

# VA.gov Mobile Accessibility Testing Plan

The VA.gov mobile app development team considers accessibility to be a priority requirement in the design and implementation of the app. The purpose of this document is to outline:

1. The requirements that will guide the the VA.gov Mobile App build
2. The accessibility tools and materials that will be used during testing
3. The testing protocol and steps that will be taken when reviewing the accessibility capabilities in the app

## Section 1: Accessibility Requirements & Approach

The following items are based on WCAG 2.0 and Section 508 accessibility standards, organized by product function. Each item description reflects both the requirement and the referenced standard, along with the corresponding implementation approach.

- Items that have “**Design**”/ “**Designs**” bolded reflect standards that must be facilitated by the design workstream
- Items that have “**Programmatically**” bolded reflect standards that focus on technical implementation
- Applicable WCAG 2.1 requirements are also included to reference best practice and may be followed in cases where added implementation scope is minimal.
- Accessibility standards that relate to non-existent functionality in the app are marked N/A. In the event that the app’s design changes to include relevant functionality, these standards will be candidates to prioritize as requirements.

Category	WCAG / 508 Section	Implementation Approach
Color	<b>Color contrast</b> (1.4.3_AA) (1.4.6_AAA)	<b>Designs</b> for the the app should consider the following:  Normal fonts (< 19px) <ul style="list-style-type: none"><li>• <b>AA compliant: 4.5:1</b></li><li>• AAA compliant: 7:1</li></ul> Large fonts (19px+) <ul style="list-style-type: none"><li>• <b>AA compliant: 3:1</b></li><li>• AAA compliant: 4.5:1</li></ul>
	<b>Color</b> (1.4.1_A)	<b>Designs</b> for the app should not use color as the sole conveyance of information.
	<b>Non-text contrast</b> (1.4.11_AA)	(2.1 requirement) <b>Designs</b> of UI components and graphical objects in the app designs that are used to understand the content should have a contrast ratio of at least 3:1.

Headings & Navigation	Heading levels (1.3.1_A)	<b>Designs</b> for each page should have only, and at least one h1 on the page. There are no skipped heading levels. (i.e. the headings don't jump from h1 to h>). <b>Programmatically</b> specify ARIA level for all text headings in React Native.
	Info and Relationships (1.3.1_A) (502.3.7) (502.3.8)	Leverage React Native accessible components so that information, structure, and relationships conveyed through presentation can be <b>programmatically</b> determined or are available in text.
	Tables (1.3.1_A) (502.3.3)	<b>N/A</b> - Assumption: Not seeing complex tables in the designs. If we need to pull in a fancier table than what we can build with react native <i>list views</i> , then we may need to pull in a third party library that is accessible and meets the designs.  All tables should be <b>programmatically</b> identified and the table heading for applicable rows and columns should be announced.
	Lists (1.3.1_A)	Leverage React Native accessible list component so lists can be <b>programmatically</b> determined.
	Page title (2.4.2_A)	Each page on the app should <b>programmatically</b> have a unique title that is descriptive and differentiated.  React Native may provide libraries that facilitate this. Screen reader should provide a breadcrumb like page read: "[level 3]   [level 2]   [level 1]" (e.g. Digital Foundry Product Designer page title: "Digital Product Designer   Careers   Digital Foundry").
	Headings and labels (2.4.6_AA)	Headings and labels <b>design</b> should describe topic or purpose and be <b>programmatically</b> determined as such.
	Navigation consistency (3.2.3_AA) (c205.2.2)	<b>N/A</b> according to section 508 (c205.2.2) <b>Designs</b> for navigation mechanisms that are repeated across several pages should be consistent. (Color, icon, text, location, etc.)
	Sensory characteristics (1.3.3_A) (502.3.8)	<b>Designs</b> for pages should not require sensory characteristics such as shape, color, size, visual location, orientation, or sound to be understood or operable.
	Bypass Blocks (Skip nav) (2.4.1_A) (c205.2.2)	<b>N/A</b> according to section 508 (c205.2.2) and no such nav exists in designs.
	Link purpose in context (2.4.4_A)	<b>Design</b> so the purpose of each link can be determined by its text (e.g. " <a href="#">Schedule your appointment now</a> ") or the context it's surrounded by. (e.g. "There are a lot of <a href="#">ways to stay active</a> during the pandemic.") " <a href="#">Click/tap here</a> " is a huge no.
Non-text content & media	Multiple ways to reach each page (2.4.5_AA) (c205.2.2)	<b>N/A</b> according to section 508 (c205.2.2) since this is a non-web format.
	Text alternatives (1.1.1_A)	<b>Content / API</b> should provide text alternatives for all non-text content
	Text alternatives for audio-only and video-only content (1.2.1_A)	<b>N/A</b> - no such content in the current designs. <b>Content / API</b> should provide text alternatives for all non-text content

