

Mobile Application Accessibility Handbook



Office of the Government Chief Information Officer
The Government of the Hong Kong Special Administrative Region

COPYRIGHT NOTICE

© 2018 by the Government of the Hong Kong Special Administrative Region

Unless otherwise indicated, the copyright in the works contained in this publication is owned by the Government of the Hong Kong Special Administrative Region. You may generally copy and distribute these materials in any format or medium provided the following conditions are met –

- (a) the particular item has not been specifically indicated to be excluded and is therefore not to be copied or distributed;
- (b) the copying is not done for the purpose of creating copies for sale;
- (c) the materials must be reproduced accurately and must not be used in a misleading context; and
- (d) the copies shall be accompanied by the words “copied/distributed with the permission of the Government of the Hong Kong Special Administrative Region. All rights reserved.”

If you wish to make copies for purposes other than that permitted above, you should seek permission by contacting the Office of the Government Chief Information Officer.

Table of Contents

1.	INTRODUCTION	4
2.	UNDERSTANDING THE DIFFICULTIES ENCOUNTERED BY PERSONS WITH DISABILITIES	7
3.	TOP 8 CONCERNS FROM PERSONS WITH DISABILITIES.....	11
4.	BEST PRACTICES.....	17
5.	TESTING STRATEGY FOR DEVELOPERS	51
6.	RELATED RESOURCES	59

May 2018

1. Introduction

According to the telecommunications statistics published by the Office of the Communications Authority, the local mobile penetration rate exceeds 247.6% with over 18 million 2.5G/3G/4G mobile subscribers as at January 2018. Mobile communications have become ubiquitous, reaching out to all segments of the community, including persons with disabilities.

Internet and mobile computing bring much convenience to most people. However, some may have a wrong perception that persons with disabilities, especially those visually impaired, are unable to use mobile devices and mobile applications. Many mobile application developers are even not aware of the special needs of the disability groups in using mobile devices. In fact, more and more persons with disabilities are using touch screen mobile devices nowadays. Mobile devices and applications enable them to use Information and Communications Technology (ICT) anytime and anywhere more effectively, live more independently and participate fully in the society.

1.1 Accessibility of Mobile Applications

Given a high degree of penetration, mobile communications are opening up new channels of doing businesses and mobile platform become an ideal platform for delivering a wide range of applications and services. Therefore, making mobile applications accessible not only benefits the persons with disabilities but also helps organisations fulfilling legal responsibility¹, building corporate image and widening customer base. Opportunities arise for enterprises and organisations to develop accessible applications on mobile platform in order to harness the technology innovation as well as the fast growing digital economy. **It is necessary for mobile application owners and developers to know about the accessibility requirements of mobile applications and the diversified needs of different segments of the community.**

¹ The Disability Discrimination Ordinance (Cap 487) has created a legal duty for organisations to ensure their services are available to everyone regardless of disability. This principle is applicable to information and services provided through mobile applications and websites.

1.2 Mobile Applications

Mobile applications are software applications designed to run on mobile devices such as smartphones and tablets. Mobile applications can be divided into three types, namely **native apps**, **web apps** and **hybrid apps**. Differences among them are as follows:

- A **native app** is a downloadable software application which is platform specific and can be used offline, in most circumstances. Native apps need to be downloaded from the market places and installed on a mobile device before running.
- A **web app** is an Internet-based application which has to be launched by a mobile device's browser. As the source files are stored on a server, Internet connection is required but downloading or installation is not necessary.
- A **hybrid app** is a combination of native app and web app. It is built using open web standards such as HTML5, CSS and JavaScript. Like native apps, hybrid apps are required to be downloaded and installed on the mobile devices.

Mobile applications are usually referred to as native apps or hybrid apps which require downloading and installation on a mobile device.

1.3 Mobile Application Accessibility Handbook

This Handbook is designed for mobile application owners and developers. It is based on the Web Content Accessibility Guidelines 2.0 (WCAG 2.0) promulgated by the World Wide Web Consortium (W3C) and feedbacks collected from the local disability groups. This Handbook covers the basic concept and best practices in making mobile applications accessible to persons with disabilities. These best practices are categorised into two accessibility levels, namely Baseline level and Advanced level.

- **Baseline level:** Best practices described at this level are considered basic and essential features for persons with disabilities to access and understand the content in mobile applications. For mobile applications to attain the Baseline level, all the best practices at this level must be adopted.
- **Advanced level:** Best practices described at this level are considered desirable features to further enhance the user experience for persons with disabilities to access the mobile applications. For mobile applications to attain the Advanced level, all the best practices at this level cum the Baseline level must be adopted.

To enable persons with disabilities to use mobile applications, organisations should ensure the mobile applications must attain the Baseline level at a minimum. Nevertheless, organisations are encouraged to attain the Advanced level in their mobile applications, where appropriate, in providing a better user experience.

For the implementation details regarding different mobile operating systems, please refer to the prevailing accessibility technical guidelines of the corresponding platforms. Reference links are provided in **Section 6 – Related Resources**.

2. Understanding the Difficulties Encountered by Persons with Disabilities

There are many different types of disabilities which in general fall into four categories, namely visual impairment, physical impairment, hearing impairment and cognitive impairment. Different techniques are required by persons with different types of disabilities when using mobile devices as well as mobile applications. This Section lists out the difficulties encountered by persons with disabilities when using a mobile application and the accessibility features commonly available in a mobile device for helping them.

2.1 Visual Impairment

Difficulties: Persons with visual impairment who are blind, having low vision, colour blindness or colour deficiency, are unable to see the screens and controls in mobile applications. Due to their visual constraints, they cannot see the buttons and use a touch screen to access and navigate the mobile application functions.

Solutions: Most persons with visual impairment rely on a screen reader to operate computers as well as mobile devices. A screen reader is software for converting information displayed on the device screen into speech or Braille on a Braille display. Some mobile device screen readers with gesture-based features enable persons with visual impairment to hear descriptions of functions on their devices and to operate the functions through touch and gestures.

Accessibility features:

- Screen reader – Some mobile devices bundle with a built-in screen reader. Third-party screen readers are also available in application market places for download.
- Adjustable font sizes – Some mobile devices offer this as a built-in feature to enlarge the font size of the mobile applications.
- Screen magnifier – Some mobile devices offer this as a built-in feature to enlarge the screen display.

- Adjustable brightness/contrast controls – This feature changes the foreground/background colour of the screen and alter the brightness to meet individual needs.
- Backlit display – This feature changes the contrast of the screen thus allows better viewing in poor lighting or outdoors conditions.
- Voice recognition – This allows users to complete a function or control the mobile devices through voice commands.

2.2 Hearing Impairment

Difficulties: Persons with hearing impairment cannot hear the caller, audio notifications, audio content and instructions which are presented in audio format.

Solutions: Most persons with hearing impairment rely on sign language and text messaging to communicate. By providing sufficient captioning and/or sign language for audio content, visual and/or vibration alert will generally enable hearing impaired persons to use mobile applications.

Accessibility features:

- Vibrating alerts/visual notification – By using this feature, the device vibrates, blinks or a notification is displayed on the screen when receiving a message.
- Captioning – Use of caption for a video can show the content of an audio conversation and describe the important cues and surrounding sound taken place in the video scenes.
- Adjustable volume control – This feature is particularly helpful for those people who need a hearing aid.
- Mono audio – This feature aids people with hearing loss affecting one ear only. Enabling this feature will combine the audio content from both left and right channels.

2.3 Physical Impairment

Difficulties: Persons with upper limb and hand mobility problem have difficulties in performing gestures (such as pinch, spread, and flick), tapping buttons and controlling sliding bars on a mobile device.

Solutions: Provision of advanced speech recognition software and accessibility design allows hand-free control of the mobile devices. Some high-end mobile devices even provide eye-ball tracking and touch-free gesture features. Some persons with upper limb and hand mobility impairment may use other body parts to operate a mobile device with the aid of a stylus pen.

Accessibility features:

- Voice recognition – This feature allows users to use voice commands for operating a mobile device.
- Speed of pressing buttons – This feature allows users to adjust the speed required to navigate through the screen pages.

2.4 Cognitive Impairment

Difficulties: Persons with cognitive impairment may have problems related to attention, memory, analytical, communication, computational and reading skills. They have greater difficulties in understanding complex text or following complicated instructions.

Solutions: Although they do not require special tools to use a mobile device, provision of intuitive user interface can always help. Some persons with learning difficulties may use screen readers to help them focus and better understand the content of the mobile applications.

Accessibility features:

- Intuitive user interface – Provision of clear and simple user interface, consistent interface elements and prominent icons make navigation easier in using mobile applications and help illiterate persons as well.
- Read aloud with simultaneous text highlighting – This feature helps persons with cognitive impairment reinforce reading comprehension while using a mobile application.

- Auto-text for input fields – Some mobile devices provide this feature which aids the replacement of text with preloaded texts to reduce the number of keystrokes and avoid typo-error.
- Adjustable time for completing an operation – Provide sufficient time for users with different capabilities to complete a mobile application function at their own pace.
- Most of the accessibility features mentioned in **Section 2.1 to 2.3** will also help persons with cognitive impairment use mobile applications with greater ease and understanding.

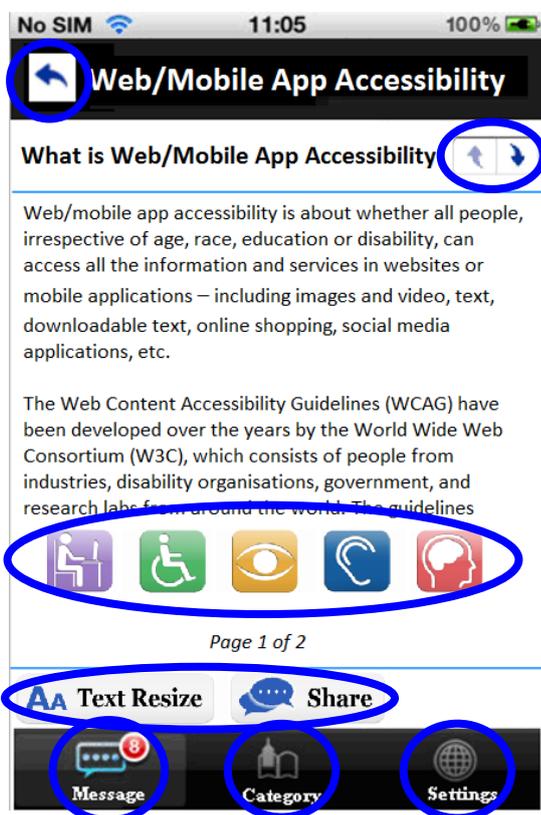
3. Top 8 Concerns from Persons with Disabilities

This Section introduces some common pitfalls in mobile applications that create barriers for use by persons with different types of disabilities. Developers should avoid the pitfalls and adopt appropriate coding techniques when developing a mobile application.

3.1 Text alternatives for non-text elements not provided

Affected Group: Persons with Visual Impairment

Meaningful and concise text alternatives should always be provided for non-text elements such as buttons, form fields, selection bars and images, etc.



For example, if you have a button on your mobile application, how would you communicate the meaning of this button to a person with visual impairment who is using a screen reader?

The best way is to provide a text alternative for this button  such as "Go back" or simply "Back".

3.2 Not function properly when using with screen readers

Affected Group: Persons with Visual Impairment

Each button and navigation menu should function properly when using with or without screen readers.

Buttons in some mobile applications do not function properly when users use the mobile application with the aid of screen readers. Developers or mobile application owners should ensure the mobile application and its functions/buttons would not clash with the prevailing screen readers. Sufficient testing should be performed to ensure the mobile application functions properly when using with screen readers.



Besides, when using with screen readers, many mobile applications are unable to proceed after reading the tutorial/acceptance statement for the first-time visit. Make sure go through all pages and test the whole mobile applications using screen readers.

3.3 Font size is too small and text resize function not provided

Affected Group: All Disability Groups

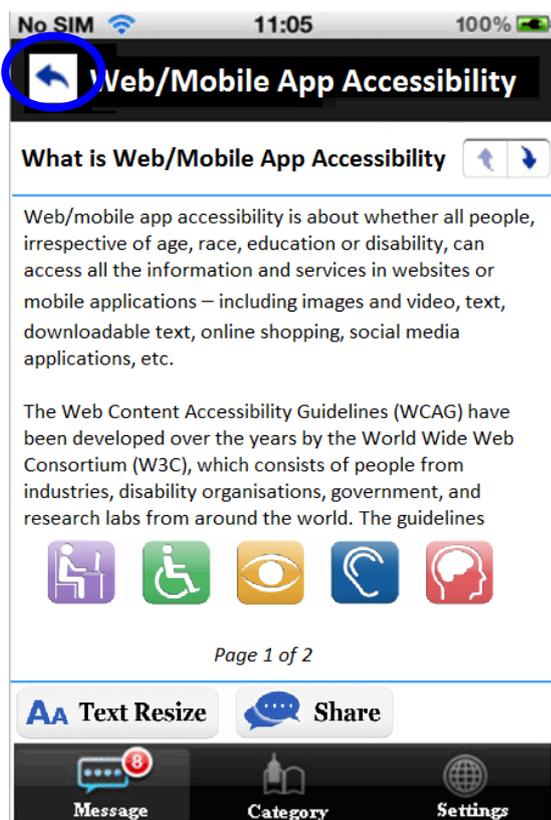
Not all users are using a large screen device. Scalable font size provided in a mobile application not only facilitates persons with disabilities to use the mobile application, but also helps elderly users and people using small screen devices. It is always a good practice to provide functions in mobile applications allowing users to enlarge the font size.



