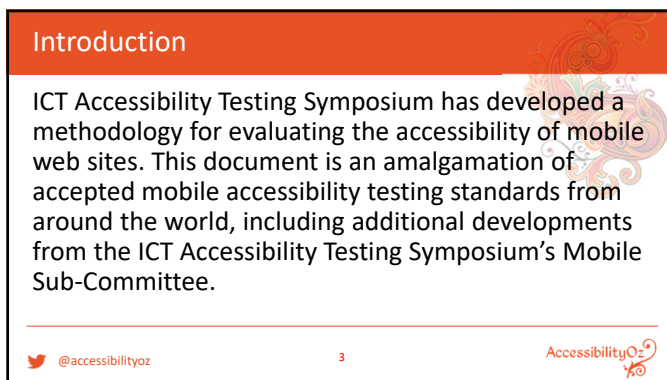




1



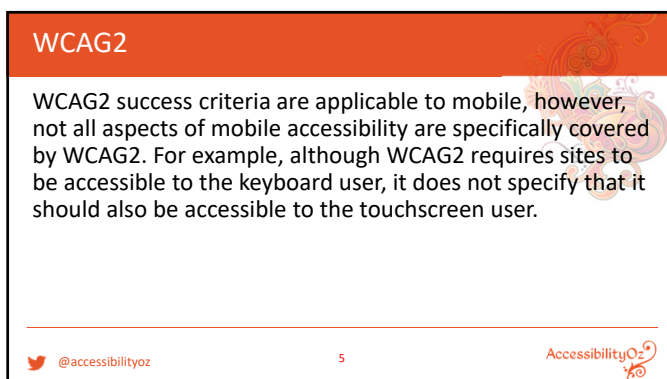
2



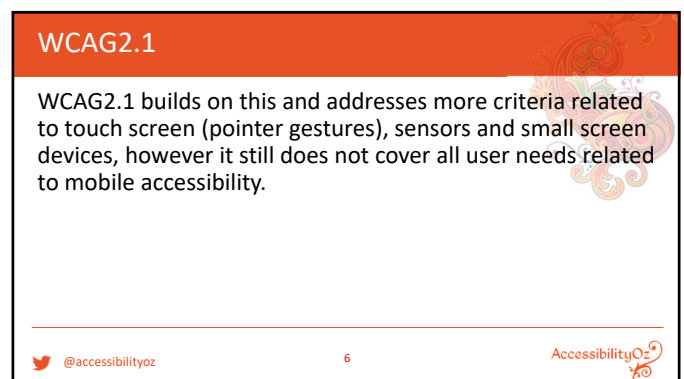
3




4



5



6



## Test all variations of mobile sites

gian@accessibilityoz.com  
accessibilityoz.com

AccessibilityOz

7

## WCAG 2.1 Page variations

Low vision users (who use the zoom function inherent in the browser) are often restricted to a mobile view of the site on their desktop. As part of WCAG2, zooming to 200% should already be included in regular testing (and therefore is not included in this methodology). It is essential that functionality is not removed due to a variation in the page.

@accessibilityoz 8 AccessibilityOz

8

## WCAG 2.1 Page variations

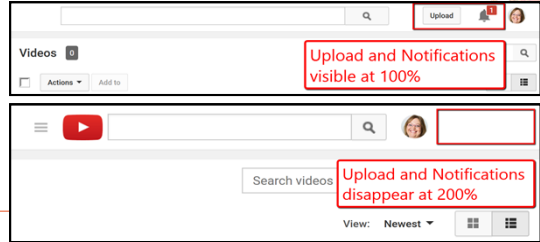
For example, previously in YouTube (this has now been fixed), the upload and notifications buttons were visible at 100% screen size but not at 200% screen size. This would mean that people browsing at 200% screen size would not be able to upload a video or view their notifications, because it was assumed that the 200% view was a mobile view and people would use the mobile app instead.

@accessibilityoz 9 AccessibilityOz

9

## WCAG 2.1 Page variations

### WCAG Conformance Requirement - Full Pages - Page variations




Upload and Notifications visible at 100%

Upload and Notifications disappear at 200%

@accessibilityoz 10 bilityOz

10



## Choosing what to test in native apps

gian@accessibilityoz.com  
accessibilityoz.com

AccessibilityOz

11

## Define application functionality

Through your understanding of the purpose of the native mobile application, define which functionality is critical to its purpose and use and that must be tested for efficacy, operability, and workflow from a user experience perspective.

