



CS 30700
Spring 1 Retrospective

Team 32:

Ali Fareed Shirazi

Sara Xiao

Jun Shern Lim

Ming Yan Ti

What went well during Sprint 1?

In general, we were successfully able to implement all user stories as we described them. Thanks to easy communication through the Discord messaging app, we were able to facilitate active updates from each team member and set up weekly meeting times for working on and implementing solutions. We were also able to test our application effectively and practice showcasing it multiple times.

User Story #1

As a user, I would like to register for an Itembnb account.

No.	Description	Estimated Time	Owner
1	Create a sign-up page with all the necessary fields (such as email, name, and password) with the submit button.	3 hours	Ali
2	Create a function in the frontend to automatically validate all the inputs in the sign-up fields on the client side.	2 hours	Ali
3	Create a profile picture section where users can upload images from their local file system.	1 hour	Ali
4	Create an API function in the backend to send the create query to the database.	2 hours	Jun
5	Create an API function to check if the email has been registered before and send a success/failure alert message to the user indicating if the sign up was successful from the backend.	2 hours	Jun

Completed

From the homepage, if the user is not logged in, they can access a “sign up” page that allows them to enter pertinent information required to create an account. If the input into the signup page is valid, upon clicking the submit button, they will be successfully registered as a new user and their account information added to the database.

User Story #2

As a user, I would like to log in and log out of my account.

No.	Description	Estimated Time	Owner
1	Create a sign-in page with all the necessary fields (such as email and password) with the submit button and a sign up button (if the user wants to register for an account instead).	2 hours	Ali
2	Validate the user login credentials in the front end after getting the stored details in the database	2 hours	Ali
3	Create an API function in the server-end to send the read query to the database to get the user data and validate.	2 hours	Jun
4	Create a success/failure alert message to tell the user if the sign-in is successful.	1 hour	Ali

Completed

The website tracks whether a user is logged in or not, and the interface changes accordingly. If the user is not logged in, they will be able to see options for signing up and logging into their account. If the user is logged in, they will be able to see options for accessing their profile page, creating item posts and requests, and logging out.

User Story #3

As a user, I would like to view/edit/delete my own profile.

No.	Description	Estimated Time	Owner
1	Create a clickable avatar profile in the homepage that redirects to my user profile.	1 hour	Ali
2	Create a user profile page with all the user data displayed.	3 hours	Sara
3	Create a function to get the necessary data via HTTP request to the server on the client side.	2 hours	Sara

4	Create an edit button that allows users to toggle to edit mode in their profile.	1 hour	Sara
5	Create a save button to save changes made to user profile	1 hour	Sara
6	Add a profile picture section where users can upload images from their local file system.	0.5 hour	Sara
7	Create a function in the front end that automatically validates all the inputs in the fields on the client side.	1 hour	Sara
8	Create a delete feature that asks for a confirmation before deleting a user's profile.	2 hours	Sara
9	Create functions in the backend to send queries and changes to the database, as well as retrieve data from the database to be sent to the front end.	3 hours	Jun

Completed

If a user is logged in, clicking their avatar in the upper right corner of the page takes them to their profile page. From there, they can see all the item posts they've made and all the item requests they've made. They can also edit their account name, description, and avatar.

User Story #21

As a borrower, I would like to create an item request for an item I want to borrow that is not listed on the site.

No.	Description	Estimated Time	Owner
1	Create a "Create Request Post" page with all the necessary fields to create a request post.	4 hours	Sara
2	Create an API function in the server-end to send the "create request post" query to the database.	1 hour	Jun
3	Create a function in the front end to validate user input.	1 hour	Sara
4	Create a success/failure alert message to tell the user if the request posting is successful.	1 hour	Sara

Completed

If a user is logged in, there is a link in the navbar that will take them to the “Create Request Post” page, and from there they will be able to fill in the valid information and submit it. If the user input is invalid, e.g. empty, the submission will not go through and error messages will tell the user they must have valid inputs in the required fields.

User Story #22

As a borrower, I would like to edit/delete my request post.

