Zachary Bachrach
Product Design - Societal Impact

Accessibility Criteria and Current Approaches for Designing User-Friendly Interfaces

Designing software applications and hardware devices with user accessibility in mind is very important due to the large role that devices with software play in our lives. In this paper, we will identify the core groupings of accessibility that a designer/developer should take into account, we will outline how those groupings can be addressed in implementation, and we will study current applications with an emphasis on reviewing their accessibility and functionality.

The following are core groupings of accessibility that a designer or developer should take into account when designing an application or a hardware device:

- Visual impairment: This can refer to blindness, color blindness (otherwise known as Color Vision Deficiency (CVD), or short/near-sightedness.
- Auditory: Related to the loss of hearing, hearing impairment, or even complete deafness.
- Mobility: This is related to the loss of movement, lack of precision, or difficulty in using fingers, hands, and arms to complete tasks. Conditions can include ALS and cerebral palsy.
- Cognitive and Learning disabilities: This can include disabilities such as dyslexia, dementia, and attention deficit hyperactivity disorder (ADHD).
- Seizures: Epilepsy can be triggered by bright flashing light and rapid motion on-screen.

Now we will outline how to address these groupings in implementation.

CVD manifests in a variety of ways, the most common being the inability to discern between green and red colors. Figure 1[1] outlines some of the types of CVD and what the visible color palette is.

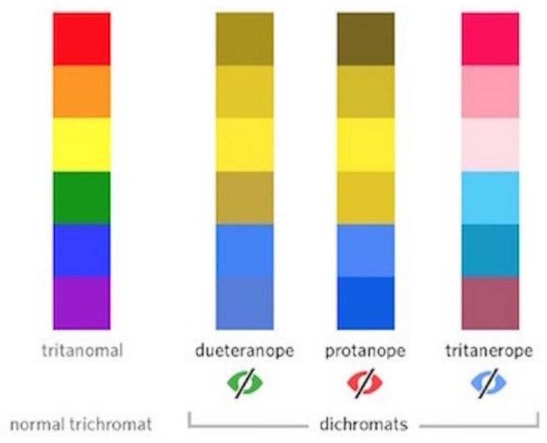
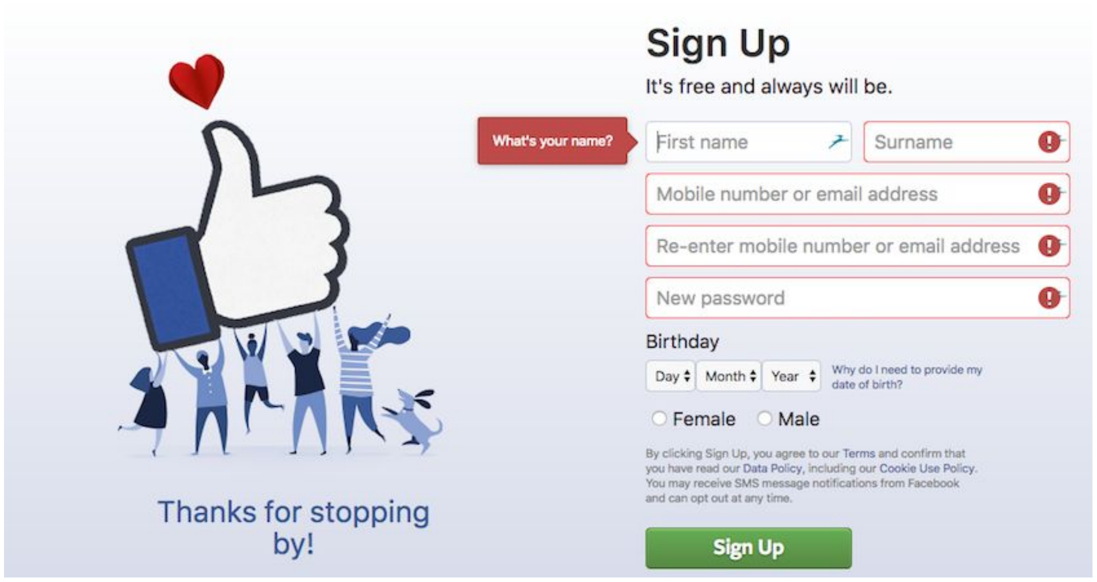


Figure 1. The leftmost bar is a non-CVD color palette, the three rightmost bars are the visible color palettes of people who have those respective CVD types.

