BOOKEROO Sprint Planning Notes

Team: 3 (Friday 10:30am Tutorial)

Sprint: 2

Date: 29/09/21

Attended: Jeffrey, Ross, Allister, Mohammad

Scrum Master: Jeffrey

Development team: Ross, Allister, Mohammad

1. Goal

Our team would like to create a shippable product for BOOKEROO.

This is defined by completing remainder of user stories in the product backlog. These user stories refer to:

(1) Listings, transactions, transaction histories, sharing books.

(2) Adding reviews on businesses or other users.

(3) Adding reviews on books.

1. Duration of the sprint

*2+ weeks*

1. What is the team’s vision for this sprint?

The product will be complete according to the product backlog. Application needs to be polished and bug-free with beautiful CSS styling.

Front and backend will be deployed to AWS.

Database connected to MySQL

1. Estimation in story points

|  |  |  |
| --- | --- | --- |
| Story | Points | Justification |
| As an admin, I want to download reports about books, transactions, and other business data in CSV format, so I can conduct stock-take and complete accounting forms. | 3 | Mohammed has already implemented a CSV export with dummy data.  Just need to integrate it with the transactions microservice when it’s complete. |
| As an admin user, I want to view a summary of all past and current transactions (sorted by date), so I can keep track of business profits. | 13 | Setup a transactions microservice in the backend.  Including model, CRUD methods.  Front-end: collects transactions data and displays. Also does displays statistics e.g., total profits (via custom math functions) |
| As a public user, I want to review books that I have purchased, so I can give my thoughts and opinions on that book. | 13 | Setup a reviews microservice in the backend.  Including model, CRUD methods.  Front-end: collects reviews data and displays them |
| As a business owner, I want to create listings for my books, so that other users can purchase those items. | 13 | Setup a listings microservice in the backend.  Including model, CRUD methods.  Front-end: collects listings data displays on book page. Listings already implemented as a dummy, just need to integrate now. |
| As a public user, I want to create share listings for my books, so that I can provide others with books I don't need. | 5 | Can restrict (in the front-end) for public users to be unable to post new book listings, only used books. |
| As a public user, I want to pay for transactions using PayPal, so I can have the book delivered to me. | 5 | PayPal payment-flow already implemented to a basic degree. Need additional front-end method to create a transaction in the backend. |
| As a public user, I want to cancel (and refund) a book purchase within 2 hours of ordering an item, so I can change my mind on a product. | 5 | Need backend support for cancel method.  Front-end, user has buttons to cancel on the condition that the time is <2 hours.  Button interacts with backend. |
| As a registered user, I want to view my transaction history (buy and sell), so I can track my sales and expenses on BOOKEROO. | 5 | Need backend support for get  Front-end: collects transactions data for given user. |
| As a user, I want to see the availability of a book, so that I can know whether or not I can order it. | 3 | Front-end:  Render “sorry we don’t have any listings for this book” |

Story Point Budget = 4 \* 20 = 80.

**Estimated Velocity = 65**

We completed all user stories last sprint. Our sprint velocity was 80 story points.

It is expected that we will easily be able to complete 65 story points in this sprint in addition to the fact that we have more experience in the frameworks, Docker and CircleCI.

The biggest challenge will be deployment to AWS as we lack experience in that field.