

MARS Accessibility Audit

Kendra Raczek (kraczek@)

UX Designer, SecOps Software

Last updated Sep 20, 2022

Summary

To help create a consistent experience across SecOpsSw products for all users, this accessibility review was performed for the MARS product. MARS is a web-based tool that focuses on streamlining WIM intakes and updates by consolidating data from outside services, while helping identify critical cases faster so the team can support individuals in crisis and mitigated targeted violence quickly, accurately and efficiently.

This testing was done using a macOS with multiple screen sizes, Chrome browser, and VoiceOver screen reader. For a more robust testing effort, it would be recommended to use multiple devices, operating systems, browsers, and screen readers, to cover all user experiences. Findings include a variety of code-based accessibility issues along with recommended fixes to remediate them. Most of the findings were identified using a combination of a screen reader and keyboard navigation.

Methods

Accessibility testing was done using both manual and automated strategies, referencing a combination of requirements and guidelines by internal and external sources, some of which are listed in the Resources section of this document.

Manual testing

- Mac VoiceOver screen reader
- Keyboard-only navigation
- Checking headings, titling in front end code
- Screen magnification
- Checking for colorblindness issues using [Colorblindly](#) Chrome extension filters

Automated testing

- [Axe DevTools Chrome extension](#)



- [HeadingsMap Chrome extension](#)

Findings ([SIM](#))

Headings ([SIM](#))

Throughout the different MARS pages, html heading tags `<h#>` are absent. Semantic headings and their rank are an important element of page structure that assistive technologies use to aid in page navigation. Many screen reader users will browse through the list of headings on a webpage as a first step to then decide where they want to navigate to; similarly to browsing a table of contents in a textbook to locate the chapter you are going to read. Every MARS page should have 1 `<h1>` that is the title of the specific page, and a tree of `<h2>`s, `<h3>`s, etc. in a hierarchical organization that reflect their relationship with the other headings, not for styling purposes.

See <https://www.w3.org/WAI/tutorials/page-structure/headings/> for best practices of headings and heading levels.

The [MARS prototype](#) has been annotated to indicate recommended heading levels.

- Go through all MARS pages and add/change html heading tags to all appropriate text content as indicated in prototype.

Titling ([SIM](#))

When browsing the different pages of MARS, the webpage html `<title>` inside `<head>` does not change from the generic "Workplace Threat Management". This title is displayed on the browser tab and is important in describing the content of each page. This is especially important for MARS users that work quickly and may have multiple tabs open, managing multiple cases or engagements at the same time. For all users, descriptive titles for every specific page or dynamic content make navigation easier and faster. To follow best practices, keep titles concise, and place the most unique part of the title first, followed by the site name, e.g. "Engagements - MARS".

See <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title> for `<title>` documentation.



The [MARS prototype](#) has been annotated to indicate titles for each major page.

- Go through all MARS pages and edit `<title>`s to be unique to each major page as indicated in prototype.

Landmarks ([SIM](#))

Most Meridian components contain semantic landmarks like `<nav>` and `<main>` out-of-the-box, but some landmarks are missing or incorrect in MARS. Landmarks are important for screen reader users to easily identify the different sections of a website and their functions. Users can then easily jump to the landmark they're looking for. For accessibility best practices, all content on a webpage should be contained within landmarks.

See https://www.w3schools.com/accessibility/accessibility_landmarks.php for documentation.

- Use a `<footer>` tag to landmark the MARS footer.
- If possible with Meridian, change top `<nav>` to `<header>` to better reflect its purpose on the page, since there are currently 2 non-unique `<nav>`s.
- Change the generic `<div>` → `<aside>` to wrap all Meridian sheets, both engagement and people sheets, to indicate it is a section of content related, but not part of, the main content.
- Use a combination of `<form>`, `<fieldset>`, and `<legend>` tags to semantically indicate forms and the sections within them, especially for long, multi-section forms like the create case or engagement forms.
 - See <https://www.linkedin.com/learning/building-great-forms-with-html-and-css/fieldsets-and-legends> for tutorial

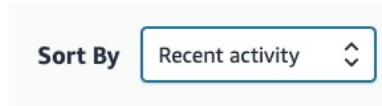
Labeling ([SIM](#))

Proper labeling is a critical accessibility requirement for all users to better understand the forms they fill out and information they see displayed. Without thorough labeling of elements and inputs, assistive technologies are unable to describe the element, and their users will have trouble performing essential actions. Form inputs and interactive elements should have explicit labels, or alternatively implicit labels through aria that are accessible to screen readers.

