Team 12 - CS 307 Sprint 1 Retrospective

Project Name: Capsule

Nathan Schneider, Kevin Jones, Evan Zimmerman, Peter Henwood

What went well?

During this sprint we finished the vast majority of our user stories in exactly the way we intended before the sprint started. Starting from almost no React / mobile development experience, we set up the base skeleton of our app, which includes profile information, friends, social media linking, google login, all their respective pages, and more.

User Stories

User Story #1

(1) As a user, I would like to be able to log in/sign up via my Google account, so that the log in/sign up process goes faster.

#	Description	Estimated Time	Owner
1	Create a login screen in React Native and add a sign in with google button	2 Hrs	Kevin
2	Figure out API calls with Google	4 Hrs	Evan

	Authenticator to allow users to sign in with google		
3	Sync email from Google Authenticator with our database	3 Hrs	Nathan
4	Check if user exists and continue to registration or main page	4 Hrs	Kevin
5	Debug and complete unit test for signing in with Google Authenticator	3 Hrs	Nathan
6	Format UI on profile page to look like the UI mockup	2 Hrs	Kevin

Completed:

Successfully implemented a feature allowing users to quickly sign in or sign up using their Google account, directing them appropriately to either the main page or registration page based on their account status.

User Story #2

(2) As a user, I would like to be able to create a username during registration, so that I can identify myself on the platform.

#	Description	Estimated Time	Owner
1	Create a functional “create username” box in React Native	1 Hr	Evan
2	Figure out how to store the username in the database (linked with google account info)	4 Hrs	Evan
3	Implement create username into profile display and other areas of the app where it will be seen.	4 Hrs	Evan
4	Complete unit testing of application to verify that the username is functional for all implemented aspects of the app	2.5 Hrs	Evan

5	Format UI on profile page to look like the UI mockup	2 Hrs	Evan

Completed:

Developed a functionality during registration for users to create a unique username, enhancing their identity on the platform and leading them to link their Spotify account subsequently. The username is properly stored in the database and is easily accessible on other pages.

User Story #3

(4) As a user, I would like to be able to have the option to link/not link my spotify account during registration, so that music can be part of my capsule account.

#	Description	Estimated Time	Owner
1	Create a link Spotify screen in React Native with a button to link Spotify and to skip linking for now	2 Hrs	Kevin
2	Figure out API calls to spotify to allow the user to login with Spotify	4 Hrs	Kevin
3	Store all relevant information necessary to retrieve future Spotify information from the Spotify API in the database	4 Hrs	Kevin
4	Navigate to the link Instagram screen after linking Spotify or skipping for now	1 Hr	Kevin
5	Format UI on profile page to look like the UI mockup	2 Hrs	Kevin
6	Test linking, skipping, and navigation features	2 Hrs	Kevin

Completed:

Provided users with the option to link their Spotify account during registration, enriching their profile with music preferences, while also giving them the flexibility to skip this step. The database contains all the spotify information so that it can be accessed later when generating capsules.

User Story #4

(5) As a user, I would like to be able to have the option to link/not link my instagram account during registration, so that recent instagram photos can be part of my capsule account.

#	Description	Estimated Time	Owner
1	Create a link Instagram screen in React Native with a button to link Instagram and to skip linking for now	2 Hrs	Kevin
2	Figure out API calls to spotify to allow the user to login with Instagram	4 Hrs	Kevin
3	Store all relevant information necessary to retrieve future Instagram information from the Instagram API in the database	4 Hrs	Kevin
4	Navigate to the main page screen after linking Instagram or skipping for now	1 Hr	Kevin
5	Format UI on profile page to look like the UI mockup	2 Hrs	Kevin
6	Test linking, skipping, and navigation features	2 Hrs	Kevin

Completed:

Provided users with the option to link their Instagram account during registration, enriching their profile with photos from the month, while also giving them the flexibility to skip this step. The database contains all the instagram information so that it can be accessed later when generating capsules. Unfortunately due to a bug with facebook the account linking doesn't work at the moment, but we have all the code implemented so that as soon as facebook fixes it our code will work.

