CS-571 Building User Interfaces, Spring 2022, Professor Yuhang Zhao

React Native *β* (6 Points)

Prototyping & Accessibility

In this assignment, you will pick one of the common features of a fitness tracking app: exercise tracking, meal tracking, health overview, and activity competition with friends. Then, you will design and prototype the feature you chose using the concepts we learned in Mobile Design and Prototyping lectures. Finally, you will explore accessibility features and assistive technologies of mobile platforms by testing your implementation of React Native α.

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.

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.

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.

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.

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 α using these features and develop specifications to implement these features into your RN components.

Submission Details

1. A completed version of this document as PDF to Canvas.
2. A link to your interactive prototype, submitted to Canvas.
3. A video recording of you demonstrating in MP4 format the intended use of your interactive prototype, saved in your Google Drive folder and shared through a link (instructions) (as video files can be too large for Canvas to handle).

Prototyping

Part 1: Select a Feature

In this part, you will choose one of the following four features that you will design and prototype in the next steps. Note that this assignment does not involve any programming for implementing a feature; it is solely focused on designing and prototyping, so feel free to be creative.

Feature 1. Your Exercises

- The user should be able to see a list of exercises recorded for the current day.
- Provide the ability to add/delete an exercise.
- An activity consists of a name (e.g., "Jogging"), duration, date, and number of calories burned.
- Allow the user to compare their goals of calories burned/activity minutes versus the current day's stats.

Feature 2. Your Meals

- The user should be able to see a list of foods eaten during the day.
- Provide the ability to add/delete a food item and specify its mealtime (e.g., breakfast)
- A food consists of a name (e.g., "Whole Wheat Bread"), and food "macros," i.e., number of calories, proteins, carbohydrates, and fats.
- Allow the user to compare their calories/macronutrients goals versus the current day's stats.

Feature 3. Health Overview

- The user should be able to see the 7-day history of their health metrics at a glance.
- Health metrics consist of the user's body weight, hours of sleep, and step counts.
- Provide the ability for the user to record body weight and hours of sleep for a specific date. Assume that step counts are automatically tracked by the app.

Feature 4. Competition with Friends

- The user should be provided with a leaderboard that allows them to compare their step counts and activity minutes with their friends.
- Assume that step counts and activity minutes of the user and their friends are automatically fetched from the server.
- Allow the user to invite a friend to compete with by specifying their username.
- When the user gets added as a friend, they should get a notification and be able to accept/reject the invitation.

Feature 4. Competition with Friends

Part 2: Paper Prototyping (1.5 Point)

In this part, you will follow the principles and methods we learned in class to develop a paper prototype of each capability listed in the feature you chose. As we discussed in class, paper prototyping is a powerful tool to rapidly develop and test ideas in an iterative fashion. When you prototype, think about how the user will interact with the capabilities you are devising, how the different screens will progress after user input, and what elements will remain on the screen and what elements will change. Therefore, your prototypes will be much more than a set of static screens.

Note that you are expected to work with real paper, and not in the digital environment, as prototyping in the physical world removes constraints and biases introduced by digital tools that are available to you. You can use white and/or colored paper. You will need scissors to cut paper into components. You can also use post-it notes in various colors or sizes/shapes to represent components. You should not use any stencils or component libraries at this point, and instead draw all the elements by hand. To draw shapes, make outlines, and write labels, using a Sharpie is recommended, although a regular pen or pencil may also be used. You can use glue or tape to attach smaller components to screens. However, note that the paper prototype should allow you to simulate user interaction with your design, which you can achieve by moving components on and off the screen.