	<b>Audio descriptions</b> <b>(1.2.5_AA) (503.4.2)</b>	<b>N/A</b> - no such content in the current designs <b>Content / API</b> for meaningful visual content should be <b>programmatically</b> described
	<b>Video &amp; live video captions</b> <b>(1.2.2_A &amp; 1.2.4_AA)</b> <b>(503.4.1)</b>	<b>N/A</b> - assumption: use native audio/video player from OS if needed <b>Content / API</b> should provide captions for pre-recorded and live content should be offered
	<b>Audio controls</b> <b>(1.4.2_A) (503.4)</b>	<b>N/A</b> - assumption: use native audio/video player from OS if needed <b>Design</b> so that all audio content has a mechanism to start, stop and adjust volume independent of the system
	<b>Pause, stop, hide - Video &amp; animation controls</b> <b>(2.2.2_A)</b>	<b>N/A</b> - assumption: use native audio/video player from OS if needed <b>Design</b> so that all media content has a mechanism to start, stop and hide
	<b>Flashing content</b> <b>(2.3.1_A)</b>	<b>Designs</b> should not include content or UI elements that have more than 3 flashes per second max. Avoid flashing content altogether if possible.
	<b>Label in name</b> <b>(2.5.3_A) (502.3.6)</b>	For UI components with a visual label, make sure the label is <b>programmatically</b> available in the beginning of the name of the component.
<b>Forms &amp; inputs</b>	<b>Identify Input Purpose - Form labels and instructions</b> <b>(1.3.5_AA &amp; 3.3.2_A)</b> <b>(502.3.6) (502.3.9) (502.3.12)</b>	Leverage React Native accessible form components so the purpose of each input field collecting information about the user can be <b>programmatically</b> determined.
	<b>Form error identification</b> <b>(3.3.1_A)</b>	Leverage React Native accessible form components so any errors in filling out a form should be <b>designed</b> descriptive, clearly visible and read <b>programmatically</b> .
	<b>Input triggers</b> <b>(3.2.2_A)</b>	<b>Designs</b> for the app when changing an input should not cause major changes in page content
	<b>Form error suggestions</b> <b>(3.3.3_AA)</b>	<b>Designs</b> for the app should include error messages that include suggestions on how to fix the error. Leverage React Native accessible form components so suggestions can be read <b>programmatically</b> .
	<b>Legal, financial and data error prevention</b> <b>(3.3.4_AA)</b>	<b>Designs</b> for app should enable users to be able to reverse, check or confirm inputs for important data
<b>Text content</b>	<b>Meaningful reading order</b> <b>(1.3.2_A)</b>	<b>Designs</b> for app should make sure everything is read in an order that makes sense and is <b>programmatically</b> read.
	<b>Images of text</b> <b>(1.4.5_AA) (502.3.8)</b>	<b>Designs</b> should not use images that are purely text, unless it's visual style cannot be accomplished via styled text. (e.g. logo text)
	<b>Text spacing</b> <b>(1.4.12_AA)</b>	<b>N/A</b> - 2.1 Requirement, but still best practice. <b>Design and set up text styles</b> so that: Line height/ spacing at least 1.5 times the font size Paragraphs at least 2 times the font size Letter spacing at least 0.12 times the font size

