Name: Upneet Bir

Date: December 5th, 2021

App Name: Counselor Aid

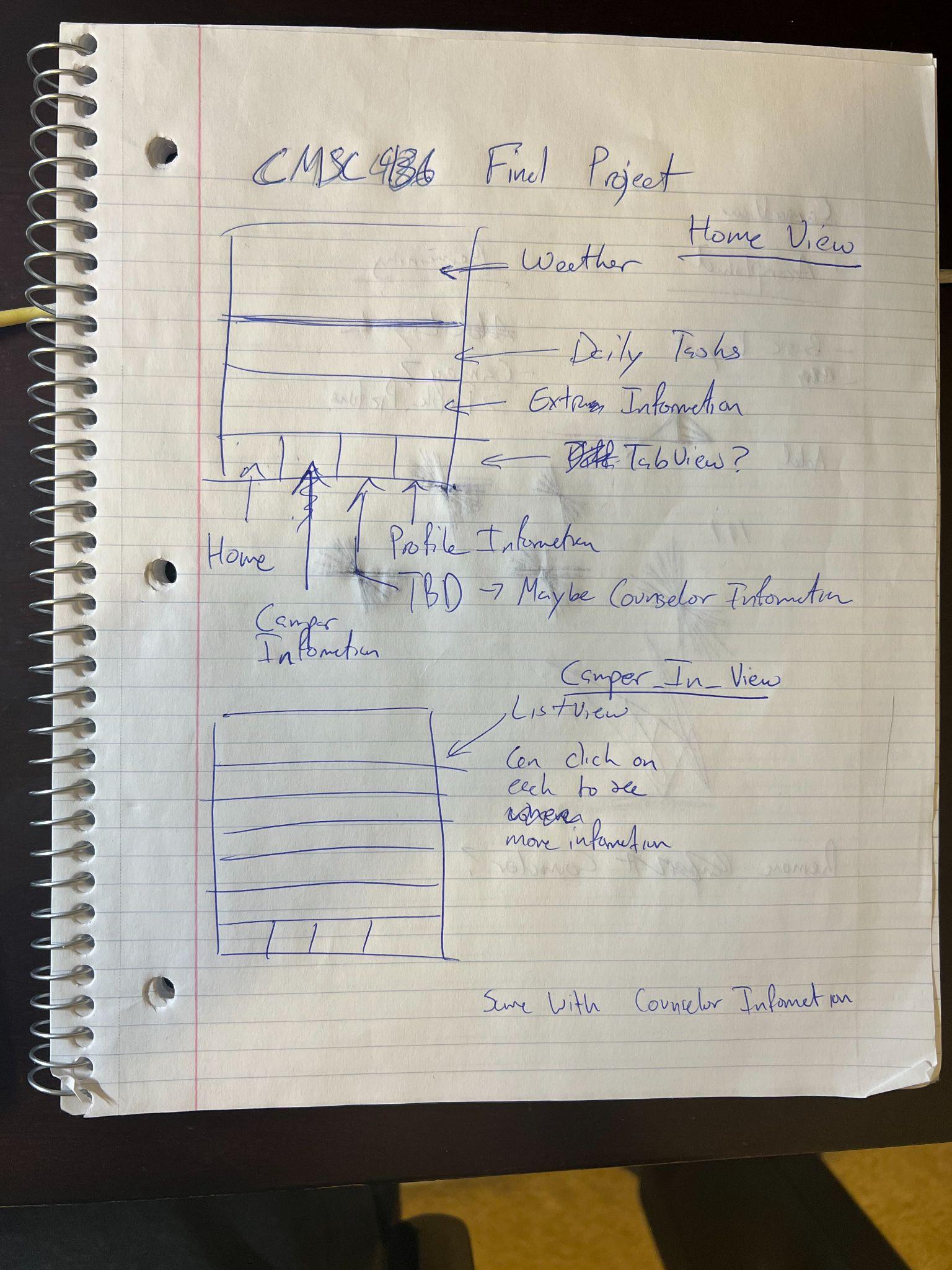
**Counselor Aid**

**Introduction**

The idea for my app comes from my personal experiences as a Camp Counselor. As a camp counselor, a lot of communication is delayed and done mostly over phone calls. This can cause problems because every instruction is mostly limited in the speed with which important information can be spread. Along with this, I have realized over the years as a counselor that every student has very specific needs and keeping track of this through the Notes app or Reminders app is much harder than expected. Adding all this on top of the fact that counselors are already low on energy throughout the day means that there is almost guaranteed to be some information that goes unreceived and could cause problems with how the camp is run. So the overall idea of my app was to create an all in one platform for counselors to better communicate with other counselors.

**Initial Ideas**

When I started the app, I decided to reach for the stars and create a very broad app that encompassed much more than the duties of a camp counselor. My initial plan was to create an app that displayed the weather, daily schedule, delays, Camper information, Counselor information, and even a photo gallery of the camp. Some of my initial sketches here show the ideas.

