PurdueParty.io

Sprint Two Planning Document - CS407 Team 4

Aaron Cox, Yu-Lun Gao, Michael Maganini, Raziq Ramli, Matthew Story, Bryan Wang

Sprint Overview

Now that we have most of our core functionality implemented, the goal of sprint two will be to expand upon and round out that feature set. This means we will largely be focusing on user stories which serve to provide more ways in which users can interact with our present offerings. To that end, backlog items that pertain to the marketplace, events board, class forum, and user profile will continue to be targeted to improve the overall user experience of our app. Additionally, while we want to refine our current features, we need to continue to introduce new ones as well via Clubs.

Scrum Master: Matthew Story

Meeting Schedule: Monday’s and Friday’s at 4:00 PM + As needed

Risks and Challenges

A significant risk for this sprint revolves around how we’ve decided to approach database management. As our project scales, we are seeing slower load times and an exponential increase in both reads and writes. Should our current approach prove unscalable, we will have to change how and where we are storing data in order to facilitate a larger project. In terms of challenges, we are continuing to struggle to find times in which the whole group is able to collaborate, damaging our overall communication abilities. We need to ensure that everyone is able to attend each stand-up and is constantly checking in on the group discord.

Current Sprint Detail

User Story #5

As a user, I would like to have a profile with a brief bio and image.

#	Description	Estimated Time	Owner
1	Add a UI field to account creation page for an account bio	2 Hrs	Michael
2	Add a UI field to account creation page for uploading a profile image	4 Hrs	Michael
3	Test input validation	2 Hr	Michael
4	Test Firebase Database Implementation	2 Hr	Michael
User Story #5 Total Hours		10 Hours	

Acceptance Criteria:

- Given that the UI field for creating an account bio on the account creation page is implemented correctly, when a user navigates to the account creation page, there should be an available field for typing in a bio.
- Given that the UI field for creating an account bio on the account creation page is implemented correctly, when a user types in their desired bio, they should be presented with an error message if their bio is greater than 256 characters upon attempting to create their account.
- Given that the UI field for uploading an image on the account creation page is implemented correctly, when a user uploads an image, they should be able to see a preview of the image they have uploaded before attempting to create their account.
- Given that the UI field for uploading an image on the account creation page is implemented correctly, when a user tries to upload an image, they should only be able to view file types JPG, JPEG, or PNG when browsing their system for photos.
- Given that the UI fields for uploading an image and for creating a bio on the account creation page were implemented correctly, when a user uploads a valid image and or creates a valid bio, the information should be stored in the Firebase database.

User Story #13
As a user, I would like to hide my profile from users.

#	Description	Estimated Time	Owner
1	Develop UI for edit hide/show profile	2 Hrs	Yu-Lun
2	Communicate with Firebase Database to update the hide/show.	2 Hrs	Yu-Lun
3	Test to ensure the UI responds to changes in the Firestore Database	2 Hrs	Yu-Lun
4	Test to ensure the UI redirects to the appropriate page	1 Hr	Yu-Lun
User Story #13 Total Hours		7 Hours	

Acceptance Criteria:

- Given that the UI of the edit profile page is adjusted correctly, when a user clicks the edit profile button, there should be a field for the user that allows them to choose to hide their profile or not.
- Given that the communication with Firestore Database is configured correctly, when a user updates his/her profile states, the relevant field should be changed in the database.
- Given that the communication with Firestore Database is configured correctly, when the user searches for a hidden profile, the hidden user’s profile should not appear.
- Given that the UI of hiding/showing user profile is injected correctly, when a user successfully edits this field, the user will be presented with a message that informs them their account was successfully hidden.
- Given that the link to a hidden user’s profile page is working correctly, when a user attempts to go to a hidden user’s profile page, the hidden user’s profile page should have a hidden or private status.

User Story #11

As a user, I would like to check other user profiles.

#	Description	Estimated Time	Owner
1	Develop search field on the navigation bar that supports searching for other user's profiles	3 Hrs	Michael
2	Develop redirection to a user's profile page when a profile is selected from the list of profiles	2 Hrs	Michael
3	Develop UI for viewing a list of profiles that are most relevant to the search	4 Hrs	Michael
4	Communicate with Firestore Database in order to show a the list of accounts relevant to the profile search	3 Hrs	Michael
4	Communicate with Firestore Database in order to show a selected profile's information	1 Hr	Michael
4	Test for responsive UI rendering	2 Hr	Michael
User Story #11 Total Hours		15 Hours	

Acceptance Criteria:

