Intern-Master

Considered your internship application done :)

Team 17 - CS 30700: Design Document

Ji Ma, Nathaniel Young, Ji Soo Cha

1. What went well?

a. Complete: As a prospective job applicant, I want to know every single job application available in my field.
    i. The python scrawler was created successfully to gather available job applications
    ii. The scrawler pushed the data to Firebase job class
b. Complete: As a prospective job applicant, I would love to have easy to use dashboard with the functionality to login in.
    i. The dashboard is very simple to use thanks to the principles in material design.
    ii. The overall functions has been indicated clearly, and the user experience is very straight forward.
c. Partly Completed: As a prospective job applicant, I would love to see the detailed company data with the ability to filter my searches (for example, by location and deadline date). Detailed company information should be rendered in the frontend nicely.
    i. Although we have company information in Firebase, we need to gather more information so that we can create a useful filter or search feature on the website.
d. Complete: As a prospective job applicant, I want to see available open source projects to work on.
    i. Another python scrawler was created to gather a number of open source projects.
    ii. This scrawler then added the data into Firebase.
e. Complete: As a developer, we want to deploy the frontend and backend and test them with ease
    i. We used package management system npm, system.js, and browser-syn c to automate the process of managing dependencies, installing them, and testing front-end result on the fly on different browser environment.

2. What did not go well?

a. Next Sprint: As a developer, we want to update the company application forms with scrawler, view user stats in the admin dashboard.
    i. For now, we could just view those things and modify things based on firebase frontend management system.
    ii. The scrawling process could also be triggered by script manually to update Firebase, in terms of saving time, we didn't implement the admin dashboard.

b. Delayed: As a prospective job applicant, I want to read a blog about relevant news topics in my industry and user success stories.
    i. Our first priority is to get our core functionalities working, so we don't have enough time to implement this feature yet.
    ii. For the next sprint, we will create the blog section in the front end. Every story will be loaded as a simple list card. The detailed content will be stored as markdown on server and rendered on the frontend.
c. Delayed: As a prospective job applicant, I would like to post on the community forum and get feedback on my job application search.
    i. The community forum feature was not a critical feature for us to get up and running for the main part to work. For example, we need to connect the authentication from the front end with the backend before we are able to use that information to log users in to the community forum. Furthermore, we ran out of time as we were already overloaded with work from the main frontend and backend features.
d. Next Sprint: As a prospective job applicant, I would like to share my job application I applied to on social media.
    i. Again, this was not a critical feature for us to get running for the main system to work. As we were crunched for time trying to get the main features in the frontend and backend working, we didn't have time to add a dynamic shared link.
e. Next Sprint: As a prospective job applicant, I want to have a profile page with my education, skills, work experience, GitHub link, etc.
    i. Although we do have a profile page for the prospective job applicant, we didn't gather the user login data through authentication through signin with Google or LinkedIn api's.
    ii. The linkedin api is kind of hard for us to get through, we have to wait for like long time to get authorized to have the api key to use the linkedin api.
    iii. So, we thinked about a work around, which is using a python scrawler to scraw user's linkedin data based on their linkedin user name to fill the company application forms.

3. How should you improve?

a. We need to be realistic on the next sprint on assigning tasks and workloads. We started our sprint 1 with the assumption that 5 of our original developers will work together, however, 2 of them dropped the class. So the workload seems very overwhelming when we try to complete them.
b. We need to allocate accurate hours for each user stories and hopefully we will be able to complete all the tasks that are given to us. Also, we should not underestimate the time it takes to complete a certain task because there is

always the possibility that an unforeseen road block comes up, and it may take longer to fix that code than originally expected.

c. Communication is the key. We need to document the functionality of every single chunk of code well in terms of easing the difficulty in collaboration. For example, for some complex api in backend, it need to be documented. Also, the front end data service, controller, and html view needs to be documented and structuralized as well. Also, if one member needs help with a certain part of the project, he needs to ask for help and be proactive so that he will not get stuck. This could eliminate potential bottlenecks when the frontend is waiting on the backend for vice versa.
d. We can hold up more meetings with our team members and share tasks that are completed and should be completed. Also, this will allow us to discuss problems that are facing individually and allow us to communicate efficiently about our project.