Boilerbazaar

Sprint 2 Planning Document

Team 30: Dung “Ryan” Doan, Xavier Huu Pham, Jeffrey Yian Wang, Shicheng Fang, Michio L Sekiguchi

Sprint Overview

During this sprint, we hope to successfully develop the UI for the algorithms that we have made in the previous sprint. This includes the ones that we were unable to complete during the last sprint. In addition, we hope to set up the login system to access the webpage and create user accounts whether that be through ITaP or our own system. We would also like to set up the messaging system between different users and add more functionality related to listings such as listing availability and watchlists.

Scrum Master: Xavier

Meeting Plan: Tuesdays/Thursdays @ 5:00 pm

Risk and Challenges:

In the last sprint, we had an issue with time. This set us back a bit in the implementation of user stories as we must catch up with the user stories that we did not complete in sprint one. It is important for us to avoid this problem in future sprints. Now that we have connected our frontend and backend of our web application, the challenge now would be testing to verify if our implementation is working properly. Another challenge we may face is getting permission from ITaP to use their login system. This could be due to privacy concerns that ITaP may have. We have an alternate plan if we cannot, but this sprint would be easier if we could use it.

Current Sprint Detail

User Story #1

As a user, I would like to be able to login through Purdue authentication.

#	Description	Estimated Time	Owner
1	Create our own login page if ITAP does not give Purdue authentication.	3 Hrs	Jeffrey
2	Create sign up page if user does not have account.	4 Hrs	Jeffrey
3	Create algorithm to add user to database.	1 Hr	Jeffrey
4	Create algorithm to verify username and password.	1 Hr	Jeffrey
5	Debug and test signing up and logging in algorithm using unit tests.	3 Hrs	Jeffrey
6	Integrate ITAP if we get permission.	2 Hrs	Jeffrey

Acceptance Criteria:

- Given that we got Purdue ITAP permission, users should be able to login with their Purdue account.
- Given that we didn’t get Purdue ITAP permission, users should see a login page when they first load the page.
- Given that the login page UI is designed correctly, users should see a place to enter email and password.
- Given that the sign-up page UI is designed correctly, users should be prompted to answer a few questions such as email, password, etc.

User Story #2

As a student, I would like to be able to see how many other users have viewed a textbook.

#	Description	Estimated Time	Owner
1	Create an icon with the number of views.	1 Hr	Michio
2	Create an algorithm to update the number of views a textbook receives.	3 Hrs	Michio
3	Create an algorithm to retrieve the number of views a textbook has received.	1 Hr	Michio

4	Use unit tests to debug and test functionality of algorithm.	2 Hrs	Michio

Acceptance Criteria:

- Given that the UI is implemented correctly, the user should see an icon with how many views a textbook has.
- Given that the algorithm is implemented correctly, when a user clicks on a listing, it should increment the number of views.
- Given that the algorithm is implemented correctly, the user should be presented with an accurate number of views a textbook has received from before the request was made.

User Story #3

As a student, I would like to be able to track a textbook (watchlist) to see any changes in price, etc.

#	Description	Estimated Time	Owner
1	Create a button on a single listing page that when clicked allows a user to track or untrack a textbook listing.	2 Hrs	Xavier
2	Create an algorithm to add the tracked textbook to the user database.	1 Hr	Xavier
3	Create an algorithm to remove the tracked textbook from the user database.	1 Hr	Xavier
4	Debug and test algorithm using unit tests.	2 Hrs	Xavier

Acceptance Criteria:

- Given that the button is created correctly, users should be able to add tracked listings from the page of a listing.
- Given that the button is implemented correctly, it should add a textbook listing to a user’s list of tracked listings when pressed.
- Given that the button is implemented correctly, a user should be able to remove a textbook listing from their list of tracked listings.

User Story #4

As a seller, I would like to be able to edit the status of a transaction for a book I am selling (available, sold).

#	Description	Estimated Time	Owner
1	Create button to mark listing as sold or available.	1 Hr	Shicheng
2	Create algorithm to update database with new status.	1 Hr	Shicheng
3	Update buy page search algorithms to prevent sold textbooks from appearing.	2 Hrs	Shicheng
4	Update the single listing page.	2 Hrs	Shicheng
5	Create unit tests to ensure proper functionality of algorithm and UI.	4 Hrs	Shicheng

Acceptance Criteria:

- Given that the button and algorithm are implemented correctly, the user should be able to update the status of a listing.
- Given that the algorithm is implemented correctly, the user should not be too able to see sold listings on the buy page.
- Given that that UI is implemented correctly, the user should be able to see the status of a listing.

