

CS 407

Spring 2018

Sprint 1 Planning Document

January 29, 2018

MakeItHire

Team 23

Avi Rakesh
Ben Weis
Nik Suprunov
Rajat Srivastava
Zack Fernandez

1 Sprint Overview

Overview

For Sprint 1, the projected goal of the team is develop a medium fidelity prototype of the web application with the basic functionality done and tested. We plan to deliver most of the front-end web pages for students, recruiters and head-recruiters, and enable the ability for them to sign-up and make accounts, as well as add their personal details to the database, like the resume and cover letter for students, and the company profile descriptions for the head-recruiters.

This sprint will be more focused on front-end and database developed to handle most of the planned features of the application, along with a basic server in NodeJS which is able to connect the two together and transfer data to and fro the database in properly parsed, secure, and usable format. This will enable us to et the Full-Stack connection working.

In addition to the user-based focus of this sprint, we will also develop method, in this sprint so that we as developers are able to add and remove student and recruiter accounts, and manage them directly from the MySQL database using queries.

Scrum Details:

Scrum Master: Rajat Srivastava

Scrum Meetings: MWF 4:30pm - 6pm

Risks and Challenges

Understanding Server Response

The response coming from the server can be misleading. A string saying "No Result" might be displayed as someone's name is asked for a 'Name' from the Server. Hence the team plans to use error codes before each response, to make sure the code means no errors before parsing the string to user feed-able information.

Working Full-Stack connection

One of the initial and most common concerns of a project is making the Full-stack connection work, and there are always certain risks involved with servers crashing and data being lost from the database. Our team will take care when doing the initial setup to make a smooth connection between the stacks.

Server-Database Data-Transfer Concerns

The Server and Database will be communicating a lot with different types of data of different types of user, and there are chances of a breach of our user privileges because of a wrong call from the Server. To tackle this, our Backend team will get a good idea on how the database is set-up and how it differentiates between information of different users, so as to avoid a breach situation.

Teams

Frontend: Zack Fernandez, Nik Suprunov

Backend: Ben Weis, Avi Rakesh, Rajat Srivastava

Database: Rajat Srivastava, Nik Suprunov

2 Current Sprint Details

User Story 1: As a Student, I should be able to make a personal account on MakeItHire

#	Task Description	Time (hrs)	Team	Owner
1	Design landing page, home page and profile page	11	FrontEnd	Zack
2	Create signup/login page	8	FrontEnd	Nik
3	Implement API routes and data validation for updating profile	4	Backend	Ben
4	Design API routes and payload for Students' personal accounts	5	Backend	Avi
5	Make 'Users' table in MySQL and add tables for personal details	2	Database	Rajat

Testing/Unit Tests

1. Create a dummy student account with a '.edu' email address
2. Create a dummy student account without a '.edu' email address

Acceptance Criteria

- The account with '.edu' should be successfully created, and the user should be redirected to Account Homepage.
- The account details added should be displayed on the database by the developer.
- The account without '.edu' email address should be denied an account, and an error should be displayed'.

User Story 2: As a Student, I should be able to upload my resume to my account

#	Task Description	Time (hrs)	Team	Owner
1	Create form for uploading resume	5	FrontEnd	Zack
2	Implement API routes and data validation for uploading a resumé	4	Backend	Ben
3	Design API routes and payloads to upload a resumé	4	Backend	Avi
4	Convert PDF resume to BLOB format in NodeJS and create schema to store resume in MySQL	4	Database	Rajat

Testing/Unit Tests

1. Upload a PDF format resume on the student account details
2. Upload an JPEG format resume on the student account details

Acceptance Criteria

- The PDF format resume should successfully be uploaded.
- The user could download the previously uploaded the resume from the same page.
- The JPEG format resume will not be accepted, and error message will be given.

User Story 3: As a Student, I should be able to upload a cover letter to my account

#	Task Description	Time (hrs)	Team	Owner
1	Create form for uploading cover letter	5	FrontEnd	Nik
2	Implement API routes and data validation for uploading a coverletter	4	Backend	Ben
3	Design API routes and payloads to upload a cover letter	3	Backend	Avi
4	Convert PDF resume to BLOB format in NodeJS and create schema to store cover letter in MySQL	3	Database	Rajat

Testing/Unit Tests

1. Upload a PDF format cover letter on the student account details
2. Upload an JPEG format cover letter on the student account details

Acceptance Criteria

- The PDF format cover letter should successfully be uploaded.
- The user could download the previously uploaded the cover letter from the same page.
- The JPEG format cover letter will not be accepted, and error message will be given.