- Given that a search field on the navigation bar is implemented correctly, when a user searches for a specific profile by their email or username, they are presented with a list of results sorted by similarity to the search.
- Given that the UI for viewing a list of searched profiles is implemented correctly, when a profile is selected from a searched list of profiles, the user is presented with a page that provides that account's relevant profile information.
- Given that a search field on the navigation bar is implemented correctly, when a user searches for a specific profile by their email or username, they will not be presented with results associated with profiles that have been deleted.
- Given that communication with Firestore is implemented correctly, when a user searches for a specific profile, profiles that are completely unrelated to the query will be ignored.
- Given that the UI renders responsively, when a user uses the search bar to query for a specific profile, the page will dynamically render the results without hard reloading.

User Story #48
As a user, I would like to be able to set a sell price and item description for my sell listings.

#	Description	Estimated Time	Owner
1	Inject UI that contains input fields for sell price and item description	1 Hrs	Bryan
2	Connect the sell-price and item description from the UI to FireBase database object	1 Hrs	Bryan
3	Test UI for accurate rendering of the sell-price and item description	1 Hr	Bryan
4	Test input validation on the input fields	1 Hr	Yu-Lun
User Story #48 Total Hours		4 Hours	

Acceptance Criteria:

- Given that the UI for create-sell-listing is functional, when the user navigates to create-listing page, there should be input fields on the page where a user can enter a price and item description for their sell listing.
- Given that the user enters invalid input, when the user hits the submit button, the user should either be notified of invalid input or the field should not allow them to enter incorrect input.
- Given that the user has submitted successfully, when the user goes to the landing page and info page, the sell-listing price should be accurately reflected on the landing page tiles, the marketplace preview tile, and the sell-listing page.

User Story #49
As a user, I would like to be able to list my contact information to potential buyers.

#	Description	Estimated Time	Owner
1	Inject UI that supports contact information on sell listings	2 Hrs	Matt
2	Inject contact information into the UI of the edit and creation pages	2 Hrs	Bryan
3	Connect owner's email to sell-listing	2 Hrs	Matt
4	Add contact info to Firestore Database	2 Hrs	Bryan
5	Test UI for appropriate rendering of contact info	1 Hr	Matt
6	Test input validation on new fields	1 Hr	Bryan
User Story #49 Total Hours		10 Hours	

Acceptance Criteria:

- Given that the UI for contact information is injected correctly into the relevant pages, when a user navigates to the edit and creation pages for sell-listings, there should be a component that allows a user to input their phone number.
- Given that the UI for contact information is injected correctly into the sell-listing, when a user inputs contact information, the owner's phone number should appear on that sell listing.
- Given that the UI for the contact information is implemented correctly, when a user navigates to the sell-listing edit and creation pages, only valid phone number formats should be accepted.
- Given that the connection between a sell-listing and its owner's email is implemented correctly, when a user navigates to the information page for a specific sell-listing, the owner's email should appear on that sell-listing.
- Given that the contact info is correctly added into Firestore Database, when a user adds their contact information, their phone number should be added onto the sell-listing in Firestore.

User Story #50

As a user, I would like to be able to include an image of the item I am selling in my sell listings.

#	Description	Estimated Time	Owner
1	Inject UI for uploading images onto sell-listing edit and creation pages	2 Hrs	Matt
2	Inject UI for rendering images onto individual sell-listing pages	2 Hrs	Matt
3	Configure Firebase Storage to store relevant images	2 Hrs	Matt
4	Link sell-listing item in Firestore to relevant Firebase Storage	3 Hrs	Matt
5	Test UI (image rendering, upload)	1 Hr	Matt
User Story #50 Total Hours		10 Hours	

Acceptance Criteria:

- Given that the UI for uploading images onto the sell-listing is implemented correctly, when a user navigates to the sell-listing edit and creation pages, there should be an option to upload an image.
- Given that the UI for rendering images onto the individual sell-listing page is implemented correctly, when a user navigates to a sell-listing's individual page, the image that the owner uploaded should be rendered along with the rest of the information.
- Given that the UI for rendering images onto the individual sell-listing page is implemented correctly, when a user navigates to a sell-listing's individual page, a default image should be rendered if no image was provided.
- Given that Firebase storage is configured correctly, when an image is uploaded, the image should be stored under a human-readable path as opposed to its direct download link.
- Given that the sell-listings in Firestore are correctly linked to their images in Firebase, when an image is linked to its sell-listing, the sell-listing should contain the downloadable (renderable) link as opposed to its Firebase Storage path.