User Story #5

As a buyer, I would like to share a seller's listing.

#	Description	Estimated Time	Owner
1	Create button to share listing.	2 Hrs	Michio
2	Create a popup for the user to share the listing in different ways (e.g., email, link, etc.).	4 Hrs	Michio
3	Implement a way of linking to the listing.	3 Hrs	Michio
4	Create unit tests to test functionality of sharing algorithm and UI.	2 Hrs	Michio

Acceptance Criteria:

- Given that the button is implemented correctly, it should present a user with options for sharing a listing when pressed.
- Given that the sharing UI is implemented correctly, the user should be able to share a listing in two or more ways.

- Given that the algorithms are implemented correctly, the user should be able to share a listing with another user.

User Story #6

As a student, I would like to be able to view the profile of sellers.

#	Description	Estimated Time	Owner
1	Create UI for seller profile.	6 Hrs	Shicheng
2	Create button to click to open seller profile.	1 Hr	Shicheng
3	Create an algorithm to access the profile information of a specific user.	1 Hr	Shicheng
4	Create unit tests to ensure proper functionality of UI and algorithm.	3 Hrs	Shicheng

Acceptance Criteria:

- Given that the UI is implemented correctly, the user should be able to see profile information in a clean and organized way.
- Given that the button is implemented correctly, the user should be able to navigate to the seller's profile using the button.
- Given that the algorithm is implemented correctly, the user should be presented with accurate profile information of a seller.

User Story #7

As a student, I would like to be able to see how many sales a seller has made.

#	Description	Estimated Time	Owner
1	Create an algorithm to update the number of sales made when transaction status is set to sold.	2 Hrs	Shicheng
2	Create algorithm to get how many sales a seller has made.	1 Hr	Shicheng
3	Update UI to represent how many sales a seller has made.	3 Hrs	Shicheng
4	Create unit tests to make sure UI and algorithm is working properly.	2 Hrs	Shicheng

Acceptance Criteria:

- Given that the algorithm is implemented correctly, a seller's number of sales should increase by one when transaction status is set to sold.
- Given that the algorithm is implemented correctly, the client should receive the correct number of sales a seller has made.
- Given that the UI is implemented correctly, the user should be able to see the number of sales a seller has made.

User Story #8

As a student, I would like to be able to view a watchlist of listings from my profile.

#	Description	Estimated Time	Owner
1	Create UI for watchlist in settings page.	6 Hrs	Xavier
2	Create button to click to route to watchlist page.	1 Hr	Xavier
3	Implement an algorithm to retrieve a user's watchlist.	1 Hr	Xavier
4	Implement an algorithm to check if the price of a listing changed since it was tracked.	2 Hrs	Xavier
5	Debug and create unit tests to check if UI and algorithm is implemented correctly.	2 Hrs	Xavier

Acceptance Criteria:

- Given that the UI is implemented correctly, the user should be able to see their watchlist from the client easily and intuitively.
- Given that the button is implemented correctly, the user should be able to navigate to their watchlist from the settings page.
- Given that the algorithm is implemented correctly, the user should be able to easily see if there was a price change on a tracked listing.

User Story #9

As a student, I would like to be able to see my viewing history.

#	Description	Estimated Time	Owner

1	Create UI for viewing history.	6 Hrs	Michio
2	Create button to click to see viewing history.	1 Hr	Michio
3	Implement algorithms to save a listing to a user’s viewed history once they view it.	1 Hr	Michio
4	Implement algorithm to retrieve a user’s viewed history from the database.	1 Hr	Michio
5	Create unit tests to test algorithm functionality.	3 Hrs	Michio

Acceptance Criteria:

- Given that the UI is implemented correctly, the user should be able to easily see their viewing history.
- Given that the algorithms are working, when a user views a listing, it should be saved to their viewed history.
- Given that the algorithms are working, the listings displayed in the viewing history should be a set amount of the most recent listings that the user previously viewed retrieved from the database.

User Story #10

As a student, I would like to be able to see which classes are associated with a textbook.

#	Description	Estimated Time	Owner
1	Add input on the sell page for class.	3 Hrs	Xavier
2	Add input on the edit page for class.	2 Hrs	Xavier
3	Adjust algorithms to account for the class field of a listing.	1 Hr	Xavier
4	Implement class filters.	1 Hr	Xavier
5	Create UI for searching listings by class.	2 Hrs	Xavier
6	Implement algorithm to search by course.	1 Hr	Xavier

7	Debug and test the searching algorithm with unit tests	3 Hrs	Xavier

Acceptance Criteria:

- Given that the UI is implemented correctly, the user should be able to add the class or course to a textbook listing.
- Given that the course filter is implemented correctly, the user should be able to filter returned textbook listings by course.
- Given that the course search option is implemented correctly, the user should be able to search for textbook listings by course.