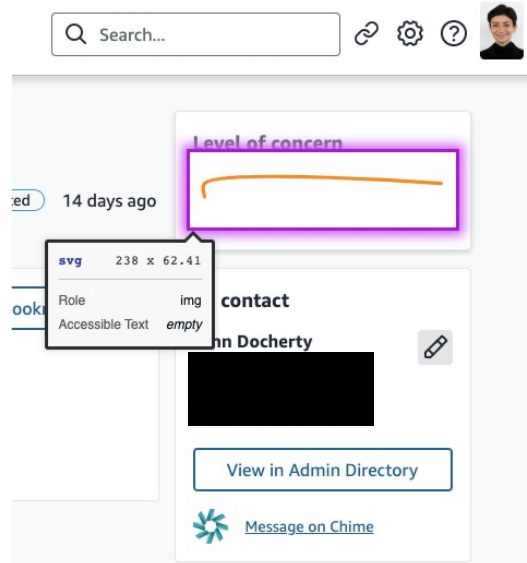
See <https://www.w3.org/WAI/tutorials/forms/labels/> for documentation.



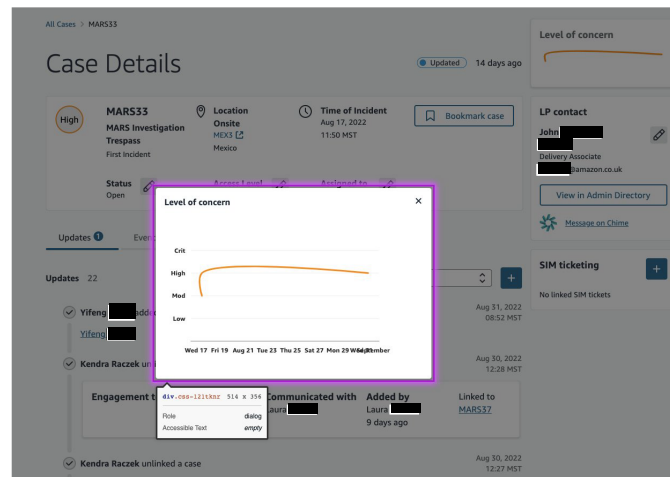
- Add `aria-labelledby="[ 'Sort by' id]"` to case lists and update thread of cases, so that the select input box has a label accessible to screen readers.



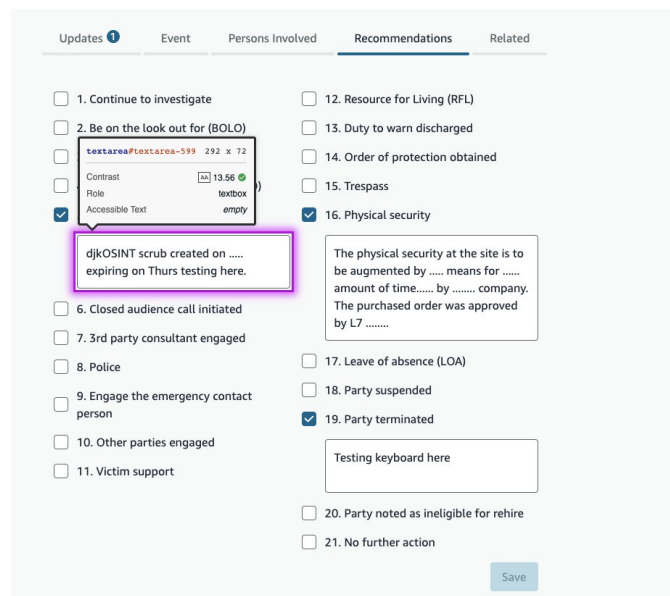
- Add `aria-label="Line graph showing change in level of concern over time"` and `aria-describedby="[legend id]"` to level of concern sparkline tile.



