

Bachelor's thesis

Degree Programme in Information and Communications Technology

PINFOS18

2023

Sanni-Anni Vuorinen

# Accessibility testing in software development



Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Degree Programme in Information and Communications Technology

2023 | 41 pages

Sanni-Anni Vuorinen

## Accessibility testing in software development

Accessibility of software affects significant portion of the population in Finland. Many different types and levels of disabilities should be taken into account during the development of any software. Law sets requirements for standardised level of accessibility that public and certain private companies must adhere to.

Testing of the software accessibility can be partly automated and even integrated within the development pipeline to eliminate the most obvious and blatant issues before they reach the users. However, to ensure that the accessibility is meeting the requirements set by the World Wide Web Consortium, additional manual accessibility testing is required.

In this thesis a case study was done to compare a public service, which is subject to legal accessibility requirements, with a private online store that operates without such mandates. The purpose of this case study was twofold: first, to demonstrate various techniques for testing the accessibility of web services; and second, to examine how legal requirements influence the accessibility of these services.

The results of the case study show that even though modern web applications usually have somewhat usable level of accessibility even without legal requirements they still often have issues that have negative impact on the usability of the service.

Keywords:

Accessibility, Software testing, Quality assurance

Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintäteknikan koulutus

2023 | 41 sivua

Sanni-Anni Vuorinen

## Saavutettavuustestaus ohjelmistokehityksessä

Verkkosivujen saavutettavuus vaikuttaa merkittävään osaan Suomen väestöstä. Ihmisillä on olemassa monia eri vammoja kuten kuulo- tai näkövamma, jotka tulisi ottaa huomioon ohjelmistokehityksessä. Laissa on määritelty tarkat vaatimukset saavutettavuudelle, joita julkisten ja osan yksityisten yritysten tulee noudattaa.

Saavutettavuutta voi testata automaattisesti ja sen voi jopa integroida ohjelmistokehitykseen, jossa viat pystytään poistamaan ennen julkaisua kuluttajille. Manuaalinen testaus on kuitenkin hyvä suorittaa automaattisen testauksen lisäksi, jotta voidaan olla varmoja, että saavutettavuus täyttää World Wide Web Consortiumin asettamat vaatimukset.

Tässä opinnäytetyössä suoritetaan saavutettavuustestaus yhden julkisen ja yhden yksityisen verkkosivun välillä. Testauksessa käytetään erilaisia testausmetodeja, joilla mitataan saavutettavuutta ja katsotaan täyttääkö julkiset sivut lain vaatimat velvoitteet ja kuinka hyvin yksityisen sivun saavutettavuutta on kehitetty.

Opinnäytetyön tulokset osoittavat, että sivujen saavutettavuudessa ja käytettävyydessä oli suuria eroja.

Asiasanat:

Saavutettavuus, Ohjelmistotestaus, Laadunvarmistus

# Content

<b>List of abbreviations (or) symbols</b>	<b>7</b>
<b>1 Introduction</b>	<b>8</b>
<b>2 Literature review</b>	<b>9</b>
2.1 Groups affected by accessibility of software	9
2.1.1 Elderly	9
2.1.2 Developmental disabilities and dementia	9
2.1.3 Vision and hearing disabilities	10
2.1.4 Motor impairment	10
<b>3 Accessibility best practices</b>	<b>12</b>
3.1 Level A	12
3.2 Level AA	13
3.3 Level AAA	13
<b>4 Legislation in Finland</b>	<b>15</b>
4.1 Who are obligated?	15
4.2 Exemptions	15
4.3 Requirements dictated by the law	16
<b>5 Automatic tools for evaluating accessibility</b>	<b>18</b>
5.1 WAVE	19
5.2 Axe accessibility tools	20
<b>6 Manual accessibility testing process</b>	<b>22</b>
6.1 Navigation	22
6.2 Media	23
6.3 Software to support manual accessibility testing	24
6.3.1 ACTF aDesigner	24
6.3.2 Colorblindly	25
<b>7 Accessibility testing public and private websites</b>	<b>27</b>

7.1 OmaVero tax card creation	27
7.1.1 Testing the accessibility	27
7.1.2 Report of the testing	31
7.2 Ordering process of Suomalainen kirjakauppa web shop	31
7.2.1 Testing the accessibility	31
7.2.2 Report of the testing	38
7.3 Conclusion and comparison	38
<b>8 Conclusion</b>	<b>39</b>
<b>9 References</b>	<b>40</b>

## Pictures

Picture 1. HP BigKeys LX accessibility keyboard.	11
Picture 2. WCAG Conformance Logo for level A.	12
Picture 3. WCAG Conformance Logo for level AA.	13
Picture 4. WCAG Conformance Logo for level AAA.	14
Picture 5. An example of a WAVE report.	19
Picture 6. An example of an axe-linter catching an accessibility issue during the development.	21
Picture 7. aDesigner visualizing a web page to aid for full blind testing.	25
Picture 8. Colorblindly simulating what a website looks like with protanopia.	26
Picture 9. WAVE report of OmaVero tax card creation.	27
Picture 10. Second page of tax card creation analysed with WAVE.	28
Picture 11. Example of an orphaned form label.	29
Picture 12. Narrator read the label "Eläkkeet" twice while navigating the page.	30
Picture 13. Tax card created successfully using only narrator for navigation.	30
Picture 14. Result of the WAVE scan on the front page.	32
Picture 15. WAVE report from a product page.	33
Picture 16. WAVE errors from the shopping cart page.	34

Picture 17. Narrator was unable to read the ad banners.	35
Picture 18. Narrator is unable to shift focus to new modal.	36
Picture 19. Narrator did not focus on the top of the page.	37

## List of abbreviations (or) symbols

CSS	Cascading Style Sheet
JAWS	Job Access With Speech
QA	Quality Assurance
W3C	World Wide Web Consortium
WAVE	Web Accessibility Evaluation Tools (WebAIM, 2023)
WCAG	Web Content Accessibility Guidelines

# 1 Introduction

Accessibility testing is used to test different types of software for the benefit of people living with different kinds of disabilities. The disabilities that most affect the ability to use modern software are issues with eyesight, hearing, or mobility. Accessibility testing aims to make software usable for anyone, no matter the circumstances.

To help disabled people use the software they need; software engineers have developed different type of tools. Such tools include speech recognition software, screen reader software, screen magnification tools or physically easier to use IT equipment. In Finland the law requires such tools to work in any official site and accessibility testing is conducted to ensure it.

The topic of this thesis was chosen out of interest in software testing and to increase knowledge on software testing tools. It was also interesting to test websites to evaluate whether websites comply with Finnish legal standards, where the aging population has more and more disabilities limiting their daily life.

The beginning of the thesis will cover the basics of accessibility testing and existing literature on the subject. The thesis will introduce Finnish legislation on accessibility and see if the case study web pages pass the current legal requirements. Different testing tools will be used for the case studies to see which one is the best.



## 2 Literature review

### 2.1 Groups affected by accessibility of software

It is a default assumption in most software that users can read the content in the application, are able to use a keyboard with standard layout, use mouse to navigate through the application and hear sounds if played by the application. However, there are multiple groups that might have difficulties performing one or more of these tasks independently. (Kavcic, 2005)

In Finland there are about one million people with varying degrees of disabilities who may require some level of help with technical devices. Most people could find help easily with already existing software but may be unaware of such technology existing. (Etelä-Suomen aluehallintovirasto, 2023)

#### 2.1.1 Elderly

The biggest group affected are the elderly, who make about 25% of the Finnish population. Most of the elderly suffer from at least one or several conditions affecting their daily life. The degree of how much the disabilities affect varies by a fair amount. The most common disabilities in the elderly are issues with eyesight, hearing, or memory.

There are a lot of accessibility tools and devices that have been specifically developed for the elderly. For example, many different types of phones and tablets are created to be as simple as possible and often have built-in accessibility tools such as the magnifying tool.

#### 2.1.2 Developmental disabilities and dementia

In Finland there are about 50000 people with developmental disabilities. Developmental disabilities can cause difficulty in learning, which can make it difficult to learn how to operate technical devices.

