Mobile Accessibility Test Approach

This document gives quick summary about the approach followed to test mobile applications accessibility.

### Audience of Mobile Applications Requiring Accessible Applications

Below are the mobile users considered as part of accessibility test approach:

* Partially sighted (especially when using a smaller screen)
* Color blindness
* Hearing loss
* Dyslexia and other cognitive issues that can affect the readability of a site
* Arthritic and other finger joint movability issues
* ADA or a combination of these issues which makes addressing accessibility even harder!

Some users will use items like Bluetooth keyboards, but does this mean a tester also needs to use extra hardware when testing a mobile site? For requiring other hardware, you are more likely to need a range of mobiles with different operating systems than extra hardware like a Bluetooth keyboard. This is due to the fact hardware used for accessibility will highlight the same issue in the software whereas different operating systems will have different screen readers and versions inbuilt.

The issues found with different screen readers will vary but the hardware will have the same common issues, e.g. the page order will match the order of items selected with left to right swipes using a screen reader. The minimum requirements to test for mobile accessibility should be an Android and an iOS mobile (as they each use different screen readers and browsers/applications). I would advise to not use emulators for testing mobile applications and websites, they are a good starting point but do not show how the hardware will work with gestures.

Some will say you should not look at the screen to do true testing like an end user, but I believe being able to see the items helps as you can tell if a label shown is not read or given focus by the screen reader.

### Accessibility Tools

To help you understand the requirements of testing web sites and applications for accessibility, it helps to understand the tools required. I am going to go through the main points and how to use the tools to cover mobile accessibility testing.

#### Screen Readers

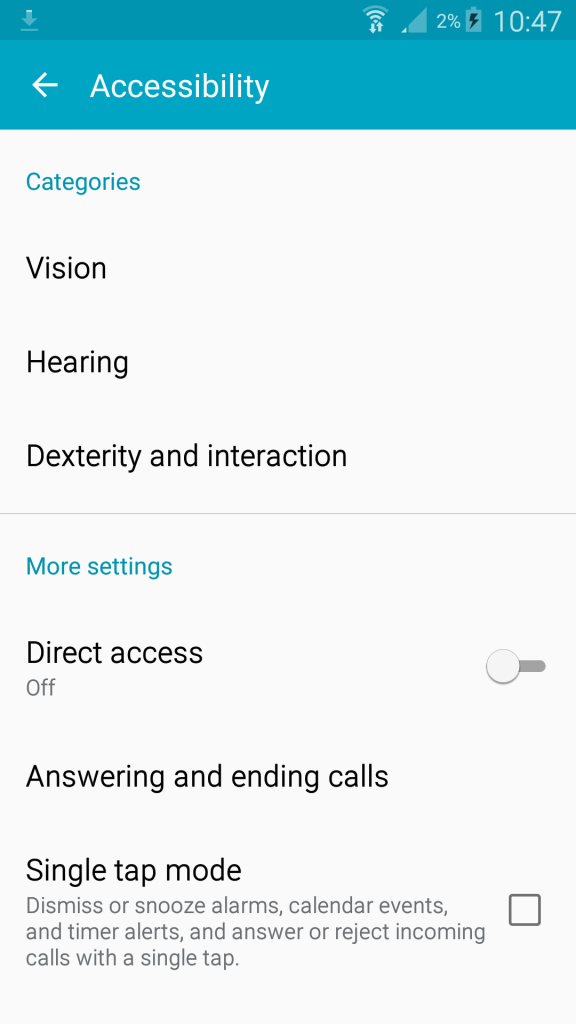
The main accessibility tool most testers seem to be aware of is screen readers. On desktops, you often need to use a paid for a tool like JAWS, but luckily on mobile, you have free screen readers built into the operating system. A screen reader is a tool used to read what a sighted person will see on the page. The screen reader for a phone is often activated via the phone settings as inbuilt to the phone operating system.

#### Using a Screen Reader:

When you start to do the testing, the screen reader testing is normally the most time-consuming and challenging part of mobile accessibility testing, but it will also highlight bad practices and bugs on both desktop and mobile.

