

Summary and Checklist: Android Native Mobile App Accessibility

Summary

Native Android apps are easy to make accessible once you've learned the Android Accessibility API. Accessibility testing with TalkBack and a keyboard are easy ways to check if your app is accessible. Google's Accessibility Scanner app is a great testing tool that also checks for low-contrast text.

Accessibility testing and requirements have to be included in all stages of design and development, starting at the beginning in wireframes and user interface specifications. When accessibility is included early and tested often, it will be less work to develop accessible apps because most problems will be discovered before they reach QA testing. Always keeping accessibility in mind during design and development will ensure that users with disabilities will have an accessible experience when they interact with the application.

Checklist

UI Controls

Buttons

- Buttons MUST be keyboard operable and show keyboard focus visibility.
- Buttons MUST convey their accessible name, role, value, and state to TalkBack users.

Links

- Links SHOULD convey a meaningful link purpose.

Progress Bar

- Progress bars MUST convey their accessible name, role, and current value to TalkBack users.
- When progress spinners appear, they MUST be conveyed to TalkBack users either through focus management or screen reader announcements.

Slider

- SeekBar sliders MUST have TextView labels connected using `android:labelFor`.

Switch

- Switch controls MUST have `android:text` labels spoken to TalkBack users.

Notifications

Notifying Users of Changes

- TalkBack users MUST be notified of dynamic content changes either through spoken accessibility announcements OR focus management.

Time Limits

- User login sessions MUST NOT timeout the user or kick the user out without first providing an accessible timeout warning and session extension method.

Alert

- Alerts MUST be conveyed through TalkBack or grab the TalkBack user's focus.

Dialog

- Modal dialogs MUST trap keyboard and screen reader focus and MUST prevent focus of the grayed-out content.

Custom Elements

Accordion

- Accordion components MUST announce their accessible name and their expanded or collapsed state.

Autocomplete

- Autocomplete suggestions MUST be announced to TalkBack users when they appear and MUST be operable for TalkBack and keyboard-only users.

Carousels

- Carousels MUST have a method to stop their movement.

Rating

- Rating controls MUST convey their accessible name and current rating value to TalkBack users.

Sortable Table

- Sortable table column headers MUST convey their sorted state to TalkBack users.

Toggle

- Toggle buttons MUST convey their toggled state and accessible name to TalkBack users.

Summary and Checklist: Android Native Mobile App Accessibility

Form Labels and Validation

Labels

- Label text **MUST** be visible at all times.
- Label and instruction text **MUST** be programmatically associated with the input.

Group Labels

- Groups of related inputs **MUST** convey their group legend text to TalkBack users.

Checkboxes

- Checkboxes **MUST** convey their label text and checked state to TalkBack.
- Checkboxes inside a group **MUST** convey their legend text to TalkBack users.

Date Picker

- Native date picker controls **SHOULD** be used over plain text inputs or custom date picker controls.

Radio Buttons

- Radio buttons **MUST** convey their individual label, group legend, and checked state to TalkBack users.

Select

- Select dropdown controls **MUST** have a programmatically associated visible text label.

Text Fields

- Text fields **SHOULD** have the correct keyboard input type specified.

Validation Feedback

- Form validation error messages **MUST** be conveyed to TalkBack users when errors are displayed and when user is focused on invalid inputs.

Navigation

Page Titles

- Each page **MUST** have a meaningful title that describes its purpose.

Tab Panel

- Tab controls **MUST** convey their selected state to TalkBack users.

Menu

- Menus **MUST** be operable with both TalkBack and the keyboard, and focus order for both TalkBack and the keyboard **MUST** be logical.

Device Orientation

Landscape & Portrait Orientation

- The orientation of the user interface and content **MUST NOT** be locked to either landscape or portrait unless a specific orientation is essential for the functionality of the app.

Semantic Structure and Meaning

Tables

- Data tables **MUST** have a logical reading order, and data cells **MUST** convey row and column header text to TalkBack users.

Reading Order and Focus Order

- Reading order and focus order **MUST** follow a meaningful sequence for TalkBack and keyboard users.

Color and Contrast

Colors that Convey Information

- Color **MUST NOT** be the only method used to convey information.

Contrast

- Small text (under 18 point regular font or 14 point bold font) **MUST** have a contrast ratio of at least 4.5 to 1 with the background.
- Large text (at or over 18 point or 14 point bold) **MUST** have a contrast ratio of at least 3 to 1 with the background.

Audio and Video

Captions

- Videos **MUST** include captions for deaf and hard of hearing users.

Audio Descriptions

- Videos **MUST** include audio descriptions or a separate audio-described version for users who have vision disabilities.

Transcripts

- Audio files **MUST** include text transcripts for users who are deaf or deafblind.

Summary and Checklist: Android Native Mobile App Accessibility

Text Content

Language Settings

- The primary language of the app **MUST** be specified, and languages that differ from the primary language within the app **MUST** be identified.

Text Resize

- Text **MUST** be resizable at least up to 200% larger.

Images

Alternative Text

- Images **MUST** have a `contentDescription` unless they are decorative or redundant.
- `ImageButton` controls **MUST** have a `contentDescription`.
- Dynamic `ImageButton` controls **MUST** update `contentDescription` when image changes.
- Images that do not convey content, are decorative, or contain redundant content **MUST NOT** be given an alternative text equivalent and **MUST** be hidden from screen readers.
- Complex images **MUST** include full text alternatives or extended text alternatives.

Text in Images

- Images of text **MUST NOT** be used, unless the design is not possible using real text.