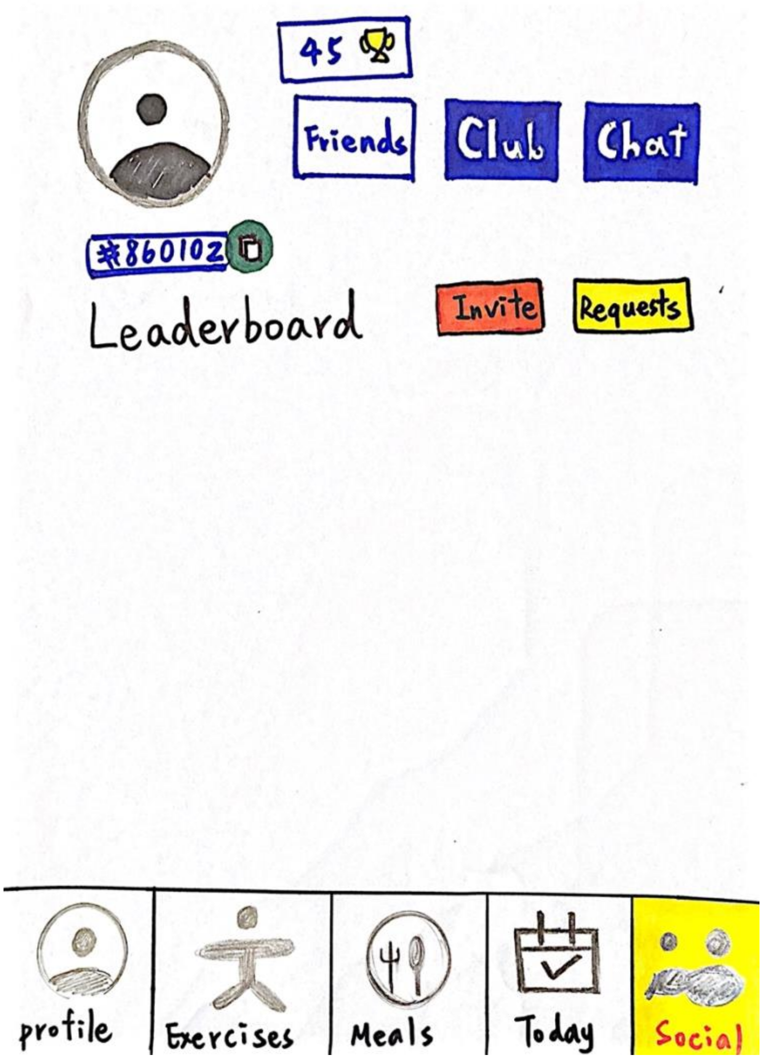
When you are done with paper prototyping, place your paper prototype on a blank paper, write down simple annotations for each major component, and take a photo of your annotated prototypes for your submission. Although a video demo is not required for this part, feel free to make one if you believe it will help us understand your paper prototypes even better; we'd love to see your paper prototypes in action!

There are no hard-and-fast rules, and you can get creative with this activity, but here are a couple of primers on paper prototyping that may be useful: A List Apart: Paper Prototyping, Patrick Thornton: Paper Prototyping: A Primer.

Video Demonstration –
https://drive.google.com/file/d/1QLgnknFocADvpJIvmFgV3y6uLjY-hFIh/view?usp=sharing