For instance, the screen reader will read out the form controls in a certain pre-defined order. It will become obvious via the screen reader whether they are structured correctly. A big difference between the desktop and mobile accessibility testing is the screen reader navigation. The desktop screen readers require keyboard shortcuts to navigate the page, but mobile screen readers can connect via a Bluetooth keyboard or specific finger gestures.

When testing with a screen reader make sure items shown in the page are read in a meaningful way. Images have a meaningful label, and/or if decorative are not read or given focus. When navigating forms, the fields have meaningful labels and errors are linked to the appropriate fields. Basically, everything you can see must be portrayed in a meaningful way via the screen reader.



For Android phones, go to Settings > My Device >Accessibility > Vision (new to Android 5.0) > TalkBack > Turn TalkBack on. All gestures for TalkBack require only 1 finger. To get you started here are the basics I would use:

* Touch or move around the screen – speak and select the item under your finger
* Swipe right or down – move to next item on screen
* Swipe left or up – move to the previous item on screen
* Double-tap – select focused item
* Swipe up then down – move to the first item on screen
* Swipe down then up – move to the last item on screen
* Swipe right then left – scroll forward on page or move slider
* Swipe left then right – scroll back on page or move slider
* Swipe up then left – Home button
* Swipe down then left – Back button



For iOS phones, go to Settings > General >Accessibility > VoiceOver > Turn VoiceOver on. VoiceOver does require up to 3 finger gestures, and although some are similar to Android, there are enough differences to cause some difficulty when switching between the screen readers. The basic gestures I would use for testing are:

**1 finger gestures:**

* Touch or move around the screen – speak and select the item under your finger
* Tap 2 times fast – activate the selected item
* Swipe left or right – selects the previous or next item
* Swipe up or down – move to the selected rotor option

**2 finger gestures:**

* Swipe up – read all from the first object in selected area
* Swipe down – read all from selected item
* Scrub back and forth – go back, cancel, close pop-up
* Pinch out/in – select/deselect
* Rotate to open the web rotor and select rotor items (see the last point for 1 finger gestures)

**3 finger gestures:**

* Swipe left or right – move to previous or next horizontal screen/page
* Swipe up or down – move/scroll down or up

#### 

#### 

#### Colour Ratio Analysers

Another essential tool is a contrast ratio analyser, however, this will be used on the desktop. The purpose of this is to ensure the colour contrast is readable for people with a variety of issues, not just colour blindness. For instance, a low colour ratio can affect people with poor visibility, and dyslexia, often not taken into consideration when checking the colour ratios.  
First you need to take screenshots of the screen from a phone, upload or send the image to somewhere accessible from a desktop and then use a validator to check there is enough contrast between two adjoining colours. I use Paint to pick the colour, then 2 websites: an RGB to Hex converter (as Paint returns the RGB colour and the analyser requires hex colours) and then a colour contrast analyser like [Luminosity Contrast Ratio](http://juicystudio.com/services/luminositycontrastratio.php).

#### HTML Validator

HTML validators only work on the desktop so when testing a mobile website. HTML validators are often used to check a web site to make sure the semantics are correct in the code. There are many available as browser add ons, e.g. [WAVE for Firefox](https://addons.mozilla.org/En-US/firefox/addon/wave-toolbar/). I recommend to use these as a starting point, but not as the definitive answer for an accessible site. In some cases the validator will raise issues that are not failures. For instance, they might highlight duplicate IDs defined in the HTML but one “duplicate” is in the HTML comment for the declaration of a variable or the code highlighted as failing is not active so not in scope to be accessible. Sometimes due to these errors, it is advisable to ask a developer to help with reviewing the results if you are not sure on them.

#### Inbuilt Accessibility Options

Accessibility options like zooming and invert colours are available on most mobiles. Most of the options are not required for audits. For instance, the zoom checkpoint in WCAG 2.0, 1.4.4, asks the tester to zoom text without using assistive technology. This means you must be able to zoom the page without using an accessibility option but to be able to zoom the page using the normal pinch action. It is like using a third party magnifying tool, if that is the only way to zoom the page it fails the checkpoint as that is requiring assistive technology and not using the default browser zoom.

