



Team 7 - Sprint 3 Retrospective

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

What went well

1. Planning and Execution

After finishing our project over the course of the semester, we are extremely comfortable with the agile methodology. We have learnt how to work in a team and divide and integrate work efficiently. We also learnt that documentation is an important tool and we're gradually improving on it.

2. Git

All the team members learnt git and used it to communicate through code with each other. We realized git is an extremely efficient tool to work in a team. We used features like branching to try out new things and then merge whatever we needed.

3. Improvement of App Usability

Once the integration was done, focusing on allowing the recruiters and students to actually be able to use the app for useful purposes was very important. This meant that, going into sprint 3, although we had the functionality by which users could see what was happening on the front end, the algorithm had to reflect those changes, so that optimizations could happen intelligently. Implementing User Stories 2-7 really gave actual "useable feel" to the app as a utility

What went poorly

1. Integration

Front end and back end were being implemented separately for the sprint 3. Algorithm was done at the last moment, therefore we had very little time to work on integration. We faced many unexpected problems at the last moment. Some things stopped working which we had to fix on the last hour before presentation.

2. Poorly Documented Code

A lot of the endpoints on our algorithm side was poorly documented. This meant that when integrating the algorithm to the backend we experienced many problems related to expected outputs and expected inputs. This made us waste a lot of time trying to reverse engineer the algorithm to understand what was done. We did have a document regarding the api for the algorithm but it did not support all the endpoints and some of the parameters were misspelled.

3. Work Division

We did not divide the work in the last sprint fairly. We simply assigned individuals in our team user stories to work on when in reality some user stories required much more work than others. Instead we should have had a better discussion deciding what work needs to be done and how much time each task will take .

How should we improve

1. Better test techniques

Improve testing techniques so that it is applicable to the entire application instead of just few components of it. Possibly robust tests that handle real world scenarios. During one of the **user stories (As a recruiter, I would like to dequeue students)**, we should have made better integration test cases.

2. Improve team-member versatility

Let each member of the team have a specific skill set which they master to help the entire team achieve their goals in that aspect of the project. (i.e. database/ networking/ etc.) It is important that team members have a diverse combined knowledge which cover wide aspects of potential development methods during a project.

3. Poor documentation code

Initially we should have set a standard for documenting every commit/piece of code. Updating documents in a timely manner so

that everyone is aware of what's going on. Having a clear and precise information regarding each commit and update will help other team members understand the changes better and everyone will stay updated on all team members' progress.