3.4 Poor navigation

Affected Group: All Disability Groups

Poor navigation makes a mobile application difficult to use for persons with or without disabilities. Provision of easy navigation method which is consistent across multiple screen pages of a mobile application helps all people, including persons with disabilities, control and navigates the mobile application easily.



Users can go back to the previous page easily by using the arrow button  associated with appropriate text alternative.

3.5 Options for notification not provided

Affected Group: Persons with Hearing Impairment

Ensure to provide notification settings with more than one option, such as “Vibrate” option and “Sound” option in mobile applications.

Some mobile applications only notify users by ringtone or sound. For persons with hearing difficulties, they cannot be alerted by this notification mode.

3.6 Instruction only available in audio format

Affected Group: Persons with Hearing Impairment

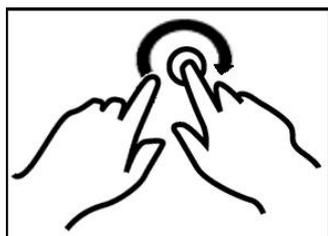
Ensure to provide instruction available in more than one format such as text format.

Some mobile applications only provide instructions in audio format. Persons with hearing difficulties cannot receive the instructions at all.

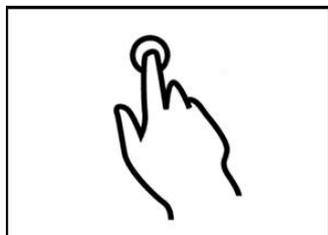
3.7 Gesture for control and sliding bar are difficult to use

Affected Group: Persons with Physical Impairment

Gesture for touch commands



Design simple gesture for controlling a mobile application is recommended. For example, the gestures on the left require a rotate gesture with two fingers. It is difficult for persons with upper limb/hand mobility problem to perform.



It is a good practice to implement a simple gesture which can be performed by one finger.

Sliding bar control (e.g. Timeline)



Sliders are small to click and require subtle movements

In addition, design larger control for sliding so that it is easier to control. Typical sliders are difficult to use because the portion that needs to be controlled is too small and must be moved in subtle increments in order to adjust values.



Individual buttons facilitate user control

A better approach is to use separate buttons for increasing and decreasing values as these can be tapped and made compatible with screen readers.

3.8 Tablet version not provided

Affected Group: Persons with Physical Impairment

Design and develop mobile applications for running on tablet devices as well.

It is a good practice to provide also a tablet version that enable users to use the mobile application on a tablet device with larger screen size for easier operation and control.



4. Best Practices

Best Practice 1 - Perceivable

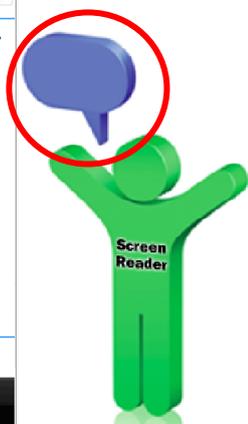
Information and user interface components must be presented to users in the ways that they can perceive.

Best Practice 1.1 – Provide text alternatives for non-text contents

Baseline level

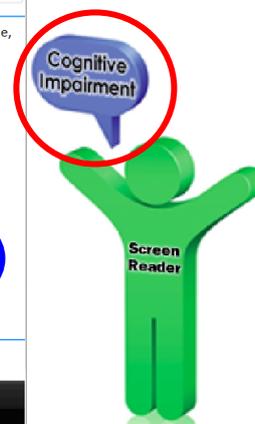
Ensure that all buttons, pictures, icons, photos and images (except CAPTCHA and images used for decoration, formatting or invisible), must have a meaningful and concise text description so that it can be read by screen readers. Text alternatives should be kept short (e.g. around four-word length) as far as possible.

Before Rectification



Screen readers are unable to read images without meaningful text descriptions.

After Rectification

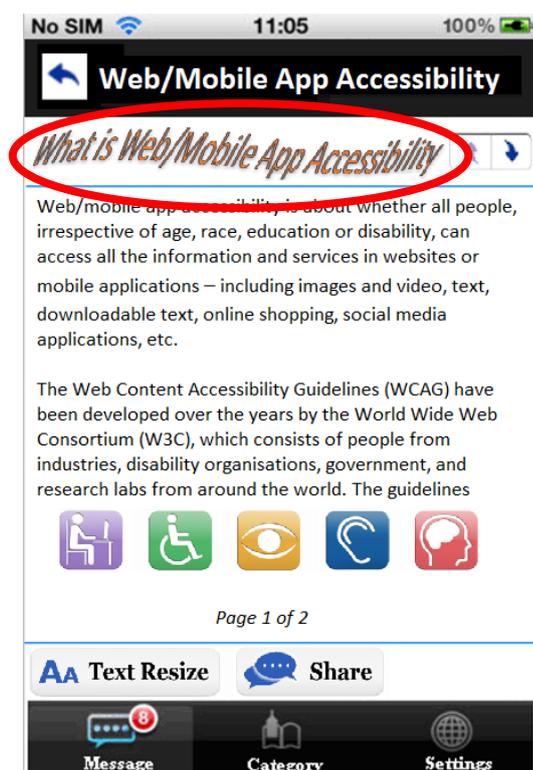


The text description enable users reading the mobile application screen page to know what the image is about and what it is supposed to illustrate.

W3C WCAG Reference: [1.1.1 Non-text Content](#)

Best Practice 1.2 – Avoid images of text**Advanced level**

Where possible, do not use images to display textual information. Accessibility tools like screen readers cannot read the text encapsulated inside an image unless a text alternative is provided for the image.

Before Rectification

The heading above is a graphic text. It cannot be read by screen readers.

After Rectification

The above text heading is not a graphic image. It can be read correctly by screen readers.

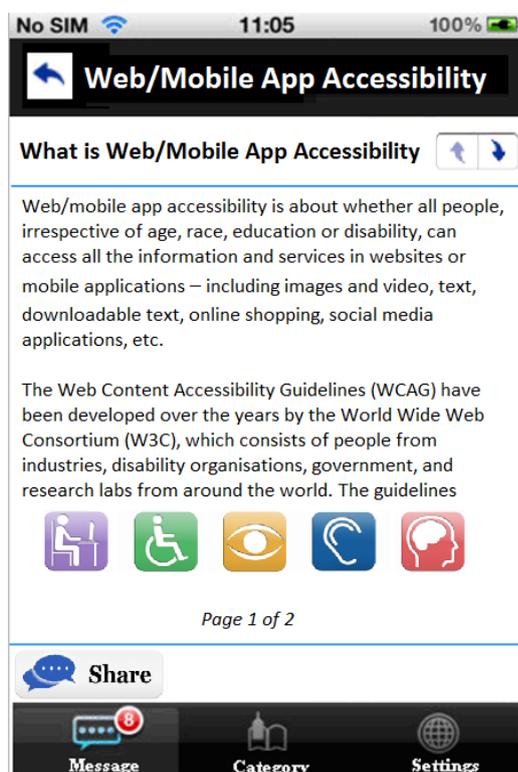
W3C WCAG Reference: [1.4.5 Images of Text](#)

Best Practice 1.3 – Provide text resize function to scale up text size or zoom support function (or work well with device’s zoom feature) without loss of content **Baseline level**

Ensure all text can be resized without loss of content or functionality by providing text resize function, or zoom support function (or work well with device’s built-in zoom feature).

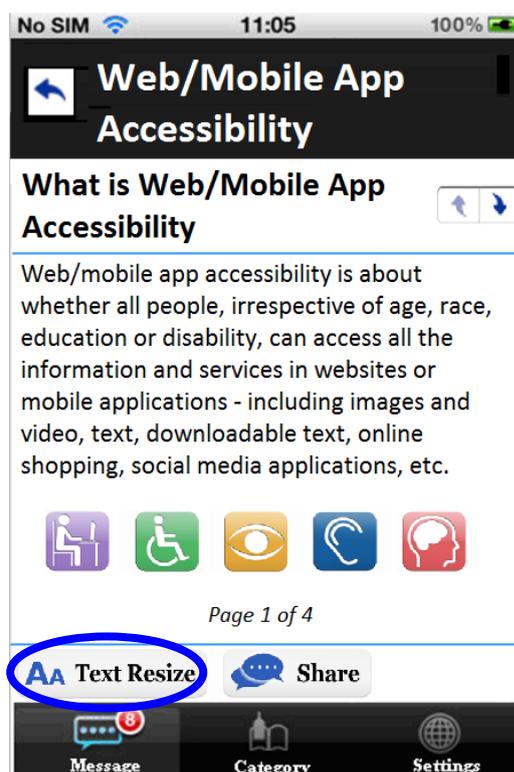
As a good practice, always provide a text resize function to facilitate reading wherever practicable. In this way, persons with mild visual impairment can read the content without using assistive technologies such as a screen magnifier.

Before Rectification



In this example, there is no text resize function.

After Rectification



By adding a text resize function, text size can be easily scaled up to meet individual needs.

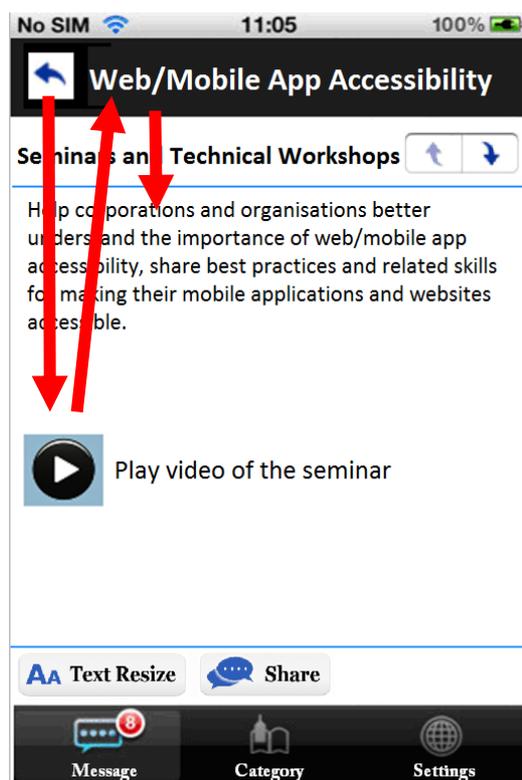
W3C WCAG Reference: [1.4.4 Resize Text](#)

Best Practice 1.4 – Provide meaningful content sequence

Baseline level

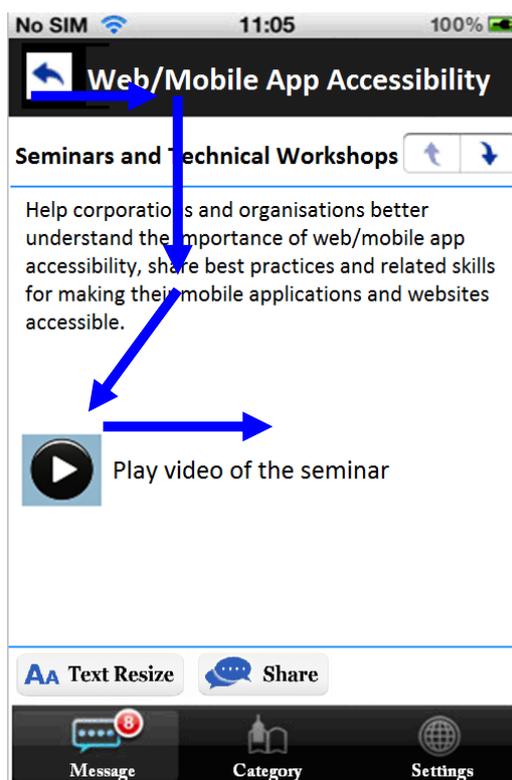
If the content needs to be read in a certain order, ensure the mobile application is designed and coded with the screen layout in a logical order.

Before Rectification



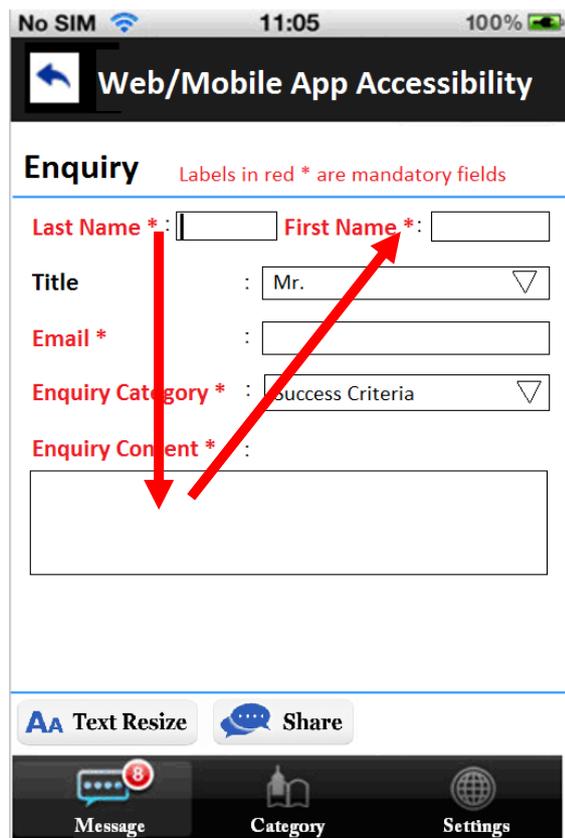
In this example, the screen page has been coded in such a way that the screen readers will read the headings and content in a wrong order.

