Boilerbazaar

Sprint 1 Planning Document

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

Sprint Overview

Since this is our first sprint, we hope to set up the frontend and the backend of our web application to fulfill the expected user stories. In addition, the database needs to be set up so that we can collect data for the different textbooks and users. We would also like to set up all the visuals for the front end so that we only need to worry about the scripts in the later sprints. We want to have the home page with a search function, the buy page with a search function and the different textbooks that are up for sale, be able to list textbooks for sale on the sell page, see information about the developers, see the locations and hours of Purdue bookstores, and see profile information.

Scrum Master: Ryan Doan

Meeting Plan: Tuesdays/Thursdays @ 5:00 pm

Risk and Challenges:

A challenge that we might come across when setting up our web application is connecting everything together. It is crucial for us to link the frontend to the backend so that we can have a foundation to develop our web application. Another challenge that we might come across is learning our framework/languages. Some of us are new to the frameworks that we will be using, and it will take some time to familiarize ourselves with the program. Finally, trying to get the project moving along and establishing a flow to development is going to be a challenge during this phase, where we are also setting up the foundation for our webapp and future features.

Current Sprint Detail

User Story #1

As a user, I would like to be able to easily navigate to any page using a navigation bar.

#	Description	Estimated Time	Owner
1	Create a UI navigation bar.	1 Hr	Jeffrey
2	Add page navigation functionality to UI navigation bar.	2 Hrs	Jeffrey
3	Create algorithm to change the background color of the page button that is active.	1 Hr	Jeffrey

Acceptance Criteria:

- Given that the navigation bar is implemented correctly, when the user attempts to click a button on the bar, the page they requested should pop up.
- Given that active page algorithm is implemented correctly, when the user clicks on a menu item, the page that the user is being directed to should have the background color of the page button changed.
- Given that UI for the navigation bar is implemented correctly, when the user opens the page, there should be 6 items on the bar.
- Given that the navigation bar is implemented correctly, when the user clicks on a new page, the navigation bar should still be at the top of the page.

User Story #2

As a seller, I would like to be able to list textbooks for sale.

#	Description	Estimated Time	Owner
1	Create a page for generating a listing.	4 Hrs	Jeffrey
2	Create an algorithm to add the listing to the database.	2 Hrs	Jeffrey

Acceptance Criteria:

- Given that the listing algorithm is implemented correctly, when the user clicks the submit button, it should send the data of the textbook to the database.
- Given that the web client is implemented correctly, it should prevent the user from generating a listing with insufficient information, and instead prompt the user to fill in the necessary information.

- Given that the UI is created properly, when users click on sell page, it should open a page with different inputs for users to fill out.

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 upload listing images.	4 Hrs	Ryan
2	Create an algorithm to compress images to JPEG to save space.	3 Hrs	Ryan
3	Create an algorithm to save images to the database.	1 Hr	Ryan
4	Create a UI for users to remove/update listing images.	3 Hrs	Ryan
5	Create an algorithm to remove/update images from the database.	2 Hr	Ryan
6	Allow users to add multiple pictures at once.	3 Hrs	Ryan

Acceptance Criteria:

- Given that the algorithm is implemented correctly, the image(s) should be sent to and saved in the database and retrieved whenever another user views the listing.
- Given that the UI is implemented correctly, when users click to add image, then they should be able to add multiple images at once.
- 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.

User Story #4

As a seller, I would like to be able to list the title and price of the textbook.

#	Description	Estimated Time	Owner
1	Implement input for users to enter data.	2 Hr	Ryan
2	Format input for price (1,000 becomes 1000), rejects invalid input (text/symbols)	1 Hr	Ryan
3	Create algorithm to add data to database.	1 Hr	Ryan

4	Make input box “required” to answer and highlight red when not answered.	1 Hr	Ryan

Acceptance Criteria:

- Given that the UI is implemented correctly, when the user clicks on an input, they should be able to enter text.
- Given that algorithm works properly, when the user has values in the input, algorithm should get the values and update the database and the listing.
- Given that the web client works properly, input to price should be formatted (remove ‘,’ and whitespaces), while invalid input should be rejected (non-numeric values).
- Given that the required field works, when the user enters a value, the input box border should not be red anymore.