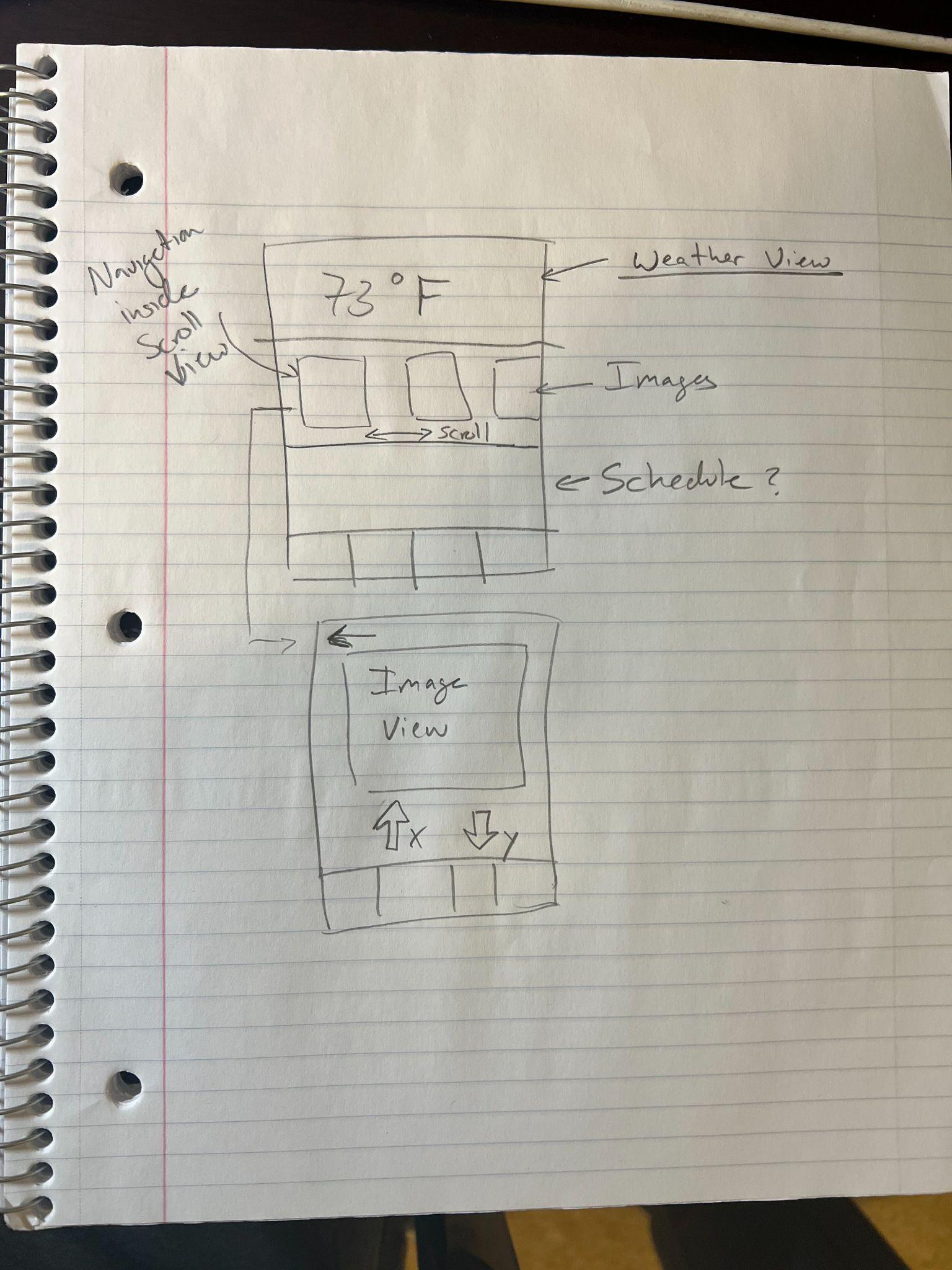


I started by creating a small timeline for myself. I have done projects in multiple week timelines and feel that I had a good amount of experience with ground up development to attempt this project on my own. When I started, I knew I would struggle the most on the backend. I also started this project before learning about the amazing services provided by Google Firebase. I decided to spend a good amount of my time creating schemas or entities for my individual objects.

This still left one problem, how would multiple users communicate with others? How would the updates they make be given to another user? Since we had just covered CoreData, I wasn’t sure how to connect another user to see my status or see what updates I make for a specific camper.

Taking a break on the backend, I decided to focus more on the front end of the project since we had covered most of these concepts already. When I first started the front end, it was very choppy and unattractive. The letters were blocky and the display wasn’t smooth. Not to mention that the color scheme was a very dull Light Mode. I still accomplished the work I set out to create a semi-functioning front end.

As I stated, at that time, I was planning to create a small image feed where counselors could add their pictures to be up and downvoted by other counselors. I experiment creating a view that would slide horizontally with small squares and implementing a small like system. Here is my basic drawing of that idea:



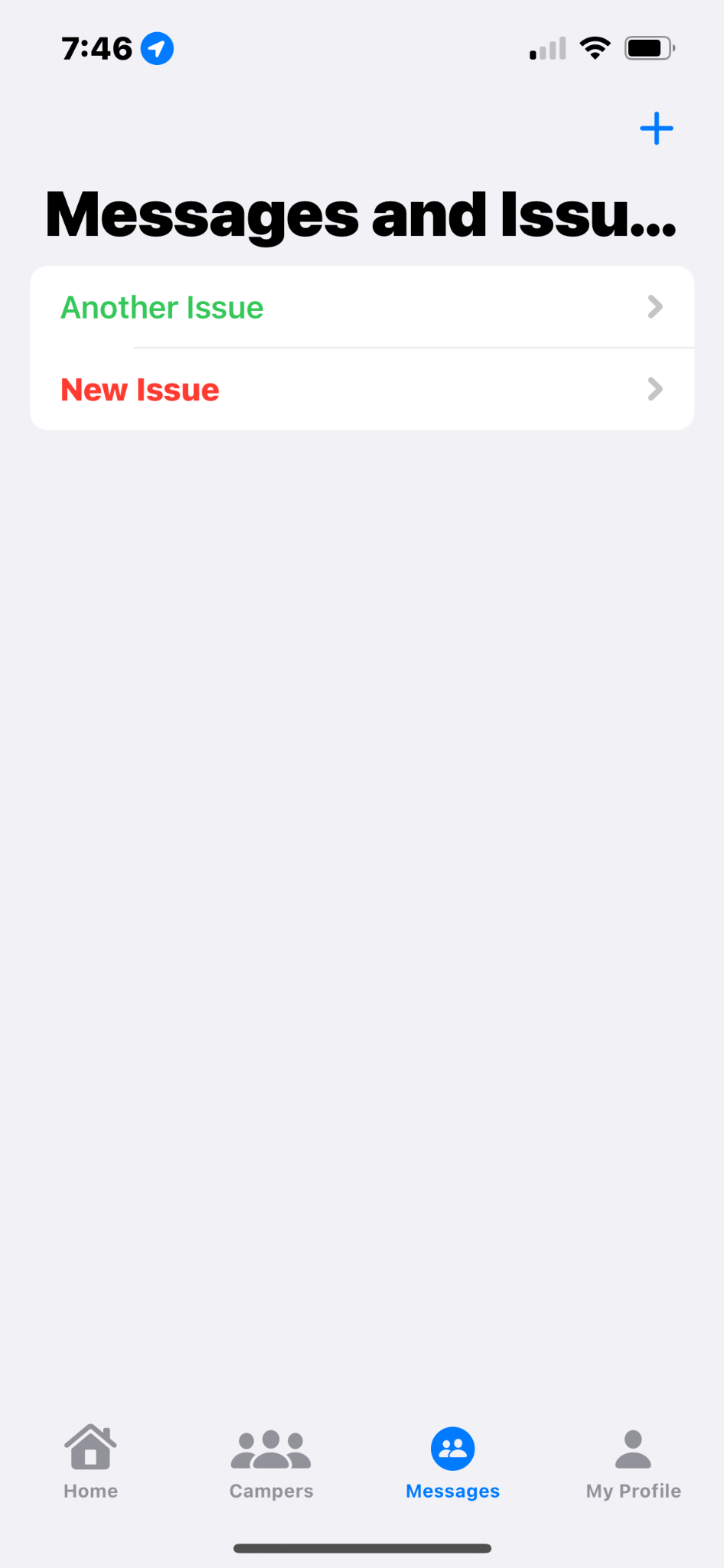