- Add `aria-labelledby="[ 'Level of concern' id]"` to the level of concern line graph modal with `aria-modal="true"` so that the modal has an accessible identifier.



- In recommendations, fill out the `aria-labelledby` attribute of each text box to be the checkbox text label it refers to, so that the text boxes have an accessible label.



Button vs link syntax ([SIM](#)) ([SIM](#))

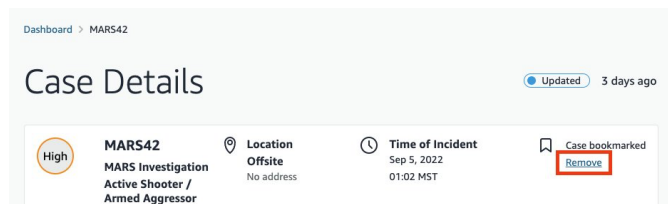
There are numerous interactive button and link elements throughout MARS that are incorrectly identified as each other, a mismatch from how they are presented visually. In common UX, links are expected to take users to another place on a page or a different page, while buttons are expected to trigger an action like opening or submitting. These two types of elements also behave differently using a screen



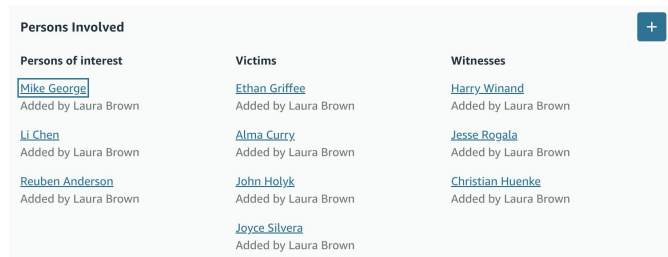
reader. Often times, screen reader users will navigate webpages by browsing elements by their type - e.g. moving through the page by headings, form elements, lists, links, etc. If buttons are written in code as a `<button>` rather than `<a>`, the screen reader will skip it when navigating through links on the page.

See <https://a11y-101.com/design/button-vs-link> and <https://ashleemboyer.com/blog/should-i-use-a-button-or-a-link> for documentation

- Remove `role="button"` from the "Remove" bookmark link so that it is correctly announced to screen readers as a link.



- Remove `role="button"` from persons involved links so that they are correctly announced to screen readers as links.



- Remove `role="button"` from the "Advanced search" link found under case and engagement list filters, so that it is correctly announced to screen readers as a link. This link is also found in the "No results" message when in the engagement list.

Category

Type

Region

Proximity

Facility code
AZA4

Flagged words

☒ New
 ☐ Updated

Clear

Apply

Advanced search

Engagements

Unlinked

Linked

Sort By

Recent activity

No unlinked engagements to display

Use [advanced search](#) to find engagements added over 30 days ago.

- Remove `role="button"` from the "Linked to" case ID links in engagement listings so that they are correctly announced to screen readers as links.
- Remove `role="button"` from the person links in a case's update thread, so they are correctly announced to screen readers as links.
- Remove `role="button"` from the attachment links in a case's update thread, so they are correctly announced to screen readers as links.

