Sprint #3 Retro - MEDS

Sprint: Sprint #3

Date: 01/10/2021

Scrum Master: Maxwell Reid (s3787033)

Development team: Ewan Breakey (s3845382), Sefanur Erciyas (s3842307), Thomas Dib (s3838765)

Things That Went Well

Development began very early

Due to the week 6/7 assignment period coming to a completion, the team members had more time to dedicate to the development. We immediately began working on the required features and were able to essentially get a head start and could tell which tasks would be more difficult than anticipated. Other tasks were quickly completed, tested, and out of the way for the rest of the sprint.

Consistent Effort

Once again due to the assignment period coming to a halt, the team was able to put a consistent amount of effort into the development rather than extreme bursts whenever we had time because we now had enough time to allocate to the project on various days within the sprint.

Pair programming

Pair programming was utilised within this sprint to allow the less experienced developers to contribute and acquire assistance from the more experienced members. This increases efficiency as any roadblocks that are reached can easily be overcome rather than spending a large amount of time searching for the answer on their own.

Communication

As with previous sprints, the communication of the team was superb. Any issues that were encountered were quickly mentioned and resolved by another team member. This included both assistance where it was required and other miscellaneous items such as reviewing a pull request on GitHub.

Jira board effectively used

The Jira board as with previous sprints was used to track the progress of the sprint and tasks for each user story. The Jira board was used arguably better than the previous sprint as everything was based on the tasks there and they were quickly and consistently updated to reflect the true progress of any work.

Things That Could Have Gone Better

More research into how to do certain things

One of the inefficiencies experienced within this sprint included the PayPal payment system. After beginning the development of this and making quite good progress, a roadblock was encountered and it was difficult to debug what was happening. Shortly after, there was a React library that was found to assist with the job, but this required restarting the progress that was done for the payment system. If this system had been more thoroughly researched, it would have been likely that this library would have been seen prior to commencement of the development and thus save the time that was wasted on restarting the feature.

Test all user types while completing a feature

For some user stories they were the same besides for which user type it involved. This meant it would take a little more effort to switch between user accounts and test them out as developing which resulted in testing all user types being overlooked. This wrong doing resulted in some bugs in the code that could have easily been fixed much earlier.

Things That Surprised Us

This sprint felt a lot more “standard” because there were not many new things the team had to do. Unlike the previous sprints where we had several new technologies we had to learn, this sprint we only had the PayPal API. This made everything feel a lot more routine.

Working on features as pairs really helped with completing them faster. When normally one team member might have spent much longer on an issue, as a pair, they are able to resolve it quickly.

Lessons Learned

Properly plan features

Due to the experience encountered with the PayPal payment system where the development of the feature essentially restarted due to finding a new way to do something, it is now apparent that the features should be more carefully planned rather than attempting to complete it straight away.

Test all user types while completing a feature

All user types should be tested while implementing a feature that involves all user types. This will avoid bugs coming up later when they could have already been fixed.

Final Thoughts

What are things to keep?

Early Development

As aforementioned, since the development began early, the team was able to better predict how the sprint should be structured and which features should be completed when. If this were not to happen, some of the features may have been underestimated and thus crammed into the end of the sprint. This way we are able to at least know if we have underestimated the particular task at hand.

Pair Programming

Pair programming has been highly beneficial to the team throughout the duration of the project. This should be kept to allow each of the team members to stay consistent with their work as well as work to their fullest potential.

Good Communication

The team should keep up with the good communication. If anyone has any issues or cannot work on the project for some reason, they should contact the rest of the team and let them know rather than leaving them without any idea. This allows the team to work as efficiently as possible and improve our rapport.

What are things to change?

Plan features properly

The features that are planned to be developed within a given sprint should be planned prior to commencement. In some cases, the development may begin before coming to a roadblock and being unable to go any further with it. For this reason it is important to explore the alternatives and possibilities in general.