About 190000 people in Finland suffer from some form of dementia. A person with dementia may have trouble with understanding language and remembering what they were supposed to be doing.

For people with developmental disabilities or dementia it is very important to create websites and software to be as accessible and easy to navigate as possible.

### 2.1.3 Vision and hearing disabilities

On estimate 55000 people have severe vision and about 5000 people have severe hearing disabilities in Finland. These people rely on accessibility tools such as screen reader software and the speech recognition software to manage their daily activities. (Näkövammaisten liitto, 2021)

People with issues with their eyesight or hearing can find help with many different accessibility tools. For the visually impaired there are screen reading software such as JAWS, Apple VoiceOver or Microsoft Narrator. Screen reading software converts any text that is displayed on a screen into audible speech. There is also software that is voice-activated. For people with deafness or hearing loss there are software that can create live captions if they are not already included in a video.

### 2.1.4 Motor impairment

Physical disabilities such as cerebral palsy, Parkinson's, spinal cord injury, muscular dystrophy, lost or damaged limb and arthritis affect the persons ability to move. In some cases, the person affected may be only able to move their mouth or eyes.

In Finland about 6500 people have cerebral palsy, of which some require help with everything in daily life. Cerebral palsy can cause issues in hand eye coordination, hearing, eyesight, and overall physical movement among many

other symptoms. (CP-liitto, 2023) About 16000 people in Finland have Parkinsons disease, most of whom are people over the age of 50. Parkinsons disease causes tremors that can make using devices difficult. (Kustannus Oy Duodecim, 2023)

For motoric disabilities there are many tools that make it easier to use technology. There are for example keyboards that have bigger and fewer keys than in a normal keyboard as seen in picture 1. An oversized trackball mouse requires less fine motor control and it's easy to grip. There is also software designed to allow a user to operate a device by just moving their head or eyes.



Picture 1. HP BigKeys LX accessibility keyboard

### 3 Accessibility best practices

The best practices in accessibility testing have been created by the World Wide Web Consortium (W3C) and they are called Web Content Accessibility Guidelines (WCAG). The goal of the WCAG is to guarantee accessibility for everyone at least at the most basic level. The WCAG is used mostly by Web designers and policy makers. The WCAG guidelines have 3 different levels: A, AA, and AAA. (W3C, 2023)

#### 3.1 Level A

The most basic level of accessibility in WCAG guidelines is level A. In level A the website should be usable with different types of assistance software, with which the user should be able to access and use the website's functions. The website should also be usable with different devices. Level A is the most basic and pretty much every website should fill the requirements in it if they want their website to be accessible to most people. Level A has 30 different requirements.

In level A the images and videos on the website must include text that can be read with screen reading software. The videos should also be able to be paused and the volume should be adjustable by the user. The website should also have colours that help the user use the website's functions. The website should be keyboard accessible so that the user can use the site without a mouse.



Picture 2. WCAG Conformance Logo for level A

(W3C, 2023)

### 3.2 Level AA

In the second level (AA) the accessibility requirements include everything from level A and 20 new ones from level AA. Level AA is where most companies should aim to have their websites accessibility. Level AA is also the legal requirement for public websites and applications in Finland.

On level AA some of the requirements are that the website is well organized and logical to navigate. Any live audio should have captions on it. The contrast on the website should be at least 4.5:1, except in large text, decorative text, user interface components, logos, or images.

Text should be resizable without assistive technology on everything except for captions and images of text. The text spacing should have line height to at least 1.5 times the font size, the space following a paragraph should be at least 2 times the font size, letter spacing should be at least 0.12 times the font size and lastly the word spacing should be at least 0.16 times the font size.



Picture 3. WCAG Conformance Logo for level AA

(W3C, 2023)

### 3.3 Level AAA

In AAA accessibility is at the highest level of WCAG, as it includes A, AA and 28 level AAA requirements. While level AAA is the highest level it may still not guarantee accessibility for the most challenging cases.

On level AAA videos need to have a sign language translation. Any background sounds are at least 20dB lower than the main speech audio, with the exemption of short sounds of a few seconds maximum. Visually the website should have option to change the foreground and background colours, the text is not justified, line spacing is at minimum a space and a half within paragraphs and the text should be resizable. The contrast in visual presentation of text and images of text should be at least 7:1 with the exceptions of large text, decorative text, user interface components, logos, or images.



Picture 4. WCAG Conformance Logo for level AAA

(W3C, 2023)

## 4 Legislation in Finland

The Finnish law called “Laki digitaalisten palvelujen tarjoamisesta” (2016/2102) requires that any official public sector website or mobile application is accessible, high quality, secure and compatible with the most common software. (Finlex, 2019) The law obligates government services and some private companies to follow the WCAG guidelines on the levels of A and AA. (Finlex, 2019) The law is based on the European Union directive on the accessibility of the websites and mobile applications of public sector bodies. (EUR-Lex, 2016)

### 4.1 Who are obligated?

Accessibility requirements set by law are obligatory for several different organizations and companies. The law applies to digital services offered by the government and other public bodies such as municipalities or wellbeing services counties. In the context of the law also state churches and learning institutions of higher education are treated as public organizations. The law also applies to some private companies. If a private company provides a digital service that is at least 50% funded by a public agency or organization that service must comply with the accessibility requirements set by the law. Also, private companies providing digital services on specific fields such as water and energy services, transportation services, postal services or monetary services are required to comply. (Finlex, 2019)

### 4.2 Exemptions

Not all services are required to follow the accessibility requirements even though the service providers are otherwise obligated by the law. Intranets or extranets initially created before 23.10.2019 and intended to be used by a limited number of people are not required to follow the accessibility

requirements. However, services created or fully updated after that are required to follow the same requirements as other services affected by the law.

Also, services created by students are not required to follow the accessibility requirements if the service is developed as a part of the studies and the service is intended to be used with limited user base and timeframe.

Historically important documents that can't be modified to meet the accessibility requirements are also exempted from following them. Live broadcasts are also exempted as it would be impossible to provide accessible captions and other accessibility features to them.

The law also includes a paragraph that allows a service provider to deviate from the accessibility requirements if they can indicate that implementing the accessibility requirements would cause unreasonable encumbrance to the operation of the service. If the service provider decides to invoke this paragraph, they need to consider the needs of the especially disabled people to use the service in question as well as the size and financial station of the service provider. (Finlex, 2019)

#### 4.3 Requirements dictated by the law

Digital services affected by the law are required to have an accessibility statement. The accessibility statement must have an explanation on which parts of the content do not meet the accessibility requirements and the reason for that. The accessibility statements content and structure must conform to the European Union accessibility directive, article 7, section 2. (Finlex, 2019)

The law dictates that the standards for the accessibility requirements are maintained by government authorities and are available at [www.saavutettavuusvaatimukset.fi](http://www.saavutettavuusvaatimukset.fi). Basically, this means that digital services must be accessible at level AA according to WCAG standard. The current version of the standard required by law in Finland is 2.1. Version 2.2 of the



standard is estimated to be required by law earliest by fall 2024. (Etelä-Suomen aluehallintovirasto, 2023)

The law specifies a notice of a conditional fine. Details of the fine are specified in the law “uhkasakkolaki (1113/1990)”.

## 5 Automatic tools for evaluating accessibility

Accessibility testing can be very difficult to do as a regular person without disabilities. The best option is to work with a disabled person when creating new software, but as that is not often possible engineers have created different tools to make the testing easier. These tools can help spot whether the website is compliant with the WCAG guidelines and find other flaws such as bad grammar, contrast issues, links that do not work, bad browser compatibility etc. Manual accessibility testing also takes a lot of time and thus is costly to do, especially regression testing during development is simply too expensive. Automated tools can be used to lower the efforts of manual testers and let them concentrate on less frequent but in-depth testing. They can check the site against a set of rules for how accessible the software should be.

