Serenova

Sprint 2 Retrospective Document

Team 24: Cristina Corley, Kara Orander, Ishwarya Samavedhi, Ava Schrandt, Mary Voorhees, Caitlin Wilson

User Stories

#	Task Description	Estimated Time	Owner
User Story #1 (From Sprint 1): As a user, I would like to be able to manage my account and take actions such as changing my password, deleting my account, and changing my username.			
1	Handling backend logic to delete a user from the database when the delete button is clicked in EditProfileView	4hr	Kara
2	Handle backend to delete user from friends in Firestore when delete is clicked	1hr	Kara
3	Add navigation back to the login page when a user deletes their account	1hr	Kara
4	Handle backend logic to make sure desired email is valid and phone number is valid	2hr	Kara
5	Add error checking to make sure the new password, username, etc. are all valid and username is not taken	2hr	Caitlin
6	Add unit tests to check functionality	3hr (each person assigned)	Caitlin
User Story #2: I would like to receive in-app rewards for meeting sleep goals.			
1	Create UI Page to allow users to view current rewards	3hr	Isha
2	Create reward system	2hr	Isha
3	Setting up database logic and validating the request	1hr	Mary
4	Add unit tests for sleep log reward.	2hr (each person assigned)	Isha, Mary

User Story #3: I would like to be able to receive rewards by using features of the app.			
1	Create UI Element for progress bar that automatically updates upon task completion	2hr	Isha
2	Implement a rewards system for using application features.	2hr	Mary
3	Provide users with rewards on completion of preset goals.	3hr	Mary
4	Add unit tests for using application features and achieving rewards.	3hr (each person assigned)	Isha
User Story #4: I would like to share my rewards with my friends.			
1	Create UI Feature to be able to display reward points with friends (on Profile Page?)	3hr	Isha
2	Handle backend for making rewards viewable to other users besides the owner.	3hr	Mary
3	Handle backend for removing visibility of rewards for other users besides the owner.	2hr	Mary
4	Add unit tests for viewing rewards by other users.	2hr (each person assigned)	Isha, Mary
User Story #5: I would like to be able to view what my sleep goals are and how much progress I have made to complete them.			
1	Create UI Element for progress rings that automatically updates upon task completion	3hr	Ava
2	Handle backend to retrieve sleep goals.	2hr	Mary

3	Connect sleep data from the API.	2hr	Ava
4	Add unit tests for comparing sleep goals to sleep data.	3hr (each person assigned)	Mary
User Story #6: I would like to be able to write and save a dream journal.			
1	Create UI Feature to be able to view a dream journal.	3hr	Ava
2	Handle backend for storing user-written content in a journal in Firebase.	2hr	Ava
3	Implement a feature for storing the date of the journal automatically to Firebase.	1hr	Ava
4	Handle backend for retrieving journal information from Firebase to display in the UI.	2hr	Ava
5	Add unit tests to check if data is stored and retrieved correctly.	2hr	Ava
User Story #7: I would like to be able to either delete or edit a journal entry.			
1	Create a UI Page for viewing/editing journal details	2hr	Ava
2	Handle backend for saving edits/updating journal entry in Firebase	1hr	Ava
3	Handle backend for deleting post	1hr	Ava
4	Add unit tests for editing and deleting journal entries	2hr	Ava
User Story #8 (From Sprint 1): As a user, I would like to be able to view posts for a forum so that I can receive sleep experiences, tips, advice, and engage in discussions on sleep-related topics.			
1	Add unit tests and manual testing to check if the correct posts are showing up.	2 hrs (each person assigned)	Cristina

User Story #9: I would like to be able to create a forum so that I can engage in discussions on topics that I am interested in.			
1	Create UI for viewing different online forums of particular topics.	3hr	Ava
2	Create UI for creating a new forum.	2hr	Ava
3	Handle backend for creating a new forum and displaying it on the UI.	3hr	Caitlin
4	Handle backend for setting the permissions of users on the created forum.	1hr	Caitlin
5	Add unit tests for creating a forum and setting user permissions.	3hr	Caitlin
User Story #10: I would like to be able to search the forum for specific posts so that I can receive sleep experiences, tips and advice about sleep-related topics on this forum.			
1	Create a UI page to allow users to search for posts based on a set of given criteria.	3hr	Cristina
2	Handle visually showing UI components for each individual post that matches the query	2hr	Cristina
3	Develop an algorithm to search Firebase for Posts based on a single keyword from the user. Implement prefix searching with the keyword.	3hr	Cristina
4	Handle navigation links from forum page to search page, from search page back to forum page, and from search page to individual posts (when clicked)	2hr	Cristina
5	Add unit tests and manual testing to check if the correct posts are showing up.	2 hr	Caitlin
User Story #11:			

