

Vision Impaired:

- Mild impairment: need large fonts, high-contrast colors, supplemented with audio feedback
 - Text no less than 11pts, ideally able to be magnified by 200% without losing functionality of the screen or content
 - A variety of visual cues, such as color, shape, size, etc.
 - For example, an error shouldn't just highlight the text field in red. Move the page to the location of the first error, include a sentence in bold that describes how to fix it, and/or use recognizable icons in addition to the red color.
 - Also keep in mind that mobile apps can be used outdoors, where the glare of the sun makes it even harder to see the screen
- Severe impairment: need screen-reader text-to-speech
 - Important to have information organized in a top-to-bottom, left-to-right manner
 - Descriptive (but not lengthy) alternate text for everything on the screen
 - If an image has a caption that the screen reader can also read, the alternative text should be different from it or left blank so as to avoid repetition.
 - Screen readers can be hard to use when the app plays a lot of music or sound effects
 - Types of screen readers:
 - Explore by touch: The user runs their finger across the screen and the screen reader describes what's underneath it. Double-tap to select
 - Linear navigation: The user swipes left and right on the screen to "read pages in a linear fashion, from top to bottom."
 - Can also navigate by landmark, such as skipping to headers if they are appropriately labelled.

Hearing Impaired:

- Closed captioning (any messages conveyed through audio need a transcription that can be read)
- Not relying on audio in the app's design
- Being able to change the volume of the app's media separate from the phone's volume
 - Volume slider, but should be easily accessible and in plain view

Motor Impaired:

- Generic input API that allows for specialized input devices to be used with it

- Support for keyboard shortcuts and mouse gestures are an important part of this. Allows for certain recognized patterns to invoke a programmed action, so that a customized accessibility input device can be used to navigate the app
- Speech recognition/dictation
- Gestures should be limited (one-finger swipe or a single tap rather than multiple in a pattern, etc.)
- Bounce key feature: Ignore rapid, repeated presses of the same key
- Design touch-targets with the size of a finger in mind
 - Average width of an index finger is 1.6-2cm (<https://mantadesign.com/accessible-mobile-app-design/>), so it's generally recommended to have buttons be at least 48x48dp on Android.
- Alternative means of data entry (such as choosing from menu options rather than typing something in)

Cognitively Impaired / Learning Disability:

- Rely more on symbols/icons for navigation
- Simple, short sentences (also helps with screen readers to cover content faster)
- A clear hierarchy of importance (having an important control at the top of its related content is much clearer than having it nested inside it)
- Minimal amount of information on each screen (limit the number of grids, tables, drop-downs, etc.)

To develop an accessible app, don't reinvent the wheel. Take full advantage of native accessibility features both to save work on the developer's side and to maximize the flexibility for the end-user.

- Android
 - Built-in screen readers
 - Google Voice Assistant (previously TalkBack)
 - Select to Speak
 - How to accommodate screen readers: <https://medium.com/mesmerhq/optimize-your-android-app-ux-for-screen-readers-454837bc4d30>
 - BrailleBack
 - Provides support for alternative, Bluetooth input devices (such as keyboards)
 - Technical Guides: <https://developer.android.com/guide/topics/ui/accessibility>
- IOS
 - VoiceOver

Guides on Accessible UI Design:

- <https://blog.usablenet.com/mobile-app-accessibility-techniques-for-inclusive-design-part-1>
 - Supporting a wide range of device screen sizes so that text appropriately wraps around rather than shrinking or getting cut off
 - Fields should go below their labels, not next to them
 - There should be inactive space between multiple small touch targets (e.g. buttons)
 - Button placement should consider all ways a phone might be held (left or right handed, limited thumb movement, etc.)
 - Repeated layout should be consistent (e.g. the order of items in a menu bar at the top of the screen. If the screen is smaller and the menu collapses, the order should still be the same when expanded)
- <https://material.io/design/usability/accessibility.html>
 - Lots of illustrations of good and bad accessible design

Avoid using CAPTCHA

Further Reading:

- Web-Content Accessibility Guidelines (WCAG): <https://www.w3.org/WAI/standards-guidelines/wcag/>
- “Ultimate Guide to Accessibility Testing”: <https://info.usablenet.com/accessibility-testing>
- Mobile Accessibility (summary of everything from big-picture thinking to technical implementation): <https://www.shopify.com/partners/blog/mobile-accessibility>
- <https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/website-app-accessibility-guidelines/>
- <https://medium.com/oberonamsterdam/how-to-create-an-accessible-app-and-why-you-should-5493f41f8bdb/>