Sprint 4 Retrospective - Team 14

Introduction

Our team developed a COVID-19 tracker web app using the React framework. The app allows us to keep track of a patient's status in terms of infections, what were their previous symptoms, and who was the doctor that they are/were assigned to? We have worked on both parts of the application and we are done with almost all the requirements provided by the professor and our project manager.

Based on our progress from Sprint 3, we were able to get more progress done. In Sprint 3, we implemented the general outline of the client application as well as started to dive into backend functionalities. This included the creation of the client dashboard, a simple profile page, as well as a symptoms page. In this sprint, we expanded more functions in the client application. We added that a client was able to update their status, and that status was able to be displayed in the database as well as the admin application getting a notification. Then also we were able to send a notification to the client application when a doctor reviewed the status. For the frontend, we mostly just worked on styling changes.

In terms of project management, it was very similar as to how we functioned in Sprint 1we divided the tasks using GitHub issues and rated them based on importance and difficulty using the Kanban Board. Our team also followed the agile methodology, where we had two weekly meetings to discuss our progress and assign teammates to different tasks.

What went wrong

1- Lack of merging to the main branch

During the sprint, we created multiple branches from our develop to implement various features. During this time, we did not merge the code changes made during the develop branch to our master branch, the branch that we needed to deploy. This sometimes lead to plenty of merge conflicts that we needed to fix at once, which always invited the danger of the code breaking. Overall, however, this did not impact us much because we were able to fix the merge conflicts at a reasonable time and did not break the overall system at all.

2- Struggle with code reusability

During the sprint, we had to finalize a lot of the backend code. This meant that many of the team members had to create different code which sometimes had to be used for similar functions. This meant that we had many functions for the Firebase utilities which only differed by their functionality only by a little bit. Overall, initially this had a big impact at the start of the Sprint since we were in danger of going over the read quota for Firebase, however, we solved this issue by going through our team meetings and accepting a common convention for the code.

What went well

1- Overall finishing of tasks

In this sprint, we had to work on the most amount of user stories in our project. In addition, since most of them had to do with backend, a topic on which a lot of team was unfamiliar with, we were still able to complete almost 77 user points worth of user stories. On top of that, at the start of the sprint, the TA mention a few bugs to us, which we promptly fixed as well. Overall, this has a good impact for us not only in this sprint, but also in the future because we were able to show that we could withstand this amount of work.

2- Good adaption to the stakeholder demands

In this sprint, we had to implement stuff on the fly, based on the TA demands. Despite this, we were able to implement all of what the TA asked in a reduced amount of time due to people having multiple midterms and trying to familiarize themselves with how to incorporate backend code into the project. Overall, again this had a good impact on our team because it boosted our confidence to be able to do more not only in this project, but also in the future potentially for 490.

3- Team communication

As in the other sprints, our team communication was again on point for this sprint. Multiple times in the sprint, we encountered bugs on the application which one teammate could not figure out how to solve. This prompted someone else from the team who could fix the issue to step in, and either call the person, or even screen share to help figure out the issue at hand. Another example of this would be the implementation of the backend. As most of the team are more comfortable with the backend, there was always bound to be some hiccups transitioning to that aspect, but once again, if someone was stuck in the team, another teammate who could help always stepped in to help the person in need. Overall, this had a very good impact on the project, as it did not derail anyone in the team, as well as letting us work on our tasks in a timely manner, which allowed us to finish the sprint stories in a timely manner as well.

Conclusion

As like the previous sprints, we found out of how important is that the whole team follow the same code standard. This will allow us less headaches in the future when we try to solve merge conflicts and use code from another one of our teammates. We did also learn, however, that despite these small problems our team will take the confidence, that we were able to do so much in so little time with us in Sprint 5 and possibly into 490 as well.