TroveSpace

Sprint 1 Retrospective

Team 13 | https://github.com/baker323/TroveSpace

Adam Baker, Kareem Elhadidi, Rei Orikata, Pritesh Kadiwala

What went well?

General

We did a good job in understanding how our product would be designed. In the beginning, we first drew out our designs and flow charts to understand the flow of our project. By doing this, we were able to make it easier on ourselves while doing the frontend and while connecting it with the backend. Furthermore, while we were working on designing the front end, a simple HTML page was created by the backend team so that they could complete the backend portion of the project simultaneously. Finally, when all the parts of the project were complete, we had to only replace the front end files to get a working project. Overall, we were able to successfully complete all of our planned user stories on time.

User Story #1

As a user, I would like to open the application from Chrome Desktop web browser (initial setup for frontend and backend).

#	Task Description	Estimated Time	Owner(s)
1	Set up frontend pages and layout	5	Pritesh
2	Set up AngularJS controllers for each module	5	Kareem
3	Set up Firebase to store data objects	5	Adam

- Completed: The app is able to be opened in Chrome and the navigation bar allows for seamless transitions from page to page.

User Story #2

As a user, I would like to create an account in the application.

#	Task Description	Estimated Time	Owner(s)
1	Implement UI for account creation	3	Pritesh
2	Set up communication with Firebase to allow account authentication	3	Adam
3	Implement error handling for UI	2	Pritesh
4	Perform tests on account creation	2	Rei

- Completed: Users are able to create accounts and receive informative error messages.

User Story #3

As a user, I would like to login to my account.

#	Task Description	Estimated Time	Owner(s)
1	Implement UI for login	1	Pritesh
2	Implement Firebase login methods	2	Adam
3	Implement error handling for UI	1	Pritesh
4	Perform tests on account login	1	Rei

- Completed: Users are able to login to their account and send a password reset link if they are unable to login.

User Story #4

As a user, I would like to logout of my account.

#	Task Description	Estimated Time	Owner(s)
1	Implement UI for logout	1	Pritesh
2	Implement Firebase logout methods	2	Adam
3	Implement error handling for UI	1	Pritesh
4	Perform tests on account logout	1	Rei

- Completed: Users are able to logout of their account and have all their information persist when they log back in.

User Story #5

As a user, I would like to change my account password.

#	Task Description	Estimated Time	Owner(s)
1	Implement UI for changing password	3	Pritesh
2	Implement Firebase methods for changing password	3	Adam
3	Implement error handling for UI	2	Pritesh
4	Perform tests on changing password	2	Rei

- Completed: Users are able to change their password and successfully login again using their new password.

User Story #6

As a user, I would like to delete my account.

#	Task Description	Estimated Time	Owner(s)
1	Implement UI for account deletion	1	Pritesh
2	Implement Firebase methods for deleting accounts	2	Adam
3	Implement error handling for UI	1	Pritesh
4	Perform tests on account deletion	1	Rei

- Completed: Users are able to delete their account and are not able to login until they create the account again.

User Story #7

As a user, I would like to have multiple distinct collections.

#	Task Description	Estimated Time	Owner(s)
1	Implement UI for creating collections	3	Kareem
2	Implement UI for viewing collections	3	Kareem
3	Implement UI for renaming collections	1	Kareem

4	Implement UI for deleting collections	1	Kareem
5	Implement UI for switching between collections	1	Kareem
6	Implement Firebase methods for creating a collection	3	Adam
7	Implement Firebase methods for fetching a user’s collection	2	Adam
8	Implement Firebase methods for renaming a user’s collection	2	Adam
9	Implement Firebase methods for deleting a collection	2	Adam
10	Implement error handling for collection creation	1	Kareem
11	Implement error handling for collection renaming	1	Kareem
12	Implement error handling for collection deletion	1	Kareem
13	Perform tests on collection creation	1	Rei
14	Perform tests on collection viewing	1	Rei
15	Perform tests on collection renaming	1	Rei
16	Perform tests on collection deletion	1	Rei

- Completed: Users are able to have multiple types of collections. They can create collections, switch between collections, rename collections, and delete collections.

User Story #8

As a user, I would like to be able to track duplicate items in a single entry.

#	Task Description	Estimated Time	Owner(s)
1	Implement UI for updating duplicates	2	Kareem
2	Implement UI for showing duplicates	1	Kareem
3	Implement Firebase methods for updating the number of duplicates	2	Rei

4	Implement Firebase methods for fetching the number of duplicates	2	Rei
5	Implement error handling for UI	1	Kareem

- Completed: Users are able to specify the number owned of a specific collectible when viewing the information page of the collectible. They can edit and save the multiple count.

