Guidelines for the development of accessible mobile interfaces

Funka Nu.

Contents

Introduction 3

About the guidelines 3

Guidelines for the development of accessible mobile interfaces 4

Choice of solution 4

Design 4

Layout and design 6

Interaction 8

Content 9

User settings 10

Funka Nu AB 11

Introduction

Funka’s guidelines for the development of accessible mobile interfaces were financed by the Internet Fund .SE. Funka's methodology is developed in close collaboration with the disability movement and everything we recommend has been tested in real life. Funk’s work and services are based on the international Web Content Accessibility Guidelines 2.0 (WCAG 2.0). Our long experience of accessibility work and usertesting with users with various needs and abilities, with and without assistive technology, show that WCAG 2.0 is not enough to provide accessibility. We have therefore developed our own test criteria that supplement the international regulations. Funka has made the authorised translation of WCAG 2.0 to Swedish.

- Web Content Accessibility Guidelines 2.0 (WCAG 2.0)
- The authorised Swedish translation of WCAG 2.0
- World Wide Web Consortium (W3C)
- Web Accessibility Initiative (WAI)

Read more about Funka under the headline ”Funka Nu AB” at the end of this document.

About the guidelines

More and more people are using touch screens and traditional mobile devices. Most are designed to work well for users with disabilities, with or without the need for assistive technology. Consequently, it is increasingly necessary for those who develop apps and mobile interfaces to know more about accessibility and the different needs that users may have.

The international accessibility guidelines, WCAG 2.0, lacks development principles for mobile interfaces.

In the work on preparing the guidelines, we made an inventory of existing guidelines and studies on accessibility in mobile interfaces. We also conducted a survey of users of smartphones to find out what the problems were and what worked well. Most important, we interviewed and made user tests with users with various kinds of disabilities.

These guidelines are open and free to be used by everyone. We would be grateful for feedback and proposals for further development, clarifications and amendments.

Guidelines for the development of accessible mobile interfaces

Choice of solution

1. **Ensure that your basic website works on mobile devices** It does not need to work perfectly but it should be as simple as possible and it must be possible to manage all functionality via one mobile interface, as far as this is practically feasible. Examples of common problems here are menus that require the mouse pointer to rest over a menu option to open a submenu. A good strategy is 'mobile first', i.e. the design of the interface is based on the display on a mobile.

2. **Do not force the user to use a mobile version, but do offer one if the pages on the basic website are large or the functions are complex** Many users prefer a mobile version of websites. This applies in particular if the website is extensive and contains many graphic objects. There may be reasons for offering a mobile version, but do not force the user to use the mobile version. Offer links between the different versions and remember the user's choice.

3. **Any mobile version of the website must, where possible, give the user access to the same information and services as the standard website unless it is a clear mobile version of a specific limited service/function** If you offer a mobile version of the website, it must be possible to use it to do and read the same things as on the standard website but in a simpler format in which, for example, large quantities of information and choices are initially hidden instead of always being visible. An exception might be if it is a mobile version of a service/function, for example for booking a ticket with a travel company. In this case, the mobile version must be seen as a simpler alternative to this service/function.

4. **Create applications for clearly defined functions to which a user may need frequent access** Applications (apps) primarily work well for clearly defined tasks, for example displaying current traffic information/problems. The application should primarily be created for well-defined tasks that a user may conceivably need to do/access frequently.

Design

5. **Follow WCAG 2.0 except where these guidelines contradict WCAG 2.0** WCAG 2.0 are applied primarily to web interfaces but several parts of the guidelines may also be applied to applications, for example success criteria that involve labelling non-text objects and success criteria relating to contrasts.

6. **When creating applications for specific devices, any design and accessibility guidelines must be followed, provided they do not contradict these guidelines** If there are specific guidelines for the development of applications for the device/operating system chosen, they must be followed. For example, Apple's guidelines must be followed when developing apps for the iPhone.

7. **If you develop an application for a specific platform, this must support the characteristic properties of the platform** For example, it must be possible, where feasible, to zoom out/in by pinching the screen with two fingers or pulling two fingers apart. Many users are used to this method of interacting with touch screens and it is, therefore, important to offer this option, where practically feasible. However, remember it must also be possible to do this type of setting with one finger. Therefore, for the example involving pinching with two fingers, the interface should also offer an alternative for zooming, for example two buttons for zooming in and out. On Android phones, there are usually physical buttons that must work as the user is used to them working, while phones with iOS use software buttons at the bottom of the screen instead.