After Rectification

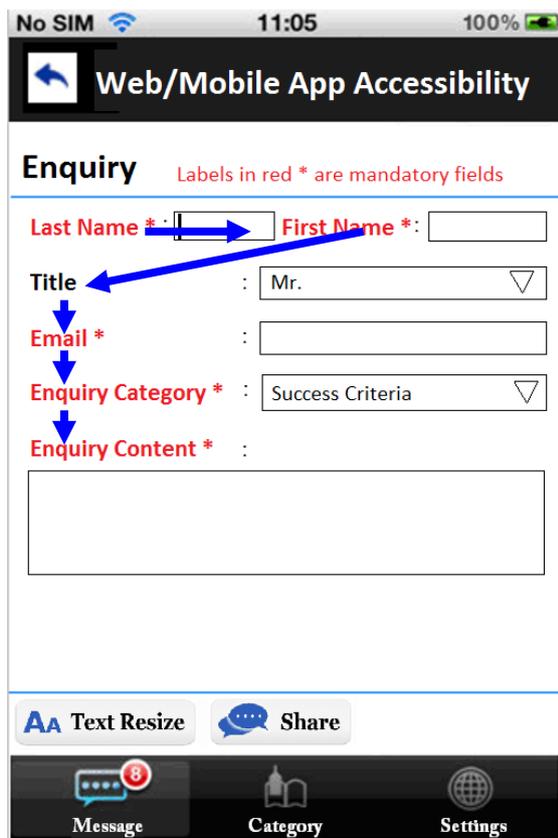


If the screen page is correctly coded, the reading order for screen readers will be from top to bottom and left to right.

Before Rectification



After Rectification



In this example, the form has been coded so that the order goes from Last Name, to Title, to Email, to Enquiry Category, to Enquiry Content, then go back to First Name. This is not intuitive to a user.

With proper coding, the form can be read aloud by screen readers in a logical manner so that the order goes from Last Name, to First Name, to Title, to Email, to Enquiry Category, then to Enquiry Content.

W3C WCAG Reference: [1.3.2 Meaningful Sequence](#), [2.4.3 Focus Order](#)

Best Practice 1.5 – Do not solely rely on sensory characteristics for instructions

Advanced level

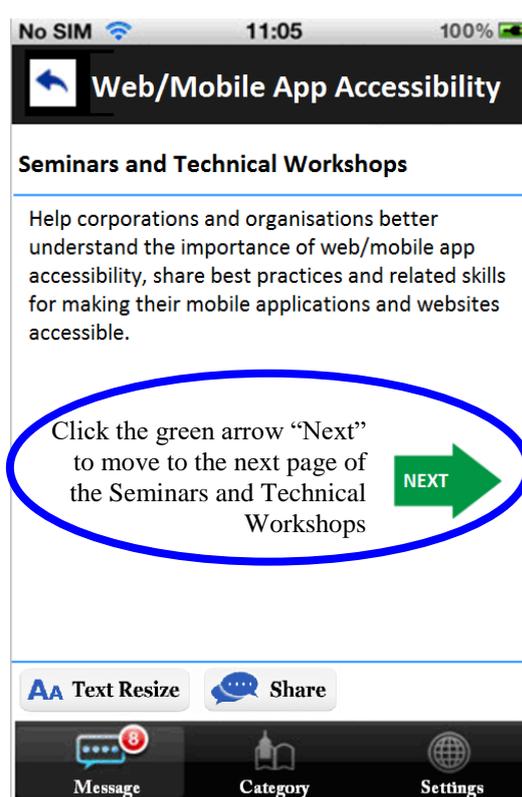
Do not rely solely on sound, shape, size or visual location to provide user instructions.

Before Rectification



In this example, it is only understandable to a person who can see that next page button. This will not be understandable to a visually impaired person.

After Rectification



The correct way to do this is to also label the button or provide a text alternative to the button and ensure clear instructions are in place to tell people which button to use and how to use it.

W3C WCAG Reference: [1.3.3 Sensory Characteristics](#)

Best Practice 1.6 – Avoid solely rely on colours to convey information

Advanced level

Do not rely solely on colours to convey information. It is impossible to be sure that everybody perceives colours in the same way (for example those persons with visual impairment or colour blindness), and information seems obvious to one person may be omitted by another.

Before Rectification

Enquiry Labels in red are mandatory fields

Last Name : First Name :

Title : Mr.

Email :

Enquiry Category : Success Criteria

Enquiry Content :

AA Text Resize Share

Message Category Settings

In this example, labels in red are mandatory fields, however, a colour blind or visually impaired person may not be able to detect this colour difference.

After Rectification

Enquiry Labels in red * are mandatory fields

Last Name * : First Name * :

Title : Mr.

Email * :

Enquiry Category * : Success Criteria

Enquiry Content * :

AA Text Resize Share

Message Category Settings

By adding asterisk (*) after each label, people who cannot perceive colours can still differentiate the mandatory fields.

W3C WCAG Reference: [1.4.1 Use of Colour](#)

Best Practice 1.7 – Provide sufficient colour contrast

Baseline level

Choose appropriate text and background colours so that they have a contrast ratio of at least 4.5:1 to make the text easy to read.

Before Rectification



In this example, the heading text in purple colour with background colour of black has poor contrast, making it hard to read.

After Rectification



When higher contrast text is used, the text is much easier to read. There are colour contrast checkers available online that can assist mobile application developers to perform colour contrast test.

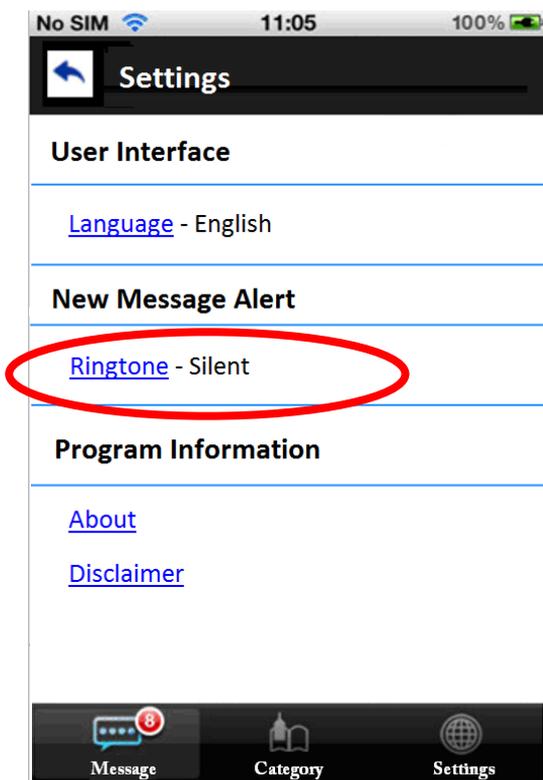
W3C WCAG Reference: [1.4.3 Contrast \(Minimum\)](#)

Best Practice 1.8 – Provide alternative means for notification

Baseline level

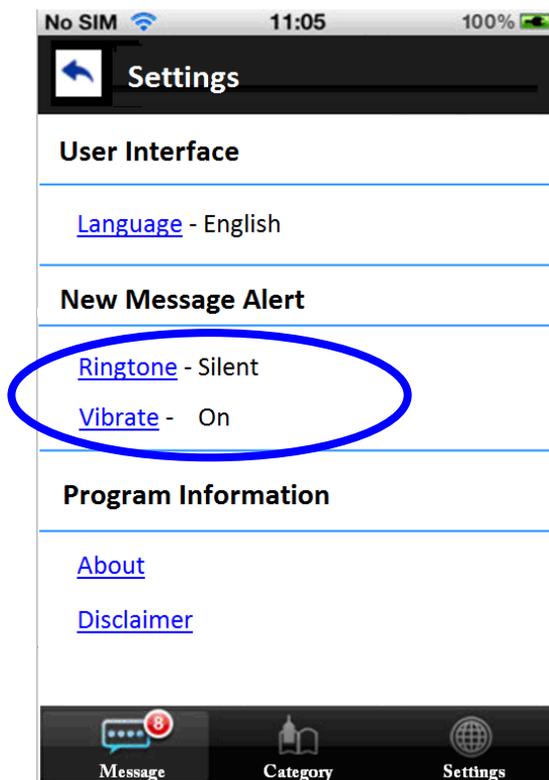
Ensure more than one means for notification that can be received by persons with visual and/or hearing impairments.

Before Rectification



This example only provides “Ringtone” option for the alert of new messages. Persons with hearing impairment cannot receive the alert.

After Rectification



This allows users to select “Ringtone” and/or “Vibrate” options for the alert of new messages. Both persons with visual and/or hearing impairments can receive the alert.

W3C WCAG Reference: N/A

Best Practice 1.9 – Provide description for prerecorded videos

Baseline level *

When a video with audio is viewed using a mobile application, a person with visual impairment can hear the audio but cannot see the picture. As a result, that person can have access to part of the information only. In fact, it should provide additional description that explains what is happening in the video. To a higher level of accessibility, an audio description of the video should be provided to describe the actions, characters, scene changes, and on-screen text that are important but are not described or spoken out in the main sound track.

* To attain the Baseline level, mobile applications are required to adopt any of the best practices 1.9, 1.10 and 1.11.

Before Rectification



This example shows a plain video without any alternative means to aid persons with visual impairment in understanding the content of the video in full.

After Rectification



An additional description that explains what is taking place in the video should be provided so that persons with visual impairment can understand the video contents by using screen readers.

W3C WCAG Reference: [1.2.1 Audio-only and Video-only \(Prerecorded\)](#),
[1.2.5 Audio Description](#)

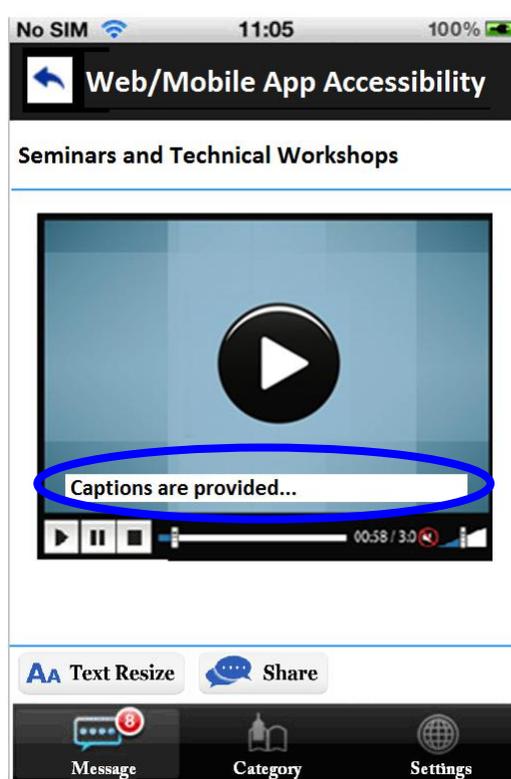
Best Practice 1.10 – Provide captions for videos **Baseline level ***

Provide captions for all videos so that they are accessible by persons with hearing impairment. Captions not only present the content of a conversation but also the important cues and surrounding noises.

* To attain the Baseline level, mobile applications are required to adopt any of the best practices 1.9, 1.10 and 1.11.

Before Rectification

For a video without caption, it is not perceivable by persons with hearing impairment.

After Rectification

Captions as shown in the example above should be provided so that persons with hearing impairment can fully understand the content of the video.

W3C WCAG Reference: [1.2.2 Captions \(Prerecorded\)](#), [1.2.4 Captions \(Live\)](#)

Best Practice 1.11 – Provide sign language for prerecorded videos

Baseline level *

Sign language is a method universally used by persons with hearing impairment to communicate. This provides the ability to reflect emotion, intonation and other audio information that may be limited when using captions. Sign language should be provided for videos as far as practicable. Resolution of the sign language picture should be sufficient to show clearly the facial expression and the finger movements of the translator.

* To attain the Baseline level, mobile applications are required to adopt any of the best practices 1.9, 1.10 and 1.11.

Before Rectification



Simply having the video with a transcript or captions may not satisfy all users especially those illiterate persons with hearing impairment.

After Rectification



An ideal way is to translate this information through sign language as shown above.

