name	Total Work Hours
Avi	37
Ben	30
Nik	30
Rajat	37
Zack	36

2 Current Sprint Details

User Story 1: As a Student, I should be able to browse all open job offers

#	Task Description	Time (hours)	Team	Owner
1	Create a page where user can see all the companies and click on each to browse offers and get more related info.	8	FrontEnd	Nik
2	Write server requests for the front-end and populate UI with the results. Write database queries to get all the appropriate jobs that the student queried	4	Frontend	Nik
3	Write database queries to get all the appropriate jobs that the student queried	3	Backend	Avi
4	Setup APIs to receive request to browse all jobs and send the information to front-end	3	Backend	Rajat

Testing/Unit Tests

1. Select a job in the table and click on it, check if relevant information is displayed.
2. Search using long/unrelated queries.

Acceptance Criteria

- All information sent by the server is being displayed.
- No query crashes the server.
- Responsive, fast UI for the table.

User Story 2: As a Student, I should be able to browse all companies with available jobs

#	Task Description	Time (hours)	Team	Owner
1	Create a page with an easy access to available jobs information.	6	FrontEnd	Nik
2	Write database queries to query the list of companies with open job applications	3	Backend	Avi
3	Setup APIs to receive search required and send results to Front-End	3	Backend	Rajat

Testing/Unit Tests

1. Double click on buttons that send Post requests
2. Fast page reload

Acceptance Criteria

- Table is properly sorted based on query
- Table is properly sorted based on query
- Fetched information is stored in local/session storage

User Story 3: As a Head Recruiter, I should be able to add job positions available for the students to apply for

#	Task Description	Time (hours)	Team	Owner
1	Create recruiter profile and include button/modal for adding job applications	8	FrontEnd	Zack
2	Write database queries to add job positions under for a company	3	Backend	Avi
3	Setup API to receive receive position details given by head-recruiter and parse it	3	Backend	Rajat

Testing/Unit Tests

1. Try to add job posting as a student
2. Try to add job posting as a student

Acceptance Criteria

- Student should not be allowed to add a job posting
- Recruiter should be able to add a job posting
- Job will be populated to the database

User Story 4: As a Head Recruiter, I should be able to add recruiters to my company profile

#	Task Description	Time (hours)	Team	Owner
1	Include functionality in company profile to allow for adding recruiters. Also include section on recruiter profile	5	FrontEnd	Zack
2	Write database queries to add recruiters for a company	3	Backend	Avi
3	Get emails of recruiters to be added and send them sign-up invitation	3	Backend	Rajat

Testing/Unit Tests

1. Add recruiter to company profile
2. Add self

Acceptance Criteria

- Head recruiter should be able to add recruiters
- Should not be able to add self
- idCompany field for recruiter account gets set as company when request accepted

User Story 5: As a Head Recruiter, I should be able to add a description for the jobs posted

#	Task Description	Time (hours)	Team	Owner
1	Allow for input of job description when application is posted	5	FrontEnd	Zack
2	From company page and from recruiter page allow for editing job descriptions	5	FrontEnd	Zack
3	Write database query to add description of an already posted job	3	Backend	Avi
4	Setup APIs to receive description for posted jobs, and parse JSON data into variables	3	Backend	Rajat

Testing/Unit Tests

1. Head recruiter attempts to add description of job posted by different recruiter
2. Recruiter tries to change other recruiter's posting description

Acceptance Criteria

- Head recruiter can make change to all postings
- Recruiter can only edit theirs
- Description is updated once changed

User Story 6: As a Recruiter, I should be able to post a job opening/open an application

#	Task Description	Time (hours)	Team	Owner
1	Create view for recruiter to add jobs	5	FrontEnd	Zack
2	Write database queries to open an application/job posting	3	Backend	Avi
3	Setup APIs to receive application details from recruiters and parse the data into application objects	3	Backend	Rajat

Testing/Unit Tests

1. Recruiter tries to post job opening
2. Student tries to find job posting

Acceptance Criteria

- Recruiter should be able to add job posting to site
- Gets stored in the database and can be found again
- Student should be able to find job posting

