Accessibility Acceptance criteria for modals

Amended from:
https://www.w3.org/TR/wai-aria-practices/#dialog_modal
https://bitsofco.de/accessible-modal-dialog
https://gist.github.com/hannalaakso/2641fc16d2158e60d551cd9da960b5da

To create a modal that is accessible, there must be some functionally to include:

1. Markup the dialog and dialog overlay appropriately
Dialogs should have an appropriate role and label. For a screen reader announce that a dialogue has opened, read the modal heading.

2. On dialog open, set focus
When first opened, focus should be set to the first focusable element within the dialog.

3. Make all interactive elements within modal are keyboard focusable
Users can access all interactive elements within the modal using keyboard only. Keyboard focus indicator is visible, and it is clear which element the user is interacting with.

4. On dialog close, return focus to the last focused element
When the dialog is closed, focus should be returned to the element that opened it.

5. While open, prevent mouse and keyboard clicks outside the dialog
Users should not be able to click on elements outside the dialog window. The underlying page must not look actionable.

6. While open, prevent users from interacting with underlying page.
User is prevented from interacting with the underlying page using both normal and specialized browsing modes such as caret browsing.
User navigating with a keyboard should not be able to TAB out of the dialog content.
The underlying page is hidden from assistive technology, including screen readers and voice control software, such as Dragon.

7. Allow the ESC Key or back browser button to Close the Dialog
When the dialog is open, pressing the ESC key or the back browser button should close it. It is clear how to close the modal.

8. Modal is always visible
Modal remains visible on the screen regardless of scrolling, screen size or orientation changes. In addition, users can scale text on the modal to 400% without horizontal scrolling (this refers mostly to scaling things using browser zoom).

For timeouts only:

9. Use aria-live region to announce remaining time
Screen reader users should be informed about the remaining time at appropriate intervals. If necessary, this can be achieved by an additional, visually hidden countdown.

Use aria-live=polite to announce updates to screen reader users. Screen readers should announce meaningful part of the message, rather than just time remaining.

10. Give user appropriate time to react
Users need at least 20 seconds to react to the timeout warning. This is the absolute minimum, consider allowing more time according to user needs.