		Word spacing at least 0.16 times the font size
	<b>Text Resizing (1.4.4_AA)</b>	<b>Design</b> so that it's possible to resize content significantly, and ideally to the max device size. <b>Programmatically</b> , all font should scale with the font scaling the user has set in the OS.
<b>Miscellaneous</b>	<b>Content on Hover or Focus (1.4.13_AA)</b>	<b>N/A</b> 2.1 Requirement <b>Programmatically</b> ensure that where content is suddenly visible, then not visible (e.g. a tool-tip) newly shown content is dismissible, hoverable and persistent. This specifically refers to actions that remove pointer hover, or when the keyboard focus triggers additional content.
	<b>Reflow (1.4.10_AA)</b>	2.1 requirement. However, best practice is to <b>design</b> content so that it's column based and thus expands vertically.  <b>Programmatically</b> leverage OS specific behavior to facilitate this as well.  <i>Make sure content doesn't require <b>both</b> vertical and horizontal scrolling when resizing.</i>
	<b>Time limits (2.2.1_A)</b>	Time limits set by the content must be able to be turned off, adjusted, extended, or be longer than 20 hours. If the time limit is essential to the experience, then it is exempt.
	<b>(4.1.3_A) Status Messages (502.3.14)</b>	<b>Programmatically</b> ensure that users are aware of status changes by declaring status messages through roles or properties. For instance, ensuring that "invalid entry", "saved", or "submitted" is communicated to the user.  <b>Design</b> should also communicate a change in status
<b>Device specific</b>	<b>Target Size (2.5.5_AAA)</b>	<b>Design</b> should ensure that target pointer/ touch inputs are at least 44px by 44px (AAA, but best practice)  <b>Programmatically</b> , React Native accessible components on tap targets will be leveraged.
	<b>Device orientation (1.3.4_AA)</b>	<b>N/A</b> - 2.1 Requirement. <b>Design</b> should ensure that the UI supports portrait and landscape mode, although this is best practice when it comes to responsive design and designing for multiple devices/ screen sizes.
	<b>Pointer gestures (2.5.1_A) (503.2)</b>	<b>Design</b> so that no inputs require gestures and are possible with taps. All functionality that uses multipoint or path-based gestures for operation can be operated with a single pointer without a path-based gesture, unless a multipoint or path-based gesture is essential.  Assumption: The only feature that will not have a tap-based alternative is scrolling, and this can be supported by platform capabilities. Both platforms have alternatives available to perform simple gestures (e.g. iOS has Assistive Touch built in.)