User Story 4: As a Student, I should be able to link my job-related accounts (e.g. Git)

#	Task Description	Time (hrs)	Team	Owner
1	Add inputs for professional accounts on profile page	3	FrontEnd	Zack
2	Add inputs for professional accounts on profile page	3	FrontEnd	Nik
3	Implement API routes and data validation for linking to external accounts	4	Backend	Ben
4	Design API routes and payloads for urls of accounts	4	Backend	Avi
5	Add columns for storing job-related accounts (links) in Student table	3	Database	Rajat

Testing/Unit Tests

1. Add a valid Github link in the job-related links option
2. Add a string without 'https://' should be rejected, or any other invalid URL

Acceptance Criteria

- The valid URL is accepted, and clicking it takes to the link provided.
- The valid URL gets stored in the database for the student
- The invalid URL does not get accepted, and error message is shown.

User Story 5: As a Head Recruiter, I should be able to create an account for my company

#	Task Description	Time (hrs)	Team	Owner
1	Create signup page for company	5	FrontEnd	Zack
2	Backend query check, account details page	5	FrontEnd	Nik
3	Implement API routes and data validation for creating a company account	4	Backend	Ben
4	Design API routes and payloads for creating a company account	4	Backend	Avi
5	Add Head Recruiter to Users table, with access of a head recruiter	3	Database	Rajat

Testing/Unit Tests

1. Try to signup for a recruiter account if the company does not exist
2. Try to signup for a recruiter account if the company does exist

Acceptance Criteria

- Database should be checked against the company with the same name, if the company already exists, send the error message.
- Head recruiter account and company page should be made if company did not exist
- Recruiter account should be added to company if company already existed

User Story 6: As a Head Recruiter, I should be able to add a description of what my company does

#	Task Description	Time (hrs)	Team	Owner
1	Make company page (w/ edit access available to only Head Recruiter)	8	FrontEnd	Zack
2	Company forms for editing/adding information	5	FrontEnd	Nik
3	Implement API routes and data validation for modifying company page	4	Backend	Ben
4	Design API routes and payloads for description of a company	4	Backend	Avi
5	Add Description column under Company table to accept long text and company logo (convert image into BLOB)	4	Database	Rajat

Testing/Unit Tests

1. Check if description contains UNICODE characters only and is not too long.
2. Try to add description to company account and verify its success

Acceptance Criteria

- Description should only be allowed if valid format
- Valid description should be updated on company profile
- Invalid description should not be allowed, and will throw error

User Story 7: As a Head Recruiter, I should be able to request recruiter to be removed from my company profile

#	Task Description	Time (hrs)	Team	Owner
1	Create form for Head Recruiter to remove recruiters	5	FrontEnd	Nik
2	Implement API routes and data validation for managing recruiters	4	Backend	Ben
3	Design API routes and payloads to remove recruiters	4	Backend	Avi
4	Make a request table viewable by admin to accept/deny requests of head-recriters	3	Database	Rajat

Testing/Unit Tests

1. Try to remove recruiter from company account
2. Check that removed recruiter doesn't show up on front end

Acceptance Criteria

- Removing recruiter from company account will deactivate their information from database
- Recruiter should not display in UI after removed
- Recruiter should not be able to access company info after removed

User Story 8: As a Recruiter, I should be able to sign up as a recruiter using my company email address

#	Task Description	Time (hrs)	Team	Owner
1	Create login for recruiter	3	FrontEnd	Zack
2	Run check against backend to see if company has already been signed up	3	FrontEnd	Nik
3	Implement API routes and data validation for creating recruiter accounts	4	Backend	Ben
4	Design API routes and payloads for company recruiters	4	Backend	Avi
5	Add the recruiter under an already existing company id. Make sure they do not get head-recruiter access	3	Database	Rajat

Testing/Unit Tests

1. Try to sign up for an account as a recruiter when the company doesn't already exist
2. Try to sign up for an account as a recruiter when the company does already exist

Acceptance Criteria

- Sign up for company that already exists should add the recruiter to that company and add their information to our database
- Sign up for company that doesn't exist will create the company, and add the recruiter as head recruiter
- If user is already signed up as recruiter, throw error message

User Story 9: As a Developer, I should be able to add/remove account for a student with a .edu account

#	Task Description	Time (hrs)	Team	Owner
1	Add and delete students from Users table, and delete their profile details	3	Database	Rajat

Testing/Unit Tests

1. Add the student account details (with '.edu') directly into the database using MySQL Workbench

Acceptance Criteria

- The new student account details should be added
- The new student account should not be accepted without an email address
- The new new student account should not be accepted with duplicate names and email

User Story 10: As a Developer, I should be able to add/remove account for a recruiter registered for the company

#	Task Description	Time (hrs)	Team	Owner
1	Add and delete recruiter accounts from Users table, and remove them from company profile	3	Database	Rajat

Testing/Unit Tests

1. Add the recruiter account details directly into the database using MySQL Workbench

Acceptance Criteria

- The new recruiter account details should be added
- The new recruiter account should not be accepted without a valid company id
- The new new recruiter account should not be accepted with duplicate names and email addresses

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.