Sprint #1 Retro - MEDS

Sprint: Sprint #1

Date: 29/08/2021

Scrum Master: Maxwell Reid (s3787033)

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

Things That Went Well

Almost all features in the sprint backlog were completed

As per the sprint review, almost all features for this sprint were implemented on time. This included features such as logging in and viewing book details. Core infrastructure was also created to support developers in the ongoing sprints.

Tasks were allocated quickly

Due to several meetings and prior experience with working as a team, development and scrum tasks were assigned based on members strengths and weaknesses allowing each member to begin work at the start of the sprint.

Collaboration for learning

Despite 2 team members having reduced experience with development, the team worked utilising pair programming allowing for teaching while developing.

Fast response times

Team communication was primarily maintained through Slack and Discord. When team members required an urgent response, members were pinged to ensure a message was not missed. Members were also able to ask questions into the channel and share resources to speed up development times.

Things That Could Have Gone Better

Microservice preparation

As the team misunderstood the requirement of utilising microservices, some of the sprint was dedicated to how multiple services could be connected together. Had the team understood the requirement earlier, more time could have been spent on the development.

Jira board task order

Due to a lack of knowledge regarding the architecture, tasks for each user story needed to be created on the fly and sometimes this led to some tasks being listed out of order.

Uncompleted features

Due to some tasks taking much longer than expected, the team ran out of time once they got up to the searching feature. Good progress was made on it however, it was not completed.

Things That Surprised Us

The lack of tutorials and resources regarding setting up microservices, connecting said microservices to the front-end and authentication of all the routes. The team decided to not use the provided base code due to the delay in release and errors present once it was released. However, because of the lack of knowledge on the topics listed above at the start of the sprint, it caused the need to restructure the code architecture during the sprint.

This caused the team to spend significantly more time on certain tasks than was initially expected.

In addition to this, there has recently been a large update to the microservice structure for spring. This resulted in partial implementations and work that the team had completed whilst researching prior to the sprint being wasted.

Lessons Learned

Updating Jira board more frequently
Team members will update the Jira board more frequently so it can be a more accurate representation of where development is at.

Begin development for complex tasks earlier
In order to reduce the impact of large development tasks such as authentication, developers will begin on them earlier in the sprint. This will allow for adjustments if large issues are encountered whilst trying to implement them. It will also allow the remaining developers to begin work on other tasks that are dependent on them.

Focus on test driven development
Due to lack of familiarity with the framework, we were unable to follow a test driven development software cycle. This led to multiple changes to the code base once the code was written and also meant that some tests needed to be pushed to a later sprint. Now that the team has built their confidence with the system, tests will be easier to create in advance and completed earlier in the sprint.

Final Thoughts

What are things to keep?

Positive attitude toward contribution
All team members were willing to contribute to the project as best they can and collaborate whenever a team member needed assistance. This ensured that the team could rely on each other to complete their assigned tasks.

Communication tools
We will continue with the communication through Slack and Discord to ensure each team member is included in discussions regarding the project. This communication will also ensure that team members remain feeling comfortable to provide their opinion even when it differs from other members.

What are things to change?

Jira updating
We will increase the frequency in which we update the Jira board to be a better representation of development. Once a task is identified, it will be added to the Jira board and its status will also be shared.

Test writing
Tests will be written before or during development rather than waiting until the end where possible. This is also a good opportunity for team members to become more familiar with

how the design is implemented.