Common examples of this are for example verifying that all input fields have aria-labels or alt texts for pictures that enable people with visual impairment to navigate the site. The automated tools can also detect if the site is using some browser specific code that might not be supported by other common browsers. They can also validate the structure of the web page so that if headings are used, there should always be a h1-level heading present in the site. Links should also always lead to something and many of these tools can automatically detect broken links that might not lead anywhere.

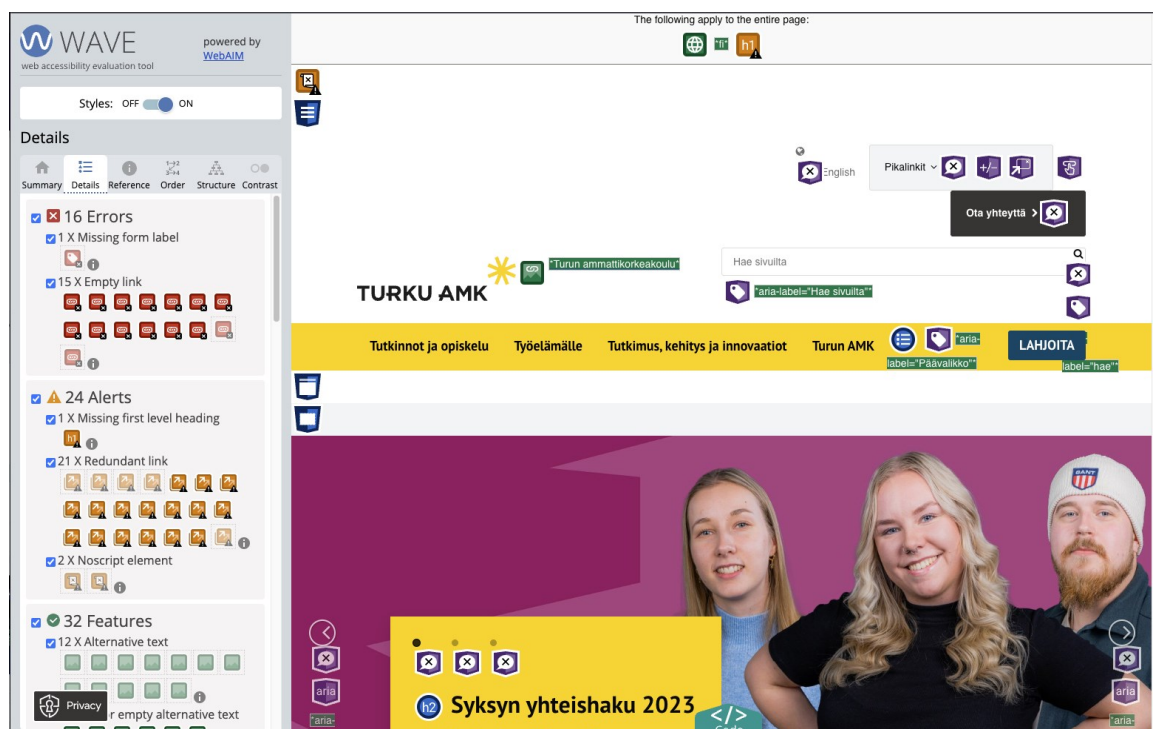
As these issues might rise in the future after some modifications or additional features are introduced to a web page or application, it is advisable to integrate some automated accessibility checks into the development lifecycle. This can greatly reduce the amount of accessibility issues in software. (Sane, 2021) An example of a popular tool to accomplish this is the axe-core JavaScript package.

Some popular tools used for accessibility testing are WAVE, Axe accessibility tools, SortSite, JAWS, QualityLogic and DYNO Mapper.

## 5.1 WAVE

WAVE (Web Accessibility Evaluation Tool) is an accessibility evaluation tool developed by WebAIM at the Institute for Disability Research, Policy & Practice at Utah State University. It is intended to be used to test the accessibility of web-based applications and websites. It is a free browser extension available to all major browsers such as Mozilla Firefox, Google Chrome and other Chromium based browsers such as Microsoft Edge. It scans the source code of a web page and can identify and report most WCAG errors. WAVE also does not send any data from the scan to any server so it can be used for projects that have strict rules regarding data privacy. (WebAIM, 2023)

WAVE should not be used as the only means of evaluating accessibility of web-based applications but to be used as a tool to accompany manual testing. It can be used by developers to quickly verify there are no major accessibility issues present in new features before they are given to the QA testers.



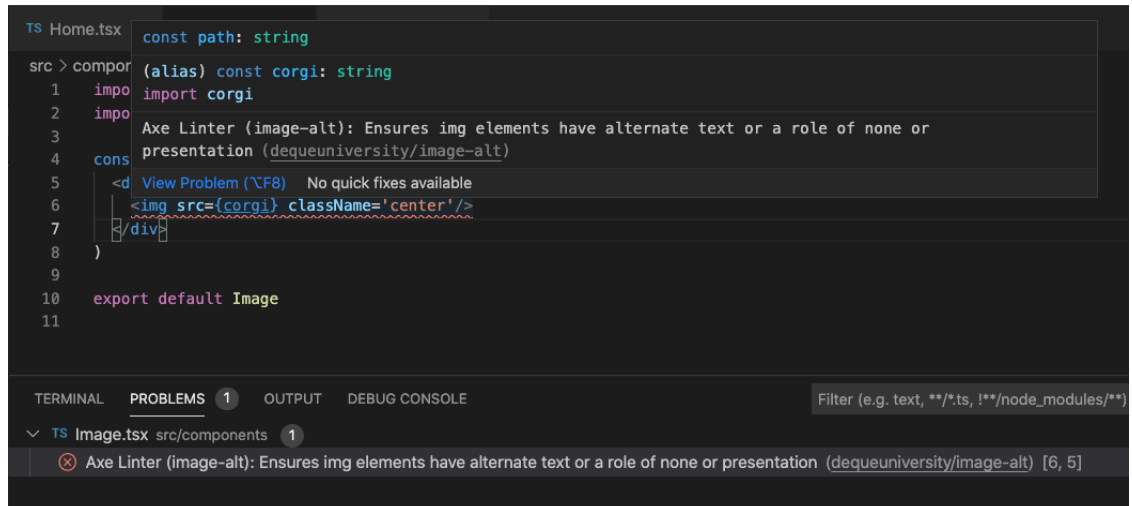
Picture 5. An example of a WAVE report

A WAVE report contains a summary page where it is possible to quickly see the overall number of errors the page has. In the details tab all the different findings are detailed by their type. For example, in picture 1 there is a field missing a form label and several empty links. By hovering or clicking a finding, the element that caused the error or alert is highlighted.

## 5.2 Axe accessibility tools

Axe accessibility tools are developed by Deque Systems, a private company focused on providing enterprise grade accessibility tools for software development. They offer some free tools for accessibility testing such as their axe DevTools which is a browser extension offering similar functionality to WAVE. Being a commercial product, the extension has additional features that can be purchased with a professional license. (Deque Systems, 2023)

In addition to the browser extension, Deque offers a wide range of other accessibility tools such as the axe-core JavaScript package that allows axe accessibility checks to be integrated in automated testing software such as Cypress. This allows automated accessibility checks in end-to-end test suites catching many easy to avoid accessibility issues to be caught during the software development process should a developer change some elements forgetting to take accessibility into account. Deque claims that using axe-core you can find on average 57% of WCAG issues automatically. (Deque Systems, 2023) To catch the issues even earlier in the development process developers could install the axe Accessibility Linter, a Visual Studio Code extension from Deque as seen in picture 6.



```
TS Home.tsx
src > compo
1  impo
2  impo
3
4  cons
5  <d View Problem (\F8) No quick fixes available
6  <img src={corgi} className='center' />
7  </div>
8  )
9
10 export default Image
11
```

AXE LINTER (image-alt): Ensures img elements have alternate text or a role of none or presentation (dequeuniversity/image-alt)

TERMINAL PROBLEMS 1 OUTPUT DEBUG CONSOLE Filter (e.g. text, \*\*/\*.ts, !\*\*/node\_modules/\*\*)

TS Image.tsx src/components 1

AXE LINTER (image-alt): Ensures img elements have alternate text or a role of none or presentation (dequeuniversity/image-alt) [6, 5]

Picture 6. An example of an axe-linter catching an accessibility issue during the development

Some paid accessibility products from Deque includes the axe Monitor that can scan entire web sites, as well as PDF documents, and provide historical data indicating what changes have decreased or increased the accessibility score of the page. Another paid product is the axe Auditor that is a manual testing tool that can be used to aid manual testing of not just web pages but also desktop software, mobile applications, and even Microsoft Office documents.