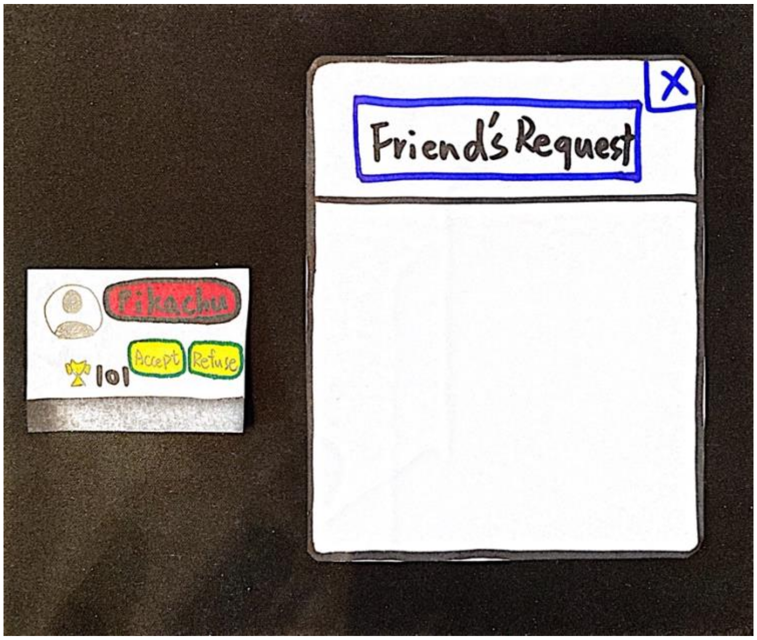
Annotations for components –

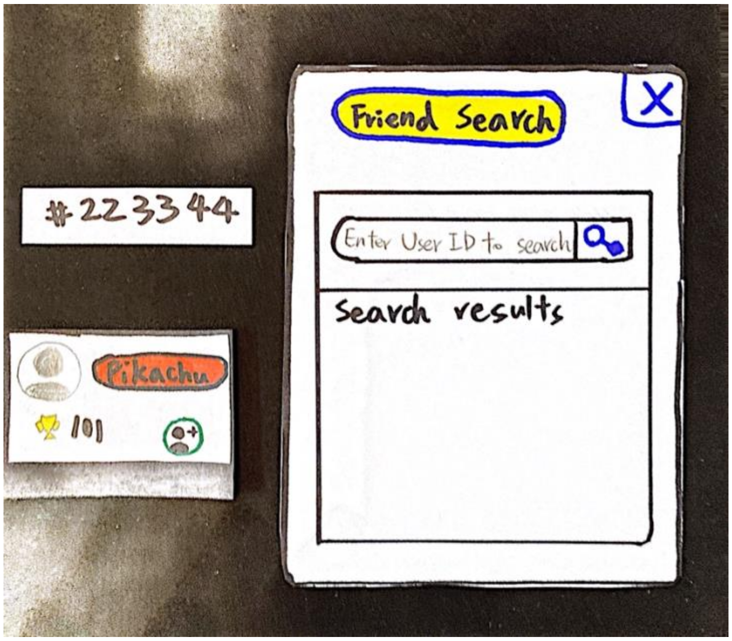
This is the main Competition view in my prototype. In this view, I have bottom selections for users. For my competition view, I decided to locate it in the “Social” tab with other socializing features, such as, Club and Chat.



These two adjustable pieces are for indicating the colors after user pressing, and it is helpful to demonstrate animations.

Friends' invitations and requests are also dynamic. After user presses 'invite' button, the application pop-up other modals to deal with the further actions for searching friends or handling accepting, refusing invitations.

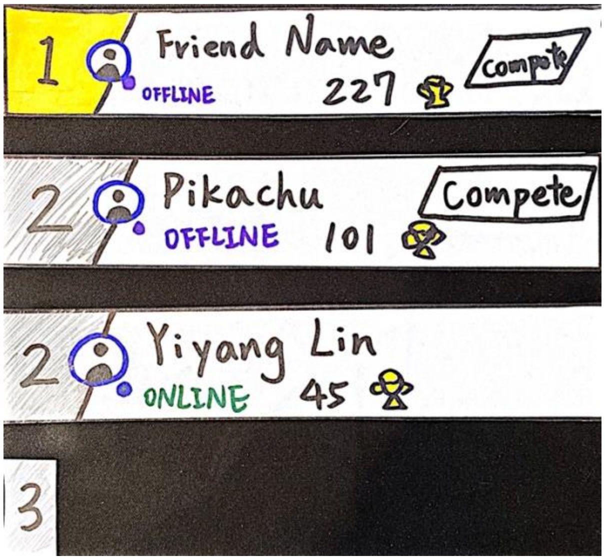


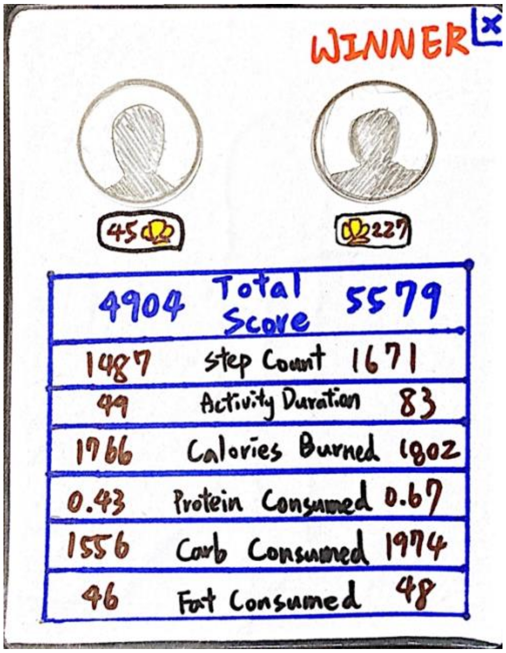


After pressing copying user ID or pressing sending invitation to friends, it will show up the current system status, and after few seconds, it gradually disappears.