To address CVD, it is important to use both colors and symbols when designing a user interface. This means not just using colors, but symbols and descriptive words. The example of the Facebook registration page (Figure 2) is very good about using a combination of colors and words, so that a person with CVD does not lose any functionality when using the interface.



Sign Up

It's free and always will be.

What's your name?

First name | Surname

Mobile number or email address

Re-enter mobile number or email address

New password

Birthday

Day | Month | Year — Why do I need to provide my date of birth?

Female Male

By clicking Sign Up, you agree to our Terms and confirm that you have read our Data Policy, including our Cookie Use Policy. You may receive SMS message notifications from Facebook and can opt out at any time.

Sign Up

Thanks for stopping by!

Figure 2. Facebook's registration page uses a combination of colors and text descriptions to ensure the user doesn't lose functionality if they have CVD.

It is prudent to use textures to differentiate between colors or objects you want to separate (Figure 3). As well, making sure the text is large and unobstructed can help people with less than perfect vision, and simplicity makes pages easier to read.

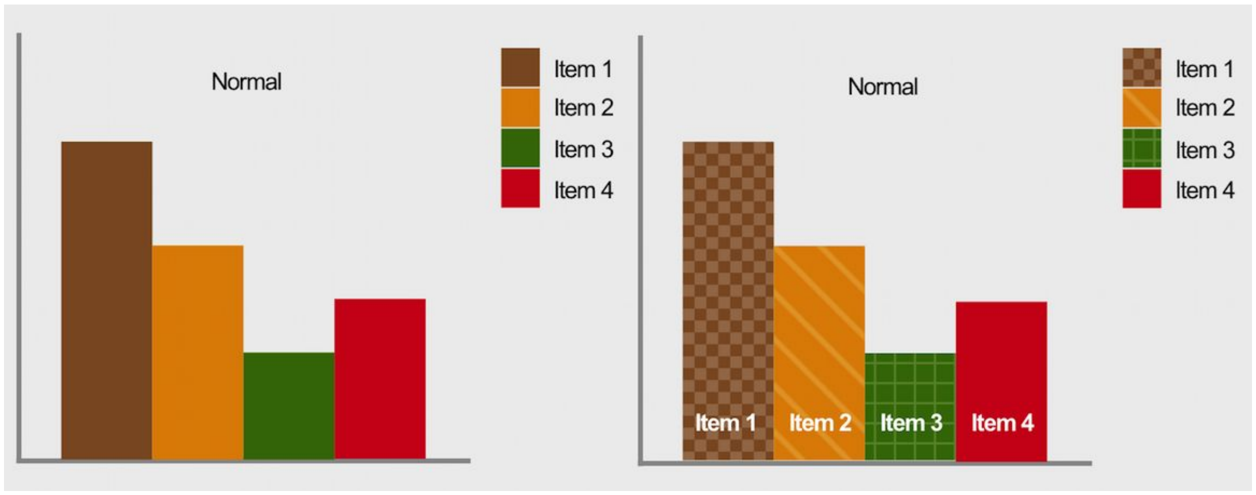


Figure 3. An example of using textures to differentiate between colors to make a chart more accessible to people with CVD.

In addition, iOS developers can use the VoiceOver[3] to allow users to interact with objects in the app if they cannot see it. Google offers Voice Access, which allows users to speak commands to the device.

Next is auditory impairments. When developing an application, iOS developers should consider using the AVFoundation[8] framework for captioning video playback. There are also third party services who offer transcription services. Google Cloud[8] and AWS[9] both offer their own speech-to-text services. In fact, my term project is focused in part on using the speech-to-text API from Google to convert video files to transcripts via a speaker diarization process. In addition, developers should keep in mind that audio cues can go unheard, and should only be one of multiple redundant ways to alert or guide a user through the interface. Functionality should not be lost because of a muted phone or a hearing impairment.

