

Project Sprint Planning Notes

Team: Moving Houses

Sprint: 2

Date: 06/09/2021

Attended: Alexander Aloï, Aili Gong, Carl Karama, Jared Song, Shannon Dann

Scrum Master: Jared Song

Product Owner: Redowan Mahmud

Development team: Alexander Aloï, Aili Gong, Carl Karama, Shannon Dann

1. Goal

By the end of the sprint, we aim to complete the unfinished user stories from sprint 1. This includes functionality such as browsing, viewing information about users and books. After this is done, we will proceed to complete the user stories assigned to Sprint 2.

We will create sub-tasks for all user stories and delegate each task with a timeframe to complete their task which is based off the points for the user story, difficulty and individual skill.

We aim to also connect our application to an Amazon RDS, a datastore which we can connect to remotely so that our final application's database is persistent.

2. Duration of the sprint

2-3 weeks

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

User stories involving more functionality with using and accessing the website will be committed to the backlog, such as sorting search queries, navigating between these results and their profile, changing user details and requesting changes to their account.

This includes searching for books by their title, author, ISBN and category, having requests for new users or business users enabled and admin functionality added for modifying user information and approving requests. Business users will now wait for admin approval to enable their account before they can register their account and log in.

A CSV of transactions and books can also be exported for business users and admins. Our code will also be deployed to CircleCi for building and continuous integration with testing, as well as being dockerised.

4. Estimation in story points

Team to estimate story points for each of the items. Provide some sort of justification.

As an admin:

I want to approve a pending registration from a business user so that they can open a new business account. - 3 points as the front end is a single button, while the backend will update a user's status to enabled.

I want to reject books to be sold by customers, so that fake or invalid books are not listed for selling – 3 points

I want to be able to download reports about books in CSV format, so that I can review what books are available on the website – 3 points as it requires converting objects in the database to a CSV file, and only needs to access the database with modifications.

I want an option to choose how the CSV files are sorted, so I can see what the most popular item is. - 1 point as assuming downloads are implemented already, then sorting the results before downloading is a simple query

I want to be able to download reports about user transactions in CSV format, so that I can review the sales history of the website – 1 point as assuming downloads are implemented already, then sorting the results before downloading is a simple query

I want to add new books, so that new items can be sold. - 13 points as this requires the full implementation of the book model, repository and controller, as well as checking for duplicate books.

I want to reject a pending registration from a business user so that they are unable to open a new business account. - 3 points as it is a direct API call to remove the request from a new user

I want to be able to add stock for a book, so that sellers can add stock to any books in their inventory. - 3 points as it is a simple API call to update the book with an extra form in the front-end

I want to modify users, so that I can suspend/unsuspend users from their account. - 5 points as the backend will need to change books sold by the seller to unavailable/available, and then update the user's status

As a business user:

I want to export a CSV of transactions, so I know if the business is making a profit - 1 point as the function of downloading csv files will already be implemented.

I want to be able to upload an item to sell, so that an admin can approve it. - 13 points as this requires the full implementation of the book model, repository and controller, as well as checking for duplicate books.

I want to be able to register a business account so that I can sell and rent my books to customers. - 8 points as the register account function been implemented but will need to be implemented for a different model

I want to be able to see all previous transactions, so I know how often transactions are happening. - 3 points as the API call does not modify anything, but it requires transactions to be fully implemented.

I want to be able to edit the books I am managing, so I can fix mistakes when originally created the book page. - 5 points as the back end requires updating the book's details, while checking for duplication errors

As a customer:

I want to view my profile page so I can change my password – 5 points because this involves multiple forms and backend hashing for the new password

I want to be able to sort my search results so that my search query starts from the lowest cost item. - 3 points as it requires a unique query

I want to be able to sort my search results so that my search query is ordered alphabetically. - 3 points as it requires a unique query

I want to be able to view my order status, so that I can see the current status of my order – 3 points as it requires transactions to be fully implemented

I want to be able to search for books in the catalogue by the category of the book so that I can find books with a specific category - 3 points as it requires a unique query

I want to be able to search for books in the catalogue by the name of the author so that I can find books written by a specific author - 3 points as it requires a unique query

I want to be able to search for books in the catalogue by the title of the book so that I can find books with a specific title or series title - 3 points as it requires a unique query

I want to be able to search for books in the catalogue by the book's ISBN so that I can find books with a specific ISBN - 3 points as it requires a unique query

I want to be able to view my transaction history, so that I can see all my recently bought books - 3 points as it requires a unique query