8. **Label graphic elements, icons and buttons with their reason or function** On websites, significant images must have a text alternative. In the same way, images and buttons in applications must have a description. Exactly how this is done varies between the different operating systems.

9. **Each form object must have either a caption or a description** Form objects, for example text fields, radio buttons and check boxes, must, where feasible, have a visible caption that is linked to the form field in a correct manner. On websites, this is done with the label element. If a caption cannot provide the full instructions, the form object's title text may be used to give the user information. However, the instructions must also always exist visually on the page as, without a screen reader, in most mobile devices the user cannot display a title text.

10. **Do not use frames in web interfaces** Frames or inline frames (iframes) work poorly in many mobile devices and should, therefore, be avoided. If an iframe has to be used for a service, you should, where possible, also offer the user a link to display the function in a separate window without frames. For example, you might have a map integrated in an iframe on the standard website but also give the user a link to display the map in a separate window without frames.

11. **Help the user input data by adapting the virtual keyboard to what needs to be entered** In web-based interfaces, this can be achieved by using html5 to label different types of input field, for example phone number, email or text.

12. **Minimise the use of scripts on the client page** Mobile devices often have poorer performance than standard PCs and the use of many scripts may cause problems. Do not demand more of the device than is necessary.

13. **Carry out practical tests on the solution** However well you follow the guidelines when developing a new application or a new web interface that is to work for mobile devices, there is a high level of complexity. This creates accessibility problems that may be difficult to anticipate. Therefore, the solution must always be tested in practice with persons who were not involved in the development. Test persons with disabilities must always be included in the user tests and accessibility experts must always be asked to interpret the results.

Layout and design

14. **Position important things higher up and less important things lower down in scrolling views** As the screens are often smaller on mobile devices, the most important information should be positioned high up to ensure that it is visible without the user needing to scroll. However, remember also that it is difficult to click objects at the very top of the screen. Therefore, important interaction should not be placed at the very top.

15. **Group elements that belong together** This applies in general but is additionally important when developing with responsive design. On many websites, related links are located to the right, which often work when displayed on a big screen. If smaller screens make the right-hand column move down below the contents column, a lot of scrolling may then be needed to find related links for part of the content. In this case, the page should, where possible, be repositioned so that the related information is positioned directly after the area to which it is related instead of all related material being placed together at the bottom.

16. **Strive to create a clean design and minimise the number of unnecessary objects** It is a big problem when sites contain many objects that the user does not perceive to be interesting/important when using the site. Websites that were designed for display on standard PCs with high-resolution screens naturally often have a design with many objects and areas. When such sites are displayed on a small screen, the user has big problems as the site takes a long time to load and requires a lot of scrolling. For example, if a site has a right-hand column with advertising, this should be repositioned on a small screen and placed at the end if it is not possible to remove it entirely.

17. **Strive to make the side header small** On mobile devices, there is often a problem with a lot of scrolling. By minimising the side header, you can reduce the problem and menus and content can be made visible as soon as the site loads.

18. **Create large clickable areas** As the devices' screen size, dpi and resolution vary, it is not possible to specify exact dimensions. There is also a difference between a website and an application, but strive for the clickable area to be at least the line height of the body text in one direction and the line height of the body text x 3 in the other direction. An icon in an app should be at least 9 millimetres wide and high.

19. **Do not place frequently used buttons out at the right/left edge unless they take up at least one third of the width of the screen** Important buttons should primarily be placed centrally and relatively far down on the screen as it is difficult to press buttons out at the edge for users who only use one hand or have to balance their mobile on their knee to use it. For example, this concerns users with reduced motor skills who may find it hard to hold the phone.

20. **Do not adjust buttons , functions or groups of buttons and functions to the right unless the group of buttons/functions extends over at least 75% of the screen in all positions** Users who cannot see the site use their index finger to scan the interface. The phone reads out what the user points to. This is done

most naturally from the top left-hand corner down the page. Buttons that are far out to the right with nothing else on the same line are very difficult to detect.

21. **Orient buttons and links in clear rows (horizontally and vertically)** This makes it easier for users who cannot see the interface to find them. If a user finds one button, it is easier to find the other buttons as well. This also creates a clearer visual overview for sighted users.

22. **Captions for input fields must primarily be positioned above the field** Check boxes and radio buttons are exceptions. The text may be to the right of the button/box. Groups of radio buttons and check boxes must, however, have a heading to indicate the function of the group. This must be positioned above the group of radio buttons/check boxes.

23. **Line lengths must be adapted to the screen width but never exceed 70 characters per row, including spaces** Where possible, it should not be necessary to scroll sideways to read a row of content. At the same time, the line length must not be so short that individual words must be divided over several lines unless there is a natural break. The objective should be line lengths of 55-60 characters, including spaces, per line.

24. **Limit the quantity of information and the number of objects displayed** To make things easier for users with small screens, it is a good idea to limit the quantity of objects and text displayed. This does not mean that you should remove parts but it can be easier for the user if you hide parts in, for example, accordion functions (you click on a heading to expand the underlying information). The user then has a quick overview and also easy access to all information and functionality. Another way of hiding objects is to place menus and link groups in fold-out menus. Remember that the functionality must be clear. It must be intuitive for the user to access hidden parts.

25. **Use known icons** Do not invent your own versions of standard icons. Use the appearance that the user has a chance of recognising from previous use.

26. **Design clickable objects so that they look clickable** Design links so that they look like links. Do not use colour alone to show that something is linked. One consequence of this is that the links then become difficult to see in direct sunlight. Make buttons three-dimensional and use known designs and locations of icons.

27. **Use high contrasts** Many users state that it is difficult to see what is displayed on the screen when the mobile is used in direct sunlight. To facilitate use, it is important always to strive to have good contrasts. Body text and icon texts should, where possible, be presented as black text on white background or the reverse unless the text is large or zoomable. Text that is zoomable or is large to begin with should at least comply with WCAG 2.0's stricter item, 1.4.6.

28. **It must be possible to use the interface with both portrait and landscape display formats.**

Interaction

29. **Use simple navigation concepts** When a website is displayed on a standard screen, there is often enough space to let the navigation use up large areas. One example is megamenus that often display 2-3 levels in the menu structure simultaneously. This works poorly on mobile devices. Here the menus need to be designed so that they take up little space and have a visually clear layout. For a web service that is to function both on a PC and on a mobile device, it may be necessary in some cases to have the menu displayed in different ways, depending on the screen width.

30. **If you develop an application for an operating system or a mobile device that may have control buttons (for example arrow keys and an OK button), it must be possible to use these to navigate in the interface** This currently applies to Android, for example. The physical back button must always work.

31. **If you develop an interface that can be used by devices in which you can connect a keyboard, the interface must, where possible, be controllable with the keyboard.**

32. **Insert shortcuts to allow the user to jump between parts of the content on long pages** A shortcut should be hidden at first but appear when it comes into focus during keyboard navigation.

33. **Minimise text input in the interface** Text input is both difficult and time-consuming in mobile devices and should, therefore, be avoided if possible. One way of avoiding it is to offer lists with choices instead of requiring text input and providing an autocomplete function (the interface suggests phrases when the user starts to enter text).

34. **If the interface permits gesture control, this should be implemented** Gestures are a way of controlling a device by making different gestures on the screen with one or more fingers. For example, an iPad user can often scroll between different pages by dragging a finger across the screen. In many interfaces, it is also possible to zoom in or out by moving two fingers apart or towards each other. Gestures must be intuitive and consistent. Use the gestures the users are used to.

35. **Do not insert functions that can only be managed via gestures. Always add a button/link**

36. **Make it possible to control the interface with just one finger** There may be situations in which it is not possible. However, where feasible, it must be possible to control all functionality with just one finger. It may be necessary for buttons to be hidden and to appear when you touch a certain area on the screen or press another button.

37. **Be consistent** For example, place buttons with a specific functionality in the same place on the screen and design them consistently.

38. **Use integrated objects as they are intended to be used and the user expects them to be used** For example, operating systems often contain integrated

components/widgets that an application can use instead of the developers implementing their own components with equivalent functionality. In devices that have physical buttons, they must, where possible, be supported by the application.