✓

Kendra Raczek added update

Aug 17, 2022
11:55 MST

Big update

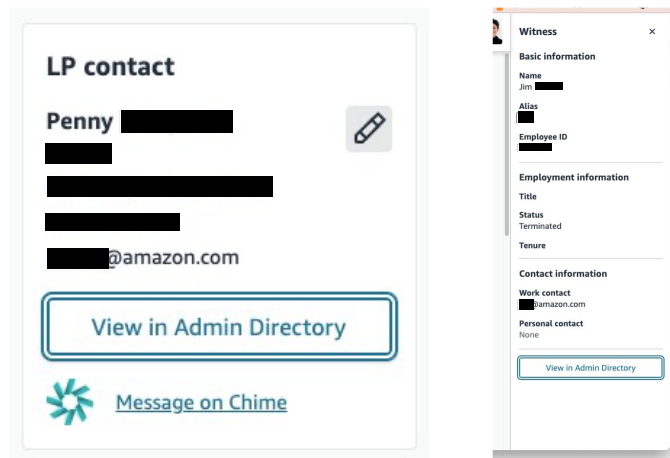
Here's an update that I will edit later

[d2f0eb7f8f1be4e75a54196afa7ad807.jpg](#)

- Change `<a>` to `<button>` for the "View in Admin Directory" buttons so that they are correctly announced to screen readers as buttons.



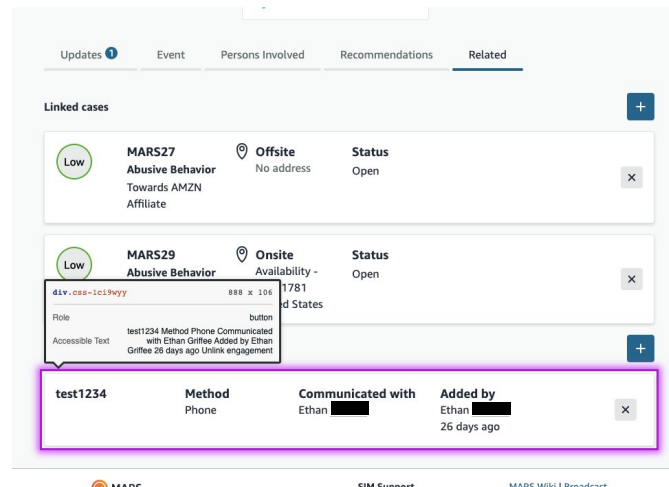
Page 7 of 14



- LP contact message on Chime link: Increase size to B300/16px, decrease margin between icon to 8px, include icon in `<a>` tag.



- Adjust linked engagement cards to match the linked case cards, so that instead of a `<div>` the card is a `<a>`. This seems to fix the problem of having a nested interactive element with the unlink button, helping screen readers and keyboard navigation to be able to access both the card itself and the unlink button.

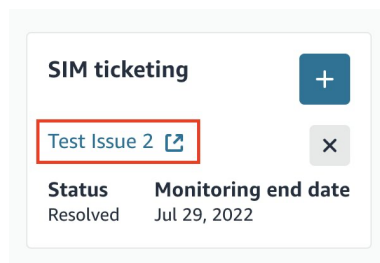


Keyboard focus ([SIM](#)) ([SIM](#))

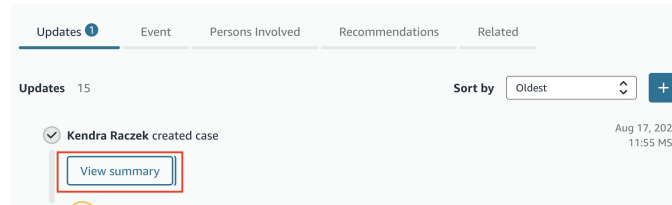
For users of assistive technologies, users with mobility impairments, and general power users, the keyboard is the primary or only medium for interaction with a computer. Often, screen reader accessibility and keyboard navigation share issues since they are used in conjunction with each other. There were two main issues identified with keyboard navigation: (1) visible keyboard focus indicators and (2) keyboard focus as users tab through and perform actions. Having an explicitly visible focus indicator on the webpage clearly identifies to the user what element they're currently on, essential for efficient page navigation. The keyboard focus order as users tab through should also reflect the logical visual order of the page and guide the user through actions.

See <https://webaim.org/techniques/keyboard/> for documentation and testing techniques.

