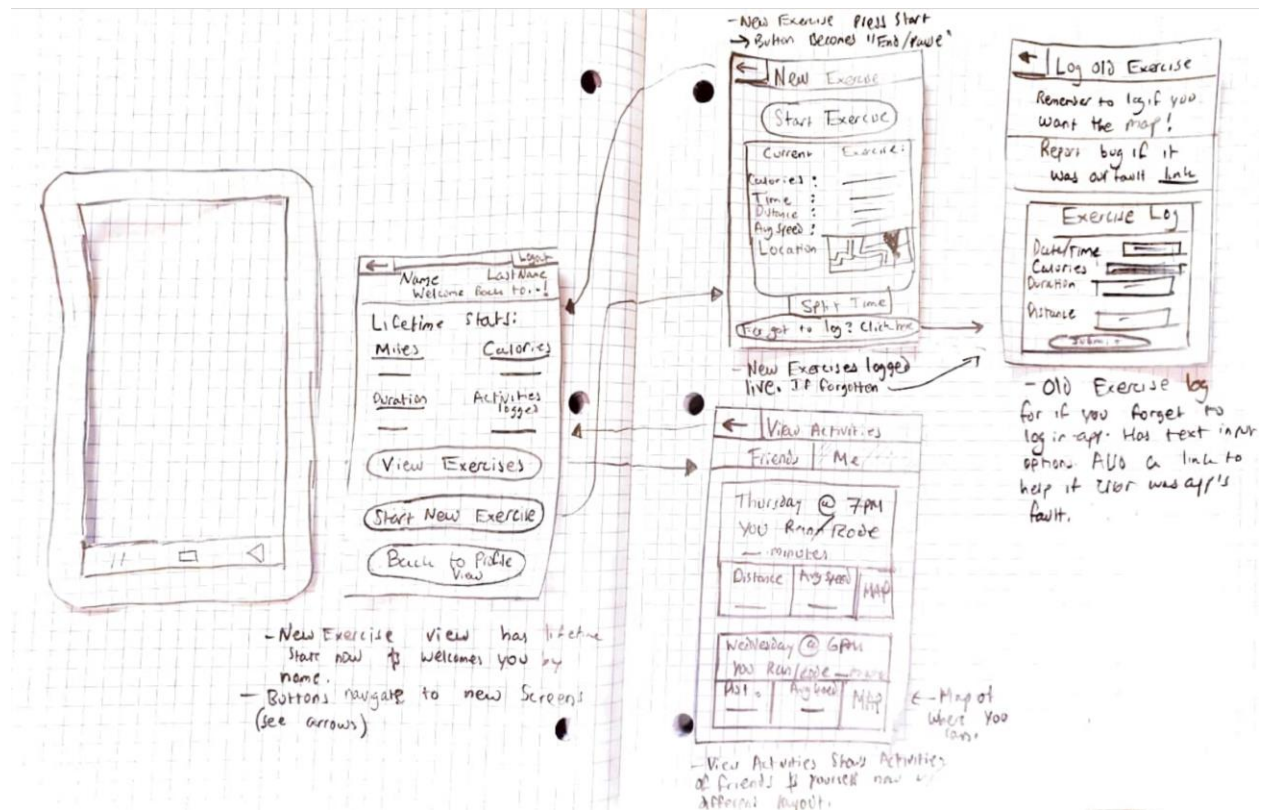


Prototyping

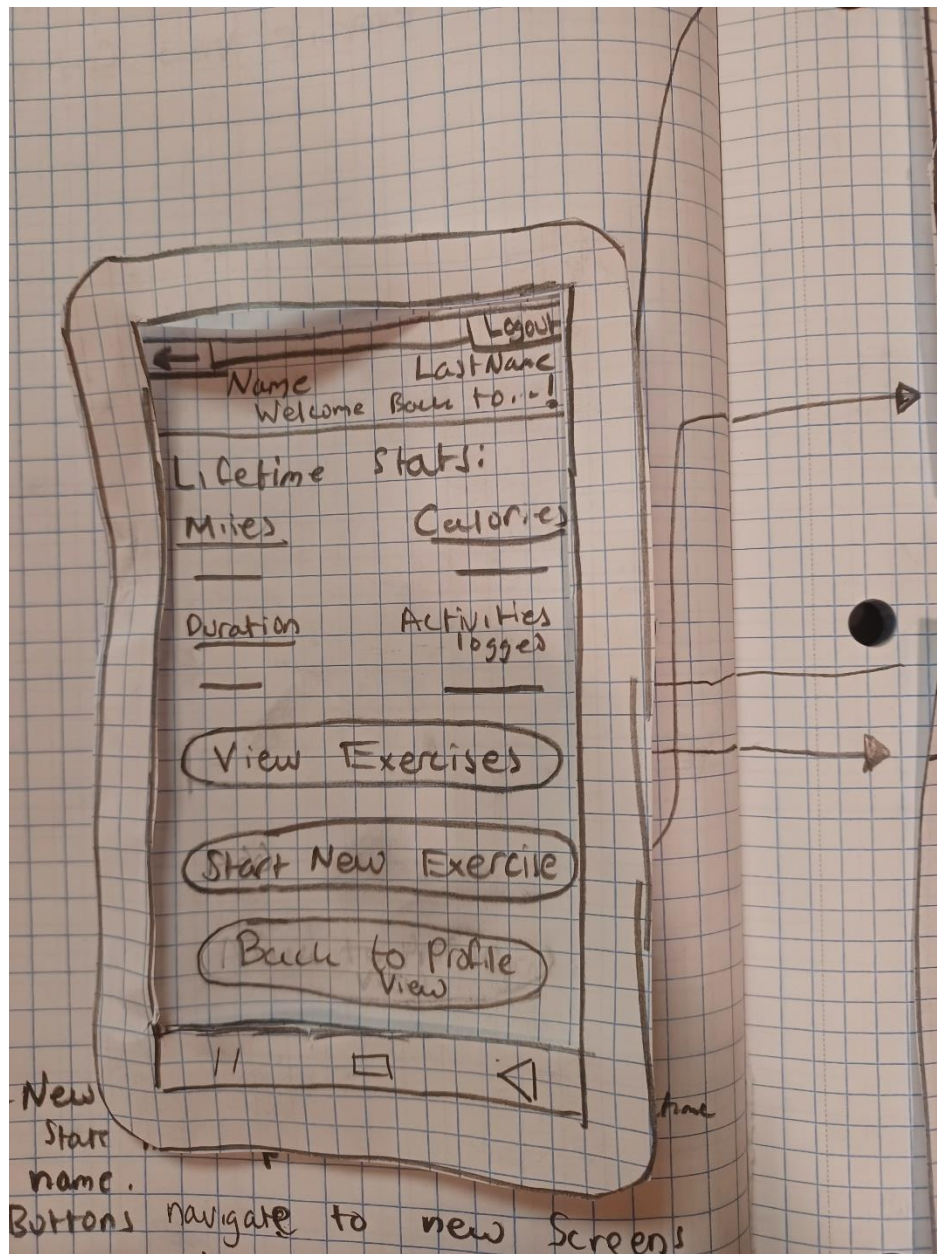
Part 1—Select a Feature: In this part, you will choose one of the four common features of a fitness tracking app, which you will be designing and prototyping in the next steps.

I would like to focus on the exercise view component of my react native alpha implementation.