User ID copied to the Clipboard
Friend Invitation Sent !

These are the leaderboard's content. The application will rank how many accomplishments the user and the friends have completed and creates this ranking.





4904	Total Score	5579
1487	step Count	1671
49	Activity Duration	83
1766	Calories Burned	1802
0.43	Protein Consumed	0.67
1556	Carb Consumed	1974
46	Fat Consumed	48

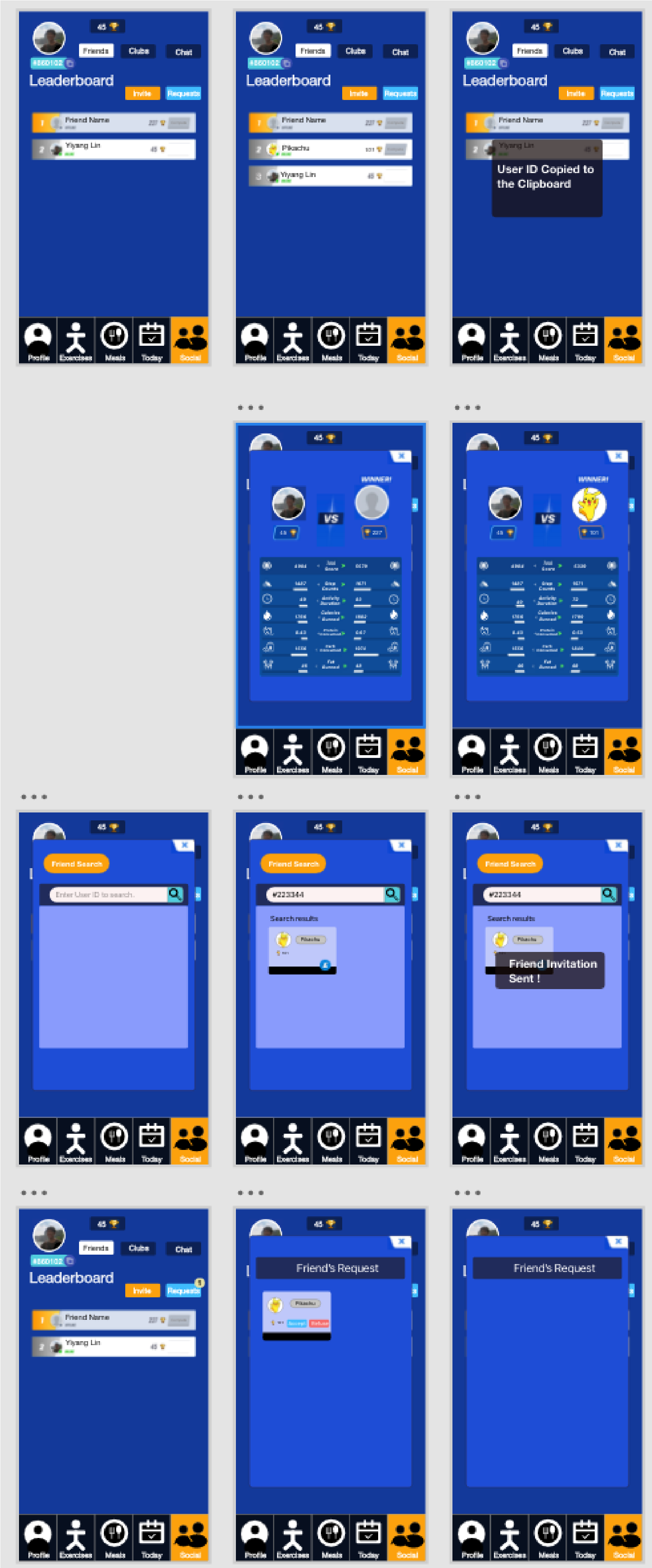
Part 3: Visual & Interaction Design (1.5 Point)

This part will involve building on the paper prototypes you created in Part 2 to develop hi-fidelity visual designs of your screens for each capability. In designing your screens, you are expected to employ the visual and layout design principles we learned in class as well as the mobile design patterns we reviewed in class, although you are not bound by these patterns and can explore different visual arrangements. Your hi-fi prototyping will also involve choosing color schemes, typography, and image use that are consistent with the conceptual design for your app. For example, what color schemes, what images, if any, and typefaces would promote healthy nutrition and exercise? Alternatively, what design choices would motivate users to meet the goals that they set for themselves?