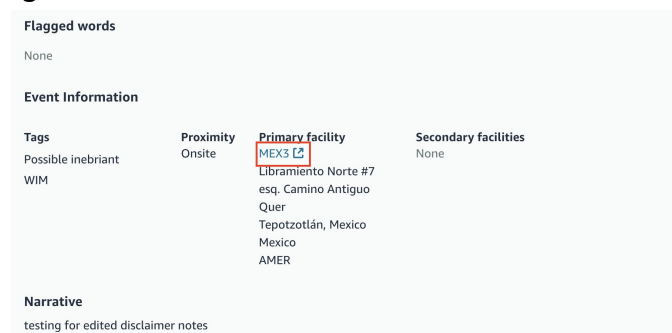
- SIM ticket link doesn't have visible focus indicator when using keyboard navigation.



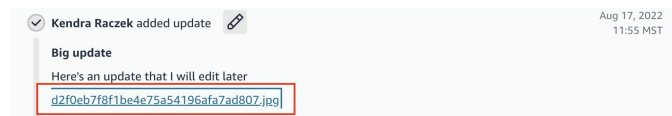
- View ~~summary~~ event button in updates thread has partial visible focus indicator when using keyboard navigation.



- Facility link in Event tab doesn't have a visible focus indicator when using keyboard navigation.



- Attachment download link in updates thread has partial visible focus indicator when using keyboard navigation.



- Person link in updates thread has partial visible focus indicator when using keyboard navigation.



- When filling out an advanced search/reporting form and selecting "See more" in 'persons involved' and 'communicated with' dropdowns using keyboard, keyboard focus moves back to the start of the page, requiring the user to tediously tab through to get back to the form field.
- When creating/editing consultation and selecting primary facility using keyboard, keyboard focus moves out of the modal, requiring user to tediously tab through to get back to the modal.

- When creating/editing consultation and selecting “See more” in the person search dropdown using keyboard, keyboard focus moves out of the modal, requiring the user to tediously tab through to get back to the modal.
- When creating/editing case and selecting primary facility using keyboard, keyboard focus moves back to the start of the page, requiring user to tediously tab through to get back to the form.
- When creating/editing case and selecting “See more” in a person search dropdown using keyboard, keyboard focus moves back to the start of the page, requiring the user to tediously tab through to get back to the form.
- After closing out of a modal, keyboard focus should return to the element that opened the modal. This might be tricky to do in code, but would aid in user experience.
 - Save linking case popover
 - Save linking engagement popover
 - Save access level popover
 - Save assigned to popover
 - Save status popover

Hidden elements ([SIM](#))

It is essential to provide the same access to all visible elements on webpages to both sighted users and screen reader users. Throughout MARS, there are a few places where elements aren’t accessible to either keyboard or screen reader, or elements that aren’t visible in the UI but accessible to keyboard or screen reader. Keeping access to elements consistent across different user experiences ensures that all users are able to access the same information and perform the same actions.

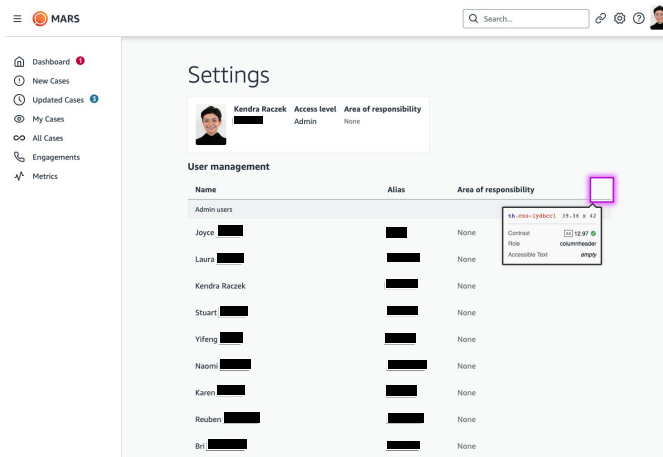
- Make the level of concern circle component a semantic `<button>`. Because it’s currently an `<svg>` element, keyboards are unable to focus on the element, skipping over it when tabbing through the interactive elements of the page.
- Add `label="Loading"` (new in Meridian 5.x) or `title="Loading"` to Meridian loader image elements so that the loading state is announced to screen reader.
 - Add `aria-live='polite'` with `role="status"` to loader elements so that they are announced once they appear on screen (added to the DOM) - reference file upload component in ‘Add update’ modal or toast alerts with a screenreader to see aria-live behavior.