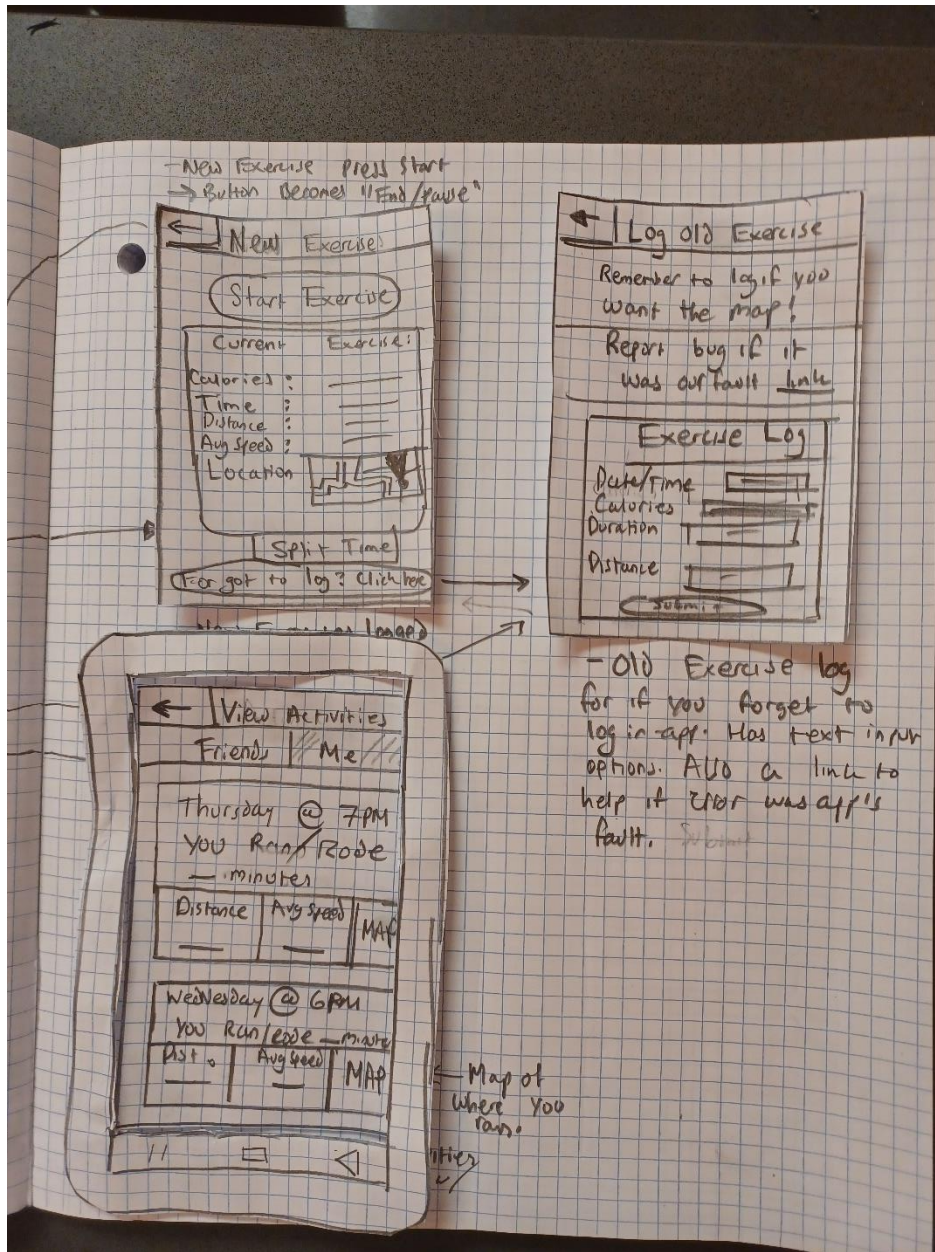
Part 2—Paper Prototyping: (1.5 Points) In this part, you will engage in paper-prototyping of the feature you chose from Part 1, using paper, pen/pencil, scissors, and tape/glue to develop and test out your ideas about how the user should interact with the capabilities required for your selected feature.



Here is a pdf scan of the paper cutout layout that I taped to some pieces of paper. There are zoom-ins below to show more detail. The exercise view is the leftmost card. From there, the options to view exercise, start new exercise, or go back to profile view will change the stack. The appropriate arrows indicate direction of travel that a user would experience when pressing the buttons. For this selected feature particularly, all of the capabilities of the previous implementation are required plus geo-tracking (so some sort of map interface) and also information on friends activities is required. Description of the screens is further explained in the scan on paper.