User Story 7: As a Recruiter, I should be able to see all the applicants who have applied to one of my job postings

#	Task Description	Time (hours)	Team	Owner
1	Build Job Applicant Page	4	FrontEnd	Ben
2	Add filtering view to applicant page	4	FrontEnd	Ben
3	Create route for getting applicants for given job	3	Backend	Avi
4	Setup APIs to receive the request for applicants' list for a particular job position	3	Backend	Rajat

Testing/Unit Tests

1. Recruiter tries to view job posting applications
2. Recruiter should be able to view a student's application to job posting

Acceptance Criteria

- Recruiter should be able to view a student's application to job posting
- Student should be able to apply to a recruiter's posting
- Recruiter can recall past applications from database

User Story 8: As a Recruiter, I should be able to filter applicants in the app

#	Task Description	Time (hours)	Team	Owner
1	Build component to view applicants via filtered criteria	4	FrontEnd	Ben
2	Built "filtering" component for selecting adequate filters	4	FrontEnd	Ben
3	Write database queries to get the applicants as requested using appropriate filtering	3	Backend	Avi
4	Setup APIs to receive filtering criteria for applications and process the information	3	Backend	Rajat

Testing/Unit Tests

1. Try to filter applications
2. Try to remove filter from applications

Acceptance Criteria

- Recruiter should be able to search/filter for an application based on applicant skills
- Recruiter should be able to search/filter for an application based on school/university
- Recruiter should be able to search/filter for an application based on applicant's location

User Story 9: As an Admin, I should be able to approve a new company profile request

#	Task Description	Time (hours)	Team	Owner
1	Make alerts page for admin	5	FrontEnd	Ben
2	Show requests to admins in alert page and allow accept/reject	5	FrontEnd	Nik
3	Make admin dashboard	8	FrontEnd	Zack
4	Write database queries to approve a new company's profile	3	Backend	Avi
5	Setup APIs to receive new compnay profile request and delete the old head recruiter	3	Backend	Rajat

Testing/Unit Tests

1. Sign in as admin
2. <Stuff>

Acceptance Criteria

- Should have access to admin dashboard
- Should be able to edit company information
- <Stuff>

User Story 10: As an Admin, I should be able to add recruiters to a company profile

#	Task Description	Time (hours)	Team	Owner
1	Company page for admin account	5	FrontEnd	Ben
2	Add or remove recruiters from company admin page	5	FrontEnd	Nik
3	Write database queries to add recruiters with the specified emails	3	Backend	Avi
4	Setup APIs to receive recruiter emails and parse to verify them	3	Backend	Rajat

Testing/Unit Tests

1. Check if a company with the email domain (of the recruiter being added) exists or not
2. Add recruiter emails to a company profile using my Admin Dashboard

Acceptance Criteria

- If the company does not exist, recruiter adding request does not add the recruiter, and gives error instead
- If the company exists, the request adds the recruiter in the company list
- The updated company list can be seen live in the database

User Story 11: As an Admin, I should be able to remove recruiters from a company profile

#	Task Description	Time (hours)	Team	Owner
1	Make manage recruiters page for admin to manage recruiters for every company	4	FrontEnd	Ben
2	Write database queries to delete recruiters from the company profile	3	Backend	Avi
3	Setup APIs to receive emails to be removed from company profiles	3	Backend	Rajat

Testing/Unit Tests

1. Check if the recruiter (wanting to be removed) exists.
2. Remove the recruiter using recruiter's email address in the Admin Dashboard.

Acceptance Criteria

- If the recruiter does not exist, do not let the admin proceed with the delete request, and show error instead.
- If the recruiter exists, delete the recruiter from the user's table.
- Make sure in the database, that removing recruiter (not a head recruiter) does not affect the company table in any way.