## 6 Manual accessibility testing process

If the accessibility testing is performed without an accessibility testing tool, the testing may take a while longer. When manually testing software, the tester should check that the website is compatible with accessibility tools such as screen reader software and the magnifying tool as they are essential for many people. It's also important to test the brightness, contrast, and the font size of the website to make sure they are all adjustable. The images and videos on the website should be checked to have captions and that the volume can be changed. If the website has some sort of form to fill in the tester should check that it includes clear instructions how to fill it. Overall, the website should have logical navigation and the style should remain consistent throughout the website. Testing the accessibility as a person with no disabilities can sometimes be tough or even impossible fully, as the tester might not be able to read braille or sign language to confirm that they are implemented properly. Consulting people with actual disabilities during the testing is a good practice to ensure that those kinds of issues won't go unnoticed.

### 6.1 Navigation

The navigation of the website or application should be tested with different disabilities in mind. The tester should try to navigate the page using only keyboard with screen reader on. The tester can easily notice if the navigation of the page is not logical, and the cursor is skipping some fields or going through them in the wrong order. With this kind of testing, it is also easy to determine if the screen reader skips some important text elements or if some of the fields are missing descriptive text or they are not descriptive enough.

The software should also scale correctly when zoom is applied. The tester should check that all the content is still visible on the page and if the content is readable without horizontal scrolling. Some buttons or other clickable elements

in the page might also lose functionality if scaling is not properly implemented in the code. (Harvard University, 2023)

## 6.2 Media

Assessing the accessibility of media in websites should include testing that all media such as images, videos or recordings have the same level of accessibility as the rest of the web site and that they also meet the legal requirements, should they apply to the site.

While checking any images, the tester should verify that they all contain descriptive alt texts in clear language so people with visual disabilities can still get the message in them.

Accessibility testing of videos and recordings should consider the requirements set for the website. If the site demands the highest-level AAA accessibility requirements the tester should confirm that the media caters to people with a variety of disabilities. For example, the videos should be checked and verified that they don't contain potentially seizure causing imagery (flashes that have a frequency higher than 3Hz). Videos and recordings should also be verified to have overlay texts that have good contrast and readability. Confirming that sign language overlays are done properly can be very hard if the tester is not familiar with sign language. Similar issue might rise when tester is trying to verify that descriptive captions for video or audio is properly readable with braille tablets. (W3C, 2023)

Videos should also be accessed with every different type of caption to ensure they fit properly. For example, the video might not be paced to account for the additional time required to fit descriptive captions. The tester should ensure that the captions or sign language overlays stay in pace with the video.

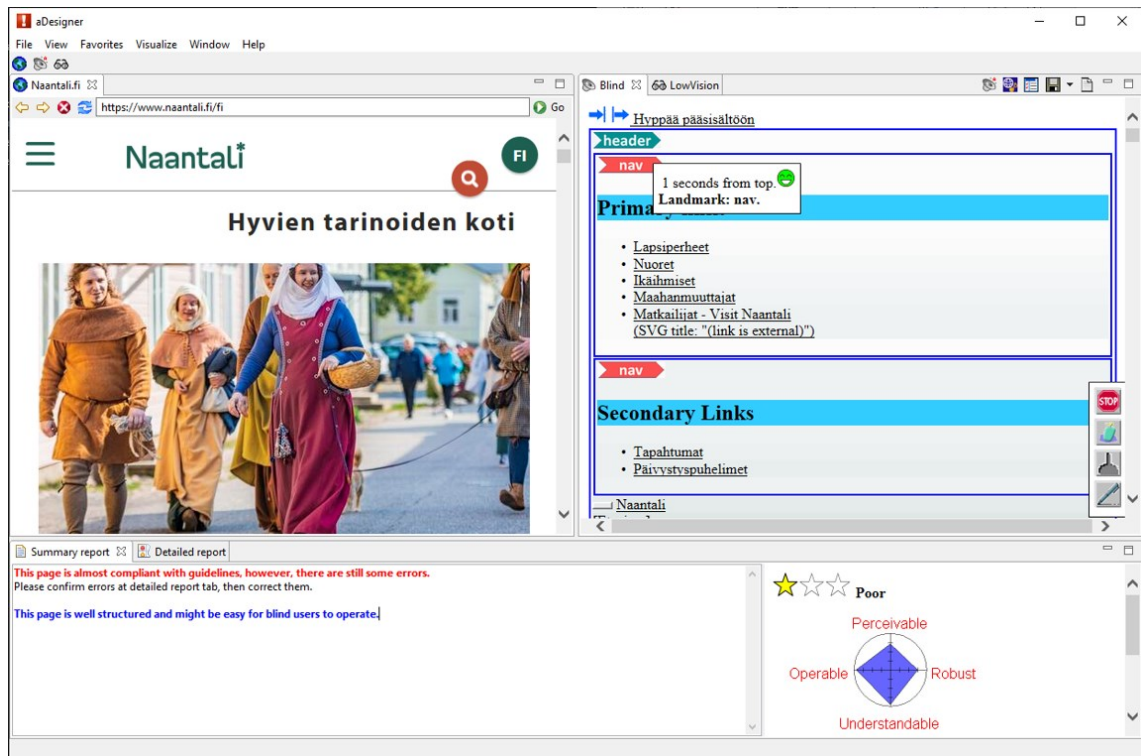
### 6.3 Software to support manual accessibility testing

Manual accessibility testing can be made easier with the use of several different software tools. These tools differ from automatic testing tools in that they don't test the WCAG requirements by themselves but can aid the manual testing process. The testing software can for example simulate colour blindness or other disabilities to allow the tester spot accessibility issues that won't be apparent to a person without disabilities.

#### 6.3.1 ACTF aDesigner

The Eclipse Accessibility Tools Framework (ACTF) aDesigner is a disability simulator that can help accessibility testers test content from the perspective of someone with decreased eyesight or different degrees of blindness. To help test for full blindness, aDesigner can visualize the structure of the web page so that the tester can easily see if a page is structured logically to navigate with screen readers. (Eclipse Foundation, 2023)