No.	Description	Estimated Time	Owner
1	Create the “Modify Request Post” page with all the necessary fields populated with the existing values from the request.	4 hours	Sara
2	Create a function in the front end that automatically validates all input fields on the client side.	2 hours	Sara
3	Create a success/failure alert message to tell the user if the item request edit is successful.	1 hour	Sara
4	Create a delete feature that asks for a second confirmation for the item request deletion.	2 hours	Sara
5	Create functions in the backend to send queries and changes to the database, as well as retrieve data from the database to be sent to the front end.	2 hours	Jun

Completed

If a user is logged in and viewing a request they created, they will have access to an Edit and Delete button. Clicking the Edit button toggles an edit mode wherein the user can change their item request. Editing a request only goes through if the input is valid, and otherwise error messages inform the user which fields are incorrect. Clicking the Delete button will create a confirm/deny deletion popup from which the user can delete their request.

User Story #25

As a lender, I would like to post an item with a price and assign it to predefined categories.

No.	Description	Estimated Time	Owner
1	Create a "Create Item Post" page with all the necessary fields with a submit button.	4 hours	Ming
2	Create a function in the front end that automatically validates all the inputs in the fields on the client side.	2 hours	Sara
3	Create a multi-select feature for items categories to allow users to easily select them.	2 hours	Ming
4	Create an API function in the backend to send the "create item post" query to the database.	1 hours	Jun
5	Create a success/failure alert message to tell the user if the item posting is successful.	1 hour	Ming

Completed

The item post page successfully takes in the item's details as inputs and verifies them, and once the details are correct and valid the user is able to create this post and send it to the backend, from which it is sent to the homepage and is displayed there whenever the page is loaded.

User Story #26

As a lender, I would like to add a thumbnail image and multiple preview images to my posted item.

No.	Description	Estimated Time	Owner
1	Create an item picture section where users can upload multiple images from their local file system.	3 hours	Ming
2	Make a scrolling feature to move between images.	3 hours	Ming
3	Create a feature that lets users delete or replace images they've uploaded.	3 hours	Ming

4	Create a feature that lets users select the order in which the images will be displayed.	3 hours	Ming
5	Implement validation to check if the input image has a valid file type.	1 hour	Ming

Completed

We developed a picture upload and preview section for users to upload and arrange their pictures. We perform validation on the uploaded images and make sure only *jpg* and *png* files are accepted. We have two bars: the picture review section and the picture sources bar. The users will first upload the images into the picture sources bar, then they can rearrange the images in the picture review section by dragging or double clicking to delete an image in the picture review section. Images from the picture source bar can be deleted by clicking on the trash icon.

User Story #27

As a lender, I would like to edit/delete my posted item.

No.	Description	Estimated Time	Owner
1	Create a "Modify Item Post" page where all the relevant fields/tags can be edited.	3 hours	Ming
2	Add a multi-select feature for item categories to allow users to easily modify them.	0.5 hours	Ming
3	Create a delete feature that asks for a second confirmation for the item deletion.	2 hours	Ming
4	Create a function in the front end that automatically validates all the inputs in the fields on the client side.	2 hours	Sara
5	Create an API function in the backend to send the update/delete item post query to the database.	2 hours	Ming
6	Create a success/failure alert message to tell the user if the item posting edit is successful.	1 hour	Ming

Completed

We developed a dedicated "edit item page" such that when the logged-in user is the owner of that item, they can edit the item page by toggling on the edit button.

When the save button is clicked, we then send the updated item data back to the database and save the latest changes. We also developed a delete feature so users can delete their posted item and the deletion will be reflected in the database.

User Story #8

As a user, I would like to see a homepage with a feed for items up for borrowing and items that people have requested.