@accessibilityoz 12 AccessibilityOz

12

## Common elements to test

- Navigation (menus, header, footer)
- Landing page
- Emergency alert pages
- Login pages
- Settings
- Account and profile
- Contact Us
- Real-time updates (eBay, Uber)
- Privacy policy, Terms & Conditions
- Interactional / transactional (select product, add to cart, payment, live chat, help, Q&A)
- Widgets (calendars, date pickers)
- Third-party integrations (geo-locational maps, chat, etc.)
- And/or High-traffic areas

@accessibilityoz

13

AccessibilityOz

13

## Common elements to test (continued)

**Ask the question:** how would the experience be impacted if the functionality failed, the content could not be reached, and or the experience caused a barrier to the user?

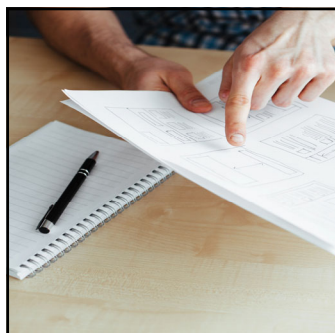
**Prioritize.** All functionality should be accessible within the native application; however, it is important to define and include the critical functionality for each individual app to be prioritized in your testing.

@accessibilityoz

14

AccessibilityOz

14

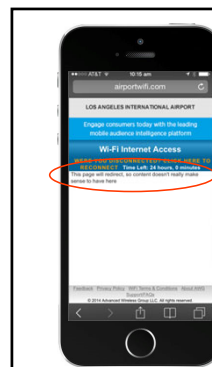


Please note that this methodology does not include those errors already included in W CAG2, but does include those errors in W CAG2.1

gian@accessibilityoz.com  
accessibilityoz.com

AccessibilityOz

15



## Test with real devices

I don't feel safe giving you my credit card details...

AccessibilityOz

16

## Mobile Testing Methodology

## Mobile Testing Methodology Overview

Step 1: Identify what needs to be tested

- Identify devices
- Identify the site type and variations of the page (Mobile site only)

@accessibilityoz

18

AccessibilityOz

18

## Mobile Testing Methodology Overview

### Step 2: Conduct specific mobile tests

- Critical issues
- Mobile-specific issues
- Mobile assistive technology and feature support
- Mobile and desktop relationship issues

@accessibilityoz

19

AccessibilityOz

19

Identify what needs to be tested

20

## Identify what needs to be tested

1. Identify devices
2. Identify the site type (mobile site only)
3. If a responsive site, determine if there are variations of the page (mobile site only)

@accessibilityoz

21

AccessibilityOz

21

You need to meet  
**WCAG2** and this  
mobile testing  
methodology

22

Test critical issues

23

## New features means new traps

**Trap:** Where a user is trapped within a component and cannot escape without closing the browser or app.

There are many more traps in mobile sites and native apps than on desktop.

@accessibilityoz

24

AccessibilityOz

24

## Hover trap

Content must be able to be dismissed if activated on touch (often these are actionable items that are activated on mouse hover on a desktop)

Applies to: Touch users

@accessibilityoz

25

AccessibilityOz

25

## Hover trap



Cannot dismiss zoomed in section

AccessibilityOz

26

## On-screen keyboard trap

Onscreen keyboard must be able to be dismissed.

Applies to: Onscreen keyboard users

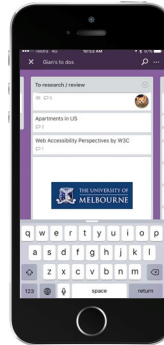
@accessibilityoz

27

AccessibilityOz

27

## On-screen keyboard trap



Cannot dismiss keyboard

AccessibilityOz

28

## Screen reader swipe trap

Screen reader users must always be able to activate an item on the current page or move back to the previous page.

Applies to: Screen reader users

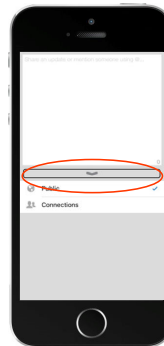
@accessibilityoz

29

AccessibilityOz

29

## Screen reader swipe trap



Cannot change or close page, access keyboard or go back

AccessibilityOz

30

## Touch trap

User must always be able to scroll / swipe to move up and down the page.