User Story #5

As a seller, I would like to be able to add the condition of the textbook.

#	Description	Estimated Time	Owner
1	Implement dropdown list for users to choose.	1 Hr	Ryan
2	Update the database with user selection.	1 Hr	Ryan
3	Make dropdown list “required”.	1 Hr	Ryan

Acceptance Criteria:

- Given that the dropdown list is implemented correctly, the user should see different options to choose from when clicking the dropdown.
- Given that the web client is implemented correctly, the user must specify their book’s condition before being able to create a listing.
- Given that there is no dropdown selection, the dropdown list should prompt the user to select a value.

User Story #6

As a seller, I would like to be able to add a description of the textbook.

#	Description	Estimated Time	Owner

1	Implement textbox for users to type bullet points.	2 Hrs	Jeffrey
2	Implement algorithm to get entered text.	2 Hrs	Jeffrey
3	Create algorithm to update the database with description values.	1 Hr	Jeffrey

Acceptance Criteria:

- Given that the textbook is implemented correctly, when the user enters more than the size of the textbox, the textbox should be able to scroll down.
- Given that the algorithm is optimally written, when the user enters text and lists textbook for sale, the algorithm should update database and listing with text.
- Given that UI is implemented correctly, the user should see a textbox to enter the description of the textbook.
- Given that the UI is implemented correctly, the user shouldn’t be able to enter more than 500 characters for the description.

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 an algorithm to check if a user is the owner of a listing they are trying to edit or delete.	3 Hrs	Shicheng
2	Create an algorithm to update the information of a listing from the database.	2 Hrs	Shicheng
3	Create an algorithm to delete a listing from the database.	2 Hrs	Shicheng
4	Create button for edit and button for delete.	1 Hr	Xavier

Acceptance Criteria:

- Given that the algorithm is working correctly, users should be able to make changes to their own listings.
- Given that the editing algorithm is working correctly, users should be able to update the information of a listing and update the information inside the database.
- Given that the deleting algorithm is implemented correctly, when users delete a listing, then it should update the listings and remove from database.

User Story #8

As a student, I would like to be able to search for used textbooks from other students.

#	Description	Estimated Time	Owner
1	Implement an algorithm to get searched input and send it to the database.	2 Hrs	Shicheng
2	Create search bar and submit button.	1 Hr	Shicheng
3	Create UI to display search results	6 Hrs	Shicheng
4	Create dropdown list for the different searches.	1 Hr	Mitch

Acceptance Criteria:

- Given that the algorithm is implemented correctly, users should be able to see search results that resembles their query.
- Given that the UI is created properly, when the user clicks submit, they should see a page with the list of textbooks that they searched.
- Given that the dropdown list is created correctly, when the user clicks on it, then it should display 3 different options of Title, Author, or ISBN.

User Story #9

As a student, I would like to be able to filter/sort through the student sold textbooks by course, edition, etc.

#	Description	Estimated Time	Owner
1	Create checkboxes for criteria that the user wants to filter by.	2 Hrs	Xavier
2	Create a dropdown list where users can select what they want to sort by.	2 Hrs	Xavier
3	Create an algorithm that records user’s filters and sorting preference, then queries the result from the database.	3 Hrs	Xavier
4	Create a filter button where it will open the list of things to filter by.	1 Hr	Xavier

Acceptance Criteria:

- Given that the database is implemented correctly, there should be a diverse range of options for the user to filter and sort by.
- Given that the algorithm is implemented correctly, the result returned should align with the user's specifications
- Given that the UI is designed correctly, when the user clicks on filter button, then it should be easy and intuitive for the user to sort and filter.