Your visual prototype does not have to look photorealistic and employ the look and feel of native OS elements, but you are free to use stencils or standard elements from libraries or UI kits as well as icons, shapes, and images (e.g., from online libraries such as freepik).

You will complete this part of the assignment using Adobe XD (you do not need to purchase it, as the assignment can be completed using its free features). If you are unable to use Adobe XD, e.g., if you only have a Linux computer, you can use alternative prototyping tools like Figma, or create your visual prototype using Microsoft PowerPoint, Keynote, or alternative presentation software that has basic drawing and interactive capabilities as well as the ability to export presentations as videos (as needed for Part 4).

The deliverable from this part of the assignment will be screenshots of the screens from your hi-fidelity prototype. You can paste cropped screen grabs or exported screens in image format.



User ID Copied to the Clipboard

Friend Invitation Sent !

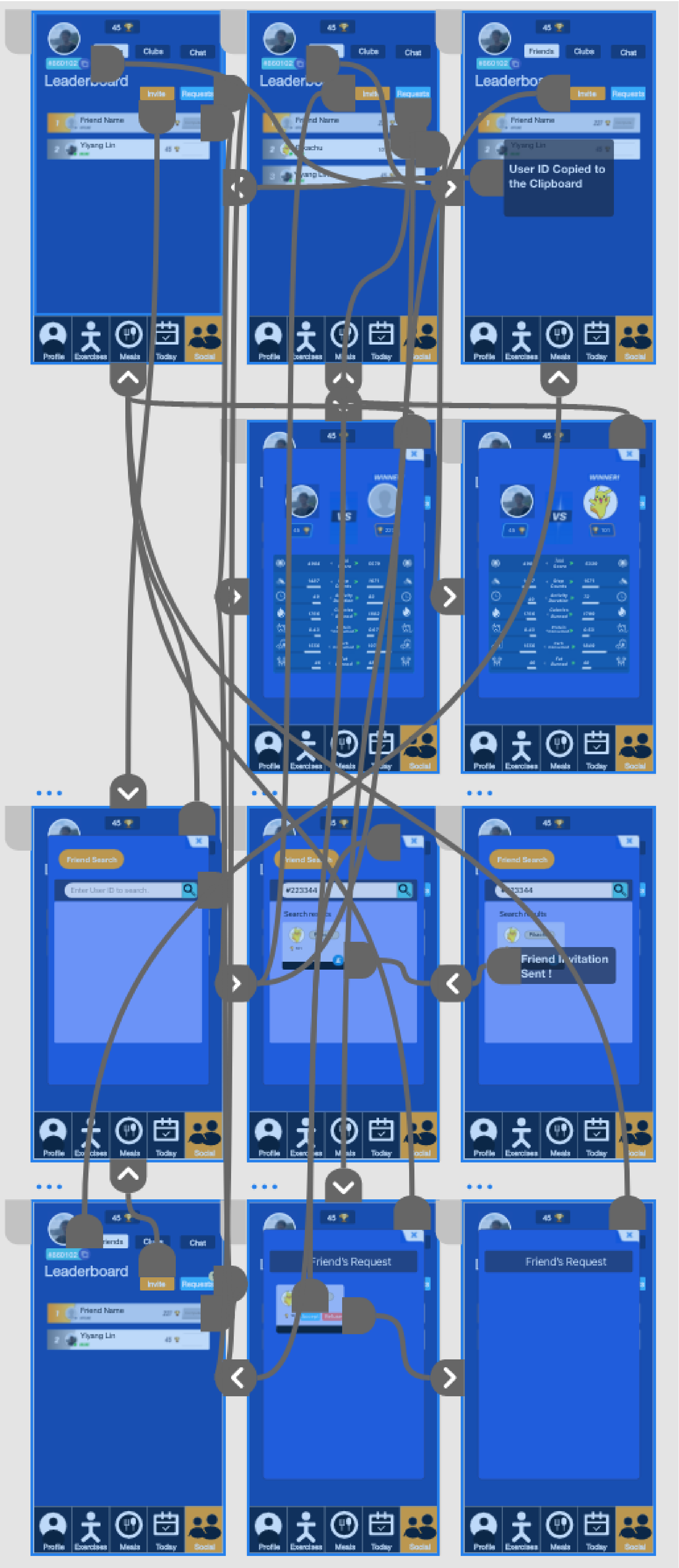
Part 4: Interactive Prototyping (1.5 Point)