User Story #11

As a buyer, I would like to be able to easily message the seller.

#	Description	Estimated Time	Owner
1	Add an option in listing to allow user to contact the seller.	1 Hr	Ryan
2	Create input field and ajax request to send messages	3 Hrs	Ryan
3	Create API gateway and lambda function to save messages to server	2 Hrs	Ryan
4	Create unit tests to test functionality of messaging algorithm.	2 Hrs	Ryan

Acceptance Criteria:

- Given that the UI is designed correctly, listings will have an option for user to contact the seller.
- Given that the algorithm is optimally written, when the user submits their message, it should be saved into the database.
- Given that the algorithm is written correctly, the recipient database should also be updated.

User Story #12

As a student, I would like to be able to see past messages.

#	Description	Estimated Time	Owner
1	Create UI to display messages	4 Hrs	Ryan
2	Create JS function to allow user to scroll	2 Hrs	Ryan
3	Create UI and JS function to load more messages	3 Hrs	Ryan
4	Create ajax request to retrieve messages	1 Hr	Ryan
5	Create js function to parse retrieved messages to UI	1 Hr	Ryan
6	Create UI and JS function to display a list of conversations that can be switched to	2 Hrs	Ryan
7	Use unit tests to test functionality of algorithms and UI.	3 Hrs	Ryan

Acceptance Criteria:

- Given that the UI is designed correctly, users should be able to view a list of their conversations.
- Given that the UI is designed correctly, users should be able to click on a conversation to view that conversation messages.
- Given that the UI is designed correctly, users should be able to scroll through messages and load more messages.
- Given that the UI is designed correctly, users should be able to switch to a different conversation.

Sprint 1 - Incomplete User Stories, Tasks, and Acceptance Criteria

User Story #3

As a seller, I would like to be able to add/remove pictures of what I am selling.

#	Description	Estimated Time	Owner
1	Create a UI for users to remove/update listing images.	3 Hrs	Ryan
2	Create an algorithm to remove/update images from the database.	2 Hrs	Ryan

3	Create unit tests to test functionality of removing/updating algorithm.	2 Hrs	Ryan

Acceptance Criteria:

- Given that the UI is implemented correctly, when users click the x button, then it should remove the image from the UI.
- Given that the remove/update algorithm is implemented correctly, when users remove/update images from the listing, it should update the database.
- Given that the UI is implemented correctly, when users add more pictures, it should append to existing pictures.

User Story #7

As a seller, I would like to be able to edit or delete my textbook listings after submitting them.

#	Description	Estimated Time	Owner
1	Create button for edit and button for delete and implement them with existing algorithms.	1 Hr	Jeffrey
2	Create a UI for users to see their listings.	4 Hrs	Jeffrey
3	Create a UI to edit the listings	3 Hrs	Jeffrey
4	Create unit tests to ensure proper functionality of UI and algorithm.	3 Hrs	Jeffrey

Acceptance Criteria:

- Given that the UI is working correctly, users should be able to see a list of all their listings.
- Given that the UI and algorithm are working correctly, users should be able to edit their listings from the client.
- Given that the UI and algorithm are working correctly, users should be able to delete their listings.

User Story #10

As a buyer, I would like to see details on the listed textbook (condition, description, etc.).

#	Description	Estimated Time	Owner

2	Create UI for a textbook listing.	5 Hrs	Jeffrey
3	Create unit tests to ensure proper functionality of UI.	2 Hrs	Jeffrey

Acceptance Criteria:

- Given that the UI is designed correctly, when the user clicks on a listing, then the listing's information should be accurate, complete, and comprehensive for the user.
- Given that the UI is implemented correctly, when the user clicks on the listing, then the textbook images, description, title, author, ISBN, condition should show up.
- Given that the correct textbook listing and its information is displayed, they should be able to traverse links to profiles and books associated with the listing.

