Team 7 – Sprint 2 Retrospective

Ashvin Lohiya, Akshat Goyal, Shubhang Kulkarni, Palina Rawat, Aakash Keswani, Sidhant Chadda

What went well

1. Planning

   At this point, I think we have got accustomed to working in the agile method. We managed to complete all user stories on time, and everything seemed to work as planned. We have all learnt quite a lot about documentation and the need to document things as we move forward.

2. Algorithm

   After much discussion over the algorithm and whether to use a more complex algorithm vs the simple algorithm discussed before, we tested various algorithms developed by team members over a wide range of test-cases, and the one that seemed to do best in most situations was the simple queue-of-queues based algorithm described in the beginning. Hence we have decided to stick to this algorithm. I guess this is one of those situations where a complex algorithm results in worsened results.

3. Teamwork

   Most of us in the team have got to know each other very well, and we know what to expect from one another. This helps us improve work efficiency as we assign work to each of the members' strengths and help each other out with our weaknesses. Hence in general our team work improved over this sprint.

What went poorly

1. Experimenting and over fitting

   We spent a lot of time discussing more complex algorithms to schedule interviews and we also spent some time designing more complex algorithms to improve our test-performance, but unfortunately we ended up over-fitting specific test cases when our scheduling algorithm got more complex. Hence we decided to

stick to the simple algorithm discussed initially as it gave the most fair results across the largest number of test-cases.

2. Messy Code

    We didn't have a standard naming convention and code-style for a lot of the project's code, hence it got a little messy when we were trying to understand the methods that each of us wrote. There were instances where group members were using functions in an incorrect way because they did not understand exactly what the function did, and this was mostly because there were no clear comments explaining it.

3. Work Division

We approached dividing the work amongst our group by assigning members the work that they were good at and what they liked. This resulted in some members needing to spend more time just because the work that they preferred was more time consuming.

How should we improve

1. Better Test Cases

    In this sprint we had a few instances when untested code was pushed onto the git-repository causing other parts of our application to fail. To fix this we will make sure to test our code (exhaustively) before pushing them to master.

2. Cleaner code and comments

    We decided that during the upcoming sprint we will add a comment before every function that describes the input parameters and output of each function. This may seem like a small problem, but it ended up taking up a lot of time while fixing a bug because one member didn't completely understand how one of the functions written by another team member worked.

3. Multi-Disciplinary Work Division

We realized this sprint, that dividing work amongst the team based on what a member is best at was not the best idea, simply because some of the work is just way more time consuming than other work e.g. in **User Story #4, *"As a student, I'd like to filter companies based on my major"*,** the work on the front end was much more than that of the backend. Hence we have decided to divide the work by the time that it takes, but we will still stick closer to what we know and are good at to improve work efficiency.