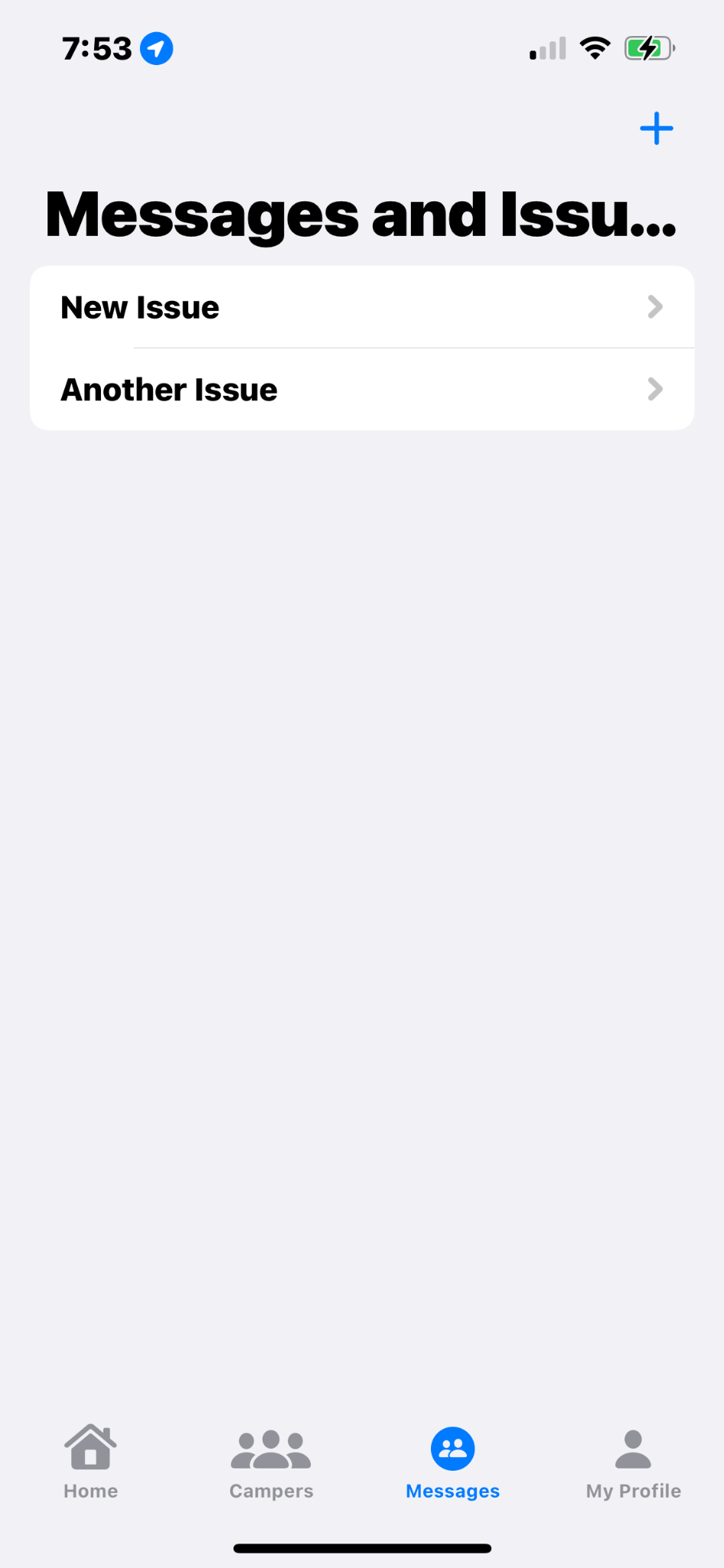
**Tabled Ideas**

