Boilerbazaar

Sprint 1 Retrospective

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

What went well?

In general, we built out the pages that were going be used in our website and integrated the important functions that were going to be used by the users. We integrated listing textbooks, searching for used textbooks, the profile page, the map page, and the about page.

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

Completed:

The navigation bar is at the top of every page, and it allows user to access different pages within our website. When they click on a menu item, they will be directed to that page. They can also see which page they are currently on by the underlined menu item.

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

Completed:

The textbook listing is stored in the database after being submitted. If the user doesn't fill out all the required inputs, then they will be prompted to enter inputs for the field. When they click on the sell page, they will see a page where they can input things about their textbook.

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
6	Allow users to add multiple pictures at once.	3 Hrs	Ryan

Completed:

Users can select multiple images for a listing. When the user submits the listing, the images are converted into base64 text strings and stored in AWS s3 database. We elected to use base64 instead of JPEG since storage is not a concern and base64 can be easily integrated back into HTML.

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

Completed:

While creating a listing, users can open a dropdown to specify the price of their textbook. This information is saved in the database when the listing is submitted. Users will not be able to submit a listing if the price is not specified.

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

Completed:

While creating a listing, users can open a dropdown to specify the condition of their textbook. This information is saved in the database when the listing is submitted. Users will not be able to submit a listing if the condition is not specified.

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

Completed:

The users have a textbox where they can enter a description of their textbook. The description text is optional and when user hits submit, it is stored in the database with the other listing inputs.

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

Completed:

The above algorithms are completed. Users can update or delete the listing if they are creator of the listings.

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
4	Create dropdown list for the different searches.	1 Hr	Mitch

Completed:

The user is presented with a search bar where they can enter text to query the database. A drop-down list also presents the user with options for searching. Pressing the submit button sends a request to the server to query the database based on user specifications and to return the results. The results are displayed to the user in a somewhat crude fashion.

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

Completed:

After a user queries something from the database, a list of filters for textbook edition and textbook condition is populated and displayed to the user. The user can then choose one filter from each respective category. After clicking the filter button, the initial query results are filtered locally based on the user's specifications. There is a drop-down for the user to select a sorting method such as title, author, or price, each ascending or descending. Once the user selects a sorting method, they can press the sort button to sort the query and filtered results locally.

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

Completed:

When the user clicks a link to a specific listing's dedicated page, the user is routed to that page. The page loads using the listing's ID as a parameter in the url. The database is then queried for the listing's information and displays it on the page. The page shows all of the details of the specified listing including condition, description and other information.

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

Completed:

When the user visits the Map page, they are presented with photos and information about all three Purdue bookstores on campus. This includes the hours, which are dynamically highlighted based on the current day of the week. It also includes an embedded Google Maps which shows the location of the bookstores and can redirect the user to a new tab for directions. There is a button at the top of the page that users can click to be informed about the closest bookstore to their current location.

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

Completed:

When the user visits the about page, they see the developers of the webpage. They can see contact information and a GitHub icon where they can click to navigate to each developer's GitHub page.

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

Completed:

When the user visits the Buy page, they are presented with a textbox that they can enter text in to search for. They can also specify what they want to search by (title, author, ISBN) using a dropdown. By pressing the submit button, the text from the textbox and the dropdown selection is sent to the database to query the listings. The listings corresponding to the query are returned to the client and displayed to the user. If the query is empty or missing a drop-down selection, it returns all the textbook listings from the database.

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

Completed:

When the user visits the page, the website style is updated to correspond to their previous dark mode preference from the database. In the top right corner, the user can press a button to change their dark mode preference. When the button is pressed, it changes the website style and updates the user's preference in the database.

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

Completed:

When the user loads the profile, the data that they had previously saved should be there. If it is a new user, their first name, last name, and PUID are filled out. Users can see their ratings and how many sales and purchases they have made. When the user clicks on the edit button, and does not save what they edited, it will not save the new information.

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

Completed:

When the user first enters the webpage, they can view their previously browsed textbooks. UI and hyperlink are finished. The user can click on the hyperlink to get to the listed textbook.

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

Completed:

Past searches made by the user are saved into the database and retrieved from the database when the user navigates to the buy page. When they are making another search, a dropdown will review past similar searches that the user can choose to autocomplete their search. An "X" button is present next to each field in the dropdown if the user chooses to remove that field from their search history.

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

Completed:

When the user clicks edit, inputs that are editable become available. A user can change their preferred name, major, and preferred meetup location.

What did not go well?

In general, the UI for some portions of our webpage is incomplete. We were also unable to complete user stories related to updating listings such as updating and removing images.

User Story #3

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

Not Completed:

We were unable to complete algorithms to update/remove images from the database. These algorithms include client-side and server-side ones. The UI to remove and update listings images is also not complete.

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
4	Create button for edit and button for delete.	1 Hr	Xavier

Not Completed:

We did not implement the UI for the button as we ran out of time to finish it before the demo. This was possibly due to poor planning as well as underestimating other user stories that relied on one another. We also realized that we would need to add a whole UI rather than just a button.

User Story #8

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

#	Description	Estimated Time	Owner
3	Create UI to display search results	6 Hrs	Shicheng

Not Completed:

The UI for the search results was completed however it only displays the books details. There was not much effort put into it.

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	Michio

Not Completed:

The user is not able to see a UI with the listing's information and actions to the listing displayed. The page only displays the queried information for the listing. This may have been caused by inadequate planning.

How should you improve?

Since this is our first sprint, there was a lot of trial and error for a lot of things. Since this is the first time for most of us using AWS, a lot of time was spent figuring out the correct settings for the API to function properly and for client-side request to succeed. This led to us underestimating the time needed to complete many of the tasks. Since we now have basic knowledge and experience with AWS, this will not take up significant development time for future sprints. In case such issues arise, we should carefully review AWS documentation to speed up our debugging process.

During our sprint, we were lacking effective and adequate communication, especially during the second week. Some group members missed weekly meetings which did not help us stay on the same page about how the completion of everyone's tasks was going. In future sprints, we need to work on making everyone's progress clearer to the rest of the team members. This includes asking for help well in advance of the deadline if a member is falling behind. We also need to work harder to attend every team meeting.

Another issue that we had this sprint was finishing things last minute. There were some user stories that were completed right before the sprint review and left us rushing to make sure everything else was working smoothly. It caused a sense of panic in our hearts. To solve this problem, we need to manage our time and finish user stories before we need to present them.

On our sprint planning document, we listed the number of hours that we thought each task would take. However, during our sprints, the tasks took longer than what we expected. Now that we know how long some tasks will take, we can solve this problem by allotting the time correctly for each task so that we don't have more user stories than we can take per sprint.