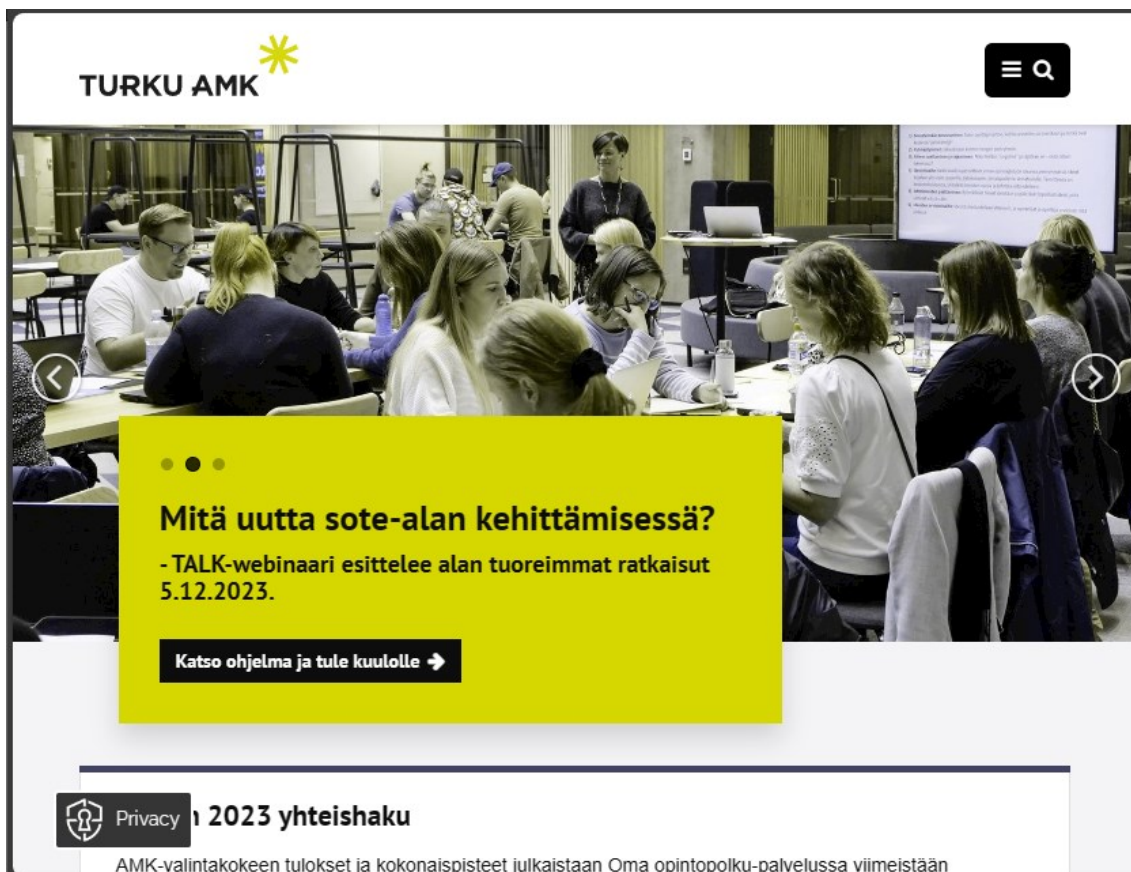




Picture 7. aDesigner visualizing a web page to aid for full blind testing

### 6.3.2 Colorblindly

Colorblindly is another great tool to simulate disabilities to assess the accessibility of web-based content. It can simulate how people with different types of colour blindness see. It supports the simulation of eight different type of colour blindness: achromatomaly, achromatopsia, deuteranomaly, deuteranopia, protanomaly, protanopia, tritanomaly, and tritanopia. It is a free open-source tool that is available as a browser extension for all chromium-based browsers. With Colorblindly the tester can easily spot obvious issues with a web page's colours and media. (VanNess, 2021)



Picture 8. Colorblindly simulating what a website looks like with protanopia

## 7 Accessibility testing public and private websites

### 7.1 OmaVero tax card creation

OmaVero is a website where Finnish citizens can see their tax information. The website requires a bank identification to login and is a public website that must have at least AA level of accessibility. Creating a tax card is essential for anyone that has a job.

#### 7.1.1 Testing the accessibility

Testing the accessibility of tax card creation with WAVE evaluation tool gave a good starting point for the testing as seen in picture 9.

The screenshot displays the WAVE (Web Accessibility Evaluation Tool) interface. On the left, a sidebar shows the tool's logo and a list of detected issues:

- 8 Errors:**
  - 2 X Missing form label
  - 3 X Empty form label
  - 3 X Empty heading
- 3 Contrast Errors:**
  - 3 X Very low contrast
- 61 Alerts:**
  - 2 X Orphaned form label
  - 1 X Noscript element
  - 8 X Redundant title text
  - 50 X Layout table

The main content area shows a preview of the website's header and navigation. The header includes the 'oma/ve' logo, a search bar, and language options (Suomeksi, svenska, English). Below the header, there is a navigation menu with items like 'Verokortit ja ennakvero'. A blue information box contains a message about automatic decision-making by the tax authority. The page title is 'Verokortti- ja ennakverohakemus'.

Picture 9. WAVE report of OmaVero tax card creation

The page unsurprisingly does not have many errors as the tax administration must comply to the accessibility requirements set by law. All the missing form label errors are from fields that are in a third-party add-on, the customer support chat box that is provided by Elisa. Both of the very low contrast errors are from the same link “Verokortit ja ennakkovero”.

The screenshot displays the WAVE (Web Accessibility Evaluation Tool) interface. On the left, the 'Details' panel shows a summary of 42 errors:

- 2 X Missing form label
- 35 X Empty form label
- 5 X Empty heading
- 2 Contrast Errors (2 X Very low contrast)
- 2457 Alerts (88 X Orphaned form label)

The main content area shows a page with a heading **h2 Muut huomioon otettavat esitetyt tiedot**. Below the heading, there are several form fields with 'aria' attributes. A code snippet at the bottom right shows the HTML for a caption element:

```

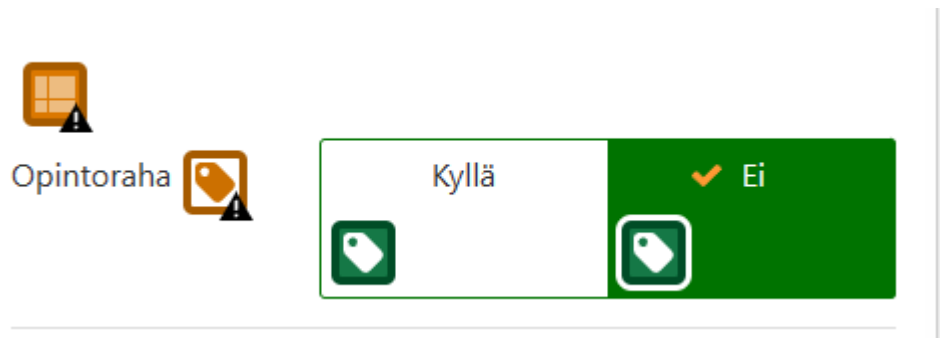
<div class="FastTitlebarCaption" id="caption2_Dn-gz3" class="FastTitlebarCaption FastHtmlLabel FHTML">
  <h3 id="caption2_Dn-gz3" class="FastTitlebarCaption FastHtmlLabel FHTML">
  </h3>
</div>
  
```

Picture 10. Second page of tax card creation analysed with WAVE

The second page of the tax card creation form has more errors though they are not severe. There seems to be multiple empty form labels and empty headings that don't contain any text. This will be further investigated with manual testing to determine if these pose actual issues when navigating with a screen reader.

Similar empty form label and heading errors were found from the third and fourth page as well with no new discoveries. All the pages also contained thousands of alerts regarding orphaned form labels. This means that labels next

to radio buttons are not directly linked to the buttons as seen in picture 11. This also will be further investigated with manual testing.



Picture 11. Example of an orphaned form label

Reviewing the page with Colorblindly did not bring out any issues regarding colours. Unfortunately testing with ACTF aDesigner proved impossible as the program was unable to go through Suomi.fi identification.

Manual testing began by going through the form with Windows narrator on. The empty heading errors indicated by WAVE were a small issue and narrator indicated that there is a heading level 3 yet did not read what the heading was about. Narrator was able to read all texts in the first page and navigate the page in a logical order.

On the later pages it also became apparent that the orphaned labels also posed a small usability issue. As the labels were not associated with the radio buttons, narrator read the label first when focusing on the label and again when switching the focus to the radio button group as seen in picture 12. This issue happened with every radio button and button throughout the form causing mild annoyance to the user.

The screenshot shows a progress bar at the top with four steps: 'Taustatiedot' (checked), 'Esitetyt tulot ja vähennykset' (checked), 'Muut tulot' (current step), and 'Muut vähennykset'. Below the progress bar is the heading 'Muut tulot' and a sub-heading 'Palkat, eläkkeet ja etuudet'. There are two main sections:

- Eläkkeet:** A question asks 'Saatko eläkettä? Ilmoita Suomesta saatava työ- ja yrittäjäeläke, kansaneläkelain mukainen eläke, takuueläke tai vakuutusin perustuva eläke, joka verotetaan ansiotulona.' The 'Ei' (No) option is selected.
- Etuudet:** A question asks 'Saatko etuusia, kuten työttömyys- tai sairauspäivärahaa, vanhempainpäivärahaa, kotihoidontukea, aikuiskoulutustukea tai kuntoutusrahaa?' The 'Ei' (No) option is selected.