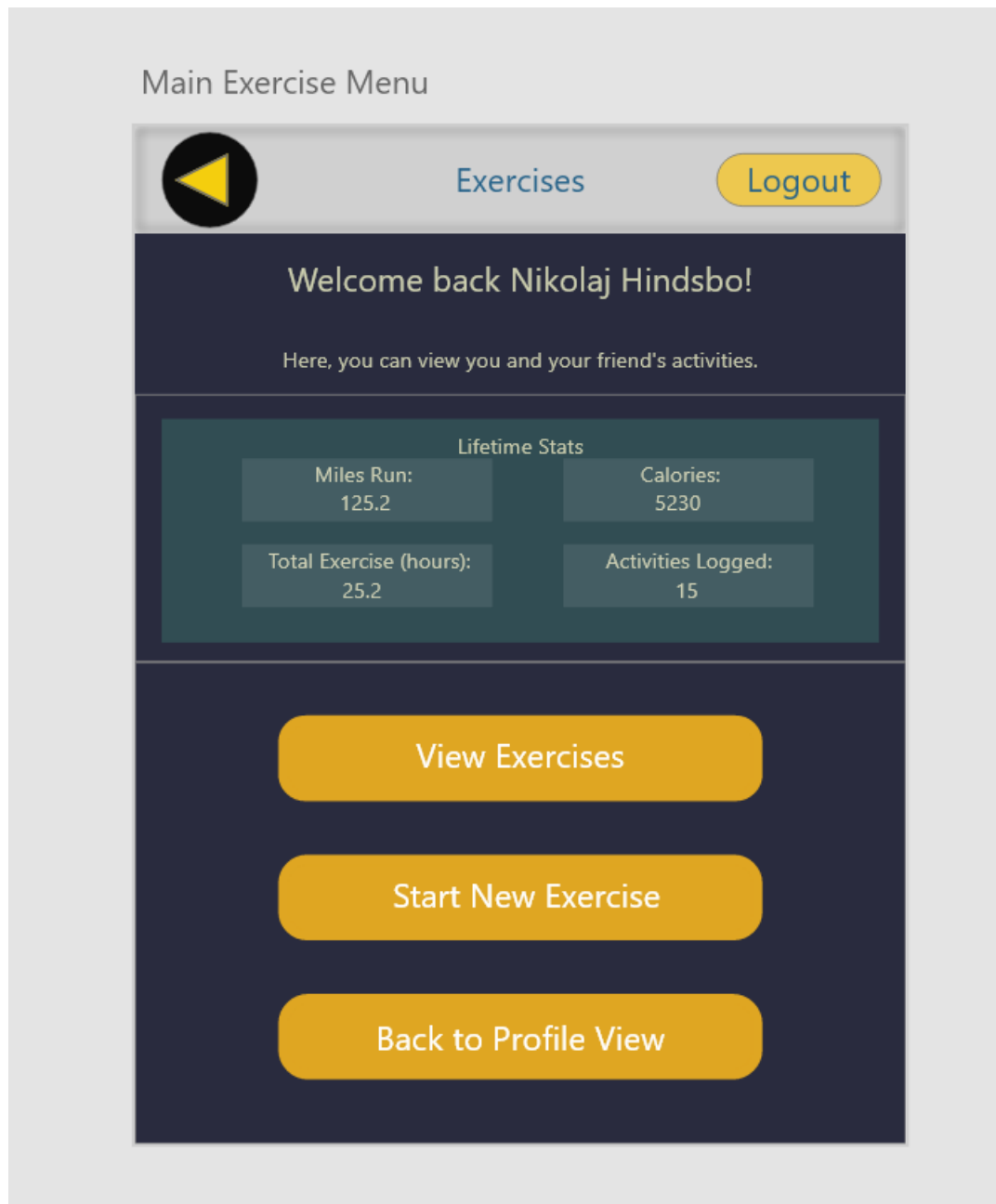


Zoom in of the main exercise mode with a phone cutout to show how it would look on the phone.



