ClapBack

Sprint 2 Planning Document

Team 1: Aidan Chen, Dawon Jeong, Ethan Buck, Ilhoon Lee, Luke Lawson, Sooha Park

Sprint Overview

For Sprint 2, the goal is to build upon the foundational social media app we built in Sprint 1 so we can just focus on implementing our friend selection (CB) algorithm for Sprint 3. These add-ons we plan to implement include more flexibility for the user's profile, more features for the messaging and notification system, and lastly more nuance in ways the user can interact with users. The hope is as we move into the final sprint we have a strong social media app that we can later add our algorithm onto.

Scrum Master: Luke Lawson

Meeting Plan: Sunday at 9:00 pm, Wednesdays at 1:30-2:20 pm as needed

Risks and Challenges:

One challenge of this sprint is to refactor the structure of how we store user messages in our database. Doing this may cause us to have to restructure our frontend code, or maybe even class structure. Another challenge is debugging past problems from Sprint 1 while working on Sprint 2's stories. Lastly, the biggest challenge is being able to polish the features of ClapBack enough so for the final Sprint all we need to focus on is the CB algorithm and light additional features. We need to be intentional during this sprint on finalizing our workflow and codebase.

Current Sprint Detail

User Story #1

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

#	Description	Estimated Time	Owner
1	Create dedicated button for account deletion	1 Hrs	Sooha
2	Create a confirmation popup	2 Hrs	Sooha
3	Create backend functionality for account deletion	3 Hrs	Luke
4	Test Acceptance Criteria	1 Hr	Luke

Acceptance Criteria:

- Given that the UI button is correctly implemented, when the user presses the account deletion button a popup should appear asking if they are sure they want to delete their account. The popup will have “yes” or “no” options.
- Given that the user presses the “no” option on the confirmation popup, the user will exit the popup and no changes will be made.
- Given that the user presses the “yes” option on the confirmation popup, the user will be logged out of their account and the account will be removed from the database.
- If the user tries to log back into their account, an error will be thrown due to the account not existing anymore.
- If the user tries to sign up with the same credentials as their deleted account, no error will be thrown due to the account not existing anymore.

User Story #2

As a user, I would like to be able to recover my account if I forget my password.

#	Description	Estimated Time	Owner
1	Create UI button on login screen to recover account if password was forgotten	1 Hrs	Ethan
2	Create UI for the user to enter their email for their password to be sent to.	2 Hrs	Ethan
3	Implement backend for the user to have a password reset emailed to them	4 Hrs	Ilhoon

4	Test Acceptance Criteria	1 Hr	Ilhoon

Acceptance Criteria:

- If the user selects the “forgot password?” button on the login screen, then the user will be presented with a popup to enter their email.
- If the user enters their email into the popup prompt, a password reset option will be sent to that email.
- If the user enters a nonexistent email into the popup prompt, then the user will be presented with an error message.

User Story #3

As a user, I would like to be able to link other platforms where I can be contacted if I wish to continue the conversation with my CB once our conversation window is closed

#	Description	Estimated Time	Owner
1	Create UI to enter/edit hyperlinks of external social media profiles	3 Hrs	Aidan
2	Create backend implementation for storing hyperlink data in database	2 Hrs	Aidan
3	Create UI so the hyperlink data is displayed on user profiles	2 Hrs	Dawon
4	Test Acceptance Criteria	1 Hr	Aidan

Acceptance Criteria:

- Given that the UI is correctly implemented, then the user should have the ability to view and select the hyperlinks to their and other external social media profiles.
- Given that the user selects the edit button, then the user should be able to edit what hyperlink they have entered into their profile.
- Given that the user selects a hyperlink and is directed outside of the app, then the user should be able to re-enter ClapBack without having to log back in (if ClapBack wasn’t closed during redirection).

User Story #4

As a user, I would like to be able to view another user’s profile (username, nickname, bio)

#	Description	Estimated Time	Owner
1	Create UI page for the user being viewed	2 Hrs	Sooha

2	Create backend implementation so user’s page is populated with their username, nickname, and bio	2 Hrs	Luke
3	Test Acceptance Criteria	1 Hr	Dawon

Acceptance Criteria:

- Given that the UI is correctly implemented, then the user should be able to view another user’s profile including their name, nickname, and description.
- Given that the user being viewed edits their profile, then the profile page should sync with the changes made by that user.
- Given that the viewed user entered their external platform links, the viewing user should be able to click said link and be sent to that page.

User Story #5

As a user, I would like to be able to receive notifications when I receive a friend request

#	Description	Estimated Time	Owner
1	Set up listeners for sent and accepted friend requests	4 Hrs	Ethan
2	Show notifications on the events	2 Hrs	Ethan
3	Make sure that notifications follow the set settings of the user	2 Hrs	Ethan
4	Test Acceptance Criteria	1 Hr	Ethan

Acceptance Criteria:

- Given that the user sends a friend request to another user, then the other user should receive a notification.
- Given that the user accepts a friend request, then the user sending the request should receive a notification.
- Given the user has notifications off, then the app will not send a notification in either of the above cases.

User Story #6

As a user, I would like to be able to remove friends

#	Description	Estimated Time	Owner
1	Design UI for friend removal, including extra confirmation	3 Hrs	Dawon

2	Hide or remove chatrooms between the users	3 Hrs	Aidan
3	Add removal of users from respective friend lists in Firebase database	2 Hrs	Luke
4	Test Acceptance Criteria	1 Hr	Luke

Acceptance Criteria:

- Given that the user clicks the button to remove a friend, they will be given an extra prompt to confirm that they want to remove this user from their friends' list.
- Given that the user selects “no” on the extra popup, then the friend removal will be canceled and nothing will take place.
- Given that the user selects “yes” on the extra popup, then the friend removal will continue as normal.
- Given that the user has removed a friend, then the chatroom between the two will not appear on either of their chatroom lists.

User Story #7

As a user, I would like to be able to report other accounts

#	Description	Estimated Time	Owner
1	Implement UI and backend for the report button in the user interface	4 Hrs	Sooha
2	Display a message box for the user to write the reason for the report	3 Hrs	Dawon
3	Create a send and cancel button	2 Hrs	Dawon
4	Design a confirmation message for the user after submitting a report	2 Hrs	Dawon
5	Test Acceptance Criteria	1 Hr	Sooha

Acceptance Criteria:

- Given that the report button is implemented correctly, when the user clicks the report button, the user should be able to see a message box to write about the report.
- Given that send and cancel buttons are implemented correctly, when the user clicks the send button, it will send the report message to the admin (one account where we collect reports from the users). When the user clicks the cancel button, the user will be able to go back to the original page where the report button is located.

- Given that the user sends the report successfully, after the user clicks the send button, the user will see a confirmation message that the message has been sent successfully.

User Story #8
As a user, I would like to block certain users from sending me friend requests

#	Description	Estimated Time	Owner
1	Implement the "Block User" button on friend request UI and backend	5 Hrs	Dawon
2	Create a list that stores blocked users	2 Hrs	Aidan
3	Design a confirmation message for user after blocking a user	2 Hrs	Luke
4	Test Acceptance Criteria	1 Hr	Luke

Acceptance Criteria:

- Given that the block function works correctly, after the user clicks the block button, the friend request will disappear and the blocked user will be successfully added to a block friend list.
- Given that the user blocks a user, when the user clicks the block button, the user will see a confirmation message that the user has been successfully blocked.
- Given that a user is added to the blocked list successfully, when the user gets a friend request from a blocked user, the user should not receive a friend request.

User Story #9
As a user, I would like to be able to reply to a specific message in a conversation

#	Description	Estimated Time	Owner
1	Add press functionality on messages	3 Hrs	Ilhoon
2	Develop a method that can access the specific message	3 Hrs	Dawon
3	Create a “Reply” button	2 Hrs	Dawon
4	Create a new message format that includes the message to reply and the user's reply	3 Hrs	Ilhoon

5	Test Acceptance Criteria	1 Hr	Aidan

Acceptance Criteria:

- Given that the press functionality works correctly, when the user presses a message, the user will be able to see a reply button to reply to that message.
- Given that the message selection functionality works properly, when the user selects a message to reply to, the text of the selected message should appear above the text bar during message composition.
- Given that a new message format for reply is implemented correctly when the user replies to a message, the user can see a message that has been replied pop up above a replied message.

User Story #10

As a user, when a friend reads my message, I should be given the time they read it under my message.