User Story #46

As a user, I would like to edit my sell listing on the marketplace.

#	Description	Estimated Time	Owner
1	Develop UI that supports editing of sell listings (fields such as description, contact info, price, and image)	4 Hrs	Matt
2	Communicate with Firestore Database in order to update the sell listing	3 Hrs	Matt
3	Fetch the original sell listing object from Firestore	1 Hr	Matt
4	Cross-reference owner of sell listing with current user in Authentication	2 Hrs	Matt
5	Test for responsive UI rendering	1 Hr	Matt
6	Test input validation	1 Hr	Matt
7	Test sell-listing owner verification	1 Hr	Matt
User Story #46 Total Hours		13 Hours	

Acceptance Criteria:

- Given that the UI for editing a sell-listing is implemented correctly, when a user edits a sell-listing, the type of acceptable data should be consistent with the type of acceptable data in create sell-listing (equivalent input validation).
- Given that the cross-reference between the sell-listing owner and Firebase Authentication is done correctly, when a user who is not the owner of the sell-listing navigates to its info page, the component that routes to the edit form of that particular sell-listing should not render.
- Given that the cross-reference between the sell-listing owner and Firebase Authentication is done correctly, when a user who is the owner of the sell-listing navigates to its info page, the component that routes to the edit form of that particular sell-listing should render.
- Given that both communication with Firestore Database and responsive rendering are implemented correctly, when an authorized user updates an existing sell-listing, the relevant fields should be changed in the database and reflected in the UI.
- Given that both communication with Firestore Database and the UI are implemented correctly, when an authorized user updates an existing sell-listing, fields that have been left blank should not update in the database or be overwritten.

User Story #47
As a user, I would like to delete my sell listing on the market place.

#	Description	Estimated Time	Owner
1	Develop UI that supports deletion of sell listings	2 Hrs	Matt
2	Address site-wide uncertainties/unconformities with deletion UI	2 Hrs	Matt
3	Communicate with Firestore Database in order to delete the sell listing	3 Hrs	Matt
4	Fetch the original sell listing object from Firestore	1 Hrs	Matt
5	Cross-reference owner of sell listing with current user in Authentication	2 Hrs	Matt
6	Test for responsive UI rendering	1 Hr	Matt
7	Test input validation	1 Hr	Matt
8	Test sell-listing owner verification	1 Hr	Yu-Lun
User Story #47 Total Hours		13 Hours	

Acceptance Criteria:

- Given that the UI for deleting a sell-listing is injected correctly, when a user attempts to delete a sell-listing, the user should be met with a confirmation message before final deletion.
- Given that the UI for deleting a sell-listing is injected correctly, when a user confirms deletion of a specific sell-listing, the user should be notified that the listing was successfully deleted.
- Given that the cross-reference between the sell-listing owner and Firebase Authentication is done correctly, when a user who is not the owner of the sell-listing attempts to navigate to its edit/deletion page via url, they should be redirected to the landing page.
- Given that the cross-reference between the sell-listing owner and Firebase Authentication is done correctly, when a user who is not the owner of the sell-listing attempts to navigate to its edit/deletion page via url, they should be able to access the page.
- Given that both communication with Firestore Database and responsive rendering are implemented correctly, when an authorized user deletes an existing sell-listing, the relevant item should be deleted in the database and removed from the marketplace UI.

User Story #29
As a user, I would like to be able to view a list of clubs available at Purdue.

#	Description	Estimated Time	Owner
1	Develop UI for club landing page	1 Hrs	Yu-Lun
2	Communicate with Firestore Database to get the clubs list	2 Hrs	Yu-Lun
3	Test to ensure the UI show the correct clubs in the database	2 Hrs	Yu-Lun
User Story #29 Total Hours		5 Hours	

Acceptance Criteria:

- Given that the UI is implemented correctly, when a user navigates to the club landing page, the user should view a dynamically rendered list of all available clubs.
- Given that the UI is implemented correctly, when a user navigates to the club list and there are no clubs available, the user should be presented with a message stating that there are no available clubs.
- Given that the UI is implemented correctly, when a user clicks on a specific club, they should be redirected to that specific club's information page.

User Story #31

As a user, I would like to be able to navigate to a club info page.