	<b>Pointer cancelation (2.5.2_A)</b>	<b>Programmatically ensure</b> that for functionality that can be operated using a single pointer, at least one of the following is true: No down-event, abort or undo, up-reversal, unless it's essential.
	<b>Motion Actuation (2.5.4_A)</b>	For a feature that requires device motion (e.g. Face ID), <b>design</b> an alternative way to use that feature that doesn't require device motion - assumption: a login username and password field is an alternative for Face ID. We are also assuming that taking a picture for document upload is an essential function that can't be done in an alternative way.
<b>Code</b>	<b>Parsing - Code validation (4.1.1_A)</b>	<b>N/A</b> - there are no code validators for react native
	<b>Page language (3.1.1_A)</b>	The default human language and be determined <b>programmatically</b> .
	<b>Name, role, value (4.1.2_A) (502.3.1) (502.3.2) (502.3.4) (502.3.5) (502.3.14)</b>	All custom UI components should <b>programmatically</b> have a name, role and value.
<b>Keyboard</b>	<b>Keyboard focus visibility (2.4.7) (502.3.12)</b>	<b>Keyboard will be supported by OS capabilities.</b> If an element is in focus, it should be clearly visible that it is.
	<b>Keyboard focus order (2.4.3_A) (502.3.10)</b>	The tab order should make logical sense. All focusable elements should be read aloud by a screen reader.  The tab order cannot be <b>programmatically</b> adjusted through React Native as it does not support indexing, therefore the tab order will be dependent on the visual order. However, we could index components using AccessibilityLabel over the indexes of the map over an array of data.
	<b>Keyboard access (2.1.1_A) (502.3.11) (502.3.13)</b>	<b>Keyboard will be supported by OS capabilities.</b> All functionality of the content is operable through a keyboard
	<b>Keyboard traps (2.1.2_A) (502.3.13)</b>	<b>Keyboard will be supported by OS capabilities.</b> Content does not "trap" keyboard focus, i.e. there are ways to exit modals and widgets using the keyboard
	<b>Character Key Shortcuts (2.1.4_A)</b>	<b>Keyboard will be supported by OS capabilities.</b> Keyboard shortcuts must be able to be turned off, remapped or active only if the component is in focus
	<b>Focus triggers (3.2.1_A)</b>	<b>Keyboard will be supported by OS capabilities.</b> Changing focus should not trigger any unexpected changes to page content

## Section 2: Test Environment

### Device Matrix

The MVP will support both iOS and Android devices. Since there is a variance between accessibility capabilities across operating systems, one device will be manually tested from each platform.

These devices will be selected from the [Device Matrix](#) based on availability during the development cycle. Specific devices from the device matrix may be used depending on the test case. For instance, when testing text/ content resizing capabilities, devices with a smaller screen size should be prioritized.

While additional configurations may exist on other devices (i.e. non-Samsung Android devices), these settings are likely to be driven by personal user preferences and to go beyond fundamental platform accessibility capabilities.

### Test Tools

Multiple testing tools will be used to ensure adequate testing coverage. Where possible, platform specific assistive technology will be used to inspect accessibility properties that manifest differently across platforms.

- Voiceover (iOS), and Talkback (Android) will be used to test Screen Reader capabilities
- Bluetooth keyboards\* will be used to test keyboard operability.
  - \*The type of bluetooth keyboard will be confirmed once devices are procured.
- Deque mobile app testing for iOS and Android may be used to audit the app as a comprehensive set of functionality is developed.
  - If needed, the results of the full app scan will be shared with the 508 Office nearing the end of the development cycle.

## Section 3: Testing Protocol

This testing protocol will be used to guide the testing process. This testing plan may be adjusted in cases where a requirement has been de-prioritized or is determined to be unfeasible for MVP.

- In the criteria section, [P] and [F] denote passing and failing criteria, respectively.

### Color

Requirement Reference	Protocol	Criteria
<b>Color</b>	Visual observation that color is not the only method used to provide information.	<b>[P]</b> if color is used but is not the only method to provide information. <b>[F]</b> if color is the only method used to provide information.
<b>Color Contrast</b>	Use a contrast comparison tool to determine contrast ratio between text and background colors	<b>[P]</b> if the contrast ratio is 4.5:1 or greater for normal fonts, and 3:1 or greater for 19px+ fonts, when comparing text to background colors. <b>[F]</b> Less than 4:5:1 contrast ratio for normal fonts and 3:1 for large fonts, when comparing text to background colors.