Applies to: Touch users

@accessibilityoz

31

AccessibilityOz

31

## Touch trap

Cannot move the screen unless you activate the up arrow

AccessibilityOz

32

## Zoom trap

Do not replace the entirety of the page with a feature that over-rides standard mobile functions such as swiping and scrolling.

Applies to: Touch users

@AccessibilityOz

33

AccessibilityOz

33

## Zoom trap

The zoom of doom

AccessibilityOz

34

## Text-to-speech trap

If the app has an ability to provide content via text-to-speech, the screen reader user must be able to pause or stop the app speaking in a simple manner, e.g. by performing a swipe on a screen.

Applies to: Screen reader users

@AccessibilityOz

35

AccessibilityOz

35

## Text-to-speech trap

Once activated, screen reader users cannot stop the TTS

AccessibilityOz

36



## Swipe trap

Any swipe gesture that is superceded by the screen reader must have an alternate gesture. You must be able to perform the same action, by using a link.

Applies to: Screen reader users or other assistive technology users which capture the swipe

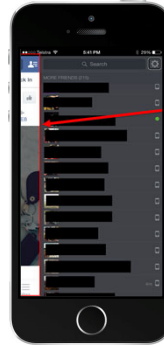
@AccessibilityOz

37

AccessibilityOz

37

## Swipe trap

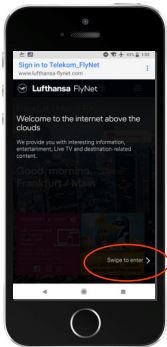


Facebook– Friends list  
Cannot go back to news feed, access keyboard or back button

AccessibilityOz

38

## Swipe trap



Screen reader user cannot swipe, as swiping is captured by screen reader only

AccessibilityOz

39

## Headset trap

Headset users must always be able to pause media (audio or video) content by using the Pause/Play control on the headset.

Applies to: Screen reader users, Headset users

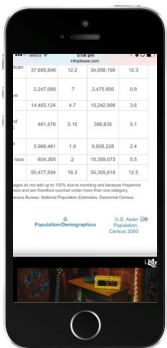
@accessibilityoz

40

AccessibilityOz

40

## Headset trap



Cannot pause the audio using headset hardware (pause on the headset pauses the screen reader)

AccessibilityOz

41

## Exit trap

Ensure there is always an accessible actionable item (eg. a close button that meets color contrast requirements and has an accessible name) that closes any feature that overlays the current page (such as a full-page ad).

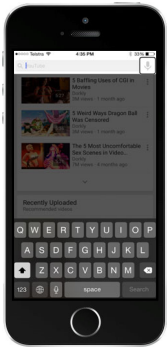
Applies to: All users

@accessibilityoz

42

AccessibilityOz

42




**Exit trap**

YouTube - homepage  
Screen reader user cannot escape from search box

AccessibilityOz

43

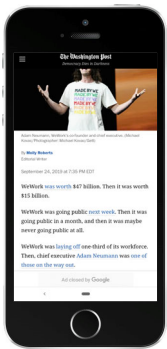


**Exit trap**

No way to close the feature (usually an ad)

AccessibilityOz

44

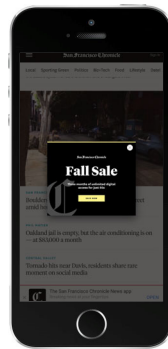


**Exit trap**

"Ad closed by Google" overlaps the "Simplified view" option on Android

AccessibilityOz

45



**Exit trap**

Close button does not meet color contrast requirements

AccessibilityOz

46

**Layer trap**

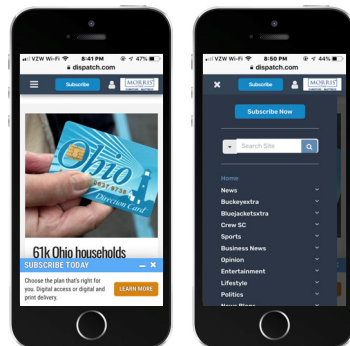
The user should not be trapped on a non-visible layer.  
Applies to: All users (but mostly encountered by screen reader users)

@AccessibilityOz

47

AccessibilityOz

47



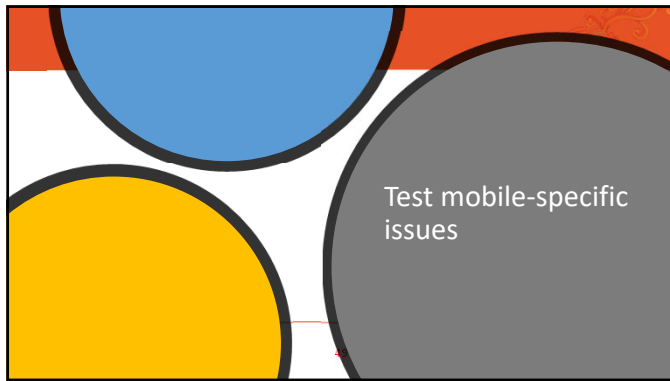
**Layer Trap**

Screen reader user cannot access the menu. Focus stays on the parent layer

AccessibilityOz

48

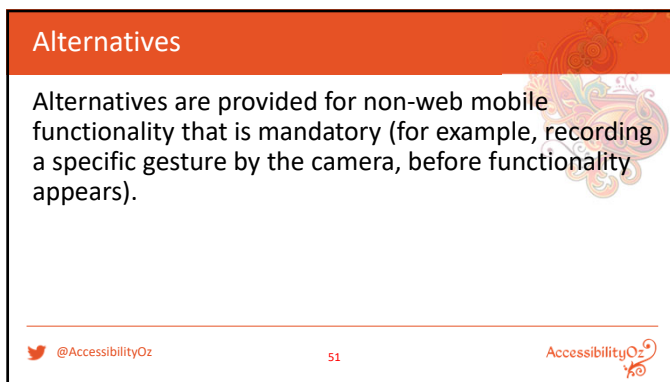




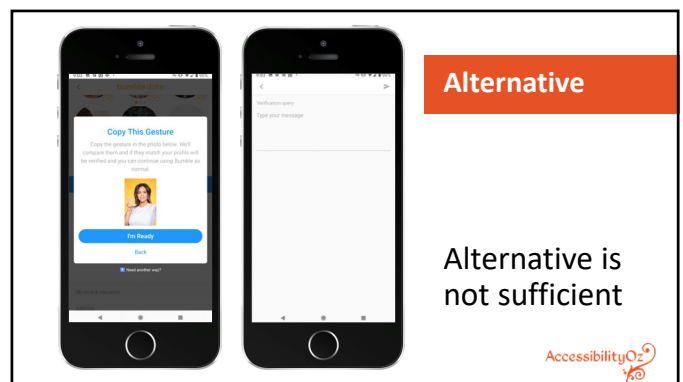
49



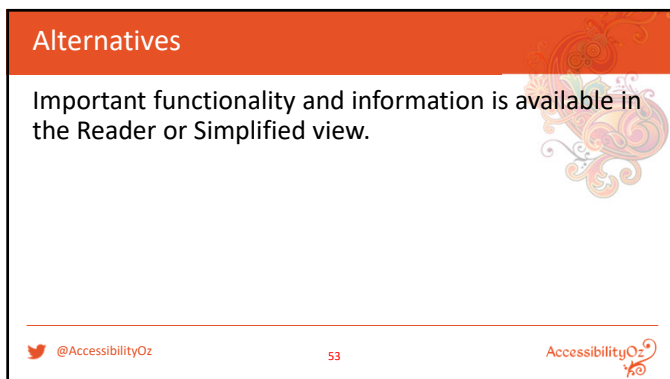
50



51



52



53



54

## Alternatives

Changes of state of non-standard controls (e.g. hamburger menu, star ratings) are clearly indicated  
Audio cues have an equivalent visual cue  
Horizontal or vertical swiping support touch gestures as a fallback (for more information see [WCAG2.1 SC 2.5.1: Pointer Gestures](#)).

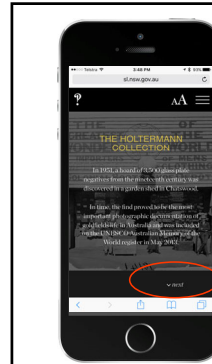
@AccessibilityOz

55

AccessibilityOz

55

## 2.5.1 example



Pass – allows for swipe or activating as a link

AccessibilityOz

56

## Alternatives

Horizontal or vertical dragging support touch gestures as a fallback (for more information see [WCAG2.1 SC 2.5.1: Pointer Gestures](#)).  
Toggle and slider elements support touch gestures as a fallback (for more information see [WCAG2.1 SC 2.5.1: Pointer Gestures](#)).

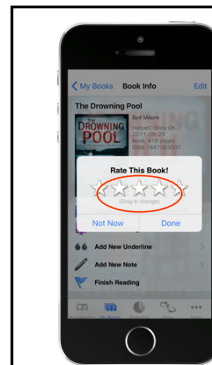
@AccessibilityOz

57

AccessibilityOz

57

## 2.5.1 example



Fail – requires the ability to drag

AccessibilityOz

58

## Alternatives

Active swipe elements support both horizontal scrolling and swipe gestures.  
Actionable elements are triggered only on removal of touch (ON TOUCH START and ON KEY DOWN have not been used) (for more information see [WCAG2.1 SC 2.5.2: Pointer Cancellation](#)).

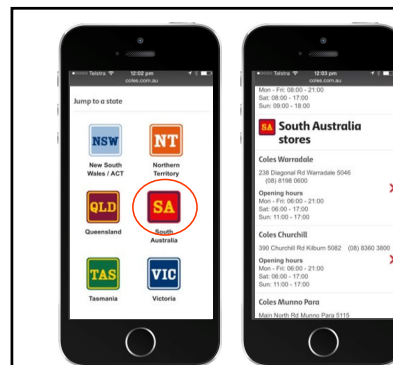
@AccessibilityOz

59

AccessibilityOz

59

## 2.5.2 example



Fail – link is activated on touch not on removal of touch

AccessibilityOz

60



## Display

gian@accessibilityoz.com  
accessibilityoz.com

AccessibilityOz

61

## Display

Do not present new content on hover over actionable element (for example, do not have a top-level menu item that displays sub-items on hover, but also when tapped opens a new page). For more information, see [WCAG2.1 SC 1.4.13: Content on Hover or Focus](#).

@AccessibilityOz

62

AccessibilityOz

62

## Display

Size of touch targets is at least 44 by 44 CSS pixels (approximately 7 to 10 millimeters). For more information see [WCAG2.1 SC 2.5.5: Target Size](#).  
Touch targets have sufficient inactive space between them (Inactive space of at least 10 pixels should be provided around active elements).

@AccessibilityOz

63

AccessibilityOz

63

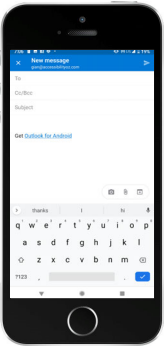


## 2.5.5 example

Fail 2.5.5 Target Size

AccessibilityOz

64

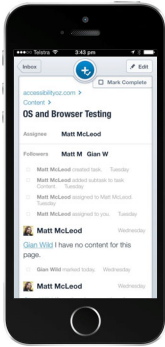


## 2.5.5 example

Pass – all touch targets

AccessibilityOz

65



## Spacing

Be careful what you select...

AccessibilityOz

66

## Display

Horizontal scroll bars do not appear at all when the page is resized. For more information, see [WCAG2.1 SC 1.4.10: Reflow](#).

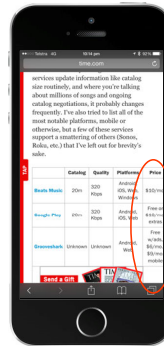
@AccessibilityOz

67

AccessibilityOz

67

## Mobile-specific interaction

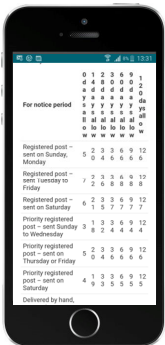


Reflow

AccessibilityOz

68

## 1.4.10 example



Fail – table with header one character wide

AccessibilityOz

69

## Display

Pinch zoom is operable, unless an accessible font resizing feature has been included in the web site that allows the user to increase the size of content at least two times the size of the standard font size. For more information see [WCAG2.1 SC 1.4.4: Resize text](#).

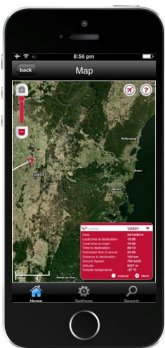
@AccessibilityOz

70

AccessibilityOz

70

## Text size

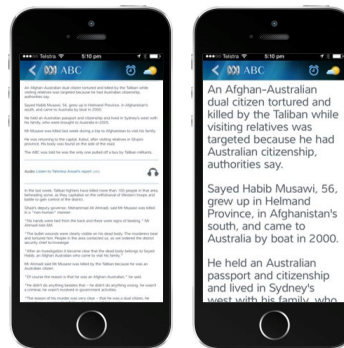


1.4.4 Resize text

AccessibilityOz

71

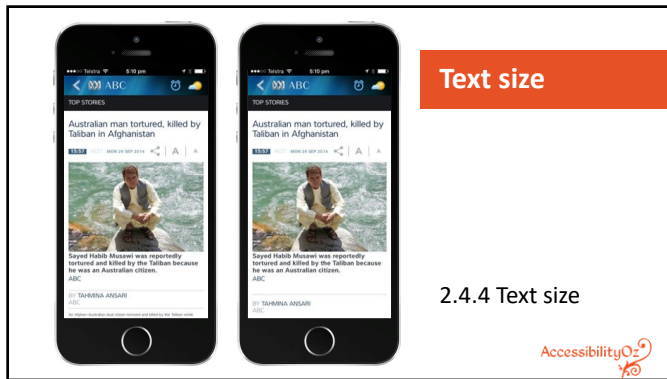
## Text size



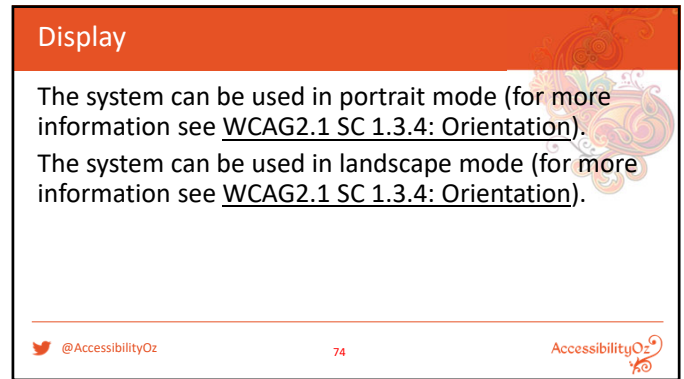
Inheriting text size from the system

AccessibilityOz

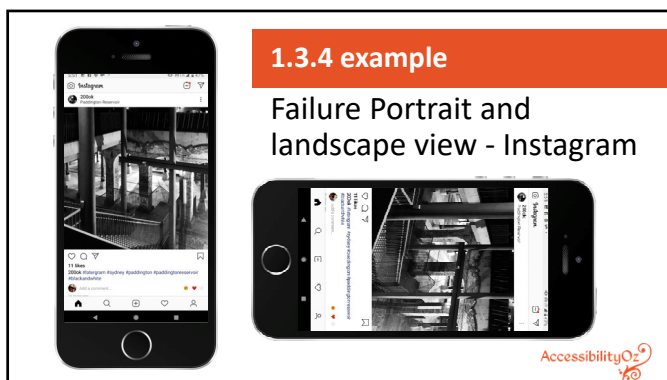
72



73



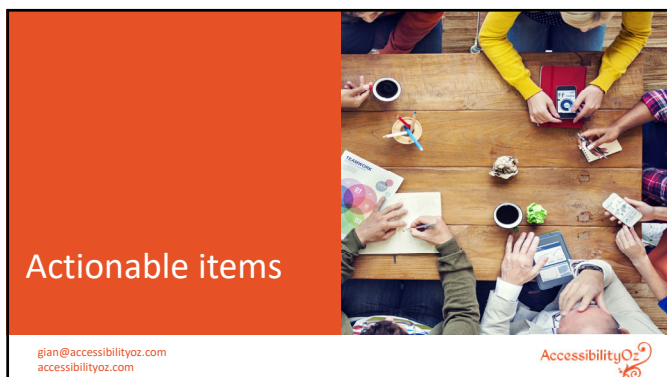
74



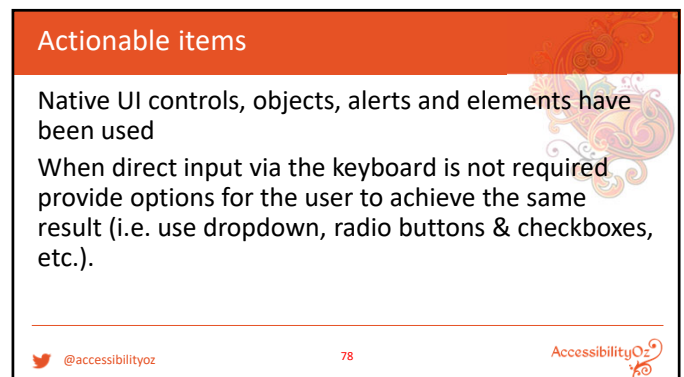
75



76



77



78



**Non-standard UI**

Non-standard UI element

AccessibilityOz

79

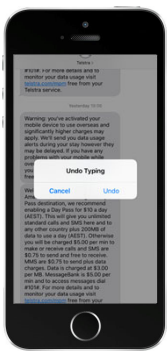
**Actionable items**

Functionality that can be operated by device motion or user motion can also be operated by user interface components, and responding to the motion can be disabled to prevent accidental actuation, except for certain situations. For more information see [WCAG2.1 SC 2.5.4: Motion Actuation](#).

Infinite scrolling has not been used

@accessibilityoz 80 AccessibilityOz

80

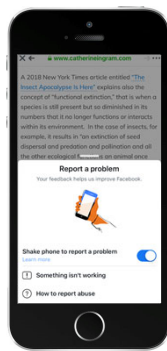


**2.5.4 example**

Fail – Undo Typing on iOS (can be turned off in Settings but no alternative)

AccessibilityOz

81




**2.5.4 example**

Fail – Report a problem on Facebook (can't be turned off)

AccessibilityOz

82

**Links**



gian@accessibilityoz.com  
accessibilityoz.com

AccessibilityOz

83

**Links**

Link text should have a minimum color contrast ratio of 3.0:1 when compared with the surrounding body text. For more information see [WCAG2.1 SC 1.4.11: Non-text Contrast](#).

@accessibilityoz 84 AccessibilityOz

84



## Links

Color alone should not be used to indicate links (if not underlined). A secondary method, such as underlines should be used, in addition to color. For more information see [WCAG2.1 SC 1.4.1: Use of Color](#).

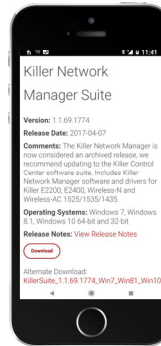
@accessibilityoz

85

AccessibilityOz

85

## Non-underlined links



1.4.1 Use of Color

AccessibilityOz

86

## Navigational aids



gian@accessibilityoz.com  
accessibilityoz.com

AccessibilityOz

87

## Navigational aids

Arrows and Next and Previous buttons have been used to indicate swipe or scroll areas (for more information see [WCAG2.1 SC 2.5.1: Pointer Gestures](#)).

Navigational aids such as back buttons, breadcrumbs, next and previous buttons are provided.

ARIA document landmarks have been used to appropriately describe document structure.

@AccessibilityOz

88

AccessibilityOz

88

## Audio and video



gian@accessibilityoz.com  
accessibilityoz.com

AccessibilityOz

89

## Audio and Video

All video and audio have an accessible transcript.

@AccessibilityOz

90

AccessibilityOz

90



91

## Forms

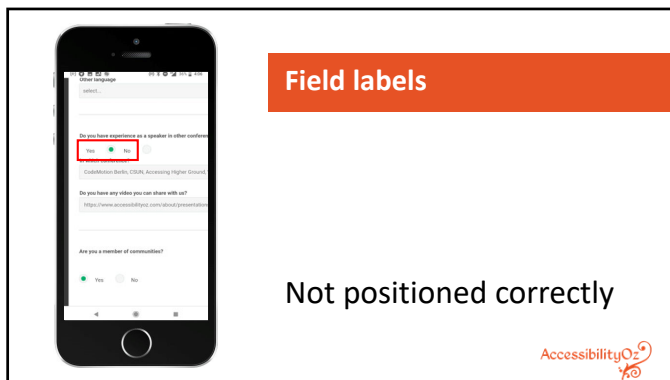
Field labels for all fields are positioned adjacent to the input field

The following HTML5 input type are used appropriately: EMAIL, TEL, DATE, DATETIME, MONTH, SEARCH

