



## Sprint 1 Planning Document

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

### Sprint Overview

During this sprint, we hope to create a basic foundation to build upon for our application. Since this is our first sprint, all classes and their attributes need to be established so we can have a robust program to expand upon. Our application follows a full-stack design, so we need to create our frontend through Android Studio, our backend with Flask, and lastly our server with Google's Firebase. After this foundation is made, which includes a basic GUI and a method of sending data through the three entities, then we will create a basic messaging service between two users. This base messaging system will help us once we implement a system that gets a user's CB (their one friend for the day).

**Scrum Master:** Luke Lawson

**Meeting Plan:** Sunday at 9:00 pm, Thursdays at 3:00 pm as needed

### Risks and Challenges:

Since this will be the very first sprint, making sure everyone is caught up on the technology we will be using, as well as delegating enough work to each person may be our biggest challenges. Since in this first sprint, we will be working on the foundations of our code, it is arguably the most important sprint as we must make sure we get a good base to work with.

## Current Sprint Detail

### User Story #1

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

#	Description	Estimated Time	Owner
1	Create UI to enter username, email, and password	4 Hrs	Dawon, Sooha
2	Set up server and database for storing user accounts	6 Hrs	Ilhoon, Aidan, Luke
3	When a user signs up, the backend processes the information and sends it to database	4 Hrs	Ilhoon
4	Make sure that a user cannot register with an existing or invalid account	4 Hrs	Ilhoon

#### Acceptance Criteria:

- Given that the UI is correctly implemented, when the user opens the app, they will have the option to create an account.
- Given the user enters their information, when they submit their information, then an account with their username, email, and password is created.
- Given an account already exists with the same username or email, when the user tries signing up with either, then they will be sent an error message and the account will not be created.

### User Story #2

As a user, I would like to be able to log into my ClapBack account

#	Description	Estimated Time	Owner
1	Create UI to log in with a username or email and password	4 Hrs	Ethan
2	Implement an established connection from client to server and sending user information to the database	5 Hrs	Luke
3	Receive server response for user authentication	4 Hrs	Luke
4	Send error message for incorrect credentials	1 Hr	Luke

5	Create landing page or error popup for logged in user	2 Hrs	Ethan
---	---	-------	-------

#### Acceptance Criteria:

- Given that the UI is correctly implemented, when the user opens the app, then they will have the option to log in.
- Given that the user is entering in their password, they have an option to mask their password
- Given that the user enters login credentials, when the data is sent to the backend server for validation, the server will return a corresponding message to the client of success or failure
- Given the user enters the correct credentials, When they click the sign in button, then they are taken to a main page within the app that shows that they have succeeded in signing in.
- Given the user entered incorrect credentials, when they click the sign in button, then they will be shown an error message and will have to try signing in again.

### User Story #3

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

#	Description	Estimated Time	Owner
1	Remove authentication for that client	4 Hrs	Aidan
2	Create a logout button	1 Hrs	Dawon
3	Return the client back to the sign in screen	2 Hrs	Dawon

#### Acceptance Criteria:

- Given that the UI is correctly implemented, when the user is signed in, then they will have an option to log out of their account.
- Given the user decides to log out, when the user taps a button, they will be returned to the sign in/sign up screen.
- Given the log in and log out systems are functional, when the user logs out, then they will be disconnected from the server and will not lose any data.

### User Story #4

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

#	Description	Estimated Time	Owner
1	Create UI to view already established user account	4 Hrs	Sootha

2	Create UI button to enable edit option	1 Hr	Sooha
3	Create a cancel edit button with functionality	1 Hr	Sooha

#### Acceptance Criteria:

- Given that the UI is correctly implemented, then the user should be able view their profile with their pre-inputted parameters (nickname, description, picture, etc.).
- Given that the UI is correctly implemented, then the user should have a button available to enable an “edit mode” for their profile.
- Given that the edit mode for the profile works, when the user is in edit mode, then the user will have the option to cancel editing.

### User Story #5

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

#	Description	Estimated Time	Owner
1	Create UI to change profile attributes (username, email, password, etc)	4 Hrs	Sooha
2	Create a button to save changes	1 Hr	Sooha
3	Create a way to send and save changes to database	4 Hrs	Ilhoon

#### Acceptance Criteria:

- Given that the user decides to edit their profile, then the user should have a specific UI to be able to change their profile attributes.
- Given that the user is satisfied with their profile edits, then they should be able to press a button and save the changes.
- Given the user decides to not save their changes, then they should be able to press a cancel button and exit the edit menu without having their changes saved

### User Story #6

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

#	Description	Estimated Time	Owner
1	Design UI for profile picture upload feature	4 Hrs	Ethan
2	Implement profile picture upload functionality	4 Hrs	Ethan

3	Send uploaded picture information to database and validate the picture	4 Hrs	Aidan
4	Integrate profile picture upload with user account database	4 Hrs	Aidan

#### Acceptance Criteria:

- Given that the UI is correctly implemented, when the user goes to the profile settings, the user should be able to see an option to upload a profile picture.
- Given that the user uploads a picture, when the user clicks the “save” button, the image should be properly processed and saved to the account.
- Given that the user uploads a picture for their profile, the user should be able to see the uploaded image displayed as the user's profile picture.
- Given that the user uploads an unsupported file format or exceeds the maximum file size, the user should receive an error message informing that the file size is incorrect or too large.
- Given that the user's uploaded picture is valid, the picture's information should be saved to the database.

### User Story #7

As a user, I would like a comprehensive messaging interface to communicate with friends (frontend story)

#	Description	Estimated Time	Owner
1	Design a chat screen	4 Hrs	Dawon
2	Design an interface to enter and send messages	3 Hrs	Dawon
3	Save and display messages that the user sends	3 Hrs	Dawon

#### Acceptance Criteria:

- Given that friend acceptance is implemented correctly, when a user clicks on a friend, they should be taken to a chat with them.
- Given that a user can open a chat with a friend, when they want to start messaging them, they should be able to type up a message and see it displayed in a text box.
- Given that a user is finished with a message, when they click send, then they should be able to see their message pop up on the chat screen.
- Given that a user has sent previous messages, when they scroll up, they should be able to see past messages sent.

### User Story #8

As a user, I would like to be able to send messages to a friend (backend story)

#	Description	Estimated Time	Owner
1	Have sent messages be sent to the server to be stored in the database	4 hours	Ilhoon
2	Have messages received by the user be retrieved from the database	4 hours	Luke

#### Acceptance Criteria:

- Given that a user is finished with a message, when the user clicks send, the message should be uploaded to the database and visible in the chat.
- Given a user is a recipient of a message, the client should retrieve the message from the database.
- Given that there is a conversation between two users, the previous messages should be saved and viewable in the chat.

### User Story #9

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

#	Description	Estimated Time	Owner
1	Implement communicating with the notification system of the phone	3 hours	Ethan
2	Implement sending a notification when a message is received and the app is not open	3 hours	Ethan
3	Implement grabbing a received message and displaying it on a notification	2 hours	Ethan
4	Design a screen where the user can indicate and toggle their notification settings	3 hours	Aidan

#### Acceptance Criteria:

- Given that the app is not open, when a user receives a message, they should be notified on their phone.
- Given that a user does not want received messages displayed, when a user receives a message, they should just be given a generic notification.
- Given that a user turned off notifications for the app, when they receive a message, they should not be notified.
- If the user toggles their notification settings, that setting should be saved in the database.

## User Story #10

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

#	Description	Estimated Time	Owner
1	Communicate with the device to select a picture	4 Hours	Sooha
2	Implement the ability to send and display image from within the chat	4 Hours	Sooha
3	Implement the ability to receive and display image from within the chat	4 Hours	Luke, Sooha
4	Convert picture to encoded PNG and send it to the server	2 Hours	Luke
5	Decode the encoded string and upload it to Firebase	2 Hours	Luke

Acceptance Criteria:

- Given that messaging is implemented as intended, when a user wants their friend to see a picture, the user should be able to select that picture from their phone.
- Given that a user can select the picture they want to send, when a user clicks send, the picture should be sent as a message.
- Given a friend has a picture they want to send to the user, when they click send, the user should be able to see any pictures sent to them.

## User Story #11

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.)

#	Description	Estimated Time	Owner
1	Design UI for profile description and media section	2 Hrs	Dawon
2	Implement profile description and media section functionality	1 Hr	Dawon
3	Integrate profile description and media section with database	2 Hr	Aidan

Acceptance Criteria:

- Given that the UI is correctly implemented when the user goes to profile settings, the user should be able to see an option to add a description and favorite media.
- Given that the user adds a description and media when the user clicks the “save” button, the information should be saved to the user’s account and visible to other accounts.
- Given that the user no longer wants a media item, when the user clicks the “remove” button, then the media item should be removed from its profile.

### User Story #12

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

#	Description	Estimated Time	Owner
1	Design UI for the walkthrough feature	4 Hrs	Ethan
2	Develop the walkthrough	1 Hrs	Ethan
3	Design the steps of the tutorial with appropriate graphics	2 Hrs	Ethan

Acceptance Criteria:

- Given that the walkthrough UI is designed correctly, after signing up the user should see a brief walkthrough that introduces to the app, its goals, and incentives.
- Given that the walkthrough feature is implemented correctly, when the user swipes left, right, or clicks the “next” button, the user should see the next screen with the relevant information.
- Given that the user finishes the walkthrough, the user should click the “finish” or “start” button to be directed to the app’s main screen.