No.	Description	Estimated Time	Owner
1	Create an API function on the backend to get recent item posts and requests from the database to be sent to the client.	2 hours	Jun
2	Create two sections to display item posts and request posts, so users can view either.	2 hours	Ali
3	Create a scrollable responsive homepage with item posts and item requests formatted with some details and clickable buttons.	4 hours	Ali
4	Create a navigation panel that redirects the user to different pages.	2 hours	Ali

Completed

When a person visits the homepage of the website, they will see an interface that has a list of existing item posts and item requests. Each item post and request links to a larger page that contains all the pertinent information about the post/request. The navbar at the top allows the user to access various features of the website if they are logged in and allows the user to login/signup if they are not.

User Story #11

As a borrower, I would like to view a posted item's details - description, lender info, date, etc.

No.	Description	Estimated Time	Owner
1	Create an item post page that contains pertinent information about the item.	4 hours	Ali
2	Create a function to get the necessary item and lender data via HTTP request to the	1 hours	Ali

	server on the client side.		
3	Create a preview image UI that lets the user click through all an item's preview images with an image carousel.	3 hours	Ali
4	Create a lender preview section in the page that has a preview of information about the lender.	2 hours	Ali

Completed

From an item post's card in the homepage, a user can click "Read More" to access a full page corresponding to the item in question, containing a lot more information about the item and the user who posted it. The page allows the user to preview the item's images and its lender.

User Story #12

As a borrower, I would like to favorite a posted item to save it for later.

No.	Description	Estimated Time	Owner
1	Create an appropriate favorite icon for each item post that can be toggled.	2 hours	Jun
2	Make the favorite icon a visual indicator of whether the item is favorited or not.	2 hours	Jun
3	Send a request to the server to update the item's favorite status for a user when the user clicks the favorite icon.	2 hours	Jun

Completed

Each item post has a favorite icon in the corner. When clicked, this icon will toggle on or off, also toggling the item's favorite status. This information gets updated live, and toggling an item's favorite status on one page will toggle it in the database and therefore its updated favorite status will be shown to the user on every page.

User Story #13

As a borrower, I would like to see a list of all my favorite items.

No.	Description	Estimated Time	Owner
-----	-------------	----------------	-------

1	Create UI of favorite items list accessible from user profile.	3 hours	Jun
2	Create a feature that lets a user switch between different view modes for the list, for ease of view.	3 hours	Jun
3	Create a feature that lets a user unfavorite an item while viewing the favorite list.	2 hours	Jun
4	Create an API function in the backend to send the select/update/delete favorite item queries to the database.	2 hours	Jun

Completed

From a user's profile, the user can go to a list of all their favorite items, which shows the items in either an expanded or minimalist view. If the user unfavorites an item from this page, the item will disappear from the list. If the user has no favorite items, then a message will display the list as empty.

What did not go well during Sprint 1?

We achieved general success, and managed to successfully complete all of our user stories and their acceptance criteria. However, we were working up to the deadline and during the last few days many changes and pull requests were made, for bug fixing and adding required features. Due to our relative unfamiliarity with some of the technologies and components used, we ended up getting delayed here and there due to having to research bugs and issues, or specific implementation details. We also had to backtrack on some decisions after having already worked on them for some time, because we found out later that there may have been an easier or more correct way of implementing those features. As a result, we had to go back and revise our work, which further added to the complexity of the project.

How should we improve?

In the future, all members should strive to be involved and organized, which includes attending meetings on time and doing work sooner rather than later.

We should also have more consistent formatting of backend-frontend integration functions, and ensure we discuss when we change things in the backend. This ensures integration between the frontend and backend is as smooth as possible. Sometimes, changes made to the backend would break things in the frontend,

which would require us to go through trying to fix bugs until we found what didn't work anymore. For example, we sometimes encountered issues with how the JSONs of server responses are formatted, and inconsistency in the chain of functions passing that information from the backend to the frontend regarding what information is kept and discarded where. Having more consistency for the backend-frontend integration functions would be helpful, and we should strive to do this during our next sprint.

We noticed that our method of splitting user stories to multiple users sometimes led to confusion regarding who was responsible for what, and additionally the fragmentation of responsibility meant some members lacked knowledge on components that were important to their tasks. In the future, we should do a better job assigning tasks to members that are closely related and not heavily dependent on other members' tasks.