Now we move to mobility impairments. This requires consideration for users who may not be able to touch small targets on a screen, or whose muscles tremor when tapping. Fortunately Google has developed accessibility services Switch Access and Voice Access to help with these issues on android apps.[2] Switch Access allows users to press physical hardware buttons on their smartphone to select different on-screen buttons. Designing apps where these services are effective requires following certain rules. One rule is to generally keep elements on the screen organized from top to bottom, as that is how a user will navigate the screen with their eyes. Make sure to have low-density layouts and large buttons. For iOS devices, there is VoiceOver as mentioned previously. Make sure alt-text is succinct and descriptive, as this will be important to users who cannot see and rely on the audio for navigation through the interface.

Moving onto cognitive impairments, iOS offers the Guided Access Protocol[4] which is designed to guide people with autism or attention/sensory challenges to stay focused on the task. The protocol allows a teacher, parent, therapist, etc. to stay on an app, restrict the home button, disable certain areas of the screen. This can be implemented by the developer using the Guided Access Protocol. For Android, Google released Action Blocks[5] which can help a user by making common actions easy. This layout and functionality should be considered by a developer when creating their interface.

On the topic of seizures, bright flashing lights and rapid movements on your application should be avoided. On iOS, the UIAccesibility[6] API can be used to reduce motion (among other things). Blinking lights are different from flashing lights - blinking refers to a distraction, while flashing (according to W3) is content that occurs more than 3 times in a second and is bright enough. It is prohibited by Section 508[11] to have larger than 3Hz (flickers per second) and lower than 55 Hz on web applications. For further specific details on the flashing laws and recommendations, a good source is the Mozilla Accessibility page on seizure disorders.[12]

Now we will look at an example web application and see how it fares in each of the accessibility categories. For this test, I chose Deque[13], a digital accessibility company. Their website is a great example of making sure that as many people as possible can access it successfully with minimal impairment (Figure 4). First off, the color scheme is a good choice in that it uses blues and colors with good contrast, keeping in mind people with CVD. The text is large and bold, and the menu bars are easy to navigate, have good contrast, and have large buttons (Figure 5).