User Story #10

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

#	Description	Estimated Time	Owner
1	Implement an algorithm that queries data from the database about a textbook listing.	1 Hr	Michio
2	Create UI for a textbook listing.	5 Hrs	Michio

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 there is a specified textbook the user can view, they should see a dedicated page to view the textbook listing and its associated information.
- 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.

User Story #11

As a student, I would like to be able to view information about the Purdue University Book Store such as locations and hours.

#	Description	Estimated Time	Owner
1	Set up UI that lists information about each Purdue Book Store location.	4 Hrs	Xavier
2	Insert images with the locations of the Purdue bookstores	1 Hr	Xavier
3	Embed Goggle Maps into the page to show the location of the different bookstores. Clicking on it would allow you to get directions.	2 Hrs	Xavier

4	Create an algorithm that finds the bookstore that is closest to the user’s location and informs them which one it is.	4 Hrs	Xavier
5	Create an algorithm to dynamically inform the user of the bookstore’s hours based on the current day.	2 Hrs	Xavier

Acceptance Criteria:

- Given that the page is setup correctly, when the user clicks on the page, they should see information about hours of the different Purdue University bookstores.
- Given that the map is set up correctly, the user should see the location of the bookstores and a dropdown pin of where it is located on the map.
- Given that the Google maps embedment is set up correctly, it should display the location of each bookstore and when the user clicks it, it should open Google maps in a new tab with the Purdue bookstore that they selected as the destination.

User Story #12

As a student, I would like to be able to view an “About” page with information about the developers and the project.

#	Description	Estimated Time	Owner
1	Create UI with information.	4 Hrs	Michio
2	Create icons with links to each person’s socials (Github, webpage, etc.).	2 Hrs	Michio

Acceptance Criteria:

- Given that the page is implemented correctly, the “About” page should display information on capabilities of the project, along with the developers’ information
- Given that icons are set up correctly, when the user clicks on an icon, it should open a new tab with the desired website.
- Given that the page and its information are displayed, they should be able to be direct to how to contact the creators in order to report an issue.

User Story #13

As a student, I would like to be able to search for a textbook based on ISBN, author, or title.

#	Description	Estimated Time	Owner

1	Create a search box.	1 Hr	Michio
2	Create a dropdown list on the left side of the search bar where users can specify what they are searching for.	2 Hrs	Michio
3	Create a submit button so that the user can search.	1 Hr	Michio
4	Create algorithm that gets submit search and sends to database.	2 Hrs	Michio
5	Implement algorithm that will load page with returned textbooks.	2 Hrs	Michio
6	Implement UI for returned textbooks.	5 Hrs	Michio

Acceptance Criteria:

- Given that the buy page UI is created properly, when the user is on the buy page, they should be able to enter text into the search box.
- Given that the UI and search box are implemented, they should see a list of options to filter the search results by.
- Given that the UI and search box and implemented, they should be able to click the submit button to request the list of search results.
- Given that the database is correctly implemented, they should be able to request listings that fit the specified criteria.
- Given that the database returns the correct listings, the client's page will be loaded with the correct listings sorted by selected method.
- Given that the client is loaded with the correct listings in the correct order, they should see the UI for the buy page with all the listings shown.

User Story #14

As a user, I would like to be able to change between light and dark mode.

#	Description	Estimated Time	Owner
1	Create an icon across all pages that when clicked toggles the algorithm.	1 Hr	Xavier
2	Create an algorithm that changes the html elements and text color of each page.	2 Hrs	Xavier
3	Implement an algorithm that updates the database with the user selected mode.	1 Hr	Xavier

4	Implement an algorithm that retrieves the user selected mode from the database.	1 Hr	Xavier

Acceptance Criteria:

- Given that the icon is created correctly, when the user clicks on the icon, the icon should change color to show that it is clicked.
- Given that light and dark mode algorithm is written optimally, when the user clicks on the icon, the algorithm should change the page from light mode to dark mode, and vice versa.
- Given that the retrieval algorithm is implemented correctly, when the user logs in again, the page should be set to the mode of what the user set previously.