When it came to the weather and schedule idea, I realized just how tricky working with JSON files can be. I have left my attempts at uploading the data into class Objects and the .json files I used inside an archives folder for your team to view. I was able to find an API online that provided a seven day forecast of the area and implemented most of the locational services required to complete this task, but parsing the raw data as it came back proved to be a bit more difficult than expected.   
 As I worked though the schedule portion of my app, I realized that it would be best to create a Schedule object in the backend. The pros were that the JSON format would allow me to store the data in a much more flexible way than other data structures in my opinion. It also made sense because the schedule isn’t something a counselor needs to change and interact with on the write level everyday. So I created a JSON file (in the archives folder) and started to create an example schedule. I tried parsing the JSON file but was met with many issues due to the fact that the Dictionary returned was of type <String, Any?> which meant that indexing deep into the schedule could prove more tedious than expected.

Thinking more, I decided that I could just directly write the schedule into the Firebase Realtime Database. This also provided a level of security if this app was to be tailored to clients because I, as a system administrator, would be one of the only users with access to the schedule, and new users/admins could be added at any time.

**What Worked**

Now onto the more successful approaches I had in the app. The first of which was the creation of a message or note board that could be broadcasted throughout the camp counselor users. For this implementation, it was not too hard, especially looking at online examples. The basic idea was to create a Message Object, which stores things such as the Subject, Message, and possibly the time the message was created. For a basic message or announcement, this would be ok. Still, in the case where there was a not too serious situation going on, it would be good to have some sense of what issues are pending and what issues are resolved. For example, a counselor needs pencils to be delivered to the main hall by around 4:00pm. To prevent that possibility of multiple counselors showing up with pencils and keeping with the theme of efficiency, I created a resolved feature in the app. This feature would simply allow people to mark these issues as resolved and unresolved as needed. In my opinion, this could be taken a bit further with a reply system. This way counselors could reply to the notes and ask for more clarification.

On the front end of this notes system, I wanted to create appealing visual aids that allow for the users to see that an issue has been resolved. On the left is the initial draft of the notes view and the right is the polished version. 

Visually, the viewer is given subtle cues as to what is impending and what is not. The bold, italics red presents a sense of urgency and provides another layer of color/contrast that appeals to the senses. Also, in my opinion viewing this board as a camper leader, I imagine seeing these contrasts provides a sense of accomplishment and reduces the overall need to check if something is in fact completed.

Throughout my time in development, I found that most of these ideas were a bit impractical and took away from the main essence of the app. While it would be great to have an app that could do anything (an app for every app), I figured it would be better that the app be great at a few sšecific things. In the end, I decided that my app would only display features that were relevant to the counselor and would help keep attention at the important tasks at hand.

**Improvements**

Looking back at my time, I see that there are many improvements that I could have made. I think the first thing that comes to mind is allowing users to receive visual aids as to how many incomplete issues/notes are remaining. Basically a more indepth notification system.