Picture 12. Narrator read the label "Eläkkeet" twice while navigating the page

Other than the minor issues with a few empty headings and orphaned labels the form was easy to fill out using only narrator for navigation. No texts were skipped, and the form was easy to navigate though.

The screenshot shows the OmaVero website interface. At the top right, there are language options: 'Suomeksi', 'På svenska', 'In English', and the website URL 'vero.fi'. Below the language options are icons for 'Apua asiointiin' (Help) and 'Omat tiedot ja uloskirjautuminen' (My information and sign out). The main content area is titled 'Verokortit ja ennakkovero' and 'Verokortti- ja ennakkoverohakemus 2023'. A green checkmark icon indicates a successful action: 'Verohallinto on vastaanottanut hakemuksesi.' Below this, it states 'Verokorttisi on valmis.' and provides instructions: 'Verokorttisi toimitetaan automaattisesti haluamallesi vastaanottajalle, jos valitsemasi toimitustapa on muu kuin "Vain OmaVeroon". OmaVerossa verokortti ja ennakkoveropäätös ovat nähtävissä Verokortit ja ennakkovero -sivulla tai Päätökset ja kirjeet -sivulla. Voit tallentaa verokortin pdf-tiedostona tai tulostaa sen.' At the bottom, there is a summary table:

<b>Ilmoitus:</b>	Verokortti- ja ennakkoverohakemus 2023
<b>Asiakas:</b>	Sanni-Anni Anneli Vuorinen
<b>Lähetysaika:</b>	22.11.2023 klo 17:20
<b>Tunnus:</b>	[Redacted]

An 'OK' button is located at the bottom right of the page.

Picture 13. Tax card created successfully using only narrator for navigation

### 7.1.2 Report of the testing

Only few minor issues were found during the testing:

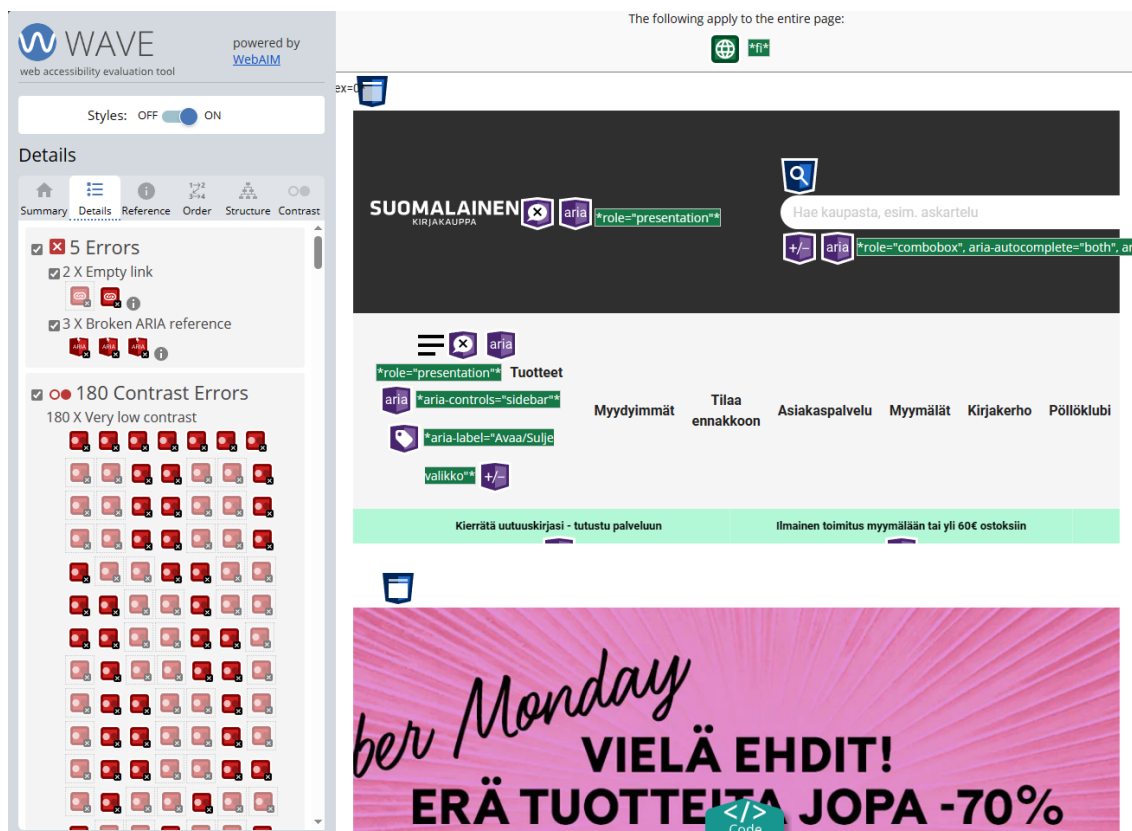
1. Low contrast in the return links  
Suggestion: increase the contrast of the texts to beyond 4.5:1 contrast ratio. For example, use colour #141414.
2. Empty headings  
Suggestion: remove all empty headings to decrease confusion with visually impaired users.
3. Orphaned labels  
Suggestion: make sure all labels envelop the form control it is associated with to reduce duplicate narrator readings and lessen the possibility to get confused when navigating with narrator.

## 7.2 Ordering process of Suomalainen kirjakauppa web shop

Suomalainen kirjakauppa is a private company that is not required by the law to take accessibility into account at all. The testing focuses only on the process of ordering a product from the web shop and is not an exhaustive test of the whole shop's accessibility.

### 7.2.1 Testing the accessibility

The testing process started again with the results from WAVE as seen in picture 14.



Picture 14. Result of the WAVE scan on the front page

The scan from the front page returns quite a lot of errors, though most of them are less severe contrast errors. The contrast errors are still quite numerous and include elements such as navigational backlinks, book author names, reviews, and the original prices of discounted products. Especially bad are the star-based reviews, as they are even a bit hard to see without any visual disabilities.

Other errors were some empty links and broken aria references. Empty links might be confusing when navigating using narrator. Broken aria references mean that aria references exist in page but the target for that reference does not exist. Both will be tested thoroughly during manual testing.



The screenshot shows the WAVE web accessibility evaluation tool interface. On the left, a sidebar displays the following summary:

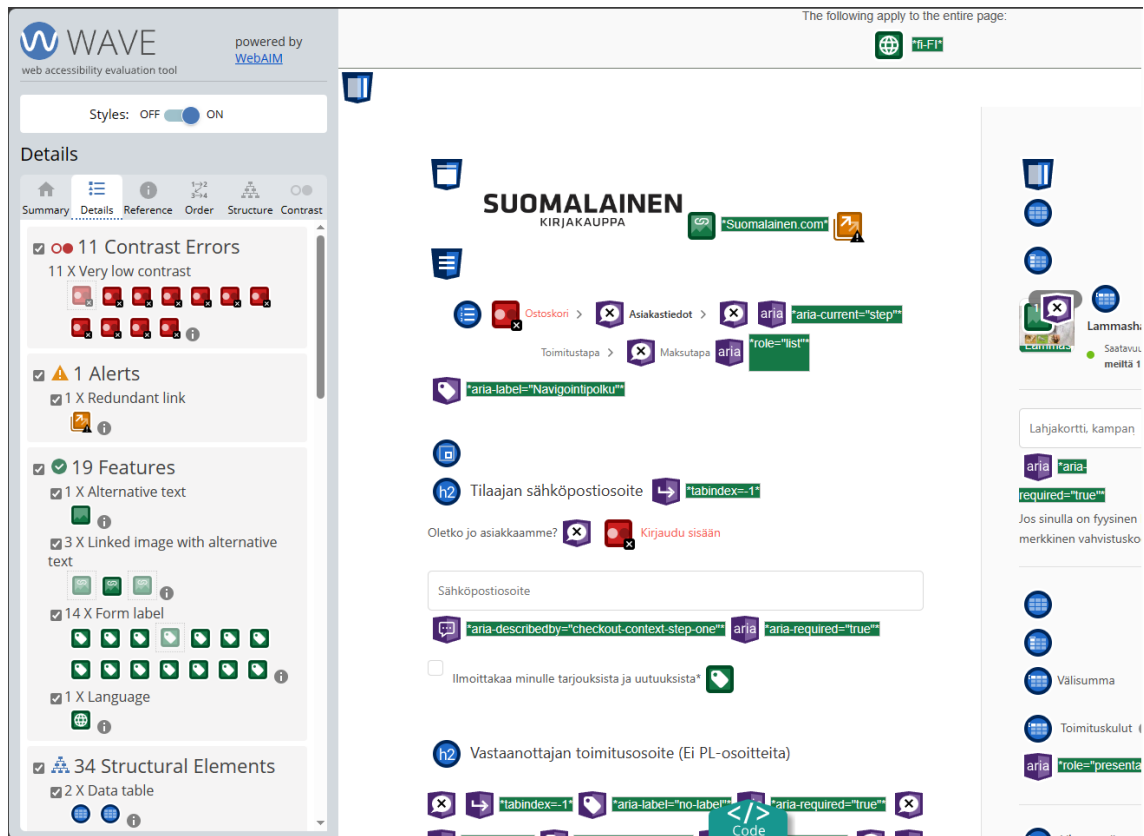
- 19 Errors:**
  - 3 X Missing form label
  - 1 X Empty button
  - 2 X Empty link
  - 13 X Broken ARIA reference
- 28 Contrast Errors:**
  - 28 X Very low contrast
- 73 Alerts:**
  - 1 X Select missing label
  - 1 X Suspicious alternative text