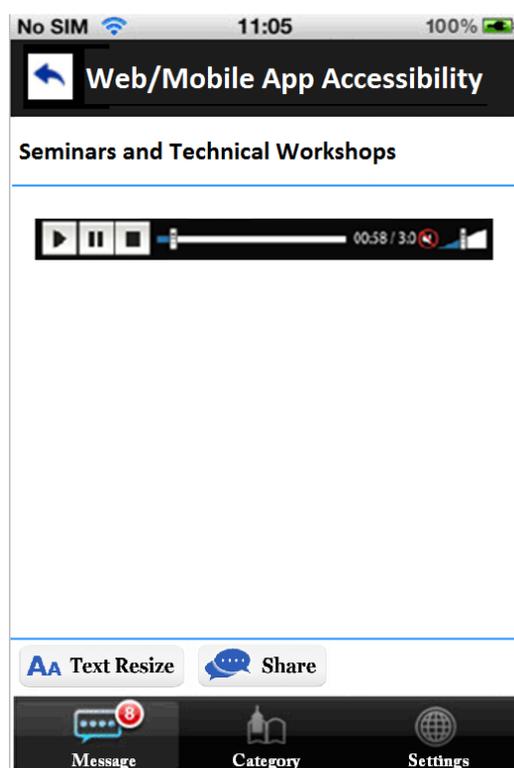
W3C WCAG Reference: [1.2.6 Sign Language \(Prerecorded\)](#)

Best Practice 1.12 – Provide alternatives for audio-only information

Advanced level

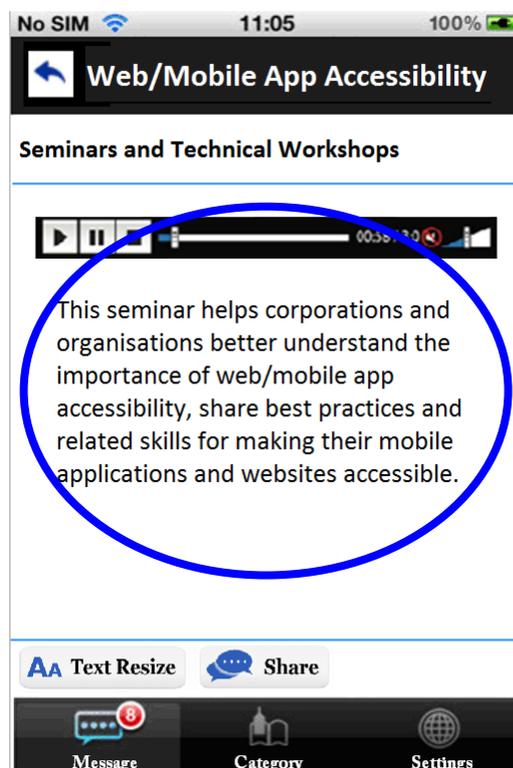
Make prerecorded audio accessible by providing alternatives that present essentially the same information to people who cannot access the original piece. For example, persons with hearing impairment cannot hear audio and need a way to get the same information.

Before Rectification



The above example provides an audio function alone which is not accessible to persons with hearing impairment.

After Rectification



In addition to the audio function, a transcript of this audio is also provided to allow persons with hearing impairment to read the audio contents.

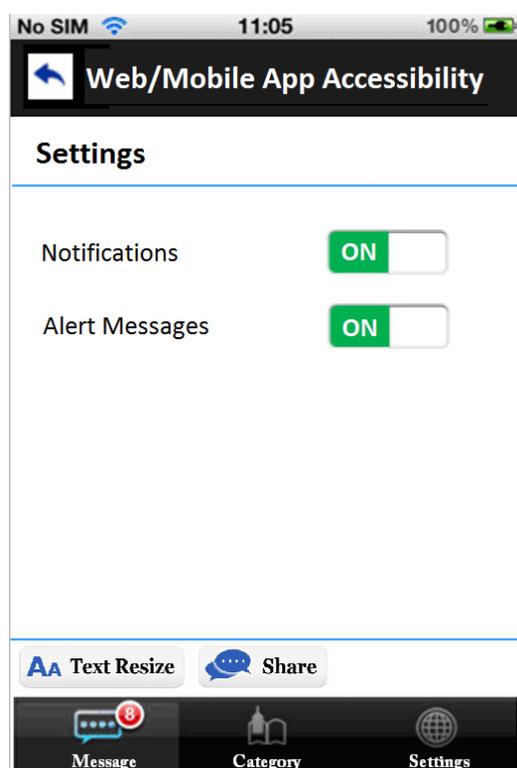
W3C WCAG Reference: [1.2.3 Audio Description or Media Alternative \(Prerecorded\)](#)

Best Practice 1.13 – Easy to turn off background sound or set as user-initiated only Baseline level

Background sound playing should be user-initiated, or at least there is a convenient navigation option to turn off the auto-playing audio or a setup function should be provided in the mobile application for user to control the audio.

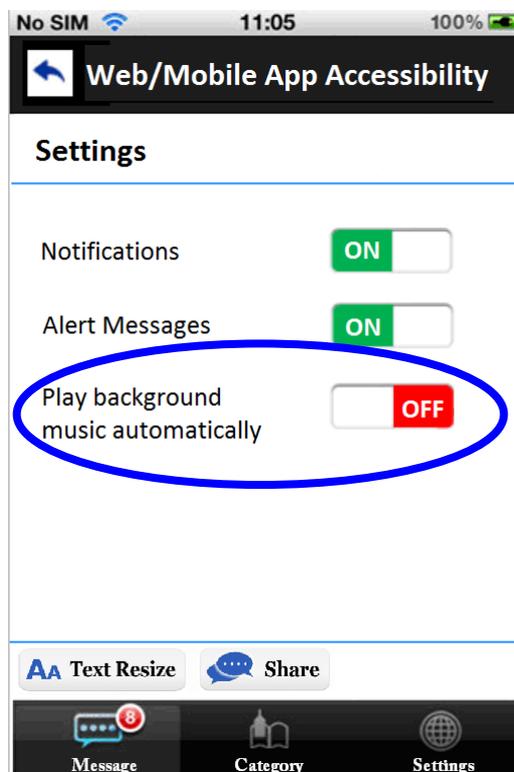
Sound that plays automatically in a mobile application can be annoying to some people, in particular for people who are using screen readers to listen to the screen contents.

Before Rectification



In this example, background sound play automatically and there is no option to control or turn off.

After Rectification



By adding an option in the "Settings" panel, user can control the background music.

W3C WCAG Reference: [1.4.2 Audio Control](#)

Best Practice 2 – Operable

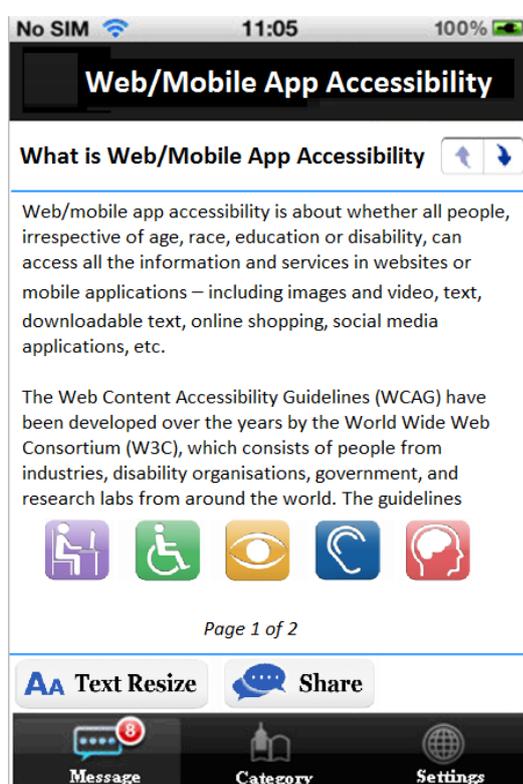
User interface components and navigation must be operable.

Best Practice 2.1 – Provide navigation controls for going backward

Baseline level

Provide navigations across all screen pages of a mobile application so that it helps users with cognitive limitations, low vision and intellectual disabilities operate in a comfortable way.

Before Rectification



In this example, navigation buttons are not provided for users to go back among the screen pages.

After Rectification

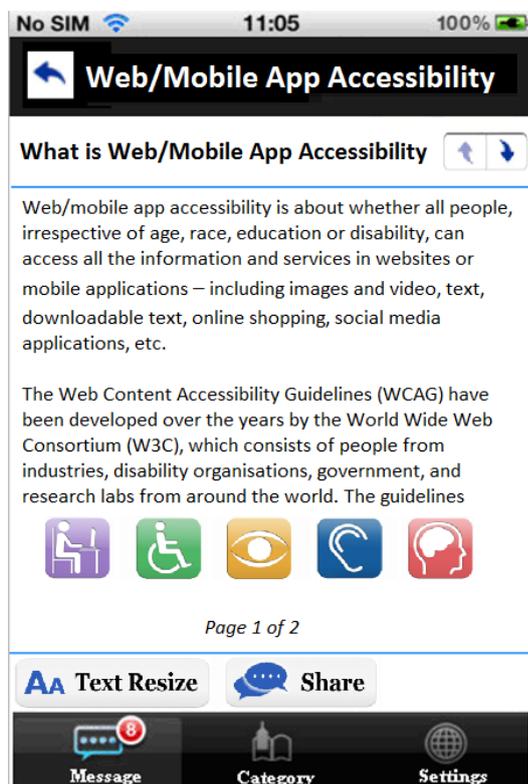


Users can go back to the previous screen page easily by using the back arrow button.

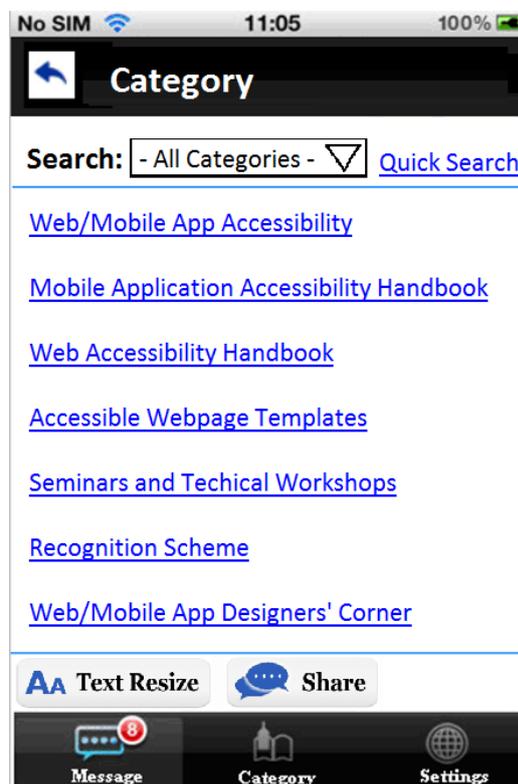
W3C WCAG Reference: N/A

Best Practice 2.2 – Provide multiple ways**Advanced level**

Ensure there is more than one way to access a page of a mobile application, for example, by using a search function, category feature, standard navigation, etc.

Before Rectification

The only way to navigate around this mobile application is through the main navigation screen.

After Rectification

It is a good practice to provide a function that allows to search and display content by category so that users can locate the required information.

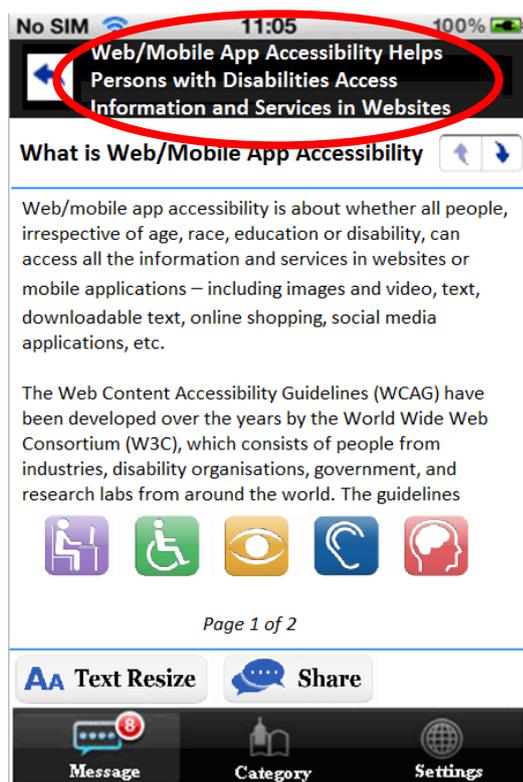
W3C WCAG Reference: [2.4.5 Multiple Ways](#)

Best Practice 2.3 – Provide clear and simple headings

Baseline level

Ensure to use clear and simple heading and content in the mobile applications so that persons with or without disabilities can easily understand.

Before Rectification



The above heading in the page is too long.

After Rectification



Due to the limitation of screen size, it is a good practice to provide simple and clear heading as well as text content in a mobile application.

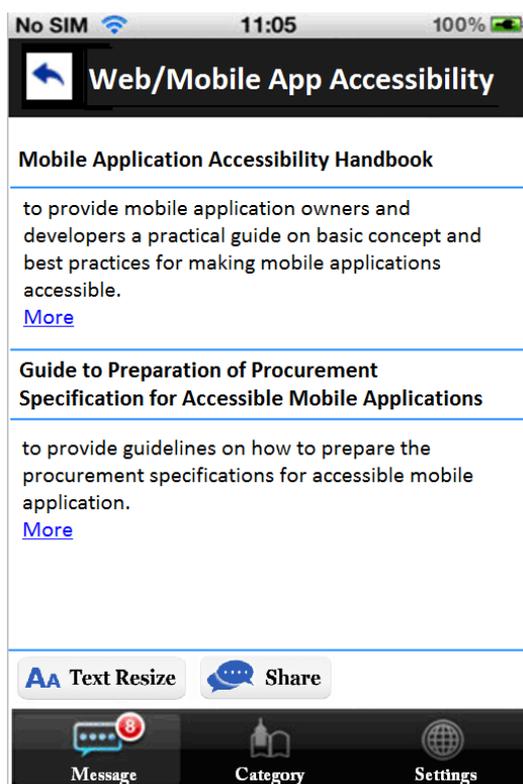
W3C WCAG Reference: [2.4.6 Headings and Labels](#)

Best Practice 2.4 – Provide clear and informative links

Baseline level

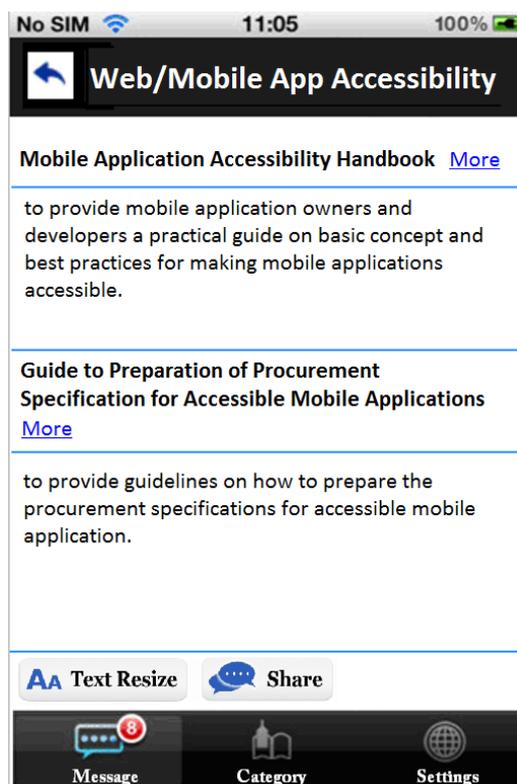
Write descriptive link text to ensure the purpose of each link can be understood by the text alone, or by the link text and the context.