### Headings & Navigation

Requirement Reference	Protocol	Criteria
<b>Heading levels, Heading and labels</b>	Use a screen reader to: <ul style="list-style-type: none"> <li>• Ensure that there is only &lt;h1&gt; on the page.</li> <li>• Ensure all heading levels are purposeful and in logical order</li> <li>• Ensure that no headings are skipped.</li> </ul>	<b>[P]</b> if there is only one <h1> on the page and all heading levels are purposeful, in logical order and no headings are skipped. <b>[F]</b> if any heading is skipped, read in an illogical order, not purposeful or there is more than one <h1> on the page.
<b>Info and Relationships</b>	Verify that the screen reader describes any visual information that denotes a relationship, e.g. a child-component relationship to a parent-component.	<b>[P]</b> if the user is informed of visually communicated information and relationships <b>[F]</b> if visually communicated information and relationships are not declared to the user
<b>Lists</b>	Ensure that the screen reader describes any list-like components as lists.	<b>[P]</b> if lists are programmatically determined. <b>[F]</b> if lists are not programmatically determined.
<b>Page title</b>	Review each page in the app and ensure that each unique page title is both descriptive and differentiated.	<b>[P]</b> if each page title is unique, descriptive and differentiated. <b>[F]</b> if a page title is misleading or undescriptive
<b>Sensory characteristics</b>	Visually inspect pages for any items that require characteristics like shape, color, size, visual location, orientation, or sound to be understood or operable.	<b>[P]</b> if all components and pages are understood and operable without needing shapes, colors, size, visual locations, orientations or sounds. <b>[F]</b> if pages require any shape, color, size, visual location, orientation, or sound to be understood or

		operable.
--	--	-----------

### Non-text content & media

Requirement Reference	Protocol	Criteria
<b>Text alternatives</b>	Review any non-text content with the screen reader to ensure it has a text alternative.	<b>[P]</b> User is able to interact with non-text content by way of text alternative. <b>[F]</b> User has no alternative for non-text content.
<b>Flashing content</b>	<b>N/A</b> The app does not contain any flashing content. In the event this is added, the tester will ensure that any flashing content does not flash more than 3 times per second (3 Hz.)	<b>[P]</b> No flashing content or if there is, it flashes 3 times per second or less. <b>[F]</b> Flashing content > 3 times per second.
<b>Label in name</b>	Review all components with visual labels and ensure the name of the label either matches, or is part of, the accessible name that is read by the screen reader.	<b>[P]</b> The label of the component either matches or is part of the accessible name. <b>[F]</b> The label name is not included in the accessible name.

### Forms & Input

Requirement Reference	Protocol	Criteria
<b>Identify Input Purpose - Form labels and instructions</b>	Use the screen reader and visual observation to ensure all fields collecting information are well labeled so the user can easily tell what is needed.	<b>[P]</b> Purpose for each input field for information collection from the user is easily determined. <b>[F]</b> Input field does not clearly convey the information needed.
<b>Form error identification</b>	Use the screen reader and visual observation to verify that any form entry or submission errors are clearly conveyed to the user. In cases where the app automatically corrects an entered value, the constraint is also communicated.	<b>[P]</b> Screen reader clearly informs users of errors that have occurred during entry or submission. <b>[F]</b> Errors messages do not exist, or the screen reader fails to read errors appearing during entry or submission.
<b>Input triggers</b>	Visually observe that filling out form fields will not change the content of the page or form in any major way.	<b>[P]</b> Filling in a form causes minor or no changes in page content. <b>[F]</b> Input by the user causes major structural changes



		to the form or page.
<b>Form error suggestions</b>	Use the screen reader and visual observation while generating errors to ensure that the user is given suggestions on how to fix their input.	<p><b>[P]</b> Input errors are accompanied by suggestions on how to fix the error.</p> <p><b>[F]</b> Upon entering information that results in an error, the user does not receive suggestions on how to fix the error.</p>
<b>Legal, financial and data error prevention</b>	<p>On the direct deposit page (and additional pages that are later determined to require legal or financial data input), ensure that a user can do at least one of the following:</p> <ul style="list-style-type: none"> <li>-Review the information they've entered</li> <li>-Confirm that the information they've entered is correct</li> <li>-Reverse or change the information they've entered after submission</li> </ul>	<p><b>[P]</b> The user is able to review, confirm, reverse, or change financial and legal inputs.</p> <p><b>[F]</b> The user is unable to do any of the aforementioned actions and all legal and financial data inputs are final.</p>