Remaining Backlog

Functional

1. ~~As a user, I would like to be able to login through Purdue authentication.~~
2. ~~As a user, I would like to be able to easily navigate to any page using a navigation bar.~~
3. ~~As a user, I would like to be able to change between light and dark mode.~~
4. ~~As a student, I would like to be able to have a list of previously searched textbooks displayed when I open the home page.~~
5. ~~As a student, I would like to be able to search for a textbook based on ISBN, author, or title.~~
6. ~~As a student, I would like to be able to search for used textbooks from other students.~~
7. ~~As a student, I would like to be able to view a dropdown of my previous searches on the search bar.~~
8. ~~As a student, I would like to be able to filter through the student sold textbooks by course, edition, etc.~~
9. As a student, I would like to be able to search with multiple filters in the home page search.
10. As a student, I would like to be able to sort textbooks by price, popularity, ratings, etc.
11. As a student, I would like to be able to see a list of websites that are selling the textbook that I searched for with the cheapest price highlighted.
12. As a student, I would like to be able to easily access the websites selling the textbooks.
13. ~~As a student, I would like to be able to track a textbook (watchlist) to see any changes in price, etc.~~
14. As a student, I would like to be able to receive desktop notifications for price drops.
15. As a student, I would like to be able to follow a seller to see if they list any other textbooks.
16. ~~As a student, I would like to be able to see how many other users have viewed a textbook.~~
17. ~~As a buyer, I would like to be able to easily message the seller.~~
18. ~~c~~
19. ~~As a buyer, I would like to share a seller's listing.~~
20. ~~As a buyer, I would like to see details on the listed textbook (condition, description, etc.).~~
21. As a buyer, I would like to be able to click on seller to see seller profile when trying to buy a textbook from them.
22. As a buyer, I would like to be able to see other textbooks that are related to the current listed textbook.
23. ~~As a student, I would like to be able to see past messages.~~
24. ~~As a student, I would like to be able to view my profile.~~
25. As a student, I would like to be able to add a profile picture.
26. ~~As a student, I would like to be able to edit my profile.~~
27. ~~As a student, I would like to be able to view the profile of sellers.~~
28. As a buyer, I would like to be able to give seller ratings based on interaction and ease of purchase.
29. ~~As a seller, I would like to be able to list textbooks for sale.~~

30. ~~As a seller, I would like to be able to add/remove pictures of what I am selling.~~
31. ~~As a seller, I would like to be able to list the title and price of the textbook.~~
32. ~~As a seller, I would like to be able to edit or delete my textbook listings after submitting them.~~
33. ~~As a seller, I would like to be able to add the condition of the textbook.~~
34. ~~As a seller, I would like to be able to add a description of the textbook.~~
35. As a seller, I would like to be able to set a meetup preference (public, in front of house, etc.).
36. ~~As a seller, I would like to be able to edit the status of a transaction for a book I am selling (available, sold).~~
37. As a seller, I would like to be able to view reviews from my buyers.
38. As a seller, I would like to give reviews about my buyers.
39. As a student, I would like to be able to block another user from messaging me.
40. As a student, I would like to be able to block another user from viewing my listings.
41. As a student, I would like to be able to search for other users.
42. As a student, I would like to be able to rate a textbook and view ratings from other users.
43. As a student, I would like to be able to add textbook information for courses that I am taking.
44. ~~As a student, I would like to be able to view information about the Purdue University Book Store such as locations and hours.~~
45. ~~As a student, I would like to be able to view an “About” page with information about the developers and the project.~~
46. ~~As a student, I would like to be able to view a watchlist of listings from my profile.~~
47. ~~As a student, I would like to be able to see how many sales a seller has made.~~
48. ~~As a student, I would like to be able to see which classes are associated with a textbook.~~
49. ~~As a student, I would like to be able to see my viewing history.~~
50. As a student, I would like to be able to submit and view GroupMe links related to courses.
51. As a student, I would like to be able to change my notification preferences (if time allows).
52. As a student, I would like to be able to upload all my syllabi to get a list of textbooks from the syllabi (if time allows).

Non-Functional

1. Performance
   As a developer,
   a. I would like to send and receive messages within 5 seconds.
   b. I would like to search and list textbooks within 10 seconds.
   c. I would like to be able to host 1 percent of Purdue’s student population at any time (~4160 students).
   d. I would like the application to run smoothly without crashing.

2. Server

   As a developer,

   a. I would like the server to communicate with the front and two databases quickly and efficiently.
   b. I would like the server to be able to store users' data in a database.

3. Appearance

   As a developer,

   a. I would like the application to be aesthetically pleasing.

4. Security

   As a developer,

   a. I would like user account to be secure and available only to the specific user.
   b. I would like to notify users of basic etiquette and security precautions when interacting with other users through the messaging system (no harassment, discrimination; passwords should not be given to anyone).
   c. I would like to ensure that the use of users' data is non-malicious and transparent.

5. Usability

   As a developer,

   a. I would like the interface to be clean and easy to navigate.
   b. I would like the application to be able to run on web browsers.