In this part, you will build on your hi-fidelity prototype from Part 2, which consisted of a set of static screens, and the ideas you developed in Part 1 for how the users will interact with the capabilities in order to develop an integrated, interactive prototype. You will continue using the same software environment and its capabilities to link screens, simulate scrollviews, etc. Adobe XD has the capabilities to create interactivity out of the box, although modern presentation packages can be used to achieve most of these capabilities (e.g., see this tutorial on how to use PowerPoint for prototyping). To get started with making interactive prototypes with Adobe XD, you might find this tutorial from Adobe XD or this video tutorial useful.

After you complete your interactive prototyping, you will create a video demonstration of how users will interact with the prototype. Again, Adobe XD has the capability to record a demonstration of your prototype, if you are not able to use Adobe XD, you can achieve the same goal using Zoom (see this tutorial on how to record a presentation using Zoom).

<link-to-your-interactive-prototype>
https://xd.adobe.com/view/ff2192f1-843c-47c9-9fd8-cbfa1b9fdf02-aa72/

<record-a-video-demo>
https://drive.google.com/file/d/1OdRQjmP3rVY95MgAMfLH4XR7rerK8kEL/view?usp=sharing



User ID Copied to the Clipboard

Friend Invitation Sent !

Part 5: Discovering and Planning Accessibility Features (1.5 Point)

In this part, you will engage in discovery of the screen reader assistive technology in your mobile platform of choice, prepare tasks for supporting accessibility in your application, and design the experience for a user with visual impairment across two steps.

Step 1. Discovery of Accessibility Features (0.5 Point). In this step, you will explore the accessibility features of the mobile device platform in which you have been testing your React Native projects. Your testing environment can be an iOS or Android device using the Expo app or an iOS or Android emulator on the computer. By enabling VoiceOver in iOS[1] (Settings → Accessibility → VoiceOver) and TalkBack in Android[2] (Settings → Accessibility → TalkBack) or accessibility testing tools in your emulator (e.g., Accessibility Inspector in Xcode), you will assess how screen readers work across two applications:

1. The latest version of your React Native α fitness tracking application
2. Another application of your choice that you frequently use

Complete or attempt to complete two common tasks for each application with the screen reader on and report below your observations. Specifically, describe what tasks you performed or attempted to in each application and how the applications supported the task. Explain in detail how well the screen reader was able to guide you through the tasks.

React Native α –

Fitness Tracking - Sign-up an account

Typing Voice Pitches –

I found out this small deficiency when I was in the very first views. The actions of first spoken touch are for recognizing the user's positions on the keyboard. After the user identifying the intended characters, the user needs to double-tap the same position on the keyboard, and in the meantime, the pitch of voice will specifically change to indicate which characters will be typed in. However, this is sometimes undifferentiable. When I tied to type on the device without see the screen, I am often to be lost and must start over again. I was not sure the characters were typed on or not because of the pitches of voice are not clearly differentiable.

Password Typing –

For the sake of security, the passwords in assistive technology were designed to be pronouncing first and then the second tap of word was pronounced in the other forms of expressions. For instance, the first tap of character 'A' will be exactly pronounced and when the second inserted 'A' will be pronounced as 'Apple'. The is a good alternative design. However, when it comes to some cultural minority, it can also be inconvenient. Some words may not be correctly understood. Some words are not always pronounced in English.

[1] You can learn more about iOS VoiceOver here: https://www.apple.com/accessibility/iphone/vision/

[2] You can learn more about Android TalkBack here: https://support.google.com/accessibility/android/answer/6283677?hl=en

Frequently Use Application –

Outlook – Sending greeting email with an attached fun picture to myself.

The first step is to compose the mail. At the top right corner, there was a "compose" icon to indicate start composing mail. When I double tapped the button, it pop-up a blank New Message template. Like most of the inputting methods on keyboard, I was able to be notified characters from VoiceOver and double tap to type in. For each function, such as, attach meeting, attach file or photo, take photo using the phone camera, reach text formatting are shown on top of the keyboard and VoiceOver correctly.

In order to selecting and scrolling the photo library, I have to utilize gestures to control each view and choosing intended photo. I was notified the start of rendering uploading image and when it was done.

After listening each voiced character and double tapping of inputting, I completed content of my greeting. Later, I finished sending to my mailbox.

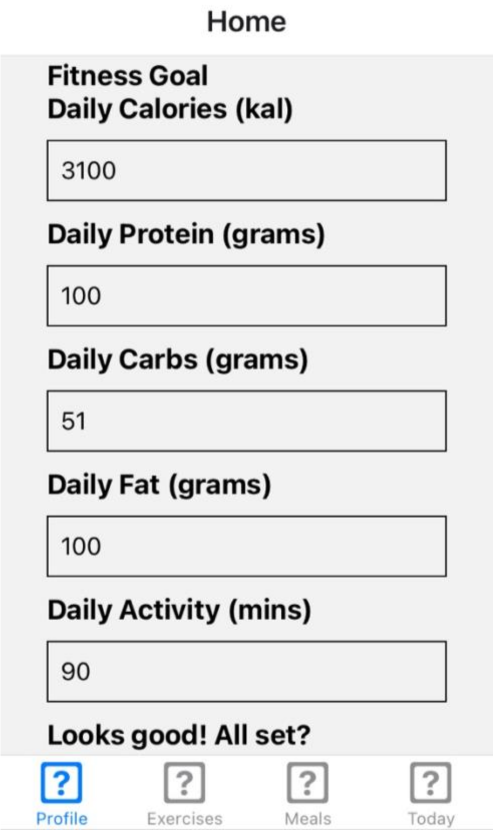
Scrolling –

At the first glance, I was stuck on scrolling views. I was not able to understand how to control and see more content in my mailbox. I could only see 6 mails on the initial page. After that I figured out, I need utilize gestures to control. This gesture is not easily be observed.

Selecting photo –

When I entered the photo library of selecting intended images, it should need more details to VoiceOver each toggled image, so that the users with visual impairment can look through each item and decided which is going to be selected and uploaded during the spoken hints.

Step 2. Planning for Accessible Design (1.0 Point). This step will involve determining how to specify accessibility features for your React Native α implementation. Focusing on the components (e.g., ExercisesView) that were involved in your tasks from Step 1, you will write out how you will enable accessibility features using React Native Accessibility (review the accessibility properties and actions), such as where accessibility features should be enabled, what labels and hints should be provided, what accessibility actions should be supported, on. It is important to put yourselves in the shoes of a user with visual impairments and consider how you would like to support user navigation and interaction, what the labels should say exactly so that they accurately and effectively communicate the functionality of each component. Note that you do not need to implement the accessibility features; focus on identifying which accessibility features from React Native you would use to make your React Native α more accessible.


Home

Fitness Goal
Daily Calories (kal)

3100

Daily Protein (grams)

100

Daily Carbs (grams)

51

Daily Fat (grams)

100

Daily Activity (mins)

90

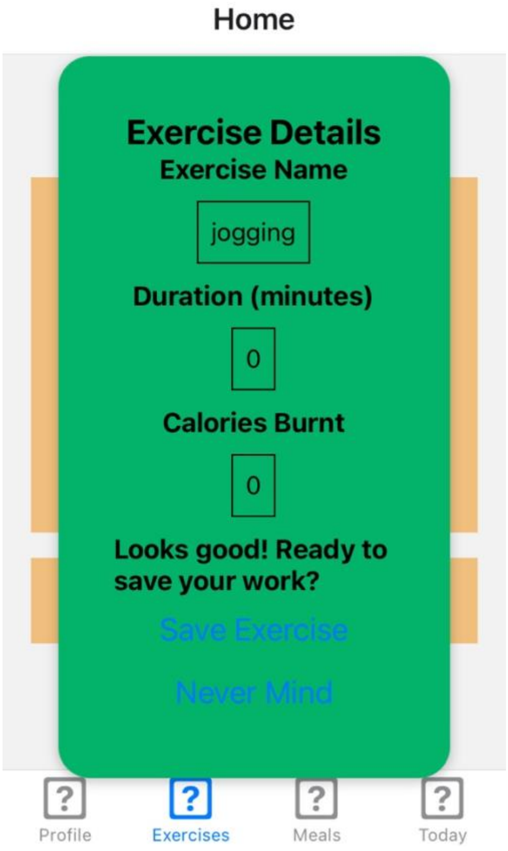
Looks good! All set?

Profile –

In my *react native alpha*, the profile view needs to be filled out information from the user that including Daily Calories, Daily Protein, Daily Carbs, Daily Fat, Daily Activity. If the users are with Sensory Impairment, they may not be able to focus their vision on a certain input box. In this design, the inputting boxes are close and stay together, it will be easily to let the users lose their control and consequentially starting over again their input.

I decided to insert *accessibilityLabel* and associate the relevant strings with the current input boxes to notify the users that information of current ingredients. For example, "This is a Daily Calories input area", "This is a Daily Protein input area" …

In addition, as I observed in the part 1, I found that assistive keyboard typing in iOS is sometimes unfriendly to users with visual impairment, I will specifically set the *accessibilityRole* with *keyboardkey* when the element acts as a keyboard key. This feature may reduce the times of starting over and lose their positions.


Home

Exercise Details
Exercise Name

jogging

Duration (minutes)

0

Calories Burnt

0

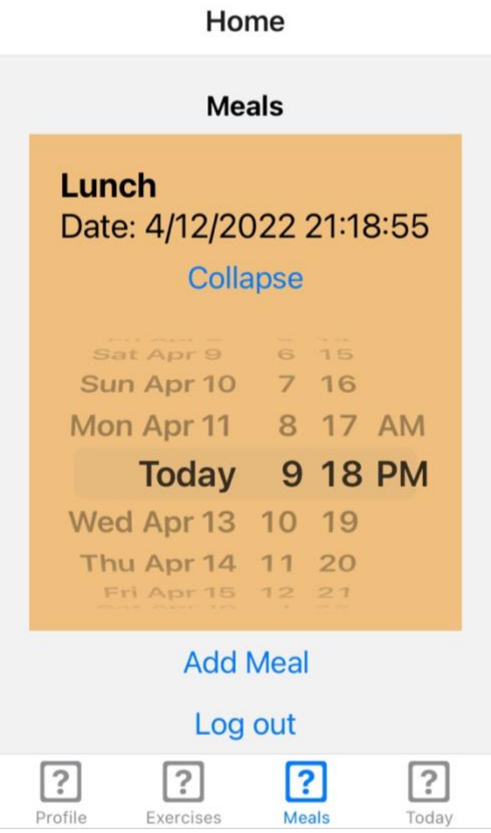
Looks good! Ready to save your work?

Exercises –

I observed that modals components in react native is sometimes show up abruptly. This may block the user's action from the previous background. For instance, when the user is in this green modal, it is not able to hit the background button, such as, Profile, Meals, and Today.

In the created modal, the popup is from "adding Exercise" button. I intentionally want to specify the *accessibilityViewIsModal*=true and *accessibilityHint="Navigates to the previous screen"* to remind the user that the "Never Mind" button is to go back and exit the current modal.

To support more assistive features, I will design two different sounds of "Save Exercise" and "Never Mind". This can give more hints to people with visual impairment.



Lunch
Date: 4/12/2022 21:18:55

Meals –

Setting time’s control is also not too obvious to users with visual impairment, especially the background color cannot be changed. The default instruction is “pick an item, adjustable…”

This widget was not designed for VoiceOver in iOS. It is hard to tell the user how to adjust the time by using VoiceOver in iOS. The animation to user is uncontrollable and hard to understand by its gestures.

When retrieving information from first tap, I decided to add more instructions and gesture details to this time settler. For example, *accessibilityHint=" using two fingers to scrolling up or down".* Besides, I will create new gestures to specifically identify it is adjusting date, hour, minute, or AM/PM. When the intended time is adjusted, it will be indicated as a vibration of the phone to be done sync with the back-end server.

Your submission will include (1) a PDF of this document, including pictures of your paper prototype from Part 2 and screenshots of your hi-fi prototype from Part 3, (2) a link to your interactive prototype that you can obtain from Adobe XD (the free version allows sharing one public link), and (3) the recorded video of your demonstration in MP4 format, shared through a Google Drive link. If you did not use Adobe XD, submit a copy of your PowerPoint (or alternative) file in Canvas.