User Story #15

As a student, I would like to be able to view my profile.

#	Description	Estimated Time	Owner
1	Create UI for profile page.	6 Hrs	Jeffrey
2	Create algorithm to retrieve users' profile data from the database.	1 Hr	Jeffrey
3	Create save button when users edit information.	1 Hr	Jeffrey
4	Create algorithm to take newly entered values and update the database.	1 Hr	Jeffrey

Acceptance Criteria:

- Given that the UI is created correctly, when users open the profile page, they should be able to see information such as their first name, last name, and PUID filled out.
- Given that the UI is implemented correctly, when the user clicks on profile, then they should be able to see their ratings.
- Given that the algorithm is implemented correctly, when the user saves changes, the algorithm should update the database with the new data from the user.
- Given that the UI is implemented correctly, when user makes edits but does not hit "save", their edited information will not be saved.
- Given that the UI is implemented correctly, users should be able to see how many purchases they made and how many sales they made.

User Story #16

As a student, I would like to be able to have a list of previously searched textbooks displayed when I open the home page.

#	Description	Estimated Time	Owner
1	Create UI for a list of previously browsed books	3 Hrs	Shicheng
2	Create a hyperlink to access previously browsed books	1 Hr	Shicheng
3	Save users' browsing history	3 Hrs	Shicheng
4	Create API to fetch users' browsing history	2 Hrs	Shicheng
5	Create an algorithm to remove duplicates from the database.	2 Hrs	Shicheng

Acceptance Criteria:

- Given that the UI is implemented correctly, users should see a list of previously viewed textbooks on their homepage.
- Given that the UI is implemented correctly, when the user clicks on a previously viewed textbook, it should link them to book listing.
- Given that previously viewed algorithm is written correctly, when the user clicks on a previously viewed textbook, it should update the database and remove the duplicate.

User Story #17

As a student, I would like to be able to view a dropdown of my previous searches on the search bar.

#	Description	Estimated Time	Owner
1	Create a dropdown that reveals when the search bar is clicked on that gives users the option to autofill the search bar when they clicked on a past search	1 Hr	Ryan
2	Create algorithm to save user searches	1 Hr	Ryan
3	Create algorithm to query past searches to display on the dropdown	1 Hr	Ryan
4	Create algorithm to display only relevant past searches as the user types into the search bar	1 Hr	Michio
5	Create algorithm that allows the user to delete their search history	1 Hr	Michio

Acceptance Criteria:

- Given that the algorithm works correctly, users should see their past searches as a dropdown when the search bar is clicked.
- Given that the algorithm works correctly, the dropdown should update as the user type into the search bar (if the user typed CS, only past searches with CS will appear on the dropdown).
- Given that the algorithm works correctly, users can specify a past search from the dropdown and delete it.
- Given that the algorithm works correctly, every search the user does should be saved into the database for
- Given that the web client works correctly, when users clicked on a past search, the search bar should be autofilled with that search.

User Story #18

As a student, I would like to be able to edit my profile.

#	Description	Estimated Time	Owner
1	Create button to edit.	1 Hr	Jeffrey
2	Create UI for the user to input their edits.	2 Hrs	Jeffrey
3	Create algorithm to change profile page inputs to editable.	1 Hr	Jeffrey

Acceptance Criteria:

- Given that the UI is created correctly, when a user hits the edit button, then all text inputs should be editable.
- Given that the UI is created correctly, a user should be able to see and change their preferred name.
- Given that the UI is created correctly, a user should be able to see and change their major.
- Given that the UI is created correctly, a user should be able to see and change their preferred meetup location.

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. As a buyer, I would like to scroll through images that a seller is trying to sell.
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, pending, completed).
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.