I would like to be able to reply to posts on a forum so that I can further the discussion on a sleep-related topic.			
1	Create UI for replying to forum posts.	3hr	Cristina
2	Handle backend for storing replies to Firebase.	2hr	Cristina
3	Handle backend for retrieving replies from Firebase to display in the UI.	3hr	Cristina
4	Add unit tests for saving and displaying replies.	3hr	Isha
User Story #12: I would like to be able to upvote and downvote forum posts and replies.			
1	Add UI element to upvote/downvote posts and comments	3hr	Cristina
2	Add variables in Firebase to store upvotes and downvotes	1hr	Cristina
3	Store upvote and downvote in firebase every time a new vote occurs	2hr	Cristina
4	Retrieve upvotes and downvotes from firebase when a post appears	1hr	Cristina
5	Add unit tests and manual testing to ensure the likes are properly updated with Firebase and the UI.	3hr	Mary
User Story #13: I would like to be able to manage a forum so that I can make sure posts are restricted to the topic of the forum.			
1	Create UI Button to remove a post from a forum.	2hr	Kara
2	Make the UI Button visible only to certain users that have the correct permissions.	2hr	Kara

3	Handle backend to remove the post from Firebase.	3hr	Kara
4	Add unit tests for removing forum posts.	3hr	Isha
User Story #14: I would like to be able to send friend requests to selected users to build my own community of friends.			
1	Develop UI for sending friend requests.	3hr	Isha
2	Handle backend that sends a request to the selected user.	3hr	Mary
3	Implement a feature that notifies the initiating user of the decision.	2hr	Isha
4	Add unit tests for recording whether another user is added as a friend.	3hr (each person assigned)	Isha, Mary
User Story #15: I would like to be able to manage my friends requests so that I can control who I am friends with. This includes being able to accept and deny friend requests.			
1	Create a UI page that displays a user's friend requests, giving the options to accept or deny.	3hr	Kara
2	Create new collections and documents in Firestore for friends, friend requests, and own requests	2hr	Kara
3	Backend to add user to Firestore friends collection on SignUp	1hr	Kara
4	When the user clicks accept, add that friend to their list of friends in Firebase and the friend array in User class	2hr	Kara
5	Handle backend that sets the user to "friend" if accepted.	2hr	Kara
6	When the user clicks deny, remove that friend from the list of requests.	2hr	Kara

7	Add unit tests for friend permissions and request permissions.	3hr	Isha, Mary
User Story #16: I would like to be able to block certain users so they cannot see my posts or add me as a friend.			
1	Implement a feature that marks other users as “blocked.”	2hr	Caitlin
2	Handle backend so a blocked user cannot see the blocker’s content.	2hr	Caitlin
3	Handle backend so that a blocked user cannot friend the blocker.	1hr	Caitlin
4	Implement a feature that allows a user to unblock other users that are marked as “blocked.”	1hr	Caitlin
5	Add unit tests for blocking and unblocking users.	3hr	Caitlin
User Story #17: I want to be able to view articles for improving sleep quality.			
1	The Home page UI should have articles the user can scroll through	3hr	Isha
2	The user should be able to see the title of each article and an associated description	2hr	Caitlin
3	The firebase should be able to handle data with the links to different articles and display them to the user	3hr	Caitlin
4	When the user clicks on an article, it should redirect the user to the article’s webpage	1hr	Caitlin
User Story #18: As a user, I would like to be able to manually log my sleep times.			
1	Send User’s sleep data sample to	1 hr	Ava

	Firestore Database		
2	Create alert notification for sleep session	1 hr	Ava
User Story #19: Use sleep analysis data to calculate a “sleep score” based on my duration of sleep and any times I was awake during the night.			
1	Create UI element for viewing current sleep score	2hr	Ava
2	Create algorithm for calculating sleep score	1hr	Ava
3	Get current sleep score from User sleep data	1hr	Ava
4	Add unit tests for sleep scoring	3hr	Caitlin
User Story #20 (From Sprint 1): I would like to be able to set sleep duration goals.			
1	Handle backend to save user sleep goals.	30min	Mary
2	Handle backend logic that will retrieve user sleep goals.	30min	Mary
User Story #21 (From Sprint 1): As a user, I would like a page that shows me a graph detailing how much sleep I got each day over the past month.			
1	Handle backend logic that will query the database for the user’s sleep logs by the specified month.	1 hr	Cristina
2	Develop an algorithm to calculate the total amount of sleep the user got per day of the specified month. This is necessary if there can be multiple sleep logs per day.	2 hr	Cristina
3	Handle backend logic that will work with Swift Charts to plot the data for display.	1 hr	Cristina

What Went Well in Sprint 2?

One thing that went well in Sprint 2 was our organization for the presentation. After last Sprint's review, we reformatted our presentation to flow better than our previous one. This time, we spoke about features according to their acceptance criteria and presented generally in order of how the pages in the application were divided. We also explicitly mentioned which user story we were covering as we were going over each feature. By making these changes, this structure was a much easier way of covering all the user stories and demonstrating that our project met the criteria in a timely manner.