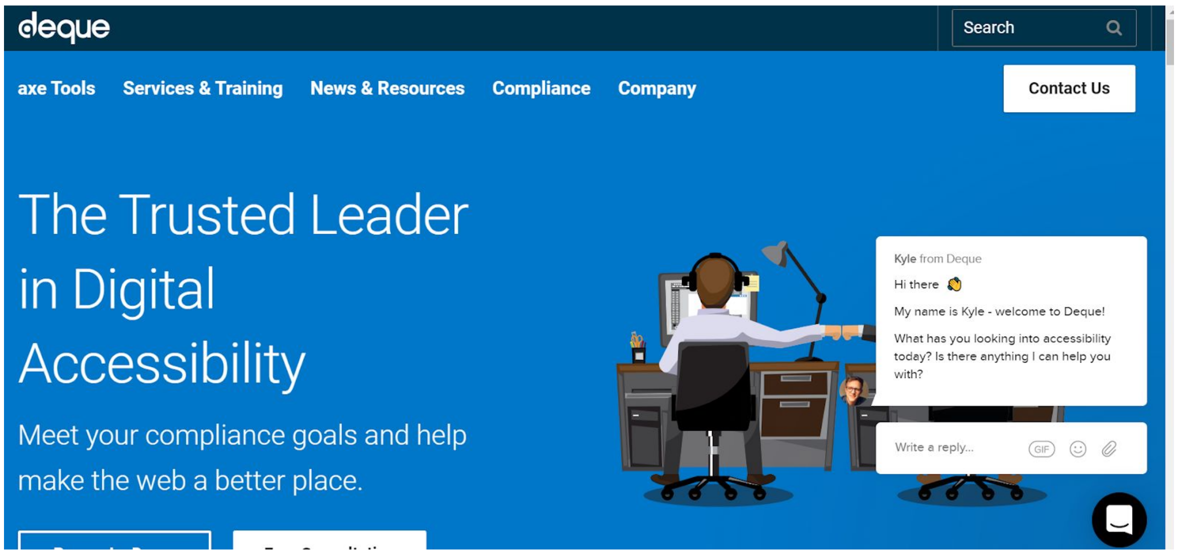

deque
Search
axe Tools
Services & Training
News & Resources
Compliance
Company
Contact Us
The Trusted Leader in Digital Accessibility
Meet your compliance goals and help make the web a better place.
Kyle from Deque
Hi there
My name is Kyle - welcome to Deque!
What has you looking into accessibility today? Is there anything I can help you with?
Write a reply...

Figure 4. Deque Systems web application is a good example of a clean user interface which has minimal accessibility issues.

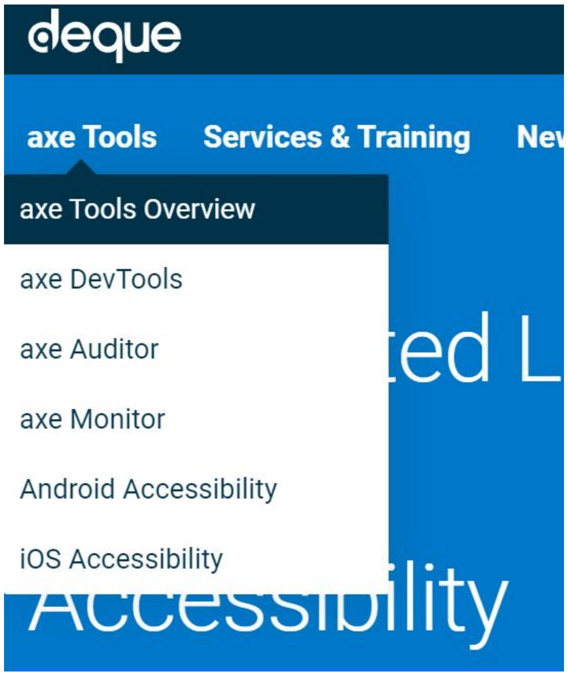


Figure 5. The menu bar has high contrast and large buttons.

I tried using the tab button on my computer to navigate through the site, and it is very easy to do this. All the non-text elements have alt-text, and navigational aids for keyboard only users. I tried the website on mobile, and the menu bar was adjusted to fit my screen and be simple to tap with large buttons. I checked the Play Store for any mobile offerings that Deque Systems might have, and they have one offering, the Accessibility Engine (axe) for Android. The app is clean and reflects a good design practice to accommodate as many accessibility issues as possible as outlined in the prior parts of this paper. In fact, the app has a 4.4 star out of 5 star rating in the store with over 5k downloads, with many reviews saying that it is simple, and really easy to use. For a company that specializes in web-accessibility consulting, they definitely have their act together!

In conclusion, the main categories of accessibility when it comes to designing software and hardware devices are visual, auditory, mobile, cognitive, and seizures. It is not only legally important to follow accessibility guidelines when designing your interface, but also good for the success of your product. Many people are affected by some form of accessibility, and it is surely in the developer’s best interest to think about some of these things during the design process.

Works Cited

1. https://usabilla.com/blog/how-to-design-for-color-blindness/
2. https://support.google.com/accessibility/android/answer/6122836?hl=en
3. https://developer.apple.com/accessibility/ios/
4. https://developer.apple.com/documentation/uikit/uiguidedaccessrestrictiondelegate
5. https://support.google.com/accessibility/android/answer/9711267?hl=en
6. https://developer.apple.com/documentation/objectivec/nsobject/uiaccessibility
7. https://developer.apple.com/av-foundation/
8. https://cloud.google.com/speech-to-text
9. https://aws.amazon.com/transcribe/
10. https://developer.mozilla.org/en-US/docs/Web/Accessibility/Seizure_disorders
11. https://www.section508.gov/content/guide-accessible-web-design-development#flashing
12. https://developer.mozilla.org/en-US/docs/Web/Accessibility/Seizure_disorders
13. https://www.deque.com/