

Sprint Planning Notes

Team: Team 01

Sprint: Sprint 2

Date: 07/09/21

Attendee:

Scrum Master: Derren Chin (s3814934)

Product Owner, Documentation: Liam Carnie (s3718562)

Lead Developer, Back-end Developer: Khoi Nguyen (s3678755)

Front-end Developer + MaterialUI + Swagger API: Sarthak Rungta (s3816605)

Back-end Developer + Hibernate + MySQL: Chen Wang (s3853260)

1. Sprint Goal:

Continue development on the user stories missed in the last sprint as well as develop the next level of user stories. On top of this, starting to look towards pushing to an online server, which requires a change of hosting programs and a look into what to use. Especially since AWS is changing their free subscription. Finally, beginning the initial testing phase in all forms (functional, JUnit and CircleCI).

2. Duration of the sprint:

2 weeks

3. What is the team's vision for this sprint?

User Story 1 (ID 1): Search by Books Name, Author, ISBN, Category

User Story 1 (ID 4): Allow **shop owner/publisher** to **add/edit/remove** his/her uploaded selling registrations

User Story 1 (ID 7): Approve a Book to upload to the website (**admin**)

User Story 1 (ID 9): Allow **shop owner/publisher** to view Transaction History of **sales**

User Story 1 (ID 21): Allow Shop Owners to create a book group (shop)

User Story 1 (ID 24): Seller Search Page: Search for copies associated with provided BookID

Why?

At this point in the project we want to incorporate core functionality but still leave out the minor details that do not affect how it progresses. What this essentially means is that in the mid-phase of the project we are focused more on getting the project to work as opposed to adding quality of life features that can be added later on. On top of this, these features work in tandem with the S1 features and therefore improve upon what we have already worked on.

4. Estimation in story points

First User story: 5

Reason: This feature is relatively simple but requires implementation on each of the layers in order to work properly, as searching requires an input on the main page, a call from the back end and a search in the database to return the items, so not complicated but a bit time consuming.

Second User story: 5

Reason: Self explanatory, as books can now be added, users will want to be able to manage said books, therefore implementation of these features is a must at this point. This is just a general update sent to the database, so again, not complicated, but takes a while to ensure that it works.

Third User story: 5

Reason: As users are now able to upload books, Admins need to approve the book to ensure that it contains suitable information and is a valid book. This is not complicated as all that is required is a send to the admin to approve the book before it is added to the database.

Fourth User story: 3

Reason: This is not too complicated, as all it is required is to call the specific book database attached to the user's account back to them, which should already be a relationship in the database.

Fifth User story: 8

Reason: This one is a bit more complicated and therefore requires more time spent, as creating a book database relationship between a group of people that already have individual book databases can get messy and bring back the wrong results.

Sixth User story: 5

Reason: A lot of this can be drawn from the other search features, but gets a bit more complicated when making sure it is only the user's books being returned, so the story points average out as an estimate.