#	Description	Estimated Time	Owner
1	Develop UI for club information page	4 Hr	Yu-Lun
2	Pull club information from club object in Firestore Database and display to UI	2 Hr	Yu-Lun
3	Test to ensure that the UI displayed mock data correctly	2 Hr	Yu-Lun
User Story #31 Total Hours		8 Hours	

Acceptance Criteria:

- Given the club information page is implemented correctly, when a user navigates to the club page, then the user will be able to see the club title.
- Given the club information page is implemented correctly, when a user navigates to the club page, then the user will be able to see the club description.
- Given the club information page is implemented correctly, when a user navigates to the club page, then the user will be able to see the club category.
- Given the club information page is implemented correctly, when a user navigates to the club page, then the user will be able to see the image of the club.
- Given the club information page is implemented correctly, when a user navigates to the club page, then the user will be able to see a list of events for that club.

User Story #30

As a user, I would like to be able to search the list of Purdue clubs.

#	Description	Estimated Time	Owner
1	Create and style search bar UI	2hrs	Raziq
2	Incorporate search bar into the clubs landing page seamlessly	1.5hrs	Raziq
3	Augment fetch clubs action to return results based on keywords and integrate it to the page component	2hrs	Raziq
4	Create necessary controllers to update clubs landing page after user finished typing search keywords and after user clicks on the search button	1hr	Raziq
5	Test that correct search results are displayed	1.5hrs	Raziq
User Story #30 Total Hours		8 Hours	

Acceptance Criteria:

- Given the search UI was implemented, when the user navigates to the clubs landing page, there will be an option to enter a search keyword.
- Given the search feature was implemented successfully, when the user enters a keyword, only clubs that match the keyword will appear.
- Given the search feature was implemented successfully, when the user enters a keyword with no matching clubs, an appropriate message will be displayed.
- Given the search feature was implemented successfully, when the user enters an empty keyword, clubs will be displayed as if there are no filters applied.

User Story #32

As a user, I would like to be able to edit a club's information if I have necessary permission.

#	Description	Estimated Time	Owner
1	Create a conditionally rendered UI on the club page to edit a club if I have permission	2 Hrs	Bryan
2	Create a page with input forms where a user can edit the relevant club information if they have permission	2 Hrs	Bryan
3	Connect the input information fields from the user to that club object in firebase	2 Hrs	Yu-Lun
4	Create a submit button that will update the object in firebase and generate message	1 Hrs	Yu-Lun
5	Test UI components rendering correctly given permission status	1 Hr	Bryan
6	Implement input validation on the input form page	1 Hr	Bryan
User Story #32 Total Hours		10 Hours	

Acceptance Criteria:

- Given the user has appropriate permission to edit a club, when that user visits a club info page, the page will conditionally render an edit-club information button.
- Given the UI for edit club info page was implemented correctly, when a user with appropriate permission clicks the edit-club page, they should be able to see input forms for club details they'd like to change.
- Given that input validation works, when a user tries to submit invalid input, that user should see an error message.
- Given that Firebase integration works, when a user submits a club's info, the changes should be reflected in Firebase.
- Given that the submission was successful, when a user navigates to the club info page after submission, the club info page should now be updated with new information.

User Story #28

As a user, I would like to be able to manage user permissions of the club if I am a creator.

#	Description	Estimated Time	Owner
1	Develop UI for edit user permissions page	2 Hrs	Yu-Lun
2	Communicate with Firebase Database to update the users permissions list	4 Hrs	Yu-Lun
3	Test to ensure the UI responds to changes in the Firestore Database	4 Hrs	Yu-Lun
4	Test input validation (users are existed)	4 Hrs	Yu-Lun
4	Test to ensure the UI redirects to the appropriate page	1 Hr	Yu-Lun
User Story #28 Total Hours		15 Hours	

Acceptance Criteria:

- Given that a UI for the club page is built correctly, when a user navigates to the club page, there should be a component that allows a user to edit a club.
- Given that a UI for the edit club page is built correctly, when a user attempts to edit a club, they should be prompted with a form that allows them to update the field of their club editors.
- Given that the communication with Firestore Database is configured correctly, when a user updates an existing club editors list, the relevant field should be changed in the database.
- Given that a basic UI for the edit club page is built correctly, when a user finishes editing a club, the user should be notified that the club editors list was successfully (or unsuccessfully) edited.
- Given that a UI for the create club page is built correctly, when a user inputs invalid information (such as a non-existing user), they should be informed that bad data was supplied.

User Story #15

As a user, I would like to be able to add events for my Purdue club.