- Because each case update thread listing div has the `aria-hidden="true"` attribute, any links or interactive content inside the updates is inaccessible to screen readers. Recommended to remove this attribute so screen reader users can access the content of each update of a case.



- Remove empty table header `<th>` until we have implemented editing areas of responsibility so it isn't read by screen readers and confusing users.



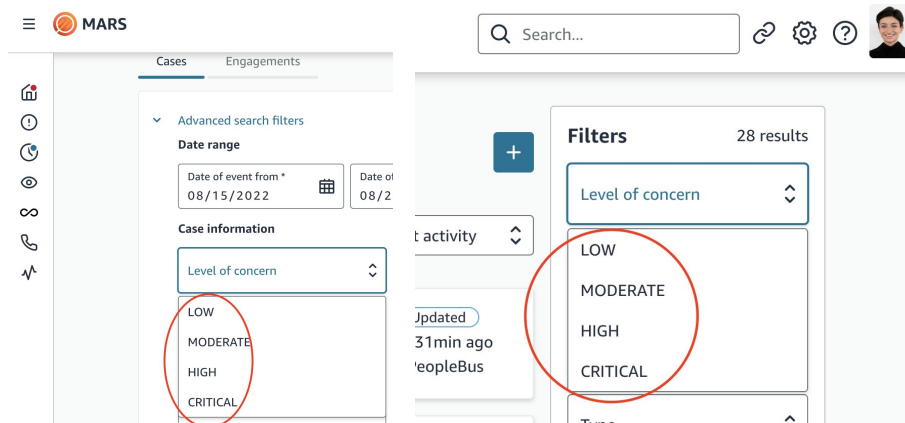
Visual feedback and treatment (SIM)

Lastly, there are opportunities where the system can more clearly communicate its status to users. [Visibility of system status](#) is one of the 10 usability heuristics by Nielsen Normal Group, and explains that greater transparency of what the system is doing allows users to feel in control and make educated decisions. Messaging should be appropriate for the action, and be available within a reasonable time.

- In consultation sheet, there is no visual feedback after linking or unlinking case
 - recommended that we implement a success message toast.
 - When users click X button to unlink, display small circular loader in its place until the operation is finished.



- After user presses “Save” button when adding/editing LP contact, disable the Save and Cancel buttons while system is loading. Otherwise, users are unsure if system is working and will press Save multiple times. This also stays consistent with other modal behavior.
- Uncapitalize level of concern options in advanced search/reporting dropdown and case lists filters. All-caps text has repeatedly been studied and proven to be less readable by both humans and assistive technologies.



Resources

[Amazon Accessibility Evaluator](#)

For earlier identification of accessibility issues, I recommend implementing the internal [Amazon Accessibility Evaluator](#) (AAE) tool to code pipelines. For all front-end code, the tool will perform automated tests and notify of accessibility issues as part of the CR pipeline process. For external customer-facing websites and applications at Amazon, it is a requirement to have AAE as a step in all code pipelines. Though it isn't required for internal-facing applications, we should still incorporate it as a highly recommended step.

[Amazon Accessibility wiki](#)

[Accessibility Learning Hub](#)

[Accessibility micro-training for engineers](#)

[Screen reader basics: Desktops and laptops](#)





Guidelines

[Internal people accessibility guidelines](#)

[External accessibility customer critical requirements](#)

[W3C / WCAG accessibility standards](#)