The main area shows a product page for 'SUOMALAINEN KIRJAKAUPPA'. The page content includes a search bar, a navigation menu with items like 'Myydyin', 'Tilaa ennakkoon', 'Asiakaspalvelu', 'Myymlät', 'Kirjakerho', and 'Pöllöklubi', and a product listing for 'Lammas-HAVAINTOJA' by Frank Glorie. Various accessibility issues are highlighted with red icons and error codes such as 'aria-\*role="presentation"', 'aria-controls="sidebar"', 'aria-label="Avaa/Sulje valikko"', and '\*tabindex=0\*'.

Picture 15. WAVE report from a product page

In the product page WAVE picked up many similar issues and some new ones as well. Contrast issues seem to be very common in the online store. Also broken aria references were even more numerous than in the main page.

The new errors present in the product page were missing form labels and an empty button. They will be investigated more during manual testing. The shopping cart page had only contrast errors.



Picture 16. WAVE errors from the shopping cart page

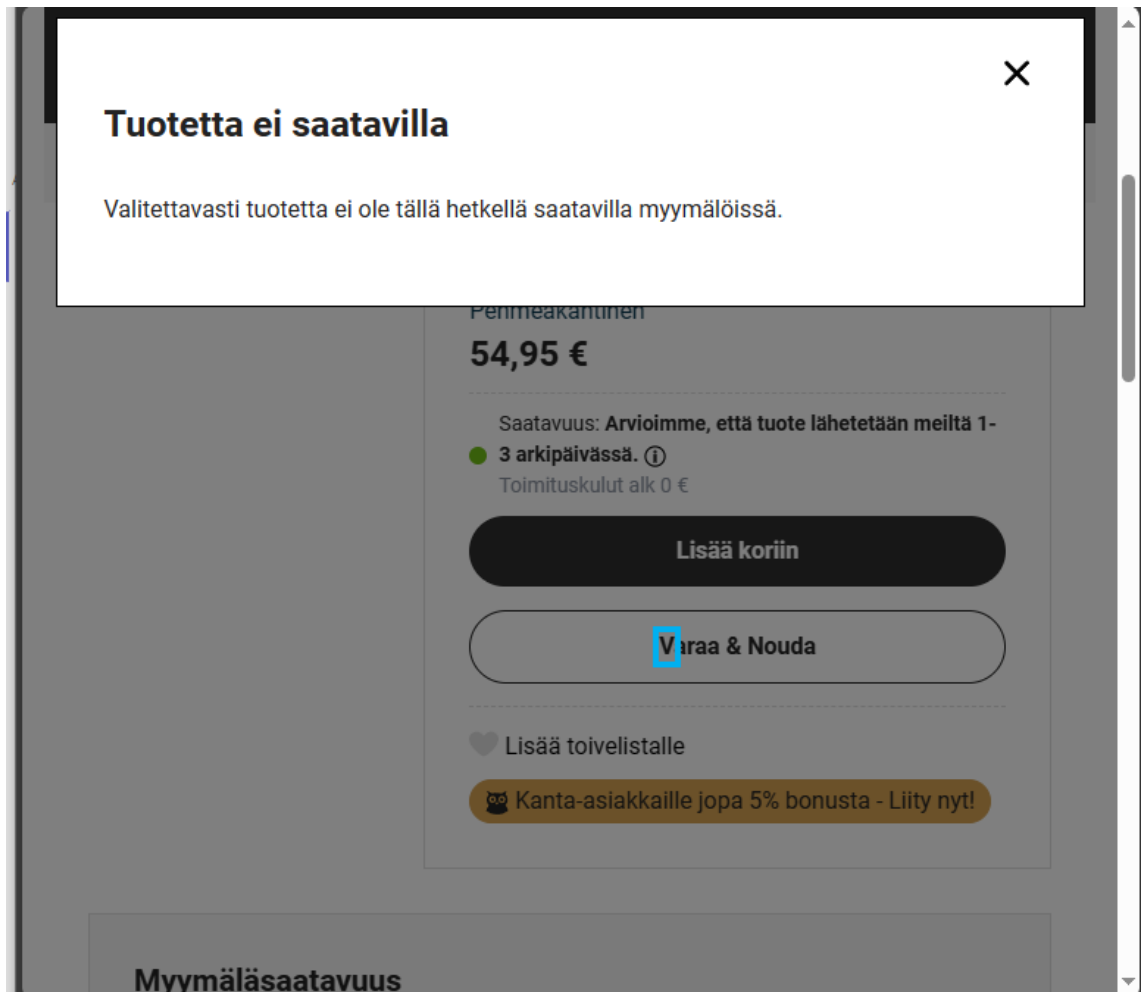
Manual testing began with a problem navigating the page properly. The focus was locked in the customer service chat window and narrator was unable to navigate out of it to the actual web page even after clicking the close button. This issue however was unable to be reproduced with subsequent testing.

Another issue uncovered with narrator was the sale ad banner which narrator was unable to read the text of. Narrator also did not read anything when navigating through a star rating until the very end. This can cause confusion as user needs to advance five times before narrator reads anything. The user might think narrator has crashed or the page has frozen.