#	Description	Estimated Time	Owner
1	Add an responsive input form on the create-event page for associated clubs with database querying	4 Hrs	Bryan
2	Cross reference userID to list of admins for a club to verify permission	2 Hrs	Bryan
3	Connect the input information fields from the user to an event object in firebase	2 Hrs	Bryan
4	Create a UI showing that associated club on the event info page	2 Hrs	Bryan
5	Test UI components rendering correctly	1 Hr	Bryan
6	Implement input validation on the input form page	1 Hr	Bryan
User Story #15 Total Hours		12 Hours	

Acceptance Criteria:

- Given that the create-event link is generated correctly, when the user clicks the create-event link and is routed to the input page, there should be an input form where they can select which club will host this event.
- Given that the user has valid permission to edit a club, when the user clicks that input form, they should only be able to see and select a club that they have edit permission for.
- Given that the input validation is working correctly, when the user has entered invalid input into the create-event fields, they should see an error message upon submission attempt.
- Given that the user has valid permission to edit a club, when the user has submitted successfully, the event object should now be visible in Firebase and linked to the club.
- Given that submission was successful, when the user navigates back to their club page, the newly-added event should be visible on the club page.

User Story #25

As a user, I would like to edit club events if I have permission.

#	Description	Estimated Time	Owner
1	Create a conditionally rendered UI on the club page to create a club event	4 Hrs	Bryan
2	Create a page with input forms where a user can input relevant event information	2 Hrs	Bryan
3	Connect the input information fields from the user to an event object in firebase	2 Hrs	Bryan
4	Create a UI showing a list of events relating to that club on the club page	2 Hrs	Bryan
5	Test UI components rendering correctly	1 Hr	Bryan
6	Implement input validation on the input form page	1 Hr	Bryan
User Story #25 Total Hours		12 Hours	

Acceptance Criteria:

- Given that the user has valid permission to edit a club event, when the user navigates to “My Events”, the club events that they can modify should appear and when the event is clicked, there should be an edit button present.
- Given that the user has valid permission to edit a club event, when the user clicks the edit event link, they should be taken to a page with input forms to edit the event for that club.
- Given that the user has entered invalid input into the edit-event fields, when the user inputs invalid input, they should see an error message upon submission attempt.
- Given that the submit button is rendered correctly, when the user hits submit, the event object should now be updated in Firebase and still linked to the club.
- Given that the user has submitted successfully, when the user visits a club page, the new event changes should be visible on the club page.

User Story #18

As a user, I would like to be able to delete an event I created.

#	Description	Estimated Time	Owner
1	Create UI link to “My Events” under events	1 Hrs	Bryan
2	Create a “My Events” page with all user’s events	2 Hrs	Bryan
3	Create delete button and dialogue box on each event row	1 Hrs	Bryan
4	Delete the event from the firebase and remove it from any club pages	2 Hrs	Bryan
5	Test that event was wiped properly and that UI reflects these changes on the site wherever that event would be	1 Hr	Bryan
User Story #18 Total Hours		7 Hours	

Acceptance Criteria:

- Given that the UI for deleting an event a user created is implemented correctly, when a user clicks on “My Events”, they should see a list of events that they have permissions for.
- Given that the UI for deleting a comment is implemented correctly, when a user clicks the delete event button, the user will be prompted with a confirmation prompt.
- Given that the delete confirmation button is rendered properly, when the user navigates to firebase, the event object should be deleted from Firebase.
- Given that the delete confirmation button works properly and the event was linked to a club, when the user checks the club object’s fields, the deleted event object’s ID should no longer be linked to anything else in Firebase.
- Given that the user deleted their event and the Firebase deletion was successful, when the user navigates to portions of the site where that event should be displayed, the deleted event should no longer be visible anywhere on the site.

User Story #60

As a user, I would like to sort posts by popularity or recency.

#	Description	Estimated Time	Owner
1	Create and style UI component for sorting	2hrs	Raziq
2	Integrate the sorting component into the class forum page seamlessly	1.5hrs	Raziq
3	Create a redux action to fetch posts based on sorting criteria and integrate it to the class forum page	3hrs	Raziq
4	Replace current implementation of fetching posts (which pulls all posts to the client-side) with the new fetch action.	1hr	Raziq
5	Test that recency sorting works	1hr	Raziq
6	Test that popularity sorting works	1hr	Raziq
7	Update current class forum page UI so that the posts show relevant information for sorting seamlessly and so that page interactions like sorting will appear intuitive.	5hrs	Raziq
8	Integrate “number of comments” field into db schema and past create post user stories.	30 minutes	Raziq
User Story #60 Total Hours		15 Hours	

Acceptance Criteria:

- Given the class page UI was updated appropriately, when the user visits the page, the page should follow a similar layout to a social media site like Reddit.
- Given the sorting UI was implemented successfully, when the user visits the class forum page, it should be clear which sorting option is currently being applied and the user should only be able to select one sorting option at a time.
- Given the post UI was updated appropriately, when the user reads a post, they will be able to see relevant information like the number of comments and timestamp.
- Given the popularity sort was implemented properly, when the user selects the option, posts will be ordered by the number of comments and then by timestamp in descending order.
- Given the recency sort was implemented, when the user selects the option, posts will be ordered by timestamp in descending order.