Zoomed in picture of the different options that you can go to from the exercise view.

Part 3—Visual & Interaction Design: (1.5 Points) This part will involve using the visual/layout design principles as well as the design patterns we learned in class to build non-interactive prototypes of the capabilities in the form of static screens.



Static version of the main Exercises view component. Lifetime stats are shown, along with buttons that allow for navigation to profile view, new exercise, and viewing exercises. I used a consistent coloring scheme for buttons and background.

View Exercises



View Exercises

Logout

Me

Friends

Thursday @ 7 PM you ran
for 30 minutes.

Distance
13.0 Miles

Avg. Speed
6.0 MPH

Map

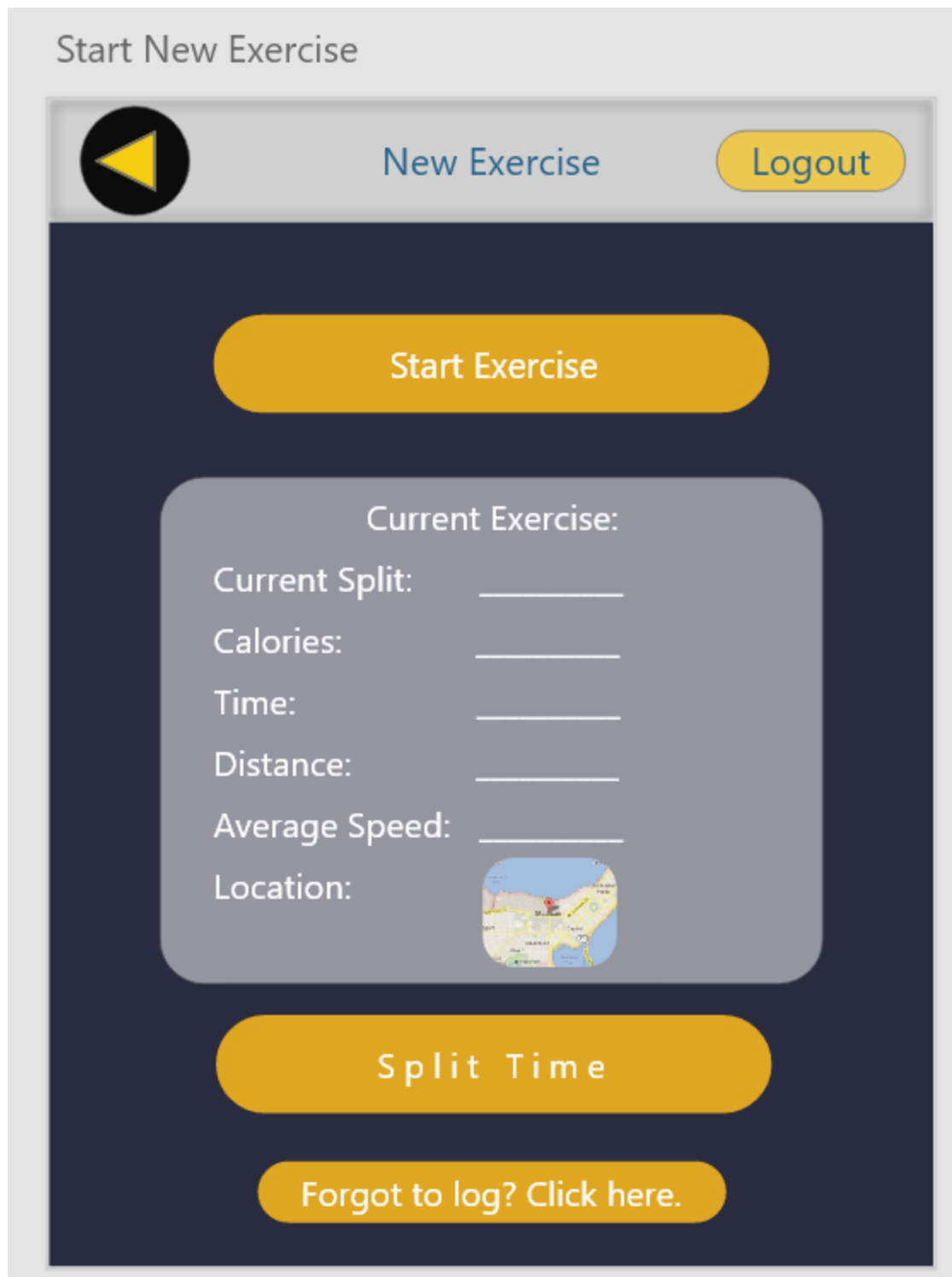
Thursday @ 7 PM you ran
for 30 minutes.