Before Rectification



In this example, the link “More” is ambiguous and does not really convey much meaning.

After Rectification



Link labels should be more descriptive and self-explanatory as shown in the rectified version above.

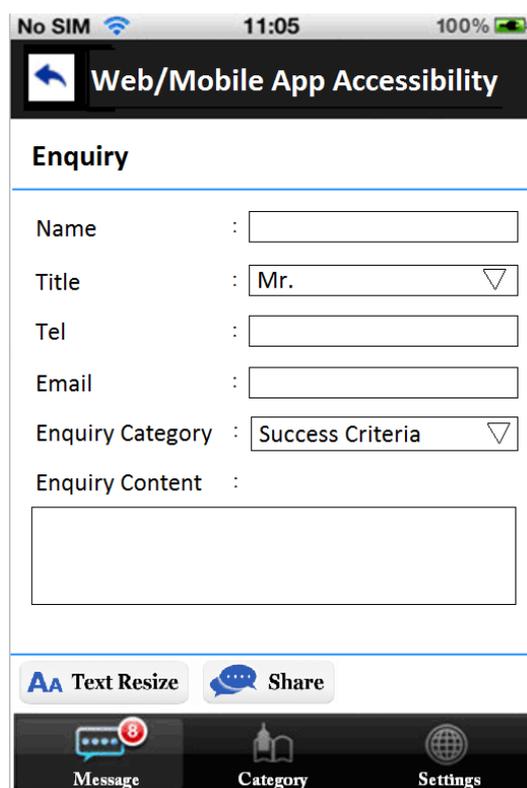
W3C WCAG Reference: [2.4.4 Link Purpose \(In Context\)](#)

Best Practice 2.5 – Provide focus visible

Advanced level

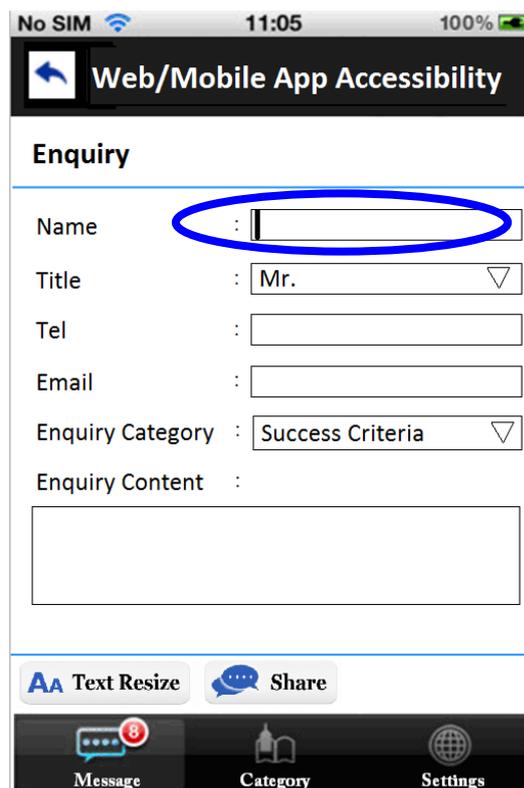
When a “text field” is selected, ensure that the focus has been moved into the “text field”. Users’ attention should be directed towards the important information and the corresponding input field.

Before Rectification



In this example, there is no way to determine which field gets the focus.

After Rectification



This example ensures that the focus is visible. This helps those users with low vision or visual impairment know where they are on an input page.

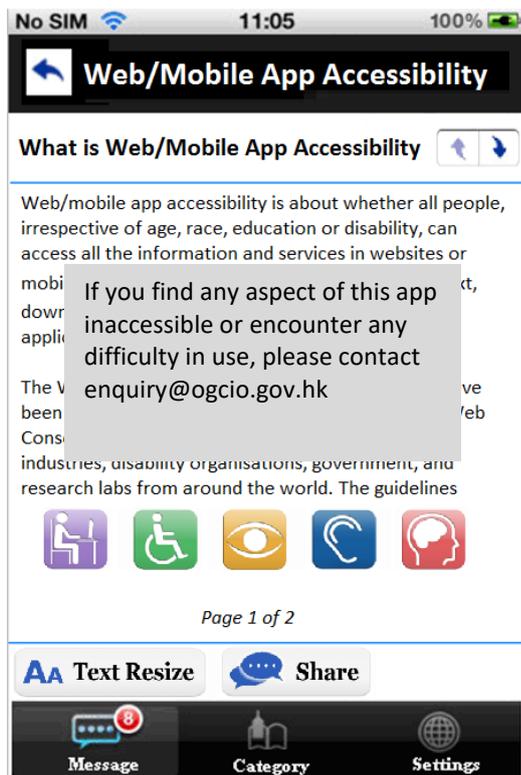
W3C WCAG Reference: [2.4.7 Focus Visible](#)

Best Practice 2.6 – Provide means to close popovers

Baseline level

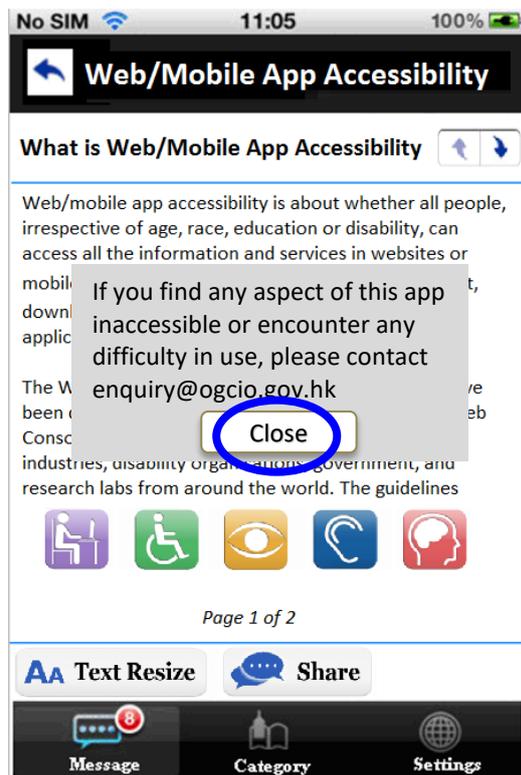
Ensure all popovers can be closed by buttons and can be accessible by screen reader.

Before Rectification



In this example, there is no means to close the popover.

After Rectification



It is a good practice to provide a close button for closing the popover.

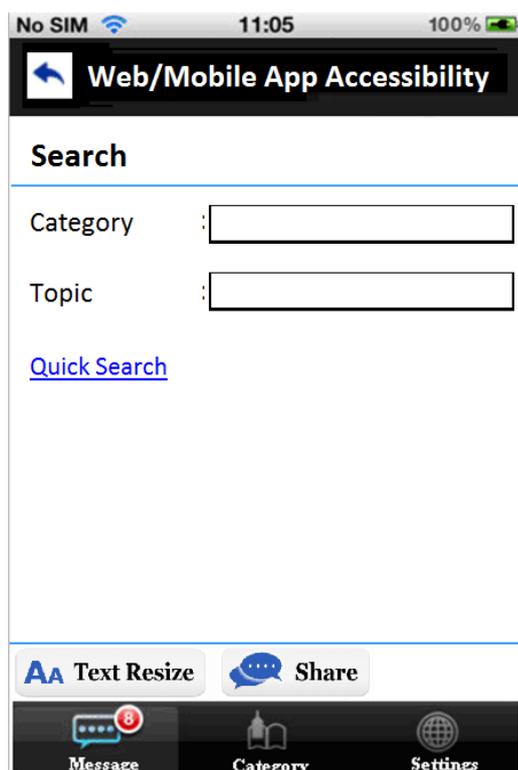
W3C WCAG Reference: [2.1.2 No Keyboard Trap](#)

Best Practice 2.7 – Minimize user input

Advanced level

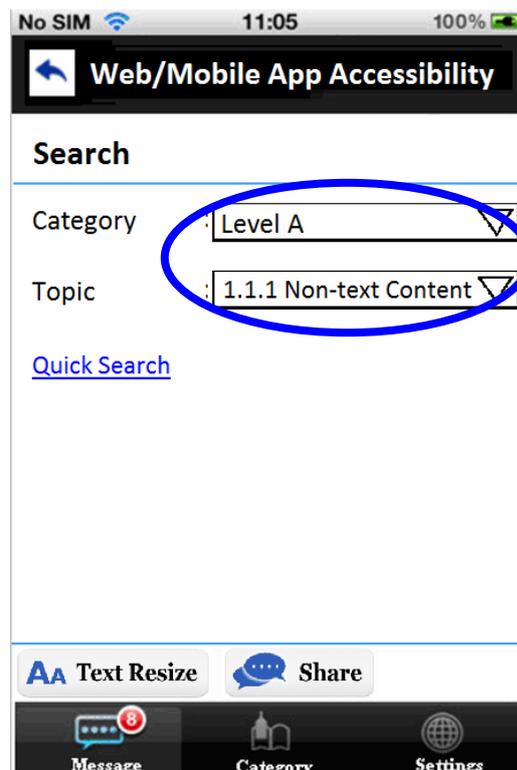
If appropriate and possible, use the components such as selection lists, pickers, default values for the text input fields and other controls that do not require much typing. Auto-display of appropriate field values for selection (e.g. range of numbers for age input field, “@” sign for email address field) can help user input with less keystrokes.

Before Rectification



In the above search form, all fields require user's input.

After Rectification



It is a good practice to use selection list in the search form in order to minimize user input and avoid unnecessary input error.

W3C WCAG Reference: N/A

Best Practice 2.8 – Make all clickable objects large enough to be tapped

Baseline level

Ensure all clickable objects such as buttons and links are large enough to be tapped by persons with upper limb and hand mobility problem. In order to provide an optimal viewing and usage experience, the size of clickable objects should not be smaller than the device's default icon size.

Before Rectification



In this example, persons with upper limb and hand mobility problem may not be able to tap the small buttons easily.

After Rectification



Bigger sized buttons allow users to tap and operate the functions of a mobile application comfortably.

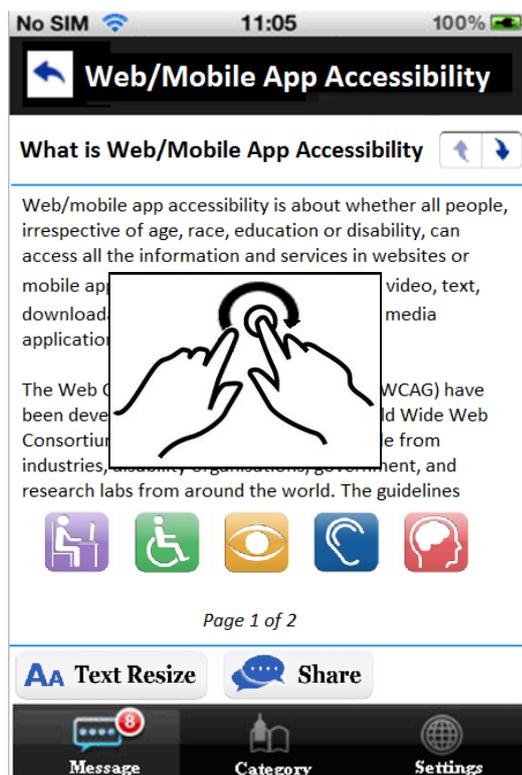
W3C WCAG Reference: N/A

Best Practice 2.9 – Provide simple gesture

Advanced level

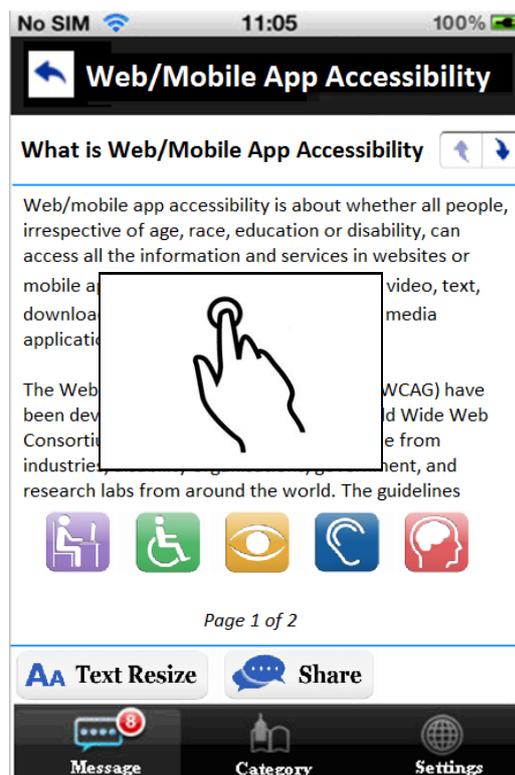
Ensure all gestures in a mobile application are simple enough to be operated by persons with upper limb and hand mobility problem.