User Story #10

As a user, I would like to add items that do not exist in the TroveSpace database.

#	Task Description	Estimated Time	Owner(s)
1	Implement the UI for creating collectibles	10	Kareem
2	Implement Firebase methods for creating collectibles	10	Rei
3	Implement Firebase methods for adding collectibles to Troves	5	Rei
4	Implement error handling for UI	2	Kareem
5	Perform tests on collectible creation	3	Rei

- Completed: Users are able to add collectibles to the database by visiting the pages for different categories and entering information specific to that category.

User Story #11

As a user, I would like to add items that already exist to one or more of my collections.

#	Task Description	Estimated Time	Owner(s)
1	Implement the UI for adding collectibles to collections	3	Pritesh
2	Implement Firebase methods for adding collectibles to collections	3	Adam
3	Implement error handling for UI	2	Pritesh

- Completed: Users are able to add collectibles to one of their collections by visiting the pages of specific categories and selecting the option from a dropdown for each item.

User Story #12

As a user, I would like to add items that already exist in the database to my wishlist.

#	Task Description	Estimated Time	Owner(s)
1	Implement the UI for adding collectibles to wishlist	1	Pritesh
2	Implement Firebase methods for adding collectibles to wishlist	2	Adam
3	Implement error handling for UI	1	Pritesh
4	Perform tests on adding collectibles to wishlist	1	Rei

- Completed: Users are able to add collectibles to their wishlist by visiting the pages of specific categories and selecting the option from a dropdown for each item.

User Story #23

As a user, I would like to create a new Trove.

#	Task Description	Estimated Time	Owner(s)
1	Implement the UI for Trove creation	3	Pritesh
2	Implement the UI for viewing Troves	3	Pritesh
3	Implement the UI for viewing all Troves	3	Pritesh
4	Implement Firebase methods for Trove creation	2	Adam
5	Implement Firebase methods for fetching Trove information	2	Adam
6	Implement Firebase methods for fetching a list of all Troves	2	Adam
7	Implement error handling for UI	2	Pritesh
8	Perform tests on Trove creation	1	Rei
9	Perform tests on Trove viewing	1	Rei

10	Perform tests on viewing all Troves	1	Rei

- Completed: Users are able to create a new trove and view the newly created trove with the information that they specified when they were creating it on the trove page.

User Story #24

As a user, I would like to name the fields in a Trove

#	Task Description	Estimated Time	Owner(s)
1	Implement the UI for naming fields	3	Kareem
2	Implement Firebase methods for naming Trove fields	3	Adam
3	Implement error handling for UI	2	Kareem
4	Perform tests on naming fields	2	Rei

- Completed: Users are able to add custom fields that can then be filled out when creating a new collectible for that specific category. Fields can easily be added and removed.

What did not go well?

Time Management

We waited until the last minute to finish the UI, so we did not have a lot of time to finish the database functionality and testing. Luckily, the backend functions were able to be created alongside the UI, but adding them to the UI and testing them was completed later than desired.

Automated Testing

While we covered a large portion of the important functionality with automated tests, we did not achieve the level of coverage we were aiming for. We had to test the remaining functionality manually, which took more time in the long run.

Version Control Management

We had some conflicts when pushing our work to GitHub that held back our progress. Since we were all using separate branches to push our completed code, not everyone was able to have the latest changes when merging branches.

User Story #8

As a user, I would like to be able to track duplicate items in a single entry.

#	Task Description	Estimated Time	Owner(s)
6	Perform tests on updating duplicates	1	Rei
7	Perform tests on showing duplicates	1	Rei

- Incomplete: We had trouble keeping our Git branches synced with the latest changes, so the duplicate count was not properly showing in the branch we used for testing.

User Story #11

As a user, I would like to add items that already exist to one or more of my collections.

#	Task Description	Estimated Time	Owner(s)
4	Perform tests on adding collectibles to collections	2	Rei

- Incomplete: We were not able to figure out how to get the automated test software to click on certain elements to enter text when the fields were dynamically generated.

How should we improve?

Time Management

We should focus on completing the front end portion before Spring Break as it would help us have more time to complete and link the entire project. Also by doing this, it would help reduce the stress put on the backend and the testing team members.

Scrum Meetings

It would be better if we were able to improve our communication by reporting our scheduling problems earlier so that we would have time to reschedule the meeting. This would help keep the project on track and eliminate any issues we have earlier.

Version Control Management

We should make sure we coordinate our Git pushes and pulls better so that everyone is on the latest version. If we have any more issues merging branches, we should look into ways to make this process easier for all members of the group.