#### WCAG 2.0 Checkpoints

I have also mentioned the most important tool for accessibility testing, the [WCAG 2.0 checkpoints](http://www.w3.org/WAI/WCAG20/quickref/). These are guidelines written by W3C and used by most companies and Countries as the standards you must adhere to. In America, there is [Section 508](http://www.section508.gov/content/learn/standards/quick-reference-guide), which almost completely matches WCAG 2.0 to level AA but missing 3 items. Therefore I recommend making sure all audits comply to WCAG 2.0 AA. You can try to meet level AAA but this is extremely difficult and not seen as a legal requirement for most countries. I have heard a few arguments of how the WCAG 2.0 checkpoints do not apply to mobile testing, but I believe if you use common sense they do apply. This has only recently been confirmed by the W3C too.

#### Checklist of Tests to Cover

* Screen reader testing: as mentioned before the way you test this on a mobile will vary from how you test on the desktop
* Zooming the site/application: this is important to a user with poor sight and those that may have difficulty moving their fingers to select a button. A larger button is easier to select. This is the same for desktop and mobile, but again using different methods. On a desktop browser you have the zoom feature, but on a mobile you have the pinch movement.
* Colour ratios: most testers verify these with a toolbar on the desktop, but I recommend using screenshots to get the actual colours shown and validating them as what is programmed in is not always what is seen. This applies to both desktop and mobile. When you have the screenshot, use the paint tool on items that look too similar in colours to grab the colour used (e.g. grey text on a darker grey background). Then convert the RGB colours to hex to work out the ratios with a tool like [Luminosity Contrast Ratio](http://juicystudio.com/services/luminositycontrastratio.php).
* Readability of the site: make sure the text is not overly complicated and easy to understand. Also make sure the text is easy to read, especially if the background is an image and/or using gradients. This is the same for both desktop and mobile testing.
* Navigation: make sure navigating the site or application is consistent, logical, structured, descriptive and not over complicated. This really applies to both mobile and desktop testing. To check the navigation make sure links are descriptive are not just “click here” ones, and easy to understand. Also if there is a mechanism, like menu items, they allow you to navigate to all pages in the site in a similar way.

### Conclusion

Mobile testing for accessibility does not need to be a big task and can be very similar to the requirements of desktop testing. There are overlaps a desktop site versus what seems to be the same on the mobile can have completely different bugs when you test both. With smart and considerate planning you can probably easily cover most accessibility tests through an effective and broad functional testing approach.

### About The Author

[Helen Burge](https://uk.linkedin.com/pub/helen-burge/23/513/99b) has a Business Information Technology degree from Bournemouth University and graduated in 2003.  After leaving university she worked in testing and part of all her roles involved accessibility testing due to her fascination at how unappreciated the need is to make websites and applications accessible.  Helen worked closely with developers to help learn the most effective way to find and report issues.  From this, she built a unique test methodology to make sure accessibility bugs are not subjective but make sure any issues raised are reproducible with actionable items.  This approach has ensured the team can resolve issues and understand the impact on the end users.

# Dependency on other applications

This section lists down the list of all applications and their latest version supported.

|  |  |  |
| --- | --- | --- |
| Application Name | Latest Version Supported | Earliest version supported |
| Orchestrator | 4.0.0.5 | 4.0.0.4 |
| EBSP | 4.0.0.5 | 4.0.0.3 |

Note: Latest code is available in SVN

While using phone with Inverted Colors active:

· The text that is from the user contrasts well against the background color

· The colors make it difficult to see anything on the page except the CalStateLA logo at the top of the page and the small map on pg. 1

While looking at the page with inverted color inactive:

· Yana’s replies contrast well with the background.

· User’s entry does not contrast well against the yellow background

· The microphone at the bottom right of each page contrasts well and is very visible

· The menu icon at the top left contrasts well against the background

· Text is small and a bit difficult to read

· On the third page, the background color right under Term/GPA/etc. causes the text to not be visible, difficult to read

· Can see the text edit area at the bottom of the page clear against the background, it is difficult to read due to the text side

· Can see the icons on the first and second page that include the thumbs up and thumbs down due to good contrast with the background

· Can see the search button on the top right corner due to good contrast.