Before Rectification



In this example, persons with upper limb and hand mobility problem may not be able to complete a complicated touch command in a mobile application with the above gesture.

After Rectification



Implementing simple gestures allows people to complete a touch command to operate a mobile application easily.

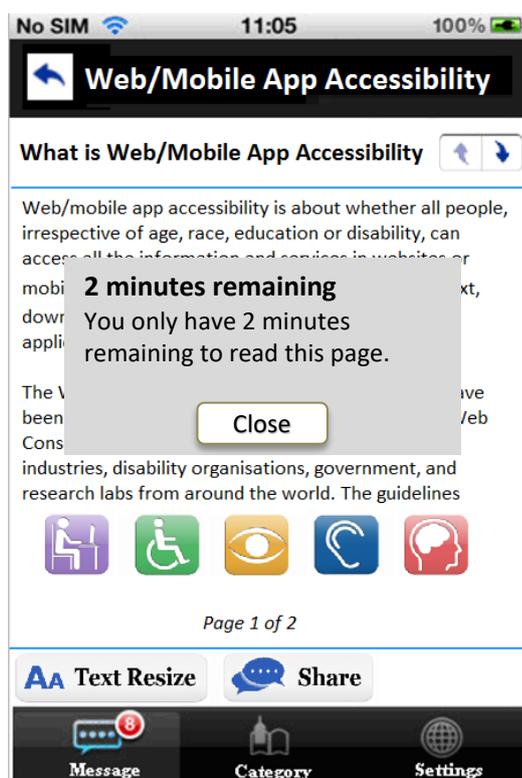
W3C WCAG Reference: N/A

Best Practice 2.10 – Provide sufficient time for users to read the content and operate a function

Baseline level

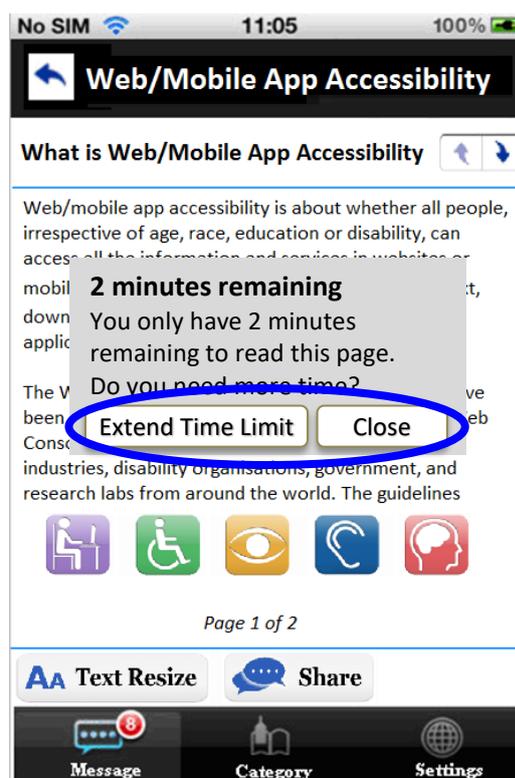
Ideally ensure all functions in a mobile application are not time dependent. If they are, ensure persons with disabilities can either adjust or stop the time limit so that they can have enough time to complete their task.

Before Rectification



This example only warns the user that time is about to expire.

After Rectification



A better approach is to allow the user to extend the time limit.

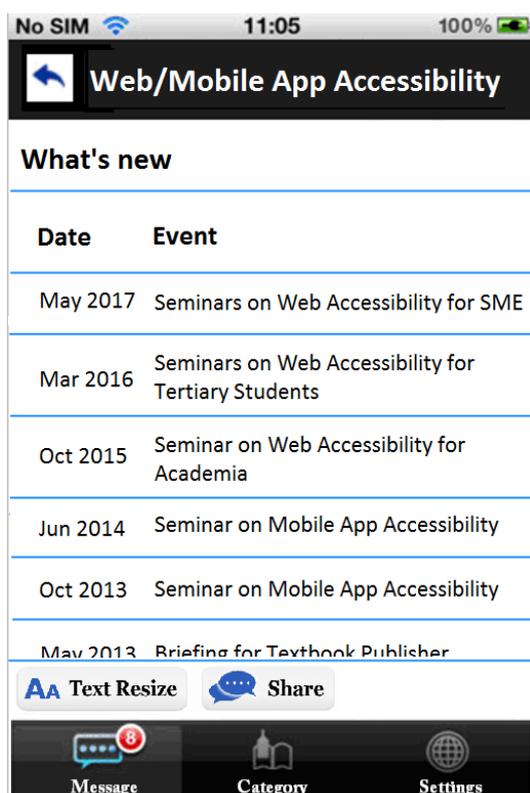
W3C WCAG Reference: [2.2.1 Timing Adjustable](#)

Best Practice 2.11 – Lists with user-initiated auto-updating

Advanced level

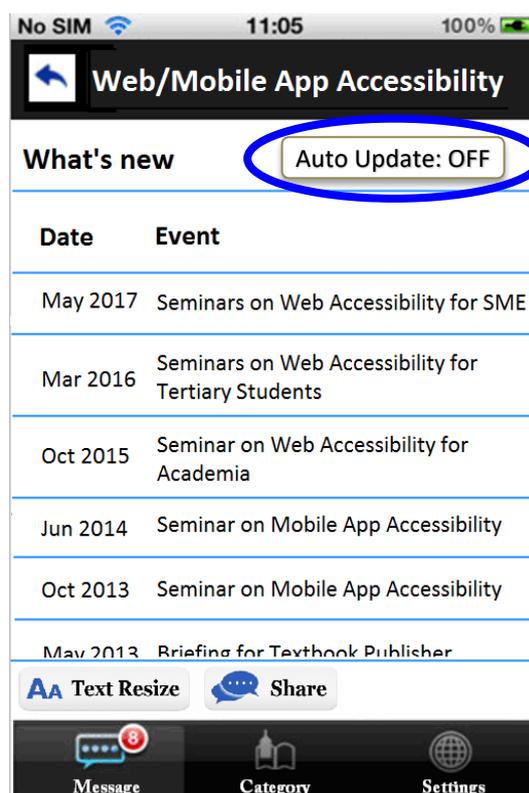
For content that moves automatically for more than five seconds or is updated automatically, there needs to be a way to stop this movement and stop the content from auto-updating, blinking or scrolling. Application should provide an option for users to control and stop the auto-updating operation.

Before Rectification



In this example, the list is designed to update automatically with content changes, which can be very frustrating for people using screen readers.

After Rectification

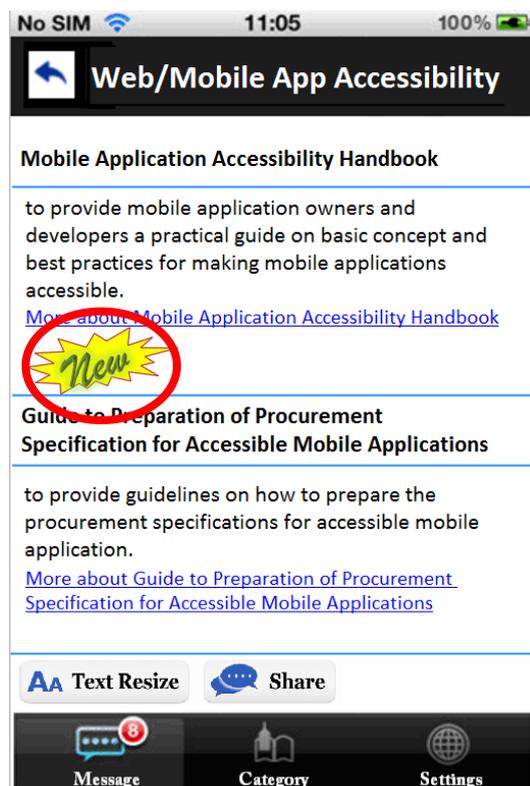


By providing a function to turn off the auto-updating, the list contents will not be refreshed frequently and easier for persons with disabilities to access.

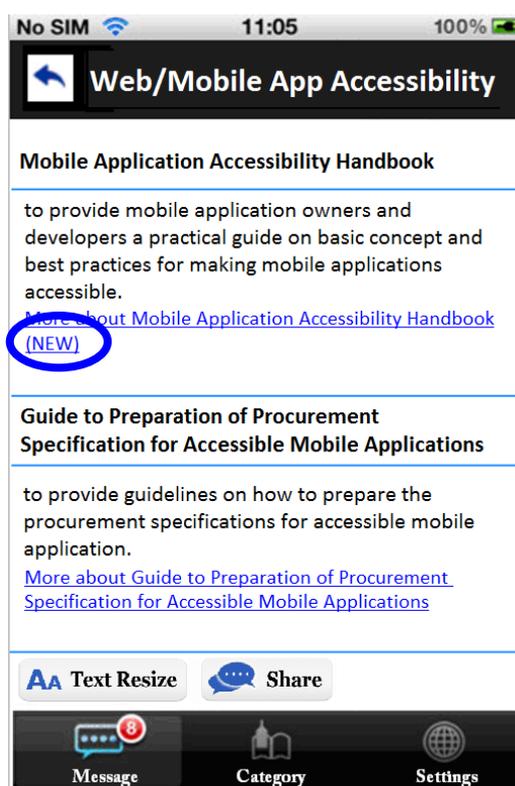
W3C WCAG Reference: [2.2.2 Pause, Stop, Hide](#)

Best Practice 2.12 – Provide three flashes or below threshold**Advanced level**

Ensure all flashing items are dimmed, and cover only a small area of the screen or the flashing rate is three times per second or less. Otherwise, this may cause problems for persons who suffer from epilepsy.

Before Rectification

In this example, the “New” image is flashing too fast and too bright. This content can cause seizures for people who are suffering from epilepsy.

After Rectification

It is better to replace the flashing content with static content, or ensure the flashing object only take up a very small portion of the page and the flashing rate is less than three times per second.

W3C WCAG Reference: [2.3.1 Three Flashes or Below Threshold](#)

Best Practice 3 – Understandable

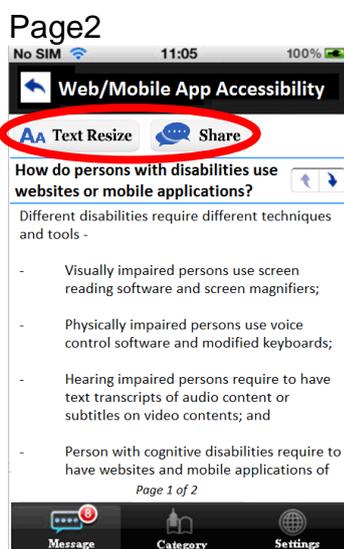
Information and the operation of user interface must be understandable.

Best Practice 3.1 – Provide consistent and simple user interface structure

Baseline level

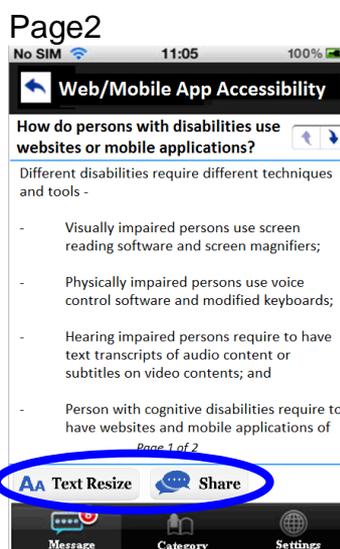
Ensure to provide consistent and simple navigation structure for the user interface.

Before Rectification



Inconsistent layout and placement of buttons across multiple screen pages will cause confusion.

After Rectification



Consistent screen layout design enhances user experience.