User Story #5

(7) As a user, I would like there to be a profile page unique to my account, so that I have a central hub for all account information

#	Description	Estimated Time	Owner
1	Add button, with good styling, to main page to access profile page	1 Hr	Nathan
2	Implement navigation to the new screen (research navigation and passing states)	3 Hrs	Nathan
3	Format UI on profile page to look like the UI mockup	4 Hrs	Nathan
4	Test application so that navigation and profile page acts as expected for each user	2 Hrs	Nathan

Completed:

Created unique profile pages for each user, serving as a central hub for all account-related information, including linked social media accounts (if they skipped it in signup) and personal details. The page can be accessed through clicking on your profile picture.

User Story #6

(10) As a user, I would like there to be a friends page with all information relevant to friends quickly accessible, so that I can complete any actions related to friends.

#	Description	Estimated Time	Owner
1	Add button on profile page to navigate to friend page	30 Min	Nathan
2	Implement navigation to friend screen	30 Min	Nathan
3	Format UI on friend page to look like the UI mockup	3 Hrs	Nathan
4	Show current friends and friend requests of the current user (Research	5 Hrs	Nathan

	MongoDB)		
5	Test application so that navigation and friend page acts as expected for each user	2 Hrs	Nathan

Completed:

Implemented a friends page displaying current friends and friend requests. All of the information is easily accessible and provided in scrollable lists so that users can choose what to do for each friend request / friend.

User Story #7

(11) As a user, I would like to be able to invite friends by username, so that we can see each other's capsules.

#	Description	Estimated Time	Owner
1	On the friend page, add a text input box and button to type in a friend username and then send the request	1 Hr	Nathan
2	When sending a friend request, add the friend request to the appropriate user object in MongoDB(research MongoDB)	6 Hrs	Nathan
3	Error checking for when the mongoDB request fails	2 Hrs	Nathan
4	Test application and ensure that data is properly stored when sending a friend request	2 Hrs	Nathan

Completed:

Implemented a search bar where you can send other users friend requests. All of the error checking is also implemented so that you can't send a friend request to a nonexistent user or someone who you already sent one to. Once you send the friend request, it is properly stored in the other user's friend request array.

User Story #8

(12) As a user, I would like to be able to view who has requested me as a friend in the friends page, so that I can visualize who wants to be my friend.

#	Description	Estimated Time	Owner
1	On the friend page, add a section for incoming friend requests	2 Hrs	Evan
2	In the friend requests section, implement a skeleton functionality for accepting/rejecting the request and make the styling look good.	4 Hrs	Evan
3	When the friend request array is updated in the MongoDB, update the user's UI to display the new request	4 Hrs	Evan
4	Test application to ensure that the client displays the information from the database properly	2 Hrs	Evan

Completed:

Users can see all of their incoming friend requests listed in a scrollable list at the top of the friends page. This friend request list is updated when they receive a new request. The styling is also implemented in a manner which is easy for the user to understand.

User Story #9

(13) As a user, I would like to be able to accept or reject a friend request in the friends page, so that I can manage who can view my capsule.

#	Description	Estimated Time	Owner
1	When accepting the friend request,	3 Hrs	Evan

	update the database for the user's current friend list		
2	When accepting the friend request, update the database for the friend's current friend list	3 Hrs	Evan
3	Error checking for when the mongoDB request fails	2 Hrs	Evan
4	Test application and ensure that data is properly stored when accepting a friend request	2 Hrs	Evan

Completed:

Implemented functionality for accepting or rejecting incoming friend requests on the friends page. When the user clicks accept/reject, it updates the screen, the database for the current user, and the database for the user who sent the request.

User Story #10

(14) As a user, I would like to be able to remove a friend that I previously accepted, so that I can choose to no longer see another person's capsule.