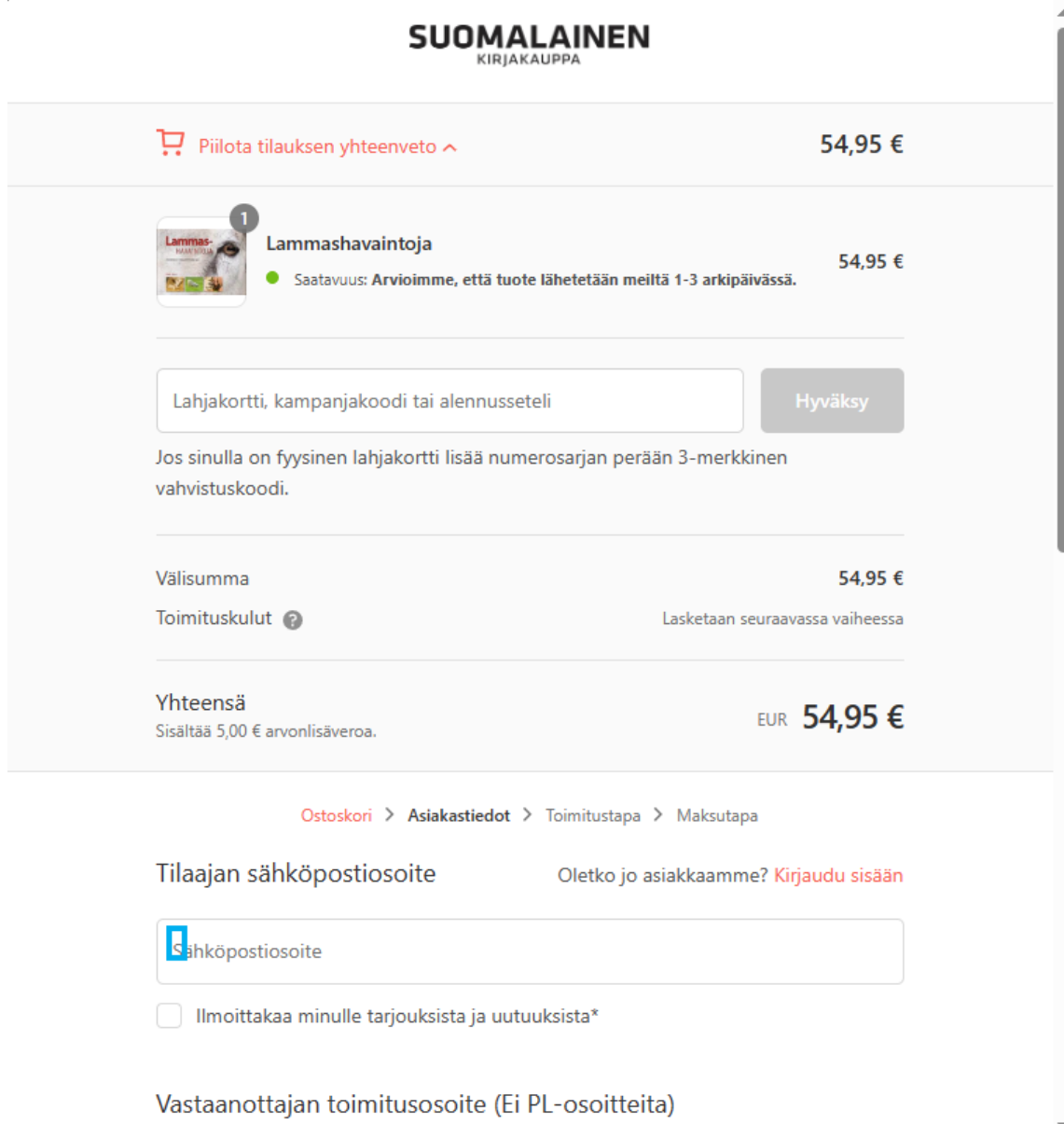
Picture 17. Narrator was unable to read the ad banners

In the product page similar issues with the star rating happened again. Another issue with navigation happened when user tries to click any button that opens a modal. Narrator is unable to move the focus to the new panel and user is left confused assuming that clicking the button did nothing.



Picture 18. Narrator is unable to shift focus to new modal

On the shopping cart page narrator was unable to focus on the top of the page and instead focused on the email field. This might cause confusion as user must manually navigate backwards to review the order details.



Picture 19. Narrator did not focus on the top of the page

Other than that navigation was quite straight forward, and the order was able to be completed. Also, the issues regarding empty links and broken aria references were not issues as narrator did not navigate into them. Testing the various pages with Colorblindly did not uncover any issues that might affect colourblind users.

### 7.2.2 Report of the testing

Few relatively bad but manageable issues were found during the testing.

Issues:

1. Multiple contrast issues were found across the site

Suggestion: increase the contrast in all the texts and graphics that are currently not meeting the recommended 4.5:1 contrast ratio

2. Banner missing descriptive texts

Suggestion: add descriptive texts to every graphical element throughout the page

3. Broken or illogical navigation

Suggestion: make the page shift focus in case a modal or panel is opened. Also make sure that when shifting focus, it is put properly in the top of the new page or element.

4. Star based rating system had navigational issues

Suggestion: group the stars in the rating system in one element so user can instantly hear the rating without confusing navigation.

### 7.3 Conclusion and comparison

Even though most private web services are not required by law to take accessibility into account, most modern web frameworks offer at least relatively good accessibility support out of the box. Still these private services often have many more issues with accessibility than public services that have certain legal requirements that bound them. Even though the site might be usable they might have issues that severely impact the user experience.

## 8 Conclusion

The purpose of this thesis was to examine accessibility testing in software development by introducing several techniques and software designed to aid in the accessibility testing of software. The thesis first introduced some different groups whose daily life is affected by the accessibility of the software they use. Next was the examination of the legal requirements as well as best practices in software accessibility. After that the actual methods for both automated and manual accessibility testing were explored. Lastly a small case study was done that compared the accessibility of a public service to a private service.

Software accessibility is extremely important for many people especially when the digitalisation of the world is only getting faster. Finland has an aging population, and most elderly have some sort of disability or issue that might impact their ability to use software efficiently. One solution to tackle this has been to legally require public organisations and some private companies that offer critical services to follow certain level of accessibility. Most private companies however have varying interest in maintaining a standardised level of accessibility.

The thesis introduces the main methods of conducting accessibility testing and following the case study section can act as a framework of accessibility tests of any other web-based application.

More work in the future would be needed to address differences in testing other types of application such as mobile or desktop applications. Also, additional research regarding the benefits of accessibility would be needed. During the information gathering phase of this thesis no studies were discovered that examined whether enhanced accessibility positively affects a service's user base.

## 9 References

CP-liitto, 2023. *Tietoa cp-vammasta*. [Online]

Available at: <https://cp-liitto.fi/cpvamma/>

[Accessed 2023].

Deque Systems, 2023. *axe-core npm documentation*. [Online]

Available at: <https://www.npmjs.com/package/axe-core>

Deque Systems, 2023. *deque.com*. [Online]

Available at: <https://www.deque.com/axe/>

Eclipse Foundation, 2023. *ACTF documentation*. [Online]

Available at: <https://eclipse.dev/actf/docs/developers/ACTF.pdf>

[Accessed 2023].

Etelä-Suomen aluehallintovirasto, 2023. *saavutettavuusvaatimukset.fi*. [Online]

Available at: <https://www.saavutettavuusvaatimukset.fi>

EUR-Lex, 2016. *Directive (EU) 2016/2102 of the European Parliament and of the Council on the accessibility of the websites and mobile applications of public sector bodies*. [Online]

Available at: <https://eur-lex.europa.eu/eli/dir/2016/2102/oj>

[Accessed 2023].

Finlex, 2019. *Laki digitaalisten palvelujen tarjoamisesta 306/2019*. [Online]

Available at: <https://www.finlex.fi/fi/laki/alkup/2019/20190306>

[Accessed 2023].

Harvard University, 2023. *Manual Testing for Accessibility*. [Online]

Available at: <https://accessibility.huit.harvard.edu/manual-testing-accessibility>

Kavcic, A., 2005. *Software Accessibility: Recommendations and Guidelines*.

Belgrade, Serbia, EUROCON 2005 - The International Conference on

"Computer as a Tool", pp. 1024-1027.



Kustannus Oy Duodecim, 2023. *Parkinsonin tauti*. [Online]

Available at: <https://www.terveyskirjasto.fi/dlk00055>

[Accessed 2023].

Näkövammaisten liitto, 2021. *Näkövammaisuus Suomessa*. [Online]

Available at: <https://www.nakovammaistenliitto.fi/fi/nakovammaisuus-suomessa>

[Accessed 2023].

Sane, P., 2021. *A Brief Survey of Current Software Engineering Practices in Continuous Integration and Automated Accessibility Testing*. Chennai, India, 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 130-134.

VanNess, A., 2021. *Colorblindly GitHub*. [Online]

Available at: <https://github.com/oftheheadland/Colorblindly>

[Accessed 2023].

W3C, 2023. *Web Accessibility Initiative*. [Online]

Available at: <https://www.w3.org/WAI/>

W3C, 2023. *Web Content Accessibility Guidelines (WCAG) 2.1*. [Online]

Available at: <https://www.w3.org/TR/WCAG21/>

[Accessed 2023].

WebAIM, 2023. *wave.webaim.org*. [Online]

Available at: <https://wave.webaim.org/>