User Story #65

As a user, I would like to be able to view a class forum post and all of its comments in a dedicated thread page.

#	Description	Estimated Time	Owner
1	Design and implement UI for main post component	5hrs	Raziq
2	Design and implement UI for comments component	4hrs	Raziq
3	Design and implement UI for nested comments component	1hr	Raziq
4	Create a redux action to fetch post object from db and integrate the action to the class forum page	3hrs	Raziq
5	Integrate fetched data with the UI	1hr	Raziq
6	Design the page layout (potentially as a modal window) and integrate all post/comment components into it.	3hrs	Raziq
7	Test that the page shows post/comment data correctly	1hrs	Raziq
User Story #65 Total Hours		18 Hours	

Acceptance Criteria:

- Given the thread page UI was implemented successfully, when the user navigates to the page, they will be able to read the full content of a post and sub-threads that spawned from it.
- Given the thread UI was implemented successfully, when the user reads a thread or a sub-thread, there should be an indicator that can help the user identify the replies and sub-threads that belong to it.
- Given the thread UI was implemented successfully, when a user deletes a parent comment, the UI will still show all of the parent comment’s sub-threads and indicate that the parent comment was deleted.
- Given the thread UI was implemented successfully, when the account of a poster or commenter was deleted, their username will no longer be associated with their past posts and comments.
- Given the page UI was implemented successfully, when the user tries to navigate to the page by passing an invalid class ID or post ID into the URL, an appropriate message will be displayed.

User Story #54

As a user, I would like to edit my post.

#	Description	Estimated Time	Owner
1	Develop UI that supports editing of posts	5 Hrs	Aaron
2	Communicate with Firestore Database in order to edit posts	3 Hrs	Aaron
3	Test for responsive UI rendering	1 Hr	Aaron
4	Test input validation	1 Hr	Aaron
5	Test Firebase Database Implementation	1 Hr	Aaron
User Story #54 Total Hours		11 Hours	

Acceptance Criteria:

- Given that the UI for editing posts is implemented correctly, when a user navigates to the post-editing page, the fields should be pre-populated with the post's current information to prevent accidental data manipulation.
- Given that the UI for editing a post is implemented correctly, when a user edits a post, the type of acceptable data should be consistent with the type of acceptable data in the created post (equivalent input validation).
- Given that the cross-reference between the post owner and Firebase Authentication is done correctly, when a user who is not the owner looks at a post, there will be no edit button on the post.
- Given that the cross-reference between the post owner and Firebase Authentication is done correctly, when a user who is the owner of the post attempts to edit that post, they should be allowed to edit that post.
- Given that both communication with Firestore Database and responsive rendering are implemented correctly, when an authorized user updates an existing post, the relevant fields should be changed in the database and reflected in the UI.

User Story #55
As a user, I would like to delete my post.

#	Description	Estimated Time	Owner
1	Develop UI that supports deletion of posts	3 Hrs	Aaron
2	Communicate with Firestore Database in order to delete the post	3 Hrs	Aaron
3	Test for responsive UI rendering	1 Hr	Aaron
4	Test input validation	1 Hr	Aaron
5	Test Firebase Database Implementation	1 Hr	Aaron
User Story #55 Total Hours		9 Hours	

Acceptance Criteria:

- Given that the UI for deleting a post is injected correctly, when the owner of a post navigates to their specific post page, there should be a component that allows deletion of that specific post.
- Given that the UI for deleting a post is injected correctly, when a user attempts to delete a post, the user should be met with a confirmation message before final deletion.
- Given that the cross-reference between the post owner and Firebase Authentication is done correctly, when a user who is not the owner of the post attempts to delete that post, they should not see a button to delete the post.
- Given that the cross-reference between the post owner and Firebase Authentication is done correctly, when a user who is the owner of the post attempts to delete that post, they should be allowed to delete that post.
- Given that both communication with Firestore Database and responsive rendering are implemented correctly, when an authorized user deletes an existing post, the relevant item should be deleted in the database and removed from the UI.

User Story #56

As a user, I would like to comment on posts.

