Mobile accessibility checklist

- Color
- Visibility
- Focus
- Text equivalents
- Handling state
- Orientation
- General guidelines

This document provides a concise checklist of accessibility requirements for mobile app developers. It is intended to continuously evolve as more patterns arise.

Color

- Color contrast must comply with WCAG 2.2 AA level requirements:
  - Contrast ratio of 4.5:1 for normal text (less than 18 point or 14 point bold.)
  - Contrast ratio of 3:1 for large text (at least 18 point or 14 point bold.)
- Information conveyed via color must be also available by other means too (underlined text for links, etc.)

Visibility

- Content hiding techniques such as zero opacity, z-index order and off-screen placement must not be used exclusively to handle visibility.
- Everything other than the currently visible screen must be *truly* invisible (especially relevant for single page apps with multiple *cards*):
  - Use the `hidden` attribute or `visibility` or `display` style properties.
  - Unless absolutely unavoidable, `aria-hidden` attribute should not be used.

Focus

- All activatable elements must be focusable:
  - Standard controls such as links, buttons, and form fields are focusable by default.
  - Non-standard controls must have an appropriate ARIA Role assigned to them, such as `button`, `link`, or `checkbox`.
- Focus should be handled in a logical order and consistent manner.

Text equivalents

- Text equivalent must be provided for every non-strictly presentational non-text element within the app.
  - Use *alt* and *title* where appropriate (read Steve Faulkner's post about Using the HTML title attribute for a good guide.)
  - If the above attributes are not applicable, use appropriate ARIA States and Properties such as `aria-label`, `aria-labelledby`, or `aria-describedby`.
- Images of text must be avoided.
- All user interface components with visible text (or image of text) as labels must have the same text available in the programmatic name of the component. WCAG 2.1: Label in name.
- All form controls must have labels (label elements) for the benefit of screen reader users.

Handling state

- Standard controls such as radio buttons and checkboxes are handled by the operating system. However, for other custom controls state changes must be provided via ARIA States such as `aria-checked`, `aria-disabled`, `aria-selected`, `aria-expanded`, and `aria-pressed`.

Orientation

- Content should not be restricted to a single orientation, such as portrait or landscape, unless essential. WCAG 2.1: Orientation

    - Examples of when an orientation is essential is a piano application or a bank check.

General guidelines

- An app title must be provided.
- Headings must not break hierarchical structure

```
<h1>Top level heading</h1>
<h2>Secondary heading</h2>
<h2>Another secondary heading</h2>
<h3>Low level heading</h3>
```

- ARIA Landmark Roles should be used to describe an app or document structure, such as `banner`, `complementary`, `contentinfo`, `main`, `navigation`, `search`.
- For touch events, ensure the following (WCAG 2.1: Pointer Cancellation):
    - The down-event should not be used to execute any part of the function;
    - Failing the above, *completion* of the function should be on the up-event, and a mechanism is available to abort the action before its completion or to undo the action after its completion;
    - Failing the above, the up-event should be able to undo any action that was triggered on a down event;
    - All the above may be violated if it is essential to trigger the action on the down event, usually to simulate real-world experiences or to provide real-time feedback. For example, game controls, piano keyboards, or virtual keyboards.
- Touch targets must be large enough for the user to interact with (see the BBC Mobile Accessibility Guidelines for useful touch target size guidelines).

> **NOTE:** The original version of this document was written by Yura Zenevich.

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:33:31 UTC