#	Description	Estimated Time	Owner
1	Implement saving the time when a user opens a chat in the database	2 Hrs	Ethan
2	Implement displaying the time read if the most recent message is the users	2 Hrs	Ethan
3	Test Acceptance Criteria	1 Hr	Ethan

Acceptance Criteria:

- Given that messaging is implemented as intended, when a friend views the user’s most recent message, the time is recorded.
- Given that a user’s friend has seen the most recent message sent by the user, then the time the friend viewed the message should appear below said message.
- Given that the user sends a new message into the chat, then the timestamp of the previous message should not be viewable anymore.

User Story #11

As a user, I would like to be able to react to messages

#	Description	Estimated Time	Owner
1	Create UI for a user selecting an image and giving it a tag	2 Hrs	Dawon
2	Implement different reactions	1 Hr	Dawon

3	Save reactions to the chatroom in the database	2 Hr	Luke
4	Test Acceptance Criteria	1 Hr	Dawon

Acceptance Criteria:

- Given that the user has been sent a message, when the user selects the message, the user will be given a small list of reactions they can add to the message.
- Given that the user can see the list of reactions, when the user selects one of the options, the message should be displayed with its reaction on both the user's and friend’s screen.
- Given that the user feels that the reaction is not the right one for the message when the user selects the message again, they will be able to remove the reaction or choose a new one.

User Story #12

As a user, I would like to be able to edit the appearance of the messaging interface

#	Description	Estimated Time	Owner
1	Create UI to change layout	1 Hr	Sooha
2	Implement the ability to change the XML file of a chat	2 Hrs	Sooha
3	Save a user’s layout preference in the database	2 Hrs	Aidan
4	Create premade layouts	2 Hrs	Sooha
5	Implement customization of layouts	3 Hrs	Sooha
6	Test Acceptance Criteria	1 Hr	Aidan

Acceptance Criteria:

- Given that the user has not set up a custom messaging interface, when the user opens a chat, they should be given a default messaging interface.
- Given that the user wants to change the style of their messaging interface, when they select the option to do so in settings, they should be given premade options for layouts.
- Given that the user does not want a premade layout, the user selects a “customize” option, they should be able to select a wallpaper from their phone and a color for the banner.
- Given that the user selects a preferred layout, the layout should persist between runs of the app.

User Story #13

As a user, I would like to be able to delete a message and have it display that it has been deleted

#	Description	Estimated Time	Owner
1	Add press functionality on messages	3 Hrs	Ilhoon
2	Develop a method that can access the specific message	3 Hrs	Luke
3	Create UI for message deletion	2 Hrs	Ethan
4	Change the text of that specific message	2 Hrs	Aidan
5	Reflect the changes on both the sender and recipient sides	2 Hrs	Ilhoon
6	Test Acceptance Criteria	1 Hr	Ilhoon

Acceptance Criteria:

- Given that the messages are correctly displayed in the chatroom, when the user presses the message to delete, the user will be given the option to delete the message.
- Given that the press functionality works on a message, when the user clicks delete, the text within the message will be replaced with a message indicating that it was deleted.
- Given that the text in the message changed, when the other user sees the message, the other user should also see the changes reflected on that deleted message.

User Story #14

As a user, I would like to be able to edit a message and have it display that it has been edited

#	Description	Estimated Time	Owner
1	Add press functionality on messages	3 Hrs	Luke
2	Implement for user to receive updated text	3 Hrs	Luke
3	Implement access for specific messages	2 Hrs	Ilhoon
4	Create UI for message editing	2 Hrs	Sooha
5	Update the text and save changes	2 Hrs	Ilhoon

6	Test Acceptance Criteria	1 Hr	Ilhoon

Acceptance Criteria:

- Given that the messages are correctly displayed in the chatroom, when the user presses the message to update, the user will be given the option to edit the message.
- Given that the press functionality works on a message, when the user clicks edit, the user will be given an editable text box to enter the updated text.
- Given that the text in the message changed, when the other user sees the message, the other user should also see the changes reflected in that updated message.
- Given that the text in the message changed, when the other user sees the message, the other user should see a text next to the message indicating that it was edited.

User Story #15

As a user, I would like to be able to see my friend composing a response in the chatroom

#	Description	Estimated Time	Owner
1	Create a method to listen to typing event	2 Hrs	Sooha
2	Design typing indicator within chat	2 Hrs	Sooha
3	Develop function to add typing indicator to chat when the other user is typing	3 Hrs	Ethan
4	Develop function to remove typing indicator from chat when the other user is done typing	3 Hrs	Ilhoon
5	Test Acceptance Criteria	1 Hr	Ilhoon

Acceptance Criteria:

- Given that the other user is typing a message when the current user is within the chat page, the user will be able to see the typing indicator.
- Given that the typing indicator shows up within the chat page when the other user is typing, the current user will be able to see the typing indicator in real time.
- Given that the typing indicator shows up within the chat page when the other user is done typing, the current user will be able to see the typing indicator disappear in real time.

User Story #16

As a user, I would like to be able to edit the nickname of my friends (visible to me)

#	Description	Estimated Time	Owner
1	Develop UI to display friend’s nickname	1 Hr	Ethan
2	Implement button to be able to edit friends nickname	<1 Hrs	Ethan
3	Implement edit friend functionality and connect to database	2 Hrs	Aidan
4	Test Acceptance Criteria	1 Hrs	Ilhoon

Acceptance Criteria:

- Given that the user can see a friend’s nickname, there should be a button for the user to click to allow the user to edit the nickname.
- Given that there is a button, the user should be able to click the button and receive the option to edit the nickname or cancel the edit.
- Given that the user inputs a new nickname for the friend, the friends name should be set to the new nickname where the old nickname used to be displayed.

User Story #17

As a user, I would like to be able to send videos to my friends

#	Description	Estimated Time	Owner
1	Create a button for video sending	1 Hr	Dawon
2	Connect button to user’s videos	2 Hrs	Aidan
3	Implement video being sent over in chat room	4 Hrs	Aidan
4	Test Acceptance Criteria	1 Hr	Aidan

Acceptance Criteria:

- Given that the chat room is entered, the user should have a button that enables them to choose a video
- Given that the photo button is pressed, the user should be given the option to choose an existing video on their device
- Given that a video is selected, the user should be able to send it in the chat room and is viewable for them and the other user
- Given the user is pressed the button, should be able to cancel by pressing back or pressing the cancel button

User Story #18

As a user, I would like to be able to turn off notifications for some or all features

#	Description	Estimated Time	Owner
1	Create notification switches in settings page	2 Hrs	Ilhoon
2	Connect switches to database/notification features	2 Hrs	Ilhoon
3	Implement off feature for notifications	<1 Hrs	Luke
4	Test Acceptance Criteria	1 Hr	Luke

Acceptance Criteria:

- Given that the user is in settings, the user should be able to click switches that correspond to features (e.g. friend requests or messages) to enable or disable notifications for that feature.
- Given that the user is in settings, the user should be able to click a master switch that enables or disables notifications for all features.
- Given that the user has disabled notifications in the device's settings, then the app's notifications settings will not override the device settings.
- Given that the user disables notifications, the app shouldn't send notifications to the user in any situation (received or accepted friend requests, new chats, etc.).

Remaining Backlog

Functional Requirements

1. User accounts

    ~~1.1. As a user, I would like to register for a ClapBack account with an email address, username, and password~~

    ~~1.2. As a user, I would like to delete my ClapBack account~~

    ~~1.3. As a user, I would like to be able to log in to my ClapBack account~~

    ~~1.4. As a user, I would like to be able to log out of my ClapBack account~~

    ~~1.5. As a user, I would like to be able to view my own account and have the option to edit it~~

    ~~1.6. As a user, I would like to be able to change my ClapBack account nickname, password, description, and email~~

    ~~1.7. As a user, I would like to be able to recover my account if I forget my password~~

    1.8. ~~As a user, I would like to be able to upload a profile picture of my choice to represent my account~~

    1.9. ~~As a user, I would like to be able to add a description about me on my profile and be able to show some of my favorite media in my profile description (games, songs, movies, TV shows, books, etc.)~~

    ~~1.10. As a user, I would like to be able to link other platforms where I can be contacted if I wish to continue the conversation with a person after the day is over~~

2. Friends list and requests

    ~~2.1. As a user, I would like to be able to search for users from my friend list~~

    ~~2.2. As a user, I would like an interface to scroll through my friend list~~

    2.3. As a user, I would like to be able to search for other users on the platform

    ~~2.4. As a user, I would like to be able to send friend requests~~

    ~~2.5. As a user, I would like to be able to view another user's profile (username, nickname, bio)~~

    ~~2.6. As a user, I would like to be able to receive notifications when I receive a friend request~~

