Boilerbazaar

Sprint 2 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 major features of our project and added messaging between users. Users can start a new conversation with a seller through the page of a seller's listing. We also cleaned up our overall UI to make it more modern and easier to use. Users can see their own listings as well as their viewing history and textbook watchlist. The watchlist consists of textbooks tracked by the user and displays price changes.

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

Completed:

When the user first opens our webpage, they are greeted with the login page. On this page they can sign in using their Purdue username or sign up with a new account. When they sign up with a new account, they are prompted with questions to answer before being able to create a new account. On success, they will be taken to the homepage.

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

Completed:

When a user selects a listing to view its dedicated page, the current view count that was already retrieved from the database is displayed on the page. When the listing is clicked the client also sends an updated view count to the database. The database then writes the updated listing data.

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

Completed:

When a user selects a listing from the Buy page, there is a button present on the UI for adding the listing to the user's watchlist. If the user clicks this button, it will be added to their watchlist if it is not already in it and the user will be notified of the change through the UI. Tracked textbooks can be removed from the Watchlist page.

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

Completed:

When going to the setting pages, users should be able to update their listings. On update from available to sold, that listing will be removed from future search results. On update from sold to available, that listing will be available on search results.

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

Completed:

When the user selects a listing on the buy page, a share button is displayed after all the listing's information. When the user clicks on the button a popup appears with working options to share the listing through a copying the link and email. The user can then click out of the popup to close it.

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

Completed:

When going to listing, user can click the link to view seller's profile including name, email address, preferred name, etc.

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

Completed:

When going to the profile of a seller, user can see number of sales of a seller. This number will be incremented after a complete transaction.

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

Completed:

When a user navigates to the Settings page, there is a tab for the user's watchlist. When the user clicks this tab, they are presented with the watchlist UI. This includes a list of all their tracked textbooks as well as any price changes that occurred on any of the listings. There is also a button present on each listing to delete the listing from the user's watchlist.

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

Completed:

On the settings page a button to direct to the viewing history is displayed. When the user enters the viewing history page the UI with all the previously views listings is shown. When a user continues to view listings on the buy page the client updates the database with the user's previously viewed listings to be displayed on the viewing history page.

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

Completed:

When a user creates a listing, they are required to add an associated class to the textbook listing. The user is also able to change the course of a listing when editing the listing. When a user searches for textbook listings, they can search and filter by class. On the page of a listing, the course associated with the textbook is displayed alongside the rest of the listing information.

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

Completed:

When a user opens the message page, an input field and a send button is shown at the bottom of the page, where the user can input their message and press send to deliver it to another user. A list of users that this user has messaged is displayed on the left. Clicking on one of these users will set whom the message is intended for. If user1 hasn’t messaged user2 before, an input field and send button is also available on any listing of user2, where upon sending the message, user2 will be added to the contacts list of user1.

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

Completed:

When a user opens the message page, a list of users that this user has messaged is displayed on the left. Clicking on one of these users will display past messages with timestamps on the page. If the messages are too big to be completely displayed, a scrollbar will appear to allow the user to scroll through messages. A refresh button is also available for the user to fetch new messages without refreshing the page itself.

User Story #3 (Sprint 1)

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

Completed:

On the edit page, the user can choose to delete certain (or all) pictures of their listing, or to add new images (cannot exceed 5 in total). On submit, the server will compile all changes that the user made and update the images accordingly.

User Story #7 (Sprint 1)

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

Completed:

Users listing is stored in the settings page. They can see all the listings that they have made. They are also able to edit the listing or delete the listing. When they delete the listing, they are asked whether they want to delete or cancel. If they edit the listing, the listing information will be filled out and they can change what they want. All changes are saved to the database.

User Story #10 (Sprint 1)

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

Completed:

When users click on a listing on the buy page, the listing opens with the information of the textbook filled out. They can also scroll through pictures of the textbook.

What did not go well?

In general, mostly everything went smoothly this sprint. All of us completed our user stories before the sprint review, besides a few mishaps. One thing that did not go well was our testing document. We did not have a completed testing document to turn in by the demo.

Another problem we encountered was a change in the primary key for our users that wasn't communicated well enough, which led to more time allocated to refactor our code.

How should we improve?

During this sprint, we had great communication, and we did not rush to finish anything at the last minute. We also had great estimates for how much time each task might take.

One thing that could be improved is to not do someone else's tasks because that can lead to some miscommunication and some holdups in our project. To solve this problem, we can have better communication letting the other person know beforehand if a part of theirs will be completed.

We should also communicate better on big changes that can potentially affect the correctness of completed user stories, either through pinging everyone on Discord, or announcing it in our meetings.

Another thing that we could improve on is the testing document. At the end of our sprint review, we were surprised that there was a testing document to turn in. For our next sprint we will complete the testing document in advance of our demo.