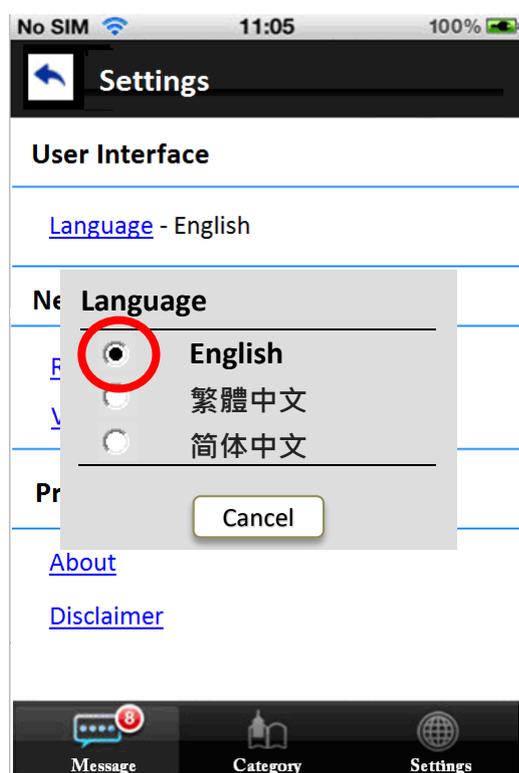
W3C WCAG Reference: [3.2.3 Consistent Navigation](#)

Best Practice 3.2 – Avoid sudden change of context

Advanced level

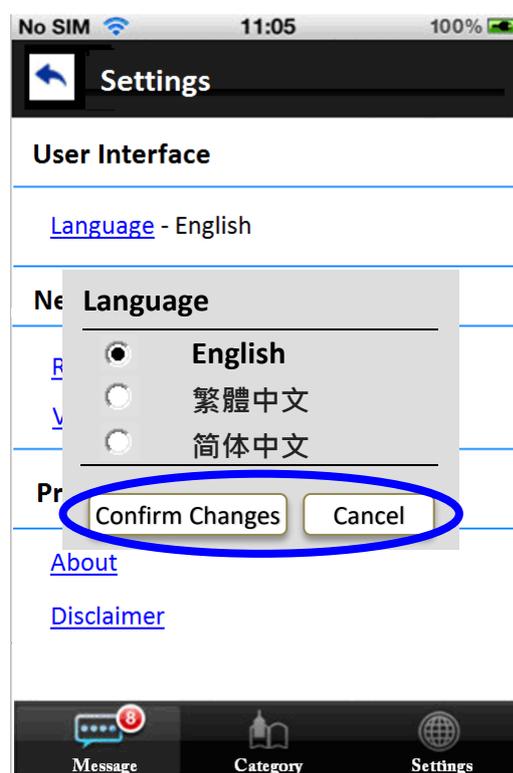
When activating a change on a mobile application such as switching to a new screen page or refreshing content, ensure to provide users with the corresponding notification or warning.

Before Rectification



This example is common in mobile applications that, when selecting an option, the request is automatically submitted.

After Rectification



This implementation is better as it gives the user total control when to submit the request to effect the change.

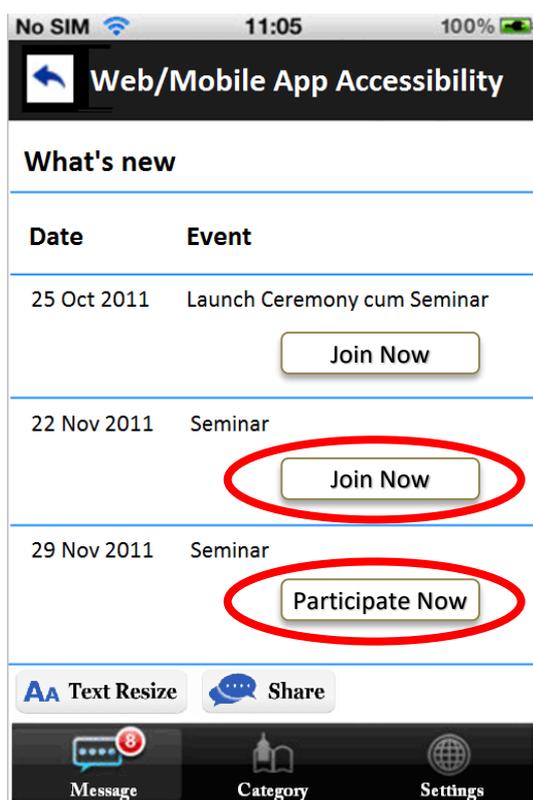
W3C WCAG Reference: [3.2.2 On Input](#)

Best Practice 3.3 – Provide consistent identification

Baseline level

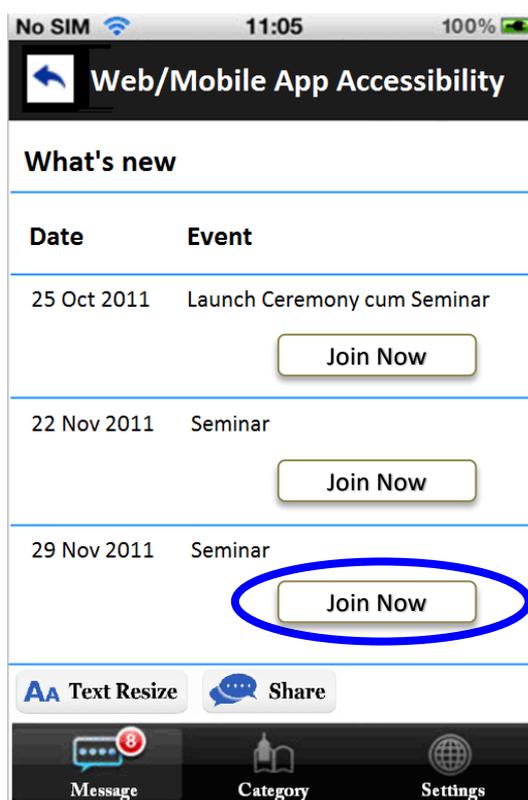
For all items that have the same functionality, ensure they use the same label.

Before Rectification



In this example, two buttons each having a different label will cause confusion to users, especially for those using screen readers, who may have doubt whether these two buttons have the similar behaviour.

After Rectification



The “Join Now” buttons are made consistent and it is clear that both would have the same behaviour.

W3C WCAG Reference: [3.2.4 Consistent Identification](#)

Best Practice 3.4 – Provide error identification **Baseline level**

If a user makes a mistake, use text to show where and what has done wrong, and provide hint for how to fix it.

Before Rectification



In this example, an error has been identified but without specific error message prompted.

After Rectification



More specific error message should be given so that users know exactly what is wrong.

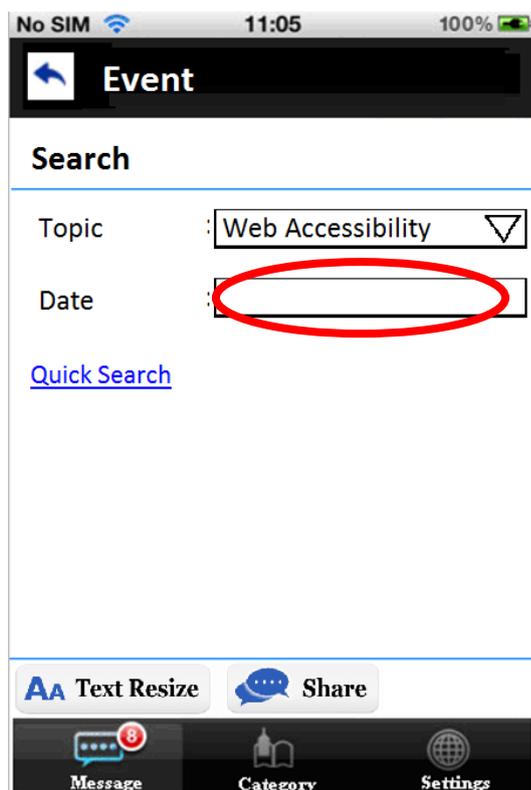
W3C WCAG Reference: [3.3.1 Error Identification](#)

Best Practice 3.5 – Provide input assistance such as proper labels or instructions for user input

Baseline level

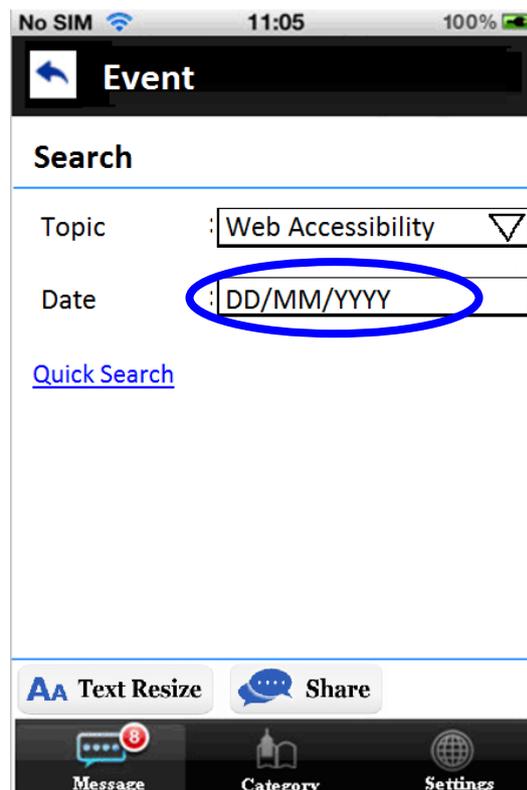
Ensure that all input components including labels, tabs, buttons and text fields must have a meaningful label or instruction so that it can be read by screen readers.

Before Rectification



There is no input hint on what is the required date format.

After Rectification



By adding input hint or instruction to the field, user can know what the mobile application component is about and what to enter.

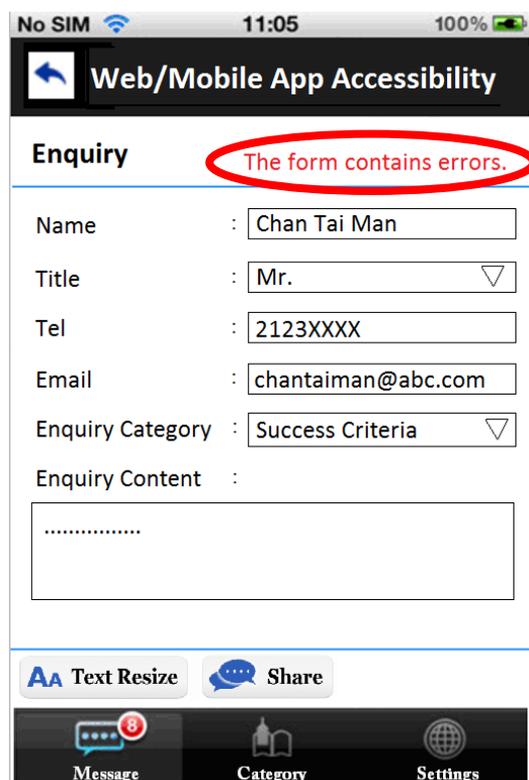
W3C WCAG Reference: [3.3.2 Labels or Instructions](#)

Best Practice 3.6 – Provide error suggestion

Baseline level

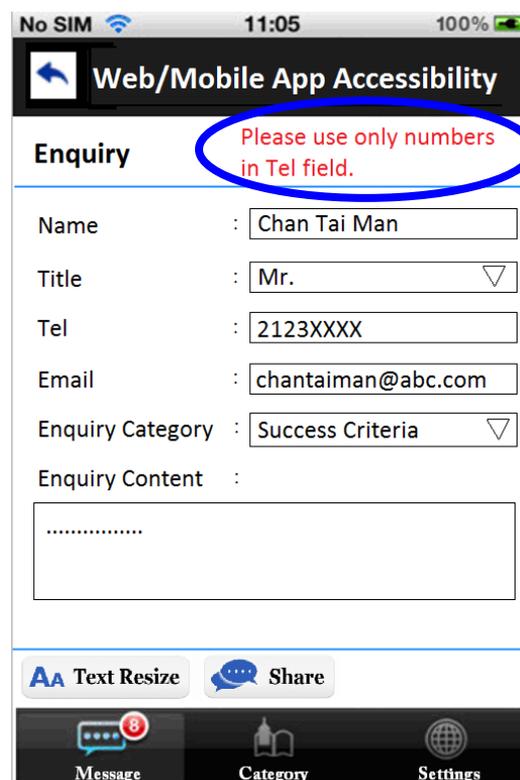
When a user makes an input error, always provide the user with a suggestion to fix the error if ever possible.

Before Rectification



The example above shows an error message that is not helpful enough because it does not provide an adequate description of what needs to be corrected.

After Rectification



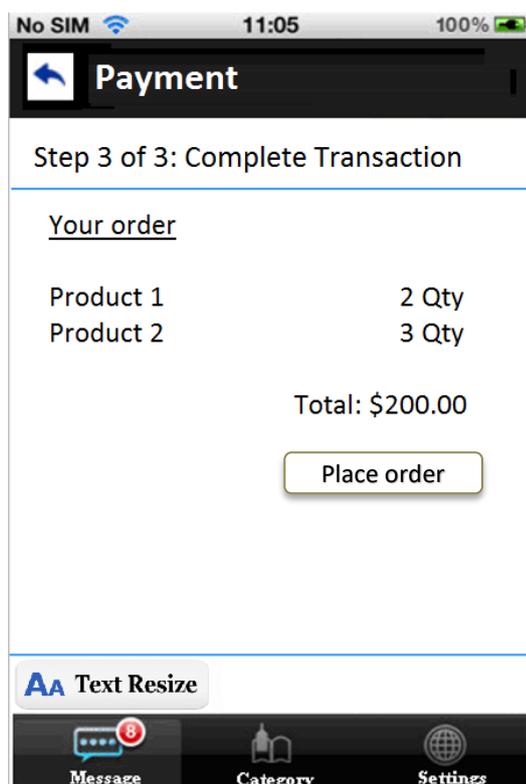
In contrast, this example shows a message that provides a good explanation of what needs to be corrected.