Distance
5.0 Miles

Avg. Speed
8 MPH

Map

The new view exercise is very different. Firstly, the view allows you to see your activity and that of friends. It is also separate from the new exercises now. If you would like to edit or delete the exercises, you should be able to by holding down with your fingers and pressing edit or delete when the options go up. This relies on user information, but makes it much neater than with an edit or delete button. The “Me” or “Friends” tab can be selected and indication with color of which tab you are in.



The new exercise view is different than before. Now, you can start your exercise and the app will track where you run. Press start will make the current exercise dynamic, changing the calories, time, distance, average speed, map, etc. with time. Split time can be added by pressing the split time during the run. If you forgot to log a run, you can click the button at the bottom to add a exercise without logging in-app.

Log Old Exercise



Log Old Exercise

Logout

Remember to log your runs live if you want geo-tracking for your maps so that you and your friends can see where you ran!

Report a bug on our website if a live run failed to log.

Exercise Log:

Calories:

Time:

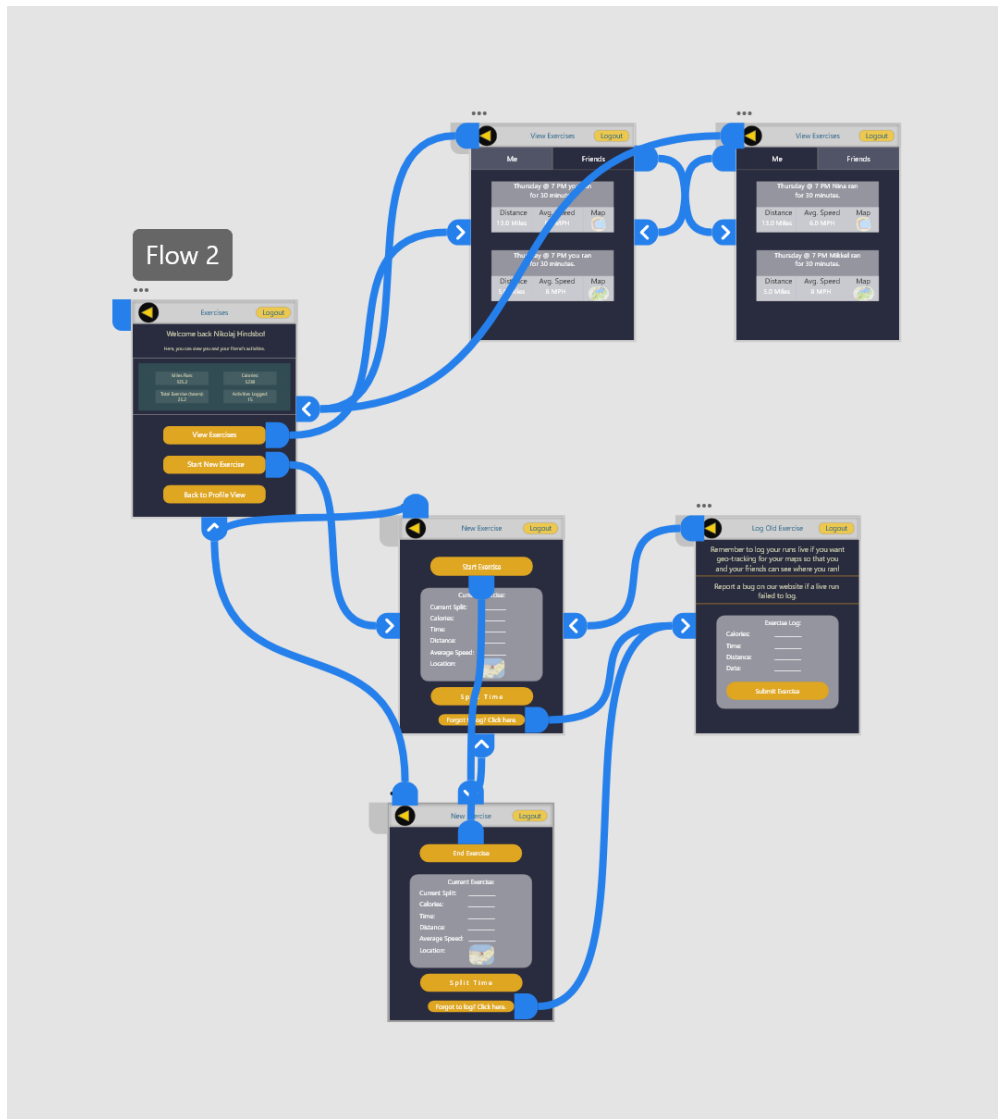
Distance:

Date:

Submit Exercise

Lastly, this view is the old exercise log. For those that forgot to log the run live, they can simply enter the values of calories, time, distance, and the date of the run. The date specifically should be a calendar app the user can interact with.

Part 4—Interactive Prototyping: (1.5 Points) In this part, you will add interactivity to your static screens by implementing an interactive prototype that integrates your designs using Adobe XD or other wireframing tools.



This is a large-scale overview of the interconnections and flow.

Adobe Link (Presentation Mode) link:

<https://xd.adobe.com/view/04c79a83-d2ba-46f7-b024-f4673cd217ac-5194/?fullscreen>

The height is set for 950 (android) and can also be used on PC. Not sure about iPhone.

Video google drive link below (I talk about the components as well):

<https://drive.google.com/file/d/1d3VprSKvsAJ1rWx4x2gO8pQ2YIXo0McG/view?usp=sharing>

Accessibility

Part 5—Discovering and Planning Accessible Design: (1.5 Points) In this part, you will discover the screen reader assistive technology features of your mobile device, plan how you might support two tasks in your React Native app using these features, and develop specifications to implement these features into your RN components.

Looking at the accessibility tools built into android, I learned that interaction control could be set on such that the phone ignores touches on the back, home, or recent button. Therefore, it may be more accessible to always provide a back button or a clear path backwards to a new stacked screen.

Therefore, two tasks that I would like to do is be able to navigate the exercise view without visual aids, and just the TextBack function if possible. For the other app, I will explore playing a chess game with just this information. Additionally, another task I will focus on is the accessibility in the form of navigation such that a user is able to navigate back and forth from stack screens without the use of the home, recent, or back buttons built into Android since that may not be accessible to everyone.

Specifically, for the two tasks I would need to set up for accessibility for text readers and alternative & augmentative communication. Firstly, for the text readers, I would need to properly set up components and buttons such that all parts of the screen allow for readability, specifically through text readers such as the TalkBack built into android devices. For the alternative and augmentative communication, I talk about that in the video provided below. Additionally, I talk a little about speech input on buttons and output, which is another task for accessibility.

Below, I made a video where I talked about different accessibility features that I had identified to support these tasks (plus a quick bonus task at the end, which was thought of since lots of the accessibility features rely on the built-in keyboard).

Link to video:

https://drive.google.com/file/d/1dE9evZUdAwDUGyZT_40Q2bRHDR0tIKF-/view?usp=sharing