### Text Content

Requirement Reference	Protocol	Criteria
<b>Meaningful reading order</b>	Use the screen reader to ensure page content is read in a logical order.	<p><b>[P]</b> Content is read in a logical order.</p> <p><b>[F]</b> Content is not read in a logical order and jumps around.</p>
<b>Images of text</b>	Use visual observation to verify images are not purely text. If images of text do exist, determine whether this is essential to the information being conveyed.	<p><b>[P]</b> Images are not purely text (unless it can't be accomplished via styled text.)</p> <p><b>[F]</b> Image is purely text despite there being a viable alternative.</p>
<b>Text Spacing</b>	<p>(Not required, but best practice.)</p> <p>Use inspection tools to verify the following:</p> <ul style="list-style-type: none"> <li>• Line height / spacing is at least 1.5 times the font size.</li> <li>• Paragraphs are at least 2 times the font size.</li> <li>• Letter spacing is at least 0.12 times the font size.</li> <li>• Word spacing is at least 0.16</li> </ul>	<p><b>[P]</b> Line height / spacing is at least 1.5 times font size, Paragraphs are at least 2 times font size, Letter spacing is at least 0.12 times font size, Word spacing is at least 0.16 times font size.</p> <p><b>[F]</b> Line height / spacing is less than 1.5 times font size, Paragraphs are less than 2 times font size, Letter spacing is less than 0.12 times font size, Word spacing is less than 0.16 times font size.</p>

	times the font size.	
<b>Text Resizing</b>	<p>Use the OS font scaling settings to verify user is able to resize content to the maximum device size.</p> <p>Note: The header and navigation bar will not be reviewed for this criteria, as resizing these components significantly would make them unusable.</p>	<p><b>[P]</b> Text may be resized with no loss of functionality.</p> <p><b>[F]</b> User experiences some loss of functionality when text is resized.</p>

### Miscellaneous

Requirement Reference	Protocol	Criteria
<b>Time limits</b>	<b>N/A as no time limits currently exist.</b> In the event this is added, the tester will use the screen reader and visual observation to ensure any time limits under 20 hours and set by the content can be turned off, adjusted, or extended.	<p><b>[P]</b> if there are no time limits or any time limits existing may be adjusted, extended, or the limit is longer than 20 hours.</p> <p><b>[F]</b> if there is a time limit (other than essential) that may not be adjusted, extended or is longer than 20 hours.</p>
<b>Status Messages</b>	Use the screen reader and visual observation to check that any errors or other status messages (like “save” or “submitted”) are communicated to the user.	<p><b>[P]</b> All status messages are communicated to the user and may be seen or heard with a screen reader.</p> <p><b>[F]</b> Status messages are not communicated to the user or read by the screen reader.</p>

### Device Specific

Requirement Reference	Protocol	Criteria
-----------------------	----------	----------

<b>Target Size</b>	Use the inspection tool to verify all touch point / target pointer inputs on a page are at least 44px by 44px in size.	<b>[P]</b> if all touch points on a page are 44px by 44px or larger. <b>[F]</b> if any touch point on a page is smaller than 44px by 44px.
<b>Pointer gestures</b>	By observation, verify no inputs (other than scrolling) require any gestures other than tapping.	<b>[P]</b> if all components (e.g. buttons, tabs, or sliders) may be operated with a simple tapping gesture to operate. <b>[F]</b> if a component besides scrolling requires a sliding or path-based gesture to operate.
<b>Pointer cancelation</b>	By observation, verify that any functionality that can be operated using a single pointer has at least one of the following true: <ul style="list-style-type: none"> <li>• No down-event, abort or undo</li> <li>• No up-reversal</li> </ul> (Unless it is essential.)	<b>[P]</b> There is no down-event, abort, undo or there is no up-reversal (unless it is essential.) <b>[F]</b> There are down-events, aborts, undos and up-reversals that are not essential.
<b>Motion Actuation</b>	By observation, verify there are alternatives to any feature requiring device motion (like Face ID) and that there is no required shaking or tilting of the device, etc. Note: Taking a picture with the device is an exception because it is an essential function with no alternative.	<b>[P]</b> if all functions using motion have an alternative that doesn't require shaking or tilting of device. <b>[F]</b> if shaking or tilting of the device is necessary without an alternative (other than taking pictures.)