#	Description	Estimated Time	Owner
1	On the friend page, add a remove friend button	1 Hr	Peter
2	Remove friend in friend list on MongoDB via request to server	2 Hrs	Peter
3	Error checking for when the MongoDB request fails (User account no longer active, User unfriended you first)	4 Hrs	Peter
4	Update client to show new friends list	1 Hr	Peter
5	Test and Debug MongoDB correctly stores information, and client accurately displays information	1 Hr	Peter

Completed:

Implemented functionality for removing a current friend on the friend page. In the scrollable list of friends, you can click the red "X" button on the ride side to remove the friend. This removes them from your client display, from the current users database, and the friend's screen/database.

User Story #11

(15) As a user, I would like there to be a history page containing a scrollable list of past capsules, so that I can be reminded of past months.

#	Description	Estimated Time	Owner
1	Add button on profile page to navigate to history page	2 Hrs	Peter
2	Request MongoDB for past capsules	3 Hrs	Peter
3	Error checking for edge cases (No past capsules, user does not exist)	1 Hr	Peter
4	Display past Capsules	3 Hrs	Peter
5	Format UI on profile page to look like the UI mockup	3 Hrs	Peter
6	Debug and Testing	2 Hrs	Peter

Completed:

Added a history page showcasing a scrollable list of past capsules, allowing users to reminisce and reflect on previous moments shared on the platform. Since our capsule system hasn't been created yet, we are currently using placeholder capsules. However, the database system works perfectly so that once our app can create capsules, the history page will already be able to display them.

User Story #12

(20) As a user, I would like to have a Story Board page, so that I can see my friend's snapshots.

#	Description	Estimated Time	Owner
1	Use React Native to create the Story Board button on the main page	3 Hrs	Peter
2	Create Story Board page	4 Hrs	Peter
3	Create all server/database interactions to get friends snapshots	3 Hrs	Peter
4	Use React Native to display snapshots, and match UI formatting	2 Hrs	Peter
5	Testing and Debugging	2 Hrs	Peter

Completed:

Established a Story Board page where users can view snapshots shared by their friends, fostering a more interactive and engaging user experience. Similar to the history page, since we don't have capsules implemented we have to use placeholder capsules. These placeholders are stored in the database, so the Story Board functionality is complete and ready for when our app we actually are able to create capsules.

What did not go well?

For this sprint, we managed to complete nearly everything we set out to accomplish. One part we couldn't complete was instagram linking. This was due to a bug in Facebook not allowing developers to add new test users in their instagram api. Online discussion boards point to this bug being fixed soon and since we finished implementing all the code, as soon as the bug is fixed our code will work.

Additionally, there were a few acceptance criteria that we had to modify towards the end of the sprint because of minor tweaks we made to how we implemented certain features throughout the sprint. Some of the acceptance criteria for the history page / story board page involved capsules which is an item we have not implemented yet, so we had to create placeholder capsules to store in the database. The current code for these pages will work with actual capsules that are generated as soon as that is implemented.

In terms of complete user stories, there weren't any that we didn't manage to finish to during the sprint.

How should we improve?

I think we did a good job during this sprint since we didn't have any previous experience with any of the technology that we used. A lot of the time at the beginning was spent learning how to use and implement react native, node servers, and MongoDB.

However, one big thing we struggled with was time management. It was difficult figuring out how to manage the time working on this project with other courses / exams. Something we plan on doing next sprint is explicitly setting aside blocks of time during the week to all work on the project. This also led to some members of our group spending more time on the project than other members. Fixing this is going to be our main focus for the next sprint so that the work is more evenly distributed.

We also found it hard to follow the sprint planning document for who was going to do which user story / task. The main cause of this was because all of us were trying to figure out how to use unfamiliar technology. As a result of this, we all ended up working on every part of the app to try and learn it. Next sprint this should go much better since not only are we more familiar with the technology, but we have a better understanding of what each person should work on.

Finally, sometimes during our weekly meetings we also found ourselves getting distracted, which occasionally left us with not enough time to properly review everything while we were together. Our plan for the next sprint is to knock out everything we need to do during the stand up meeting right at the start so that we don't run out of time.