#	Description	Estimated Time	Owner
1	Develop UI that supports commenting on posts	4 Hrs	Aaron
2	Communicate with Firestore Database in order to comment on posts	3 Hrs	Aaron
3	Test for responsive UI rendering	1 Hr	Aaron
4	Test input validation	1 Hr	Aaron
5	Test Firebase Database Implementation	1 Hr	Aaron
User Story #56 Total Hours		10 Hours	

Acceptance Criteria:

- Given that a UI for a class forum page is built correctly, when a user navigates to a specific class page, there should be a button to create a new comment on a post.
- Given that a UI for the create comment page is built correctly, when a user clicks on the button to make a new comment, there should be a text box to enter information for their new comment.
- Given that a Firestore Database functions to add comments are implemented correctly, when a user adds a new comment, the newly-created comment object should now be visible in the database.
- Given that the comment creation feature was implemented correctly, when the user created a new comment, the user should be notified that their comment creation was successful.
- Given that input validation works as intended, when the user enters invalid input for a comment, the user should be notified that their respective field contained inadequate information.

User Story #57
As a user, I would like to comment on comments.

#	Description	Estimated Time	Owner
1	Develop UI that supports commenting on comments	4 Hrs	Aaron
2	Communicate with Firestore Database in order to comment on comments	3 Hrs	Aaron
3	Test for responsive UI rendering	1 Hr	Aaron
4	Test input validation	1 Hr	Aaron
5	Test Firebase Database Implementation	1 Hr	Aaron
User Story #57 Total Hours		10 Hours	

Acceptance Criteria:

- Given that a UI for a class forum page is built correctly, when a user navigates to a post on the class forum page, there should be a button to create a new comment on a comment.
- Given that a UI for the create comment page is built correctly, when a user clicks on the button to make a new comment, they should be met with a form that allows them to fill out their comment information.
- Given that a Firestore Database functions to add comments are implemented correctly, when a user adds a new comment, the newly-created comment object should now be visible in the database.
- Given that the comment creation feature was implemented correctly, when the user created a new comment, the user should be notified that their comment creation was successful.
- Given that input validation works as intended, when the user enters invalid input for a comment, the user should be notified that their comment creation was unsuccessful.

User Story #58
As a user, I would like to edit my comment.

#	Description	Estimated Time	Owner
1	Develop UI that supports editing of comments	4	Michael
2	Communicate with Firestore Database in order to edit a comment's data	2	Michael
3	Test input validation	1	Michael
4	Test for responsive UI rendering when editing a comment	1	Michael
User Story #58 Total Hours		8 Hours	

Acceptance Criteria:

- Given that the UI for editing a comment is implemented correctly, when a user is viewing a comment, they should be presented with an option to edit the comment.
- Given that the UI for editing a comment is implemented correctly, when a user clicks the edit comment button, they can only edit the comment if they are authenticated with the profile that created the comment.
- Given that the UI for editing a comment is implemented correctly, when a user clicks the edit comment button, they are presented with a field which is populated with the current comment's information which is editable.
- Given that the UI for editing a comment is implemented correctly, when a user editing a comment, the comment cannot be empty.

User Story #59

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

#	Description	Estimated Time	Owner
1	Develop UI that supports deletion of comments	4	Michael
2	Communicate with Firestore Database in order to delete a comment's data	2	Michael
4	Test for appropriate UI rendering after deleting a comment	1	Michael
User Story #59 Total Hours		7 Hours	

Acceptance Criteria:

- Given that the UI for deleting a comment is implemented correctly, when a user is viewing a comment, they should be presented with an option to delete the comment.
- Given that the UI for deleting a comment is implemented correctly, when a user clicks the delete comment button, they can only delete the comment if they are authenticated with the profile that created the comment.
- Given that the UI for deleting a comment is implemented correctly, when a user clicks the delete comment button, the comment is deleted from the Firestore database.
- Given that the UI for deleting a comment is implemented correctly, when a user is done deleting a comment, it is no longer associated with its relevant post.