### Code

Requirement Reference	Protocol	Criteria
<b>Page language</b>	<b>N/A - the app only supports English</b> If an app supports multiple languages, ensure that the language can be accurately identified by the device by checking that: <ul style="list-style-type: none"> <li>• Screen readers can load the correct pronunciation rules.</li> <li>• Characters and scripts are displayed correctly.</li> </ul>	<b>[P]</b> Page is written in human language and is easily read, both visually and with the screen reader. Media players and characters/ scripts are also correct. <b>[F]</b> Page is unable to be read visually or by screen reader. Characters and script do not match the selected language.

	<ul style="list-style-type: none"> <li>Media players can show captions correctly.</li> </ul>	
<b>Name, role, value</b>	Use the screen reader and visual observation to verify all custom UI components have a discernible, logical name and role, (and value when applicable)	<p><b>[P]</b> The name, role, and when applicable, value of each custom component are clear and read by the screen reader.</p> <p><b>[F]</b> Any custom UI components are missing a name, role or value.</p>

### Keyboard

Requirement Reference	Protocol	Criteria
<b>Keyboard focus visibility</b>	Use a keyboard connected to the device and verify ability to tab through page elements, ensuring that focus is clearly visible.	<p><b>[P]</b> if focus is clearly visible while tabbing through elements.</p> <p><b>[F]</b> if focus is difficult to determine while tabbing through elements.</p>
<b>Keyboard focus order</b>	Use a keyboard connected to the device to tab through page elements. Verify the following: <ul style="list-style-type: none"> <li>Tab order is logical.</li> <li>All focused elements are read aloud by the screen reader.</li> </ul>	<p><b>[P]</b> if tab order is in logical order AND all elements are read aloud by screen reader.</p> <p><b>[F]</b> if tab order jumps around the screen in an illogical order OR all elements are not read aloud.</p>
<b>Keyboard access</b>	Use a keyboard connected to the device and verify that all functionality of content is accessible and operable through use of the keyboard.	<p><b>[P]</b> if all elements on the screen are accessed and operable through the keyboard.</p> <p><b>[F]</b> if any element on the screen may not be operable through the keyboard.</p>
<b>Keyboard traps</b>	Use a keyboard connected to the device and verify that all functions of the page are operable without becoming trapped in any particular element, modal, or widget (i.e. stuck and not able to leave the element.)	<p><b>[P]</b> if a user is able to leave or exit all operations on the screen without becoming trapped or “stuck”.</p> <p><b>[F]</b> if a user gets caught in any sort of loop or element that they may not exit using any key or combination of keys on the keyboard.</p>
<b>Character</b>	Use a keyboard connected to the	<b>[P]</b> if a custom keyboard shortcut is able to be turned

<b>Key Shortcuts</b>	device to verify that any custom keyboard shortcuts can be turned off, remapped or only active while the component is in focus.	off, remapped or is only active when the component is in focus <b>[F]</b> if a custom keyboard shortcut cannot be modified or is active beyond in-focus components.
<b>Focus triggers</b>	Use a keyboard connected to the device to tab through and verify that no unexpected changes appear to page content by simply changing focus.	<b>[P]</b> if no unexpected changes occur when changing focus <b>[F]</b> if any unexpected changes occur to page content when changing focus