User Story 12: As a Developer, I should be able to shut down the registration process

#	Task Description	Time (hours)	Team	Owner
1	Switch on the maintenance mode which sends Maintenance status to the Front-End and does not let any more people sign up	4	Backend	Avi
2	If registration is closed, proper message should appear and "sign" up window be disabled.	2	Frontend	Nik

Testing/Unit Tests

1. Sending the request to the Front-End to stop the "Sign up" process for any more users.
2. Sending the appropriate error status to the front end

Acceptance Criteria

- Manually testing if the Front-End receives the request to minimize the display content.
- Should have the "permission-denied" from the back-end with appropriate error code

User Story 13: As a Developer, I should be able to add/remove accounts from our database when companies join/drop our services

#	Task Description	Time (hours)	Team	Owner
1	Add and delete rows from Company table, and their associated users (Head-Recruiters) from the User Table, along with their applications, to get totally remove a company from our product	4	Database	Rajat

Testing/Unit Tests

1. Send dummy query from server to delete a row in Company database table.
2. Send dummy query from server to add a row in Company database table.

Acceptance Criteria

- When deleting, the company, row should be deleted from the Company table
- When deleting the company, its head recruiter and recruiters should also be deleted from the user's table
- When adding a new company, a new row should be created in the company table.

3 Backlog

3.1 Functional Requirement

As a Student:

1. ~~I should be able to make a personal account on MakeItHire.~~
2. ~~I should be able to upload my resume to my account.~~
3. ~~I should be able to upload a cover letter to my account.~~
4. ~~I should be able to link my job-related accounts (e.g. Git).~~
5. I should be able to answer supplemental questions on the jobs I have applied for if they require me to do so.
6. ~~I should be able to browse all open job offers.~~
7. ~~I should be able to browse all companies with available jobs.~~
8. I should be able to apply to multiple jobs at once, not limited to one company.
9. I should be able to chat with recruiters who have contacted me.

As a Head Recruiter:

10. ~~I should be able to create an account for my company~~
11. ~~I should be able to add a description of what my company does~~
12. ~~I should be able to add job positions available for the students to apply for~~
13. ~~I should be able to add recruiters to my company profile~~
14. ~~I should be able to request recruiter to be removed from my company profile~~
15. ~~I should be able to add a description for the jobs posted~~

As a Recruiter:

16. I should be able to directly text students I am interested in
17. ~~I should be able to sign up as a recruiter using my company email address~~
18. ~~I should be able to post a job opening/open an application~~
19. I should be able to add supplemental questions to my job postings
20. ~~I should be able to see all the applicants who have applied to one of my job postings~~
21. I should be able to contact applicants through the app
22. ~~I should be able to filter applicants in the app~~

As an Admin:

- ~~23. I should be able to approve a new company profile request~~
- 24. I should be able to delete a company profile
- 25. I should be able to modify a company profile
- ~~26. I should be able to add recruiters to a company profile~~
- ~~27. I should be able to remove recruiters from a company profile~~

As a Developer:

- ~~28. I should be able to shut down the registration process~~
- ~~29. I should be able to add/remove account for a student with a .edu account~~
- ~~30. I should be able to add/remove accounts from our database when companies join/drop our services~~
- ~~31. I should be able to add/remove account for a recruiter registered for the company~~

3.2 Non-Functional Requirements

Stress Testing

- 32. This includes creating numerous dummy users in the database, and making them perform activities. It includes the following:
 - (i) Response Time: With a lot of users, that app should not become slow to respond.
 - (ii) Scalability: A lot of users should be able to use the application at the same time.
- 33. Along with this our server should be able to handle a reasonably large amount of users and events
- 34. Servers must be up 24/7

Usability

- 35. It should be an easy to use application, an application should be intuitive and work smoothly.
- 36. The iOS part of an app should be supported by all iOS and modern iOS devices.

Security

- 37. The application should keep the user's account secure using good authentication method attached to a server. It should also secure user's contact information, and share only what he/she prefers to share.

Performance

- 38. The app should not be slow to work with, and should not affect the other apps working on the device as well.
- 39. Response time must be minimal with updates occurring in real time when the application is open.

Size of the application

- 40. The application should not take too much space in the user's device. The app should limit its cache size to 32MB max.