Current Total: Hours

Name	Estimated Time
Aaron Cox	40 Hrs
Yu-Lun Gao	40 Hrs
Michael Maganini	40 Hrs
Raziq Ramli	41 Hrs
Matthew Story	40 Hrs
Bryan Wang	44 Hrs

Remaining Backlog

Profile

1. ~~As a user, I would like to create a new account with my own email address.~~
2. ~~As a user, I would like to edit my profile.~~
3. ~~As a user, I would like to delete my account.~~
4. ~~As a user, I would like to have a unique username.~~
5. ~~As a user, I would like to have a profile with a brief bio and image.~~
6. ~~As a user, I would like to be able to change my password for my account.~~
7. ~~As a user, I would like to be able to log in.~~
8. ~~As a user, I would like my password to be obscured when I type it to log in.~~
9. ~~As a user, I would like to be able to log out.~~
10. ~~As a user, I would like to reset my password if I forget it.~~
11. ~~As a user, I would like to check other user profiles.~~
12. As a user, I would like to choose between light and dark mode (if time allows).
13. ~~As a user, I would like to hide my profile from users (if time allows).~~
14. As a user, I would like to choose specific features from the site as part of my homepage (if time allows).

Events Board

15. ~~As a user, I would like to be able to add events to my Purdue club~~
16. ~~As a user, I would like to be able to create public personal events.~~
17. ~~As a user, I would like to be able to edit an event I created.~~
18. ~~As a user, I would like to be able to delete an event I created.~~
19. ~~As a user, I would like to be able to view a list of Purdue events.~~
20. As a user, I would like to be able to view public personal events (if time allows).
21. ~~As a user, I would like each event to have an information page.~~
22. ~~As a user, I would like to be able to filter some events based on some criterias.~~
23. As a user, I would like to be able to RSVP to club events (if time allows).
24. As a user, I would like to see past events be hidden from my feed (if time allows).
25. ~~As a user, I would like to edit club events if I have permission.~~

26. As a user, I would like to see the events posted on the boiler links website to be listed on this application as well (if time allows).

Club Pages

27. ~~As a user, I would like to be able to create a new club.~~
28. ~~As a user, I would like to be able to manage user permissions of the club if I am a creator.~~
29. ~~As a user, I would like to be able to view a list of clubs available at Purdue.~~
30. ~~As a user, I would like to be able to search the list of Purdue clubs.~~
31. ~~As a user, I would like to be able to navigate to a club info page.~~
32. ~~As a user, I would like to be able to edit a club's information if I have necessary permission.~~
33. As a user, I would like to have access to delete the events that automatically added into the events board.
34. As a user, I would like to see a list of events that a club is holding on their page.
35. As a user, I would like to see the clubs created in the boiler links official website to be listed on this application as well (if time allows).

Living

36. As a user, I would like to be able to view a labeled map with bus stops and routes that run through the Purdue campus.
37. As a user, I would like to be able to view a bus route's next stop.
38. As a user, I would like to be able to view a list of all Purdue dining halls.
39. As a user, I would like to be able to view daily menus for each dining hall.
40. As a user, I would like to be able to check how crowded each dining hall is.
41. As a user, I would like to be able to check how crowded different sections of the Purdue CoRec are.
42. As a user, I would like to be able to see which washing machines are currently in use in each dormitory.
43. As a user, I would like to be able to see which dryers are currently in use in each dormitory.

Marketplace

44. ~~As a user, I would like to be able to create sell listings on the marketplace.~~
45. ~~As a user, I would like to be able to see a list of sell listings on the marketplace.~~
46. ~~As a user, I would like to edit my sell listing on the marketplace.~~
47. ~~As a user, I would like to delete my sell listing on the market place.~~
48. ~~As a user, I would like to be able to set a sell price and item description for my sell listings.~~
49. ~~As a user, I would like to be able to list my contact information to potential buyers.~~
50. ~~As a user, I would like to be able to include an image of the item I am selling in my sell listings.~~
51. As a user, I would like to be able to message other students on the marketplace about their listings (if time allows).

Class Forum

52. ~~As a user, I would like to be able to view class forums.~~
53. ~~As a user, I would like to post on class forums.~~
54. ~~As a user, I would like to edit my post.~~
55. ~~As a user, I would like to delete my post.~~
56. ~~As a user, I would like to comment on posts.~~
57. ~~As a user, I would like to comment on comments.~~
58. ~~As a user, I would like to edit my comment.~~
59. ~~As a user, I would like to delete my comment.~~
60. ~~As a user, I would like to filter posts by popularity or recency (if time allows).~~
61. As a user, I would like to join class forums (if time allows).
62. As a user, I would like to leave class forums (if time allows).

Other/Newly Added

63. As a user, I would like to install this web application on my desktop and mobile devices as a progressive web app (if time allows).
64. ~~As a user, I would like to be able to navigate to various pages on this website from a landing page.~~
65. ~~As a user, I would like to be able to view a class forum post and all of its comments in a dedicated thread page.~~
66. As a user, I would like to see a tutorial on all the features that the site has.