Another successful part of this sprint was our communication (maintained from the last sprint). Once again, all group members were actively engaged in individual work and team meetings. This was especially important for the sprint review, as was mentioned above. Everyone on the team was very good at setting expectations for when work would be completed. Also, there were many instances where, if one group member was adding on to another's code, the other group member would walk through how their code works.

Lastly, we were able to get through all the user stories outlined in the revised sprint planning document, unlike in the last sprint. However, we also did not have too few user stories so that we ran out of work before the end of the sprint. This will be a better setup for the beginning of the next sprint compared to this one, as we don't have to worry about unfinished user stories and rather focus completely on new ones.

User Story #15: I would like to be able to manage my friends requests so that I can control who I am friends with. This includes being able to accept and deny friend requests.			
1	Create a UI page that displays a user's friend requests, giving the options to accept or deny.	3hr	Kara
2	Create new collections and documents in Firestore for friends, friend requests, and own requests	2hr	Kara

3	Backend to add user to Firestore friends collection on SignUp	1hr	Kara
4	When the user clicks accept, add that friend to their list of friends in Firebase and the friend array in User class	2hr	Kara
5	Handle backend that sets the user to “friend” if accepted.	2hr	Kara
6	When the user clicks deny, remove that friend from the list of requests.	2hr	Kara
7	Add unit tests for friend permissions and request permissions.	3hr	Isha, Mary

For this user story (and user story #14): Implementing a friends feature for users was an integral part of this sprint and is a feature that will be used in further tasks. After completing the UI and storing friend information for a user in Firestore, we were able to successfully implement accepting/denying friend requests and accessibility to friend-specific information

What Did Not Go Well in Sprint 2?

The main thing that did not go well in Sprint 2 is we began experiencing a particular issue between our IDE (XCode) and our Version Control (Github). There would be instances where, when some group members would push to the repository, these changes would cause an error with building the project. This happened in the middle of week two, and, while we were trying to resolve the issue, certain group members could not build or run the project. Because of this, many members were not able to contribute to the project at all while the issue was being fixed. This also majorly contributed to the delay in progress of the project, leading to changes in acceptance criteria and a time crunch. We started to struggle to finish user stories on time, as we started losing a lot of time to fixing this problem. Eventually, we figured out how to resolve the issue, but it still appears occasionally, so we have been fixing it repeatedly.

Largely due to the time spent on the reason above and because of midterms, the end of this sprint was very intense. Since we had all turned our focus to midterms and other projects during the first two weeks of the sprint as well, we did not have as much time as we wanted to finish the user stories during the last week, and we had to revise our original planning document acceptance criteria to reflect that. While the revisions ended up being enough of a relief, it is unfortunate they were necessary.

Another issue we had was that most of us hadn't used the Firestore Database feature in Firebase. During Sprint 1, we primarily stored user information in the Realtime Database. Basically all team members had to work with the Realtime Database in some way during Spring 1. However, new parts of the application, such as Journals and FriendRequests, were implemented with Firestore. There was a needed learning curve for working with Firestore and learning its differences from the Realtime Database, which took up some time during this sprint. As all members were working on different parts of the project in Firestore or were not able to meet up to learn together, most of the learning curve was on our own, taking up even more time than previously planned. At this point we are fairly familiar with both the Firestore and Realtime Databases in Firebase.

In spite of these setbacks, we were able to successfully complete all of our user stories and have none to list in this section.

How Should We Improve?

In order to improve, first we need to maintain the things we know we have been doing well: communicate as much as possible, select the right amount of user stories for the sprint, organize the presentation for the sprint review, and so on. We can continue many of these by basing it off of our work from the last sprint. For example, the hours from our previous user stories can be a reference for how we estimate the hours for our next users stories.

Regarding the building error mentioned above, while we have not found a permanent solution, we know how to fix the issue when it appears. We are still looking to find the exact cause of this issue. Although we will still lose time to this problem during the next sprint, it will not be nearly as much as compared to this sprint.

Additionally, we now have experience using Firestore commands in Swift when most members of our team previously had worked very little with them before (if at all). Next sprint, regardless of whether we are developing a feature using Firestore or Realtime, we won't have to spend much time learning how the databases work. This will allow us to have more time for other user stories, as implementing database features is easier for all members of the team now than in previous sprints.

For the last sprint, in order to be successful for the final presentation, we will have to be more selective with the user stories that we choose to complete. As of right now, there are multiple user stories left to be completed, and not all of them will be able to be finished. Going forward for the next sprint, we will have to communicate with each other and decide which user stories should be the primary focus in order to make our app complete, rather than just simply choosing user stories that may be the easiest to implement or ones that we want to implement. This will include finishing up any features and implementing the ones that are integral to the app.

Overall, we have learned a lot from this sprint to improve ourselves for the next. By using these strategies, the next sprint should go even better than these past two. Although it is unreasonable to say we won't experience any issues in the next sprint, we are more prepared to handle the issues we have previously experienced.