### User Story #13

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

#	Description	Estimated Time	Owner
1	Develop a method that tracks the user’s usage	4 Hrs	Dawon, Aidan
2	Increment streaks value for the user	3 Hrs	Dawon, Aidan
3	Make streak value visible for the user	1 Hrs	Dawon



#### Acceptance Criteria:

- Given that the user created an account, when the user signs in to the app then the method will respond and increment the count value.
- Given that the server and database is intact, when the app sends the count value, then the server will find the streaks fields from the database and increment the value by count.
- Given that the interaction with the database was successful, when the server receives the updated streaks, then it will respond back to the app with a success code and updated streak value.
- Given that the streak functionality works successfully, when the user wants to check its streak, then the user can see the visualized streak value.

#### User Story #14

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

#	Description	Estimated Time	Owner
1	Display the selected CB on the main page with the CB's nickname and user profile	2 Hrs	Dawon
2	Create a button that allows the user to start chat with the CB	1 Hrs	Dawon
3	Display the time that is left before the algorithm selects a new CB	2 Hrs	Ethan, Ilhoon
4	Develop the swipe functionality that allows the user to swipe left and right to see a different screen	1 Hrs	Dawon

#### Acceptance Criteria:

- Given that the CB selecting algorithm and user authentication works, when the user opens the app, then the first screen will contain the CB's nickname, profile picture, time remaining, and a start chat button.
- Given that the server is properly set up, when the user clicks the start chat button, then the user will enter a chatroom.
- Given that both users have entered the chatroom, when the user types a text and clicks the send button, then the other user should be able to see that text in realtime.
- Given that the friend list page has been successfully implemented, when the user swipes left, then the user will see the friend list page.
- Given that the user profile page has been successfully implemented, when the user swipes right, then the user will see the user profile page.

### User Story #15

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

#	Description	Estimated Time	Owner
1	Create UI to display list of friends	3 Hrs	Sooha
2	Retrieve friend list from database	4 Hrs	Sooha, Ilhoon
3	Render friend list in order of CB chat generation	2 Hrs	Sooha

#### Acceptance Criteria:

- Given that the friendlist was properly retrieved from the database, then the user should be able to view and scroll through their friendlist using a UI.
- Given that the friend list page is completely functional, then the user should be able to sort their friendlist.
- Given that the user has no friends to display, then the user will receive a message saying that they need to add friends for them to be displayed.

### User Story #16

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

#	Description	Estimated Time	Owner
1	Develop and display a search bar on the top of the friend list page	1 Hrs	Ethan
2	Implement search action when user clicks search button or clicks enter	2 Hrs	Ethan
3	Query the database and return a list of users that match the search keyword and substrings	2 Hrs	Aidan
4	Render the list of users that match the search keyword on the friend list	4 Hrs	Ilhoon

#### Acceptance Criteria:

- Given that the friend list page renders all the user's friends, when the user swipes left from the main page, then the user will see the full list of friends and a search bar on top of the screen.
- Given that the friend list page is completely functional, when the user types a keyword in the search bar and clicks the search button, then the list will render the users that match the keyword.

- Given that the search results are correct, when the user types on the search bar, then the app will listen to any changes to the text the user types and render the users that match the keyword in real time.
- Given that the user selects a friend that they searched for, then the user should be able to view their profile.

### User Story #17

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

#	Description	Estimated Time	Owner
1	Create a button on the profile of other users that sends a friend request and cancels a friend requests	1 Hrs	Dawon
2	Send friend request to the server to be saved to the database per user	2 Hrs	Aidan
3	Find the correct user in database and save request	2 Hrs	Aidan
4	If request is canceled, remove request from the database and UI	2 Hrs	Aidan, Dawon

#### Acceptance Criteria:

- Given that the profile of other users are visible, the user should have a button to request a friend request to the user who's profile the button is clicked on.
- Given that the 'request' button is clicked, the event should add the requester to the friend request list of the requestee.
- Given that the 'request' button is clicked, the button should now indicate that a request has been sent
- Given that the 'request' button is clicked after indicated the request has been sent, the request should be canceled and the request should be removed from the friend request list

### User Story #18

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

#	Description	Estimated Time	Owner
1	Implement a notification inside the app to let user know they have a friend request	2 Hrs	Ilhoon
2	Create a UI for pending friend requests	2 Hrs	Ilhoon
3	Implement accept and deny buttons	<1 Hrs	Luke

4	Implement updates to the backend based on acceptance of or denial of friend request	6 Hrs	Luke
---	---	-------	------

Acceptance Criteria:

- Given that a friend request has been sent, then the user should see the request on a separate UI that lists friend requests the user has received.
- Given that a friend request has been sent and can be seen by the user, then the user should be given a prompt to either accept or deny the friend request via buttons to select.
- Given that the friend request is accepted, then the friend request should be removed and the two users should be added as 'friends' and added to their respective friendlists and the request button should now indicate they are friends.
- Given that the friend request is denied, the friend request should simply be removed and the request button should be reset.

# 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, description)
- 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 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 unsend a message and have it display that it has been unsent
- 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 c(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.