CS-639 Building User Interfaces, Fall 2020, Professor Mutlu

React Native 2 ⍰ (3 Points)

Accessible Design

Early Due Date (+1 Point of Extra Credit) Tuesday, November 17th @ 11:59 PM CDT

Regular Due Date Sunday, November 22nd @ 11:59 PM CDT

In this assignment, you will build on your React Native 2 α assignment to explore accessibility features and assistive technologies of mobile platforms. Specifically, you will integrate React Native accessibility features into your fitness tracking app to support screen reader use.

Part 1—Discovery, Planning, & Specifying: 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 fitness app using these features, and develop specifications to implement these features into your RN components.

Part 2—Implementation: This part will involve implementing the specifications developed in the previous part as well as ensuring that other components do not distract a user with visual impairments.

Part 3—Testing & Demonstration: In this part, you will demonstrate the two tasks you are supporting with your implementation and capture your demonstration in the form of a narrated screen recording.

Submission Details

GitHub Classroom Starter Code

React Native 2 β will build on your implementation of React Native 2 α. You should copy your code from your React 2 α project to the React 2 β repository linked above, as that will be your starter code. When you commit/push, ensure that you are committing/pushing to the react_native2_beta repository, not react_native2_alpha. To complete the assignment, you will need to submit:

1. A completed version of this document as PDF to Canvas;
2. Your repository name and latest commit hash from GitHub Classroom E.g., "react_native2_beta-ctnelson1997, 2b0ef83"
3. A video recording of you demonstrating in MP4 format the intended use of your prototype, saved in your Google Drive folder and shared through a link (instructions) (as video files can be too large for Canvas to handle).

Part 1: Discovery, Planning, & Specifying (1.4 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 three steps.

Step 1. Discovery of Accessibility Features (0.3 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 fitness tracking application
2. Another application of your choice that you frequently use

Complete or attempt to complete two common tasks in both applications 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.

The two tasks that I performed were signing up and logging in. For the fitness tracking application, everything started on the Login screen. Completely relying on the TalkBack, I had to swipe from left to right to reach the signup button which is the very last component on the screen. The screen reader will only read out loud the text on each component, which didn't help the user to know its functionality and what they should do next to activate that functionality, such as a button press or a text input. The same happened to the login screen of my fitness app, and the only thing that is different from Signup screen is the header and the buttons. Otherwise I still found it difficult to fully understand what to do with merely the screen reader.

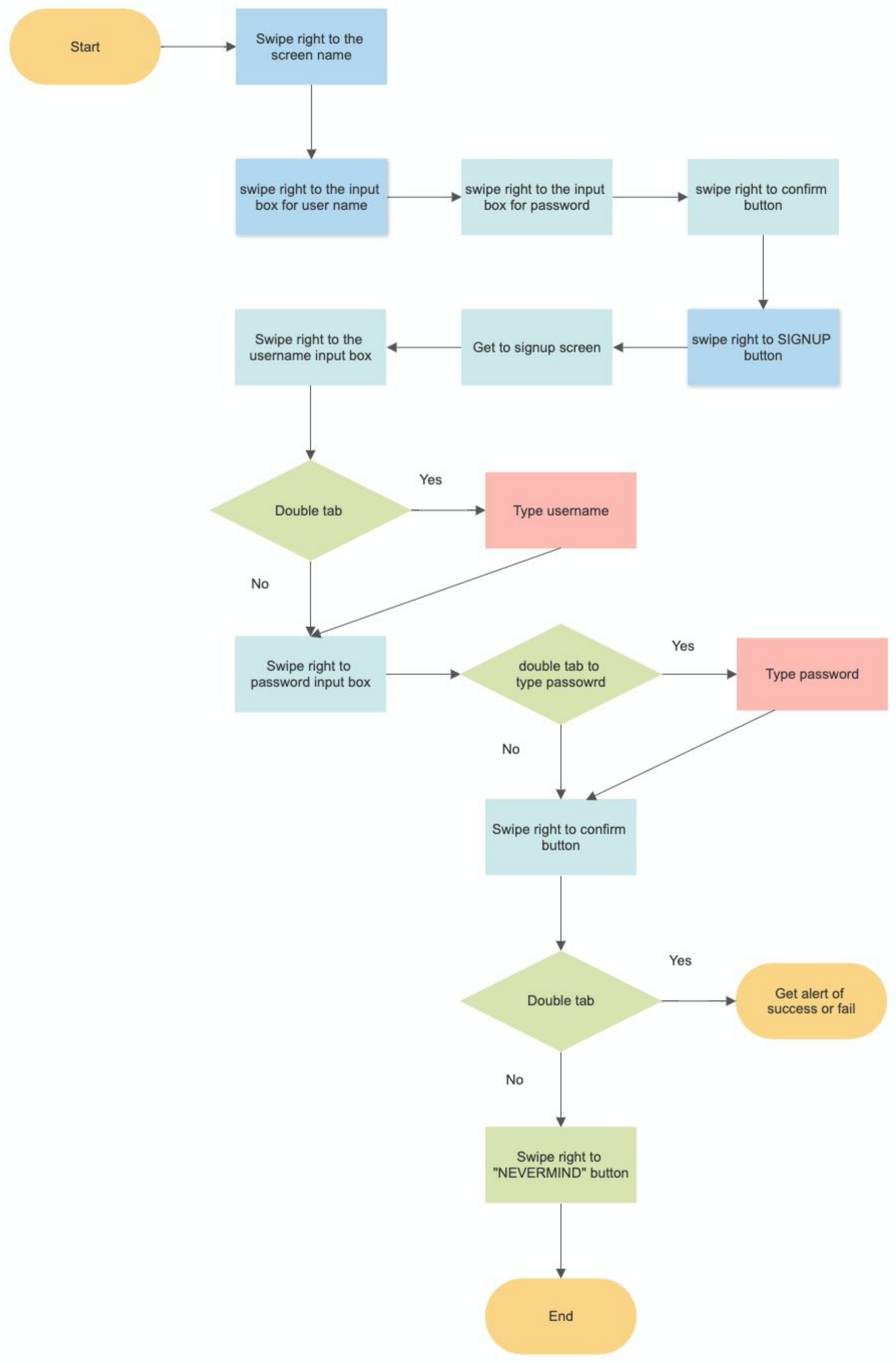
For the second app, I chose Facebook to test the signup and login screen, it works much better than my fitness app. For input fields asking username and password, the screen reader would directly notify the user that this current field could be tapped on and edited. It also numbered different fields that could be activated, such as editable areas and buttons. It informs me what to do with each active component, whether get text input, double tap, or hold to activate the functionality.