    ~~2.7. As a user, I would like to be able to accept or deny a friend request~~

    ~~2.8. As a user, I would like to be able to remove friends~~

    ~~2.9. As a user, I would like to be able to report other accounts~~

    2.10. As a user, I would like to have mutual friends recommended to me

    ~~2.11. As a user, I would like to block certain users from sending me friend requests~~

3. Messaging

    ~~3.1. As a user, I would like to be able to send messages to a friend.~~

    3.2. As a user, I would like to be locked out of messaging other people who are not my randomly selected friend

    ~~3.3. As a user, I would like a comprehensive messaging interface to communicate with friends~~

    3.4. ~~As a user, I would like to be able to reply to a specific message in a conversation~~

    ~~3.5. As a user, I would like to be able to send read receipts~~

    ~~3.6. As a user, I would like to be able to react to messages~~

    ~~3.7. As a user, I would like to be able to edit the appearance of the messaging interface~~

3.8. ~~As a user, I would like to be able to delete a message and have it display that it has been deleted~~

3.9. ~~As a user, I would like to be able to edit a message and have it display that it has been edited~~

3.10. As a user, I would like to be able to see previous conversations with past chosen individuals

~~3.11. As a user, I would like to be able to see my friend composing a response in the chatroom~~

~~3.12. As a user, I would like to be able to edit the nickname of my friends (visible to me)~~

~~3.13. As a user, I would like to be able to send pictures to my friends~~

3.14. As a user, I would like to have a conversation prompt selected from a database to be sent to me and my friend

3.15. As a user, I would like the database of conversation prompts to be regularly updated

~~3.16. As a user, I would like to be able to send videos to my friends (If time allows)~~

4. Friend-choosing system (placeholder name)

4.1. As a user, I would like to have an algorithm choose a friend based on my activity and the time since my friend and I last connected

4.2. As a user, I would like my friend to be selected daily at the same time as all other users

5. Gamification system

~~5.1. As a user, I would like to be able to keep track of consecutive days I have used the app (i.e. streaks)~~

5.2. As a user, I would like to be rewarded through incentives for my consistency in-app (if time allows)

5.2.1. As a user, I would like to track the number of messages I sent in order to access additional features (see next)

5.2.2. As a user, I would like to customize the app's theme, e.g. dark mode interface

5.3. As a user, I would like to be rewarded with other options of interaction for continued conversation with my chosen individual (See stories next)

5.4. As a user, I would like to play in-chat "Would-You-Rather", where it's a simple prompt with two buttons that provides the asker information

5.5. As a user, I would like to play in-chat "20 Questions", in which one player has a secret word and the other player must guess the secret word in 20 yes or no questions

6. Miscellaneous

~~6.1. As a user, I would like a brief walkthrough of the app to understand its goals and incentives~~

~~6.2. As a user, I would like to be given the option to either start chatting immediately upon opening the app or to access other functions first~~

~~6.3. As a user, I would like to receive a notification when I receive a message~~

~~6.4. As a user, I would like to be able to turn off notifications for some or all features~~

6.5. As a user, I would like to be able to receive notifications on when my friend for the day is chosen

6.6. As a user, I would like to get a copy of my chat logs

6.7. As a user, I would like to be able to sync my contacts with ClapBack (If time allows)

6.8. As a user, I would like some sort of time indicator to inform me of my remaining time to contact my chosen friend

Non-Functional Requirements

1. Platform and Performance
   a. I want the application to work as an Android mobile application.
   b. I want the application to be responsive and handle several concurrent users.
2. Architecture
   a. I want the application to use Flask/Firebase as the backend service to facilitate the development process and increase performance.
   b. I want the application to communicate through Flask with the real-time database in Firebase to create the main chat service and authentication system.
   c. I want the front-end of the application to be written in Kotlin to target not only Android but other platforms such as iOS and Java desktop applications.
3. Usability
   a. I want the application to be intuitive for users and easily navigable so that a user can contact their friend.
   b. I want the application to have a short tutorial of the app so that users can quickly learn its features.
   c. I want the action of messaging a friend for the day to be as easy as possible.
   d. I want the application's GUI to be very simple and not cluttered.
4. Security
   a. I want the application to encrypt and hash user passwords in a database with a password salt.
   b. I want to store the number of messages sent between users to implement gamification, but not store the content of the messages.
   c. I want the application to recover my forgotten password if I correctly answer preset security questions.
   d. I want the application to allow me to edit my password.