39. **Give feedback to the user** When the user makes an entry, this should be confirmed both with a sound and with a short vibration if the device supports this. However, it should also be possible to switch off the feedback. Please note that input need not necessarily be text via a keyboard. It may be a voice command, a photo that has been taken, a gesture or a movement with the mobile. Feedback should be given in most cases but there may be exceptions as too much feedback would be a nuisance (for example, an application that functions as a pedometer must not give feedback for every step registered).

40. **Give the user clear status information** Many people use mobile devices in stressful situations. In such situations, it is important for the user to know what is happening all the time, in particular when the user is waiting for the application/website. If, for example, the application/website is loading data, it is good to display how the loading is progressing. Always provide clear status information. It is an advantage to offer both visual feedback and feedback with sound.

41. **Give the user enough time and give warnings before time limits are reached** Many people use their mobile while travelling. Usage is often interrupted in such situations on account of external circumstances. This may, for example, be because the user is using the mobile while waiting for a bus. When the bus arrives, the user stops using the mobile when he or she boards the bus. During this time, the mobile is idle. It is important for the application/service to give the user enough time and to warn the user when the time is running out. If possible, there should also be a simple option for extending the time. The most common example of time limits is automatic logouts.

42. **Help the user avoid errors and correct any errors** This is particularly important in mobile devices where it is easy to press the wrong button. Techniques for helping the user avoid making errors include autocomplete and search suggestions. If errors still arise, this must be clearly communicated to the user both at the top of the page and where the error occurred. Where possible, you should also provide suggestions for solutions.

Content

43. **Use images only if it really helps the user** Images are good for communicating information but on mobile devices they often work less well. This is partly because they are small and partly because they take longer to load. Images should, therefore, only be used when they really help the user. Decoration images should be minimised and, where possible, be placed in the css code.

44. **Use short, descriptive headings to structure the information** If it is possible to indicate what are the headings in the code, this must be done. For websites, this means the html elements H1-H6. Another basic rule is to keep headings

relatively short. However, they must not be so short that they do not make clear to the user what he or she can read about below the heading.

45. **Avoid abbreviations** Although mobile devices with a small screen may tempt you to use abbreviations, this should be avoided. Abbreviations of organisation names and similar may be used if they are explained the first time they are used.

User settings

46. **Ensure that it is possible to zoom the interface** In an application, an enlargement function must be integrated, while a website must be zoomable in a manner that is natural for the web browser.

47. **Consider providing a setting for inverting colours** If, for example, the application has dark text on a light background, the user should have the option of inverting this so that there is light text on a dark background and visa versa.

48. **Consider providing a setting for changing the font**

Funka Nu AB

Funka started as a project within the handicap movement. Today we hold pole position within the field of accessibility with over 80 percent of Sweden's government authorities as customers. Since 2000 we are a privately owned company and our close relationship with the handicap movement guarantees a unique quality control. We have offices in Stockholm, Sweden and Oslo, Norway.

Funkas work sets standards for both development and analysis as well as issues demands for accessibility. We are part of most international work groups of importance and we regularly perform surveys on our own initiative. Because of that Funka actively promotes accessibility in Sweden, Norway and the EU. During 2012, Funka will meassure e-inclusion in all EU member states on behalf of the commission.

Within the EU Funkas experts can, among other committees, be found in the committee for mandate 376. That's the committee that works to include compulsory rules as demands of accessibility in public purchases and within that work also come to terms with common rules of certification to ensure that accessible websites and accessibility experts live up to their requirements. We also work towards standardization and usability at a national- as well as at an international level.

Funka regularly performs larger surveys regarding accessibility, for esample an interview survey regarding young patient's outlook on life. Assigned by European Patients Forum, a non-profit organization representing 150 million European patients, Funka has interviewed young people who have a regular and lifelong contact with the medical service.

In Sweden Funkas consultants had a major role in developing Handikappombudsmannens guiding principles for an accessible administration of state as well as national guidelines for web accessibility. Funkas very own methodology regarding accessibility has been included on many levels. Funka are also responsible for recommendations regarding keyboard shortcuts and icons.

For the WC3 Funka, among other assignments, made the authorised translation of WCAG 2.0 into Swedish and are active members of the WAI Age Task Force.

Funka are EPiServer Solution Partners, Microsoft Partners and Adobe Certified Training Providers.

To learn more about Funka, please visit our website www.funkanu.com.