Step 2. Planning for Accessible Design (0.5 Point). In this step, you will choose <u>two</u> tasks supported by your React Native 1 α deliverable (e.g., sign up for an account) or your React Native 2 α deliverable (e.g., add an account for the current day) and map out how you expect users with blindness or severe visual impairments to interact with them given what you learned in Step 1. You can repeat the task you specified in Step 1. Specifically, you will create flowcharts of what components the user must interact with and in what order to perform the task. This activity will help you choose the right groupings for your React Native elements in

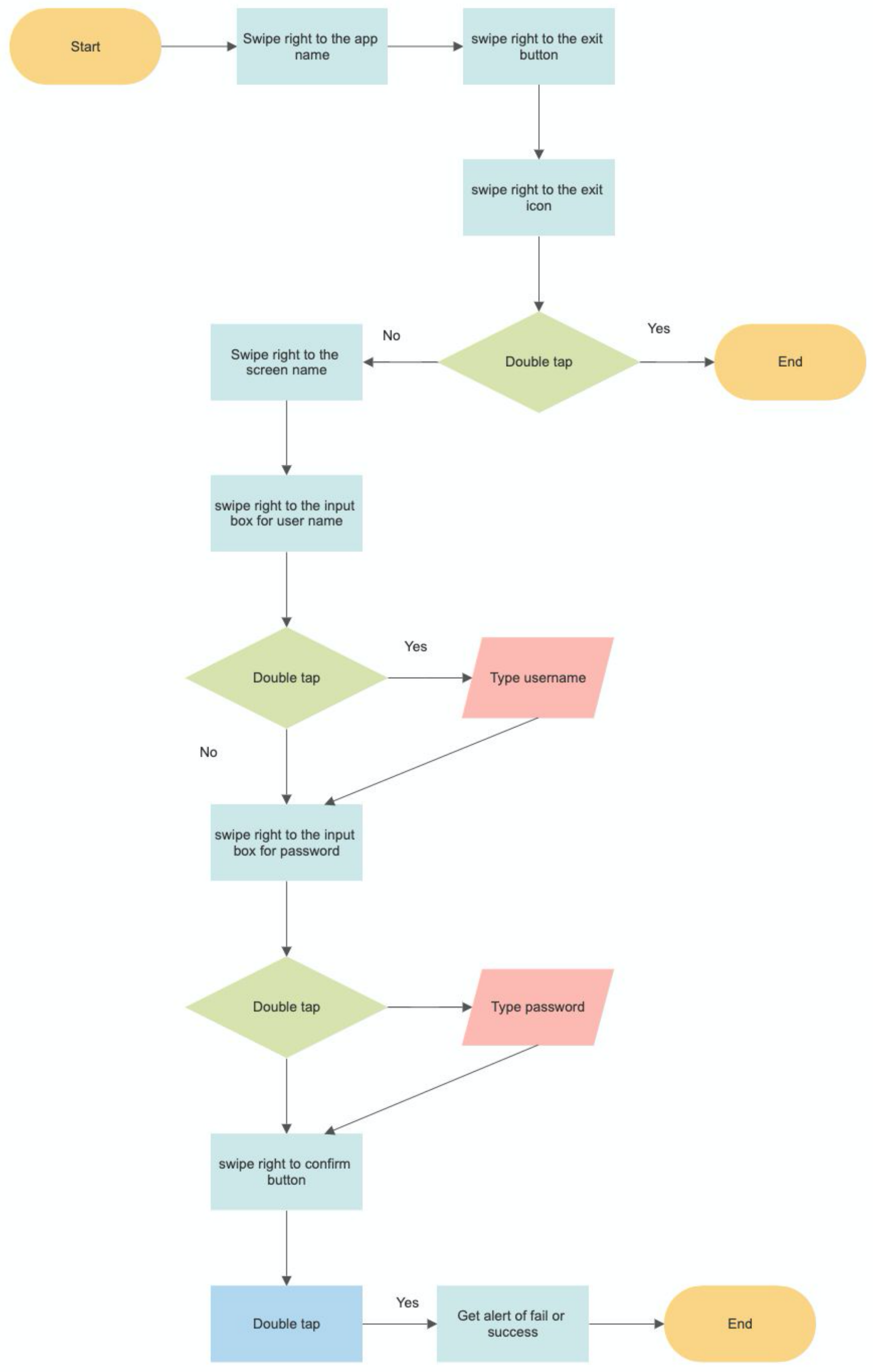
[1] You can learn more about iOS VoiceOver here: https://www.apple.com/accessibility/iphone/vision/
[2] You can learn more about Android TalkBack here: https://support.google.com/accessibility/android/answer/6283677?hl=en

order to define the accessibility features you will need. To generate flowcharts, you can use SmartDraw (using your NetID login) or free versions of other tools, such as LucidChart or Creatly.

Sign up:


Start
Swipe right to the screen name
swipe right to the input box for user name
swipe right to the input box for password
swipe right to confirm button
swipe right to SIGNUP button
Get to signup screen
Swipe right to the username input box
Double tab
Yes
Type username
No
Swipe right to password input box
double tab to type passowrd
Yes
Type password
No
Swipe right to confirm button
Double tab
Yes
Get alert of success or fail
No
Swipe right to "NEVERMIND" button
End

Log in:


Start
Swipe right to the app name
swipe right to the exit button
swipe right to the exit icon
Double tap
Yes
End
No
Swipe right to the screen name
swipe right to the input box for user name
Double tap
Yes
Type username
No
swipe right to the input box for password
Double tap
Type password
swipe right to confirm button
Double tap
Yes
Get alert of fail or success
End

Step 3. Specifying Accessibility Features (0.6 Point). This step will involve determining how to specify accessibility features for the components you included in your flowcharts in Step 2. For each component, you will write out how you will enable accessibility features using React Native Accessibility (review the accessibility properties), such as where accessibility features should be enabled, what labels and hints should be provided, what accessibility actions should be supported, and so 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.

For the LOGIN step, I will specify the usage of the two buttons, once for logging the user in and the other for switching to the signup screen. They will have accessibilityLabel and accessibilityHint notifying the purpose and providing extra information for the user to know what would happen if the buttons are clicked. And the two text input boxes will also need to have accessibilityLabel and accessibilityHint, as well as setting the accessibilityLiveRegion to be polite so that the user is confirmed about whether the input username and password are corrected. This could avoid situations of typo resulting in incorrect username or password that prevent user from successfully entering their information.

For the SIGNUP step, the username and password fields will have the same attributes as they do in LOGIN to make sure the user is confirmed about what they have typed in those fields so that after signing up they won't remember the wrong information because of typo in the signing up process. At the same time, the two buttons will keep accessibilityLabel and accessibilityHint to notify the user about their actions. If user decides to quit the app, the system will let know beforehand what would happen when user tap the button. Since we can only go to the SIGNUP screen through LOGIN screen, the button "SIGN UP" in the login screen is critical and should hint user beforehand that they would be directed to a new page.

Part 2: Implementation (0.8 Point)

The outcome of Steps 2 and 3 in Part 1 provides you with exact specifications for implementing the accessibility features into your code of the React Native application. In addition to carrying out these specifications in your code, you will also have to disable accessibility features for components that do not support your tasks and might be distractions for users with visual impairments. The deliverable of this part of the assignment is the code you will submit into GitHub Classroom.

Part 3: Testing & Demonstration (0.8 Point)

In this part of the assignment, you will perform the tasks you chose in Step 2 of Part 1 with the screen reader on and capture a video of your demonstration. You can use the iOS in-built screen recorder, one of the various screen recording options on the Android, or another device (e.g., your friend's phone, or a tablet computer) to record yourself demonstrating the tasks. Save this recording into your Google Drive, set permissions such that the video is viewable for anyone with the link, and include the link in your submission on Canvas. You can save two separate video files for the two tasks, or a single video file that demonstrates the tasks back to back. In addition to capturing your screen, screen recorders can also capture your voice, and you will be asked to provide narration along with your demonstration.

https://drive.google.com/file/d/1ML1KYEXCjZPPi-i9DpLRmOOMhwSsjRtk/view?usp=sharing

https://drive.google.com/file/d/1XYYo9FPZVeZgCDzgBRjas3JJobTJEQyo/view?usp=sharing