W3C WCAG Reference: [3.3.3 Error Suggestion](#)

Best Practice 3.7 – Provide means for error prevention (legal, financial, data) Baseline level

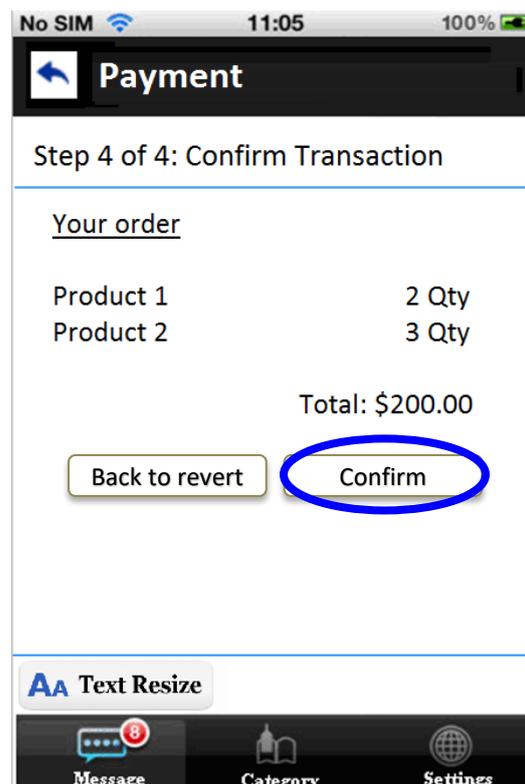
If a user has to submit data that have legal or financial consequences, make sure the system allows the user to check and confirm the information before submission, or reverse the transaction after submitting.

Before Rectification



This indicates the last step of a transaction, in which the user is forced to place the order without a confirmation process.

After Rectification



It is better to allow the user to first confirm and give an option to change any of the details before the final submission.

W3C WCAG Reference: [3.3.4 Error Prevention \(Legal, Financial, Data\)](#)

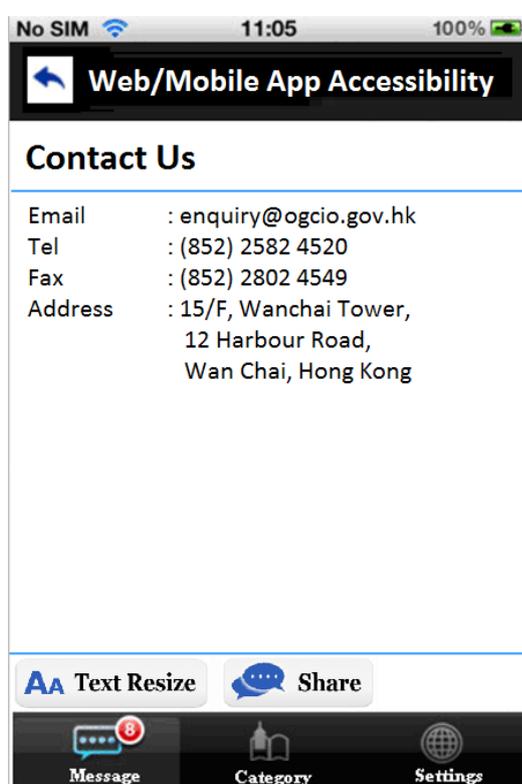
Best Practice 4 – Other

Best Practice 4.1 – Provide contact points or email feedback as well as an accessibility statement

Baseline level

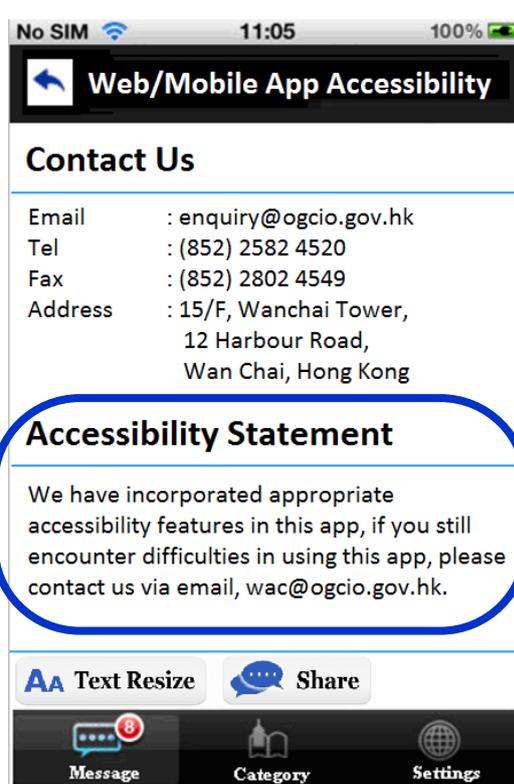
Ensure an accessibility statement with contact point is provided in the mobile applications. Declare your mobile application as accessible and provide relevant description in application market places for ease of searching.

Before Rectification



In this example, there is no accessibility statement.

After Rectification



It is a good practice to provide an accessibility statement with contact point.

W3C WCAG Reference: N/A

5. Testing Strategy for Developers

5.1 Accessibility Inspection Tools

Some of accessibility issues can be detected using inspection tools and/or simulators. It is recommended to test the mobile application with these tools during development stages as early as possible.

Other testing techniques introduced below are also necessary to ascertain the accessibility of a mobile application.

5.2 Visual Review

A great deal can be learnt about the accessibility of a mobile application just by visual review while having in mind the following questions:

- Can the contents be easily read?
- Can the functions be easily controlled?

Walk-through a mobile application in full can reveal many potential accessibility issues faced by persons with disabilities. A checklist on Visual Review is provided in **Section 5.6 – Best Practice Checklist for Developers**.



Example Tools:

- Colour Contrast Analyser
- WCAG Contrast Checker

5.3 Manual Testing with Screen Readers

An easy way to experience how persons with visual impairment use a mobile application is to simply navigate and operate the mobile application using a screen reader. This testing also ensures compatibility of the mobile application with the screen reader.

- Navigate the mobile application and determine just how much information we can access through the use of screen readers.
- Try reading the headings, navigation menus, images and buttons and also test more complex features such as input forms, if applicable.



Example Tools:

- Narrator for Windows Mobile
- TalkBack for Android
- VoiceOver for iOS

A checklist on testing using screen readers is provided in **Section 5.6 – Best Practice Checklist for Developers**

5.4 Human Testing

The most thorough approach to ensure mobile application accessibility is to test a mobile application by persons with various disabilities. It is recommended to have the human testing before your mobile application is released. As this testing method requires more time and resources, it is better to first undertake the Visual Review and Manual Testing with Screen Reader to rectify as many accessibility issues as possible, and then adopt human testing at a later project stage to uncover more subtle issues.

Some non-government organisations (NGOs) supporting persons with disabilities can help by providing free or affordable human testing services. Mobile application owners and developers may contact these organisations for assistance.

Some web/mobile app design companies and NGOs providing testing services can be found at the following link -

<http://www.webforall.gov.hk/en/wdc>

5.5 Continual Improvement

Accessibility technology, testing tools and technical features in different operating systems are evolving steadily, verification test for accessibility conformance is required for continual improvement when there is a new release of your mobile application or an update of the mobile operating systems.

5.6 Best Practice Checklist for Developers

How to Use This Checklist

1. **Review** each of the best practice and “check off” all the best practices that DO NOT APPLY to the mobile application under testing, using the N/A column.
For example, if a mobile application does not have any multi-media content, then *Best Practice 1.9 – 1.13* can be marked with “N/A” and the Visual Review can be skipped. Other items marked as “Skip” can be ignored for the corresponding testing.
2. **Perform Visual Review** by checking all items listed in the visual review column.
3. Test using **Screen Readers**.

(A) Best Practice Checklist **Advanced level**

Best Practice	N/A	Visual Review	Screen Readers
1 Perceivable			
<i>Text related</i>			
1.1 Provide text alternatives for non-text contents	<input type="checkbox"/>	Skip	<input type="checkbox"/>
1.2 Avoid images of text	<input type="checkbox"/>	<input type="checkbox"/>	Skip
1.3 Provide text resize function to scale up text size or zoom support function (or work well with device’s zoom feature) without loss of content	<input type="checkbox"/>	<input type="checkbox"/>	Skip
1.4 Provide meaningful content sequence	<input type="checkbox"/>	Skip	<input type="checkbox"/>
<i>Sensory</i>			
1.5 Do not solely rely on sensory characteristics for instructions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.6 Avoid solely rely on colours to convey information	<input type="checkbox"/>	<input type="checkbox"/>	Skip
1.7 Provide sufficient colour contrast	<input type="checkbox"/>	<input type="checkbox"/>	Skip
1.8 Provide alternative means for notification	<input type="checkbox"/>	<input type="checkbox"/>	Skip
<i>Multi-media related</i>			
1.9 Provide description for prerecorded videos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.10 Provide captions for videos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.11 Provide sign language for prerecorded videos	<input type="checkbox"/>	<input type="checkbox"/>	Skip
1.12 Provide alternatives for audio-only information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.13 Easy to turn off background sound or set as user-initiated only	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 Operable			
<i>Navigation related</i>			
2.1 Provide navigation for going backward	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.2 Provide multiple ways	<input type="checkbox"/>	<input type="checkbox"/>	Skip
2.3 Provide clear and simple headings	<input type="checkbox"/>	<input type="checkbox"/>	Skip
2.4 Provide clear and informative links	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.5 Provide focus visible	<input type="checkbox"/>	<input type="checkbox"/>	Skip
<i>Control related</i>			
2.6 Provide means to close popovers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.7 Minimize user input	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Best Practice		N/A	Visual Review	Screen Readers
2.8	Make all clickable objects large enough to be tapped	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.9	Provide simple gesture	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.10	Provide sufficient time for users to read the content and operate a function	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.11	Lists with user-initiated auto-updating	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.12	Provide three flashes or below threshold	<input type="checkbox"/>	<input type="checkbox"/>	Skip
3 Understandable				
<i>User interface related</i>				
3.1	Provide consistent and simple user interface structure	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.2	Avoid sudden change of context	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.3	Provide consistent identification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Input related</i>				
3.4	Provide error identification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.5	Provide input assistance such as proper labels or instructions for user input	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.6	Provide error suggestion	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.7	Provide means for error prevention (legal, financial, data)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Other Best Practice				
4.1	Provide contact points or email feedback as well as an accessibility statement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(B) Best Practice Checklist **Baseline level**

Best Practice		N/A	Visual Review	Screen Readers
1 Perceivable				
<i>Text related</i>				
1.1	Provide text alternatives for non-text contents	<input type="checkbox"/>	Skip	<input type="checkbox"/>
1.3	Provide text resize function to scale up text size or zoom support function (or work well with device's zoom feature) without loss of content	<input type="checkbox"/>	<input type="checkbox"/>	Skip
1.4	Provide meaningful content sequence	<input type="checkbox"/>	Skip	<input type="checkbox"/>
<i>Sensory</i>				
1.7	Provide sufficient colour contrast	<input type="checkbox"/>	<input type="checkbox"/>	Skip
1.8	Provide alternative means for notification	<input type="checkbox"/>	<input type="checkbox"/>	Skip
<i>Multi-media related</i>				
1.9*	Provide description for prerecorded videos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.10*	Provide captions for videos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1.11*	Provide sign language for prerecorded videos	<input type="checkbox"/>	<input type="checkbox"/>	Skip
1.13	Easy to turn off background sound or set as user-initiated only	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 Operable				
<i>Navigation related</i>				
2.1	Provide navigation for going backward	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.3	Provide clear and simple headings	<input type="checkbox"/>	<input type="checkbox"/>	Skip
2.4	Provide clear and informative links	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Control related</i>				
2.6	Provide means to close popovers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.8	Make all clickable objects large enough to be tapped	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.10	Provide sufficient time for users to read the content and operate a function	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 Understandable				
<i>User interface related</i>				
3.1	Provide consistent and simple user interface structure	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Best Practice		N/A	Visual Review	Screen Readers
3.3	Provide consistent identification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<i>Input related</i>				
3.4	Provide error identification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.5	Provide input assistance such as proper labels or instructions for user input	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.6	Provide error suggestion	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.7	Provide means for error prevention (legal, financial, data)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Other Best Practice				
4.1	Provide contact points or email feedback as well as an accessibility statement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Notes:

- * To attain the Baseline level, mobile applications are required to adopt all the best practices in the above table while adopting any of the best practices 1.9, 1.10 and 1.11.

6. Related Resources

Technical Guidelines and Reference Information

Android - Designing for Accessibility

<http://developer.android.com/guide/topics/ui/accessibility/index.html>

Android - Accessibility Design Patterns

<http://developer.android.com/design/patterns/accessibility.html>

BlackBerry Java SDK Accessibility Developer Guide

http://developer.blackberry.com/bbos/java/documentation/intro_accessibility_198461111.html

iOS - Accessibility Programming Guide for iOS

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/iPhoneAccessibility/Introduction/Introduction.html>

iOS - Verification on Accessibility

<https://developer.apple.com/library/ios/technotes/TestingAccessibilityOfiOSApps/TestingtheAccessibilityofiOSApps/TestingtheAccessibilityofiOSApps.html>

iOS - Human Interface Guidelines

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html>

Windows Mobile - Accessibility

<https://msdn.microsoft.com/en-us/windows/uwp/accessibility/accessibility-overview>

<http://windows.microsoft.com/en-hk/windows-10/getstarted-make-your-device-easier-to-use-mobile>

W3C Mobile Web Application Best Practices

<http://www.w3.org/TR/mwabp/>

Accessibility Testing Tools

<https://www.paciellogroup.com/resources/contrastanalyser/>

<https://addons.mozilla.org/en-us/firefox/addon/wcag-contrast-checker>

<http://www.apple.com/accessibility/ios/>

<https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback>