Create Accessible Apps Overview

In this code lab, I learned how to create accessible apps. Accessibility in Android apps is a set of design, implementation, and testing techniques that allows everyone to access the app including people with disabilities. Common disabilities Android device users may have include blindness, low vision, color blindness, hearing loss and restricted motor skills. This code lab shows how to create apps with accessibility in mind to make better user experience for both regular users and users with disabilities. Android provides several accessibility related features to help developers optimize app’s user interface to accommodate user with disabilities. This code lab taught me how to test the app for accessibility and to make changes to the XML layouts by adding attributes in order to make the app more accessible.

I also learned that adding accessibility to an app doesn’t necessarily require significant code changes. Accessibility features can be added to an app through attributes in the XML layout and through Android’s TalkBack feature to test an app for users with low vision. Android Studio can highlight missing attributes for accessibility in the layout. One of the accessibility attributes in Android is the contentDescription, which provides readable description of buttons and image views. Focus and focus order are used to provide the ability to navigate an app’s UI. Another useful attribute is the focusable attribute for ImageView. For EditText views, use android:hint instead of android:contentDescription.

Answer these questions

Question 1

Which of the following attributes should you add to ImageView and ImageButton elements to enable screen readers to describe the image?

* android:text
* **android:contentDescription**
* android:hint
* android:labelFor

Question 2

When should you add a content description to an ImageView or ImageButton?

* **When the image's role on the screen is solely decorative and does not provide any function or meaning.**
* When an image's dimensions are very small and the image is difficult to see.
* When the image is meaningful to the user in their use of the app.
* When the image is also a button.

Question 3

When do you NOT need to add a content description to a view element?

* When the view is a TextView or EditText.
* When the view is an ImageView that serves only a decorative purpose.
* When the view is a checkbox or radio button.
* When the view has an associated text view that includes with the android:labelFor attribute.
* **All of the above**.