---

@accessibilityoz 92 AccessibilityOz

92



93

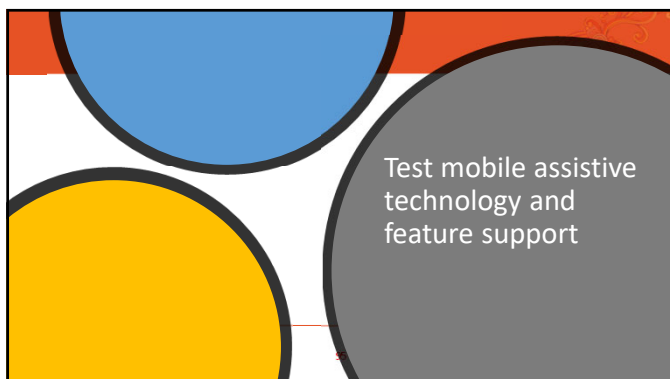
## Forms

The user should be able to dismiss on-screen keyboards that display when the user focuses on an input field. Additionally, any pop-up dialog or warning must be able to be dismissed. For more information, see [WCAG2.1 SC 1.4.13: Content on Hover or Focus](#).

---

@accessibilityoz 94 AccessibilityOz

94



95

## Test mobile assistive technology and features

All actionable items can be accessed and activated by the following assistive technologies (or when the following feature is enabled)

All important content can be accessed by the following assistive technologies (or when the following feature is enabled)

---

@accessibilityoz 96 AccessibilityOz

96

## Mobile assistive technology and features

- VoiceOver (iOS)
- Keyboard (iOS)
- Keyboard and switch (iOS)
- Zoom (iOS)
- Invert colors (iOS)
- Grayscale (iOS)
- Reader view (iOS)
- TalkBack (Android)
- Keyboard (Android)
- Keyboard & switch (Android)
- Magnification (Android)
- Invert colors (Android)
- Grayscale (Android)
- Increase text size (Android)
- Simplified view (Android)

@accessibilityoz

97



97

Test mobile and desktop relationship issues (mobile site only)

98

## Testing

Item labelling across mobile and main site is consistent  
Links between mobile and full version of the web site have been provided

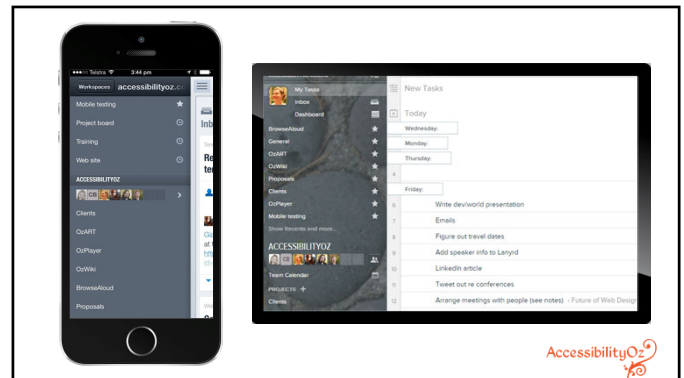
Users are not restricted to a particular version dependent on device (i.e. cannot use mobile version on desktop and vice versa)

@accessibilityoz

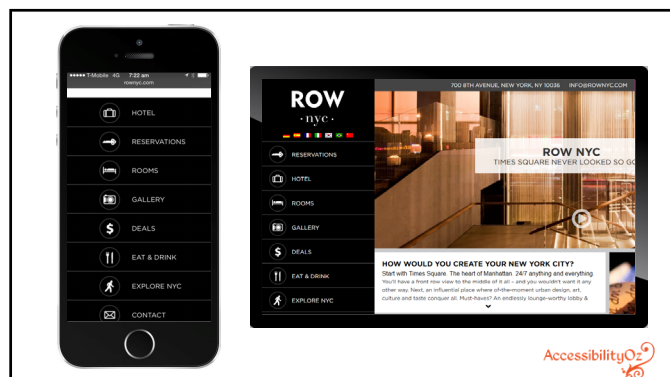
99



99



100



101

## Thanks to our committees


Brent Davis, Corbb O'Connor, Gian Wild (Co-Chair Mobile Site and Native App), Jennifer Chadwick (Co-Chair Native App), Karen Herr, Kathryn Weber-Hottelmann, Kathy Eng, Laura Renfro, Megha Rajopadhye, Michael Keane, Morgan Lee Kestner, Peter McNally (Co-Chair Mobile Site), Rafal Charlampowicz, Ryan Pugh, Shane Anderson, Steve Sawczyn, Sunish Gupta, Tom Lawton

@AccessibilityOz

102



102



What do you think  
of the  
methodology? Is  
there anything  
missing?

Slides: [pz.tt/ict19](https://pz.tt/ict19)

gian@accessibilityoz.com  
accessibilityoz.com

AccessibilityOz

103