

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Mari-Ell Mets 142840

**SUUNISED TAGAMAKS VEEBI SISU  
JUURDEPÄÄSETAVUST PUUTETUNDLIKEL  
MOBIILSEADMETEL**

Bakalaureusetöö

Juhendajad: Liina Martõnjak

Magistrikraad

Gunnar Piho

Doktorikraad

Dotsent

Tallinn 2017

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Mari-Ell Mets

21.05.2017

## **Annotatsioon**

Antud töö eesmärgiks on erivajadustega inimeste ja arendajate vajadustega arvestavate suuniste välja töötamine tagamaks veebi sisu juurdepääsetavust puuetundlikel mobiilseadmetel. Tulem annab vajaliku sisendi loomaks võimalikult vähese vaevaga veebilehti, mis on kasutatavad võimalikult paljudele kasutajatele.

Töös otsitakse vastust küsimustele, millised suunised tagaksid piisava juurdepääsetavuse puuetundlikel seadmetel ja milliste tugitehnoloogiatega peaksid suunised arvestama. Lisaks uuritakse, milliseid suuniseid ootavad veebiarendajad - milline on hea ja arusaadav suunis.

Töö eesmärgi püstitamiseks viiakse läbi intervjuud veebiarendajatega ja erinevate puuetega inimestega, analüüsitakse olemasolevaid juurdepääsetavussuuniseid ja nende kasutatavust, leitakse vastused uurimisküsimustele ning töötatakse välja kergesti kasutatavate, hoomatavate ja funktsionaalsete nõuete komplekt. Lisaks luuakse arendajatelt saadud soovitude järgi veebileht suuniste esitamiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 37 leheküljel, 5 peatükki, 45 joonist, 0 tabelit.

## **Abstract**

### **Web content accessibility guidelines for touchscreen mobile devices**

The purpose of this study is to develop a set of guidelines for designing and developing webpages that would be accessible on touchscreen mobile devices. A set of easy and manageable guidelines presented on a practical website would simplify the workflow for front-end developers and make websites more accessible for everyone.

The thesis focuses on three research questions:

- Which guidelines would provide sufficient accessibility on touchscreen mobile devices?
- What kind of assistive technologies would have to be considered?
- What sort of guidelines does the developer expect – which is a good and comprehensible guideline?

In order to answer the research questions, the author conducts interviews with six front-end developers and six people with different disabilities. Guidelines are gathered from existing accessibility documents, researches and articles, sorted and combined. To present the complete set of guidelines, a simple and practical website is created that best suits the developers' needs.

The thesis is in Estonian and contains 37 pages of text, 5 chapters, 45 figures, 0 tables.

## Lühendite ja mõistete sõnastik

CAPTCHA	<i>Completely Automated Public Turing test to tell Computers and Humans Apart</i> . Robotilõks ehk veebivormidele lisatav Turingi test, mis peab tagama selle, et antud vormi saab täita ja ära saata ainult inimene, mitte robot.
em	relatiivne ühik, mis on seotud tema lähima vanema teksti suurusega
<i>front-end</i> arendaja	veebikehe kasutajaliidese arendaja
hüperpilt	graafiline kujutis, mille eri osadele on vajutades avanevad erinevad lingid, inglise keeles <i>image map</i>
laadileht	dokument, mis teatab veebilehitsejale, kuidas HTML dokumenti kuvada, tuntud ka kui kaskaadilaadistik, inglise keeles <i>Cascading Style Sheets (CSS)</i>
märgistuskeel	käsukeel, mis koosneb dokumendi teksti sisse kirjutavatest märgenditest, inglise keeles <i>markup language</i> , näiteks XML ja HTML
punktkiri	kuuspunktil põhinev kirjaviis, mida saavad lugeda ka pimedad
rem	relatiivne ühik, mis on seotud dokumendi juurelemendi teksti suurusega
<i>trackball</i>	osutusseade, mis liigub mööda kahte telge ning võimaldab ekraanil kursorit juhtida
veebilehtseja	programm veebilehtede lugemiseks, tuntud ka kui brauser või veebisirvija, inglise keeles <i>browser</i>
W3C	<i>World Wide Web Consortium</i> , veebi standardite looja
WAI	<i>Web Accessibility Initiative</i> , W3C juurdepääsetavuse töörühm
WebAIM	<i>Web Accessibility In Mind</i> , Utah juurdepääsetavuse MTÜ

## Sisukord

1. Sissejuhatus.....	10
1.1. Probleem .....	10
1.2. Töö eesmärk.....	11
1.3. Töö struktuur.....	12
2. Suuniste loomise protsess .....	13
2.1. Intervjuud arendajatega.....	13
2.2. Intervjuud erivajadustega inimestega .....	14
2.2.1. Intervjuud liikumispuudega kasutajatega .....	15
2.2.2. Intervjuud nägemispuudega kasutajatega .....	16
2.2.4. Tuvastatud probleemid ja soovitused .....	18
2.3. Kirjanduslikud allikad.....	19
3. Loodud suuniste komplekt.....	22
3.1. Üldised põhimõtted.....	22
3.2. Lehe andmed.....	24
3.3. Navigatsioon .....	25
3.4. Pealkirjad .....	26
3.5. Sisu paigutus .....	27
3.6. Tekst.....	27
3.7. Lingid.....	29
3.7.1. Link, mis laeb alla faili .....	29
3.7.2. Link, mis avaneb uues aknas .....	30
3.7.3. Link, mis viib sama lehe teise kohta.....	31
3.8. Värvid .....	31
3.9. Pildid.....	31
3.9.1. Pilt, mis ei kannu edasi vajalikku infot .....	32
3.9.2. Taustapilt .....	32
3.10. Ikoonid .....	32
3.11. Heli ja video .....	33
3.12. Vormid .....	34

3.12.1. Sisestuskastid .....	35
3.12.2. Raadionupud .....	36
3.12.3. Valikukastid .....	37
3.12.4. Kohustuslikud väljad .....	37
3.13. Veateated.....	37
3.14. Nupud.....	38
3.15. Tabelid .....	39
3.16. Loodud suuniste esitus veebilehena.....	40
4. Loodud suuniste komplekti analüüs.....	42
4.1. Võrdlus olemasolevate suuniste komplektidega.....	42
4.2. Soovitused.....	44
5. Kokkuvõte.....	46
Kasutatud kirjandus .....	47
Lisa 1: küsimustik arendajatele.....	50
Lisa 2: küsimustik erivajadustega inimestele .....	51
Lisa 3: head ja halvad näited.....	52

## Jooniste loetelu

Joonis 1. HTML struktuurielementide kasutamise näide. ....	23
Joonis 2. CSS-is objekti peitmine. ....	23
Joonis 3. HTML lehe tiitel. ....	25
Joonis 4. HTML järjerada. ....	26
Joonis 5. Kahetasemeline HTML menüü. ....	26
Joonis 6. HTML põhisisuni liikumise link. ....	26
Joonis 7. HTML pealkirjade järjestuse näide. ....	27
Joonis 8. Lõiguvahe, reavahe ja teksti suurus CSS-is. ....	28
Joonis 9. Võõrkeelse fraasi keel on HTML-is ära märgitud. ....	29
Joonis 10. Faili allalaadimist alustav HTML link koos selgitusega. ....	29
Joonis 11. Uuel lehel avanev HTML link koos ikooni ja ekraanilugejatele nähtava selgitusega. ....	30
Joonis 12. Uuel lehel avanev HTML link koos ikooniga. ....	30
Joonis 13. HTML ülehüppamislink. ....	31
Joonis 14. HTML pilt tekstilise alternatiivi ning allkirjaga. ....	31
Joonis 15. Illustreeriv HTML ikoon on ekraanilugejate eest peidetud. ....	32
Joonis 16. HTML ikoon koos CSS-is peidetud kirjeldusega. ....	33
Joonis 17. HTML link, mille sisuks on ikoon. ....	33
Joonis 18. HTML video koos tekstilise kirjeldusega. ....	34
Joonis 19. HTML sisestuskast koos pealkirjaga. ....	34
Joonis 20. Grupeeritud ja ühise pealkirjaga HTML raadionupud. ....	35
Joonis 21. HTML sisestuskast vahemiku, ekstremaalsete väärtuste ning sammuga. ....	35
Joonis 22. HTML sisestuskast koos mustrikontrolliga. ....	36
Joonis 23. Otsingunupul olev kiri selgitab sisestuskasti funktsiooni. ....	36
Joonis 24. HTML juurdepääsetav otsingukast. ....	36
Joonis 25. HTML raadionuppudel on ühine nimi. ....	36
Joonis 27. Veateate kuvamine HTML-is. ....	38
Joonis 28. Nupu fokuseeritud ja aktiivne olek on eristatud CSS-iga. ....	39
Joonis 29. Juurdepääsetav HTML sulgemisnupp. ....	39



Joonis 30. HTML tabel kirjelduse, pealkirja, päise ja kehandiga.....	39
Joonis 31. Loodud veebilehe avakuva ekraanitõmmis. ....	40
Joonis 32. Loodud veebilehe avatud akordioniga ekraanitõmmis.....	41
Joonis 33. (a) halb näide: kasutajat sunnitakse lohistama, (b) hea näide: kasutajalt küsitakse loogiline küsimus.....	52
Joonis 34. (a) halb näide: kõik vahelehed algavad sama sõnaga, (b) hea näide: vahelehed algavad alamlehe nimega.....	52
Joonis 35. (a) halb näide: menüü ikooni võib pidada logo osaks, (b) hea näide: lisaks ikoonile on olemas ka tekst.....	52
Joonis 36. (a) halb näide: ei ole selge, kuidas otsingut käivitada, (b) hea näide: on selge, et vajutada tuleb nuppu.....	53
Joonis 37. (a) halb näide: graafiku sektorid on eristatud vaid värviga, (b) hea näide: graafiku sektoritel on lisaks värvile juures ka sildid.....	53
Joonis 38. (a) halb näide: liiga madal kontrastsus 2.21:1, (b) hea näide: kõrge kontrastsus 14.79:1. ....	53
Joonis 39. (a) halb näide: tekstiline alternatiiv puudub ja info läheb kaduma, (b) hea näide: tekstiline alternatiiv on olemas ning kannab endas sama infot, mis pilt.....	54
Joonis 40. (a) halb näide: menüüikoon ei ava menüüd, vaid lisafiltrid, (b) hea näide: tööriistaikoon avab lisafiltrid.....	54
Joonis 41. (a) halb näide: pealkirjades on palju sõnakordust, mis on visuaalselt segadusseajav, (b) hea näide: kokkukuuluvad väljad on grupeeritud ja ühise pealkirja all.....	54
Joonis 42. (a) halb näide: telefonivälja tüübiks on määratud <i>text</i> ning mobiilseadmes kuvatakse vaikimisi klaviatuur, kus numbrit sisestamiseks peab tegema lisaliigutuse ning esmalt on näha tähteklahvid, mida antud välja puhul vaja ei ole. (b) hea näide: telefonivälja tüübiks on määratud <i>tel</i> ning mobiilseadmes kuvatakse numbriklaviatuur, mis on loodud just telefoninumbrate sisestamiseks. Lisaks pakutakse otseteed oma telefoninumbri sisestamiseks.....	55
Joonis 43. (a) halb näide: selgitus täрни koha asub peale välju, (b) hea näide: selgitus täрни kohta asub enne välju.....	55
Joonis 44. (a) halb näide: vigane väli on eristatud vaid värviga, (b) hea näide: vigane väli on lisaks värvile eristatud ka ikooniga.....	56
Joonis 45. (a) halb näide: ainult tekst on klikitav, (b) hea näide: kogu nupu pind on klikitav.....	56

# 1.Sissejuhatus

Veebiruum on vaieldamatult võimsaim infovaramu, mis on seotud meie igapäevaeluga pea igas valdkonnas. Veebikeskkondade loomisel ei kiputa aga alati mõtlema puuete ja erivajadustega kasutajatele, kelle jaoks info saamine veebist võib olla tõeline katsumus. Veebi kasutamise võimaldamine on osa võrdsete võimaluste tagamisest ning aitab ka erivajadustega inimestel aktiivsemalt ühiskonnaelus kaasa lüüa.

*World Wide Web Consortiumi* (W3C) loodud juurdepääsetavuse töörühm *Web Accessibility Initiative* (WAI) on defineerinud veebi juurdepääsetavuse kui olukorra, kus puuetega inimesed suudavad veebi tajuda ja mõista, veebis navigeerida ja sellega suhelda ning sellesse ise panustada [1]. Juurdepääsetav veeb on paindlik ja sobib erinevate kasutajate vajaduste, eelistuste ja olukordadega. Seega ei too juurdepääsetavus kasu ainult puuetega inimestele, vaid näiteks ka nendele, kelle võimed on vananemise tõttu muutunud, kellel ajutine trauma nagu näiteks käeluumurd ning ka inimestele, kes kasutavad vananenud seadmeid või aeglast internetiühendust.

## 1.1. Probleem

Aastal 1999 avaldas WAI juurdepääsetavussuuniste dokumendi *Web Content Accessibility Guidelines* (WCAG) 1.0 [2]. Üheksa aastat hiljem avaldati täiendatud dokument WCAG 2.0, mis on tänaseni W3C ametlikuks soovitusel [2]. Mõned riigid on loonud omaenda juurdepääsetavuskriteeriumid - näiteks USA-s on standardiks *Section 508* [3], Prantsusmaal RGAA 3 [4] ja Saksamaal BITV 2 [5]. Üldiselt aga tunnustatakse WCAG-d kui rahvusvahelist standardit [6]. Aastast 2012 on Eesti riigi ja kohaliku omavalitsuse veebilehtedel kohustus vastata WCAG 2.0 AA taseme edukriteeriumitele. [7, lk 19]. Majandus- ja kommunikatsiooni ministeeriumi 2015. aasta uuringust selgus aga, et ainult 6% avaliku sektori veebilehtedest vastavad WCAG 2.0 miinimumtaseme nõuetele [8, lk 4]. Osaledes erinevates veebiarendusprojektides, on autor tähele pannud, et kui kohustus puudub, siis juurdepääsetavussuuniseid ei järgita, sest neid on palju ja

erinevaid, need on valdavalt keeruliselt esitatud ja raskesti hoomatavad. Lisaks sellele pole arendusprojektidesse tihti arvestatud juurdepääsetavuse tagamisele kuluvat aega. Mõnede juurdepääsetavuskriteeriumitele vastavuse kontrollimine on lihtne, kuid nende põhjalik hindamine on keerukas ja vajab spetsialisti ülevaatus. On olemas erinevaid automaatkontrolli tarkvarasid, mis kontrollivad märgendite hierarhia korrektsust, tekstiliste alternatiivide ja vormiväljade pealkirjade olemasolu, kuid ei saa kontrollida näiteks interaktiivsete elementide visuaalset eristatavust ja sisu loogilist järjekorda. Seega pole ükski neist piisav kinnitamaks, et veebileht on tõesti juurdepääsetav.

Mobiilseadmed on vaieldamatult populaarsemaks muutuvad veebitarbimisvahendid. Oktoobris 2016 kasutati mobiilseadmed esmakordselt ajaloos interneti tarbimiseks rohkem kui arvuteid - vastavalt 51,3% ja 48,7% [9]. Sellest lähtuvalt oleks mõistlik, et kõik veebilehed, mida loome, kohandaksid ka tahvelarvutitele ja mobiilidele, olles seejuures kasutajasõbralikud, täisfunktsionaalsed ja juurdepääsetavad. Veebilehti ei looda enam suurele ja väiksele ekraanile eraldi, vaid pannakse paika murdepunktid (inglise keeles *breaking points*), mis määravad elementide suuruse ja paigutuse erinevate ekraanisuuruste korral. Kuigi see ei lahenda kõiki probleeme, peetakse sellist lähenemisviisi tehnoloogia arenedes tulevikule kõige vastupidavamaks ning kõige kergemini hooldatavaks [10]. Kohanduva veebi loomisel ei piisa vaid kontrollimisest, et kõik elemendid mahuksid väikesele ekraanile ära. Tagamaks võrdväärset kasutajakogemust suurel ekraanil ja mobiilseadmel, tuleb lähtuda veebilehe sisust. Elemente tuleb ümber tõsta, muuta menüüsid ja info esitusviisi. Arvestada tuleb, et mobiilseadmetes kasutatakse tihti piiratud andmemahuga ja aeglast internetti.

Olemasolevad juurdepääsetavussuuniste komplektid nagu WCAG 2.0 [2] ja *Section 508* [11] on olemuselt normatiivsed dokumentatsioonid, mitte aga mugavalt arendusprotsessi lülitavad tööriistad. Juurdepääsetavuse tagamise tehnikaid mobiilseadmetele on kirjeldatud eraldi dokumentides, kuid ajal, mil mobiilidel kasutatakse veebi rohkem kui arvutites, pole mõistlik luua veebilehti, mis on juurdepääsetavad ainult suure ekraani, klaviatuuri ja hiirega seadmel.

## **1.2. Töö eesmärk**

Bakalaureusetöö eesmärgiks on välja töötada erivajadustega inimeste ja arendajate vajadustega arvestavate suuniste komplekt, mis tagaks veebi sisu juurdepääsetavuse ka

puuetundlikel mobiilseadmetel. Selline suuniste komplekt annaks vajaliku sisendi loomaks võimalikult vähese vaevaga veebilehti, mis on kasutatavad võimalikult paljudele kasutajatele.

Eesmärgi saavutamiseks leitakse vastused järgmisele uurimisküsimustele:

- Millised suunised tagaksid piisava juurdepääsetavuse puuetundlikel seadmetel?
- Milliste tugitehnoloogiatega peaksid juurdepääsetavussuunised arvestama?
- Milliseid suuniseid soovivad veebiarendajad - milline on hea ja arusaadav suunis?

Töö eesmärgi püstitamiseks viiakse läbi intervjuud veebiarendajatega ja erinevate erivajadustega inimestega, analüüsitakse olemasolevaid juurdepääsetavussuuniseid ja nende kasutatavust, leitakse vastused uurimisküsimustele ning töötatakse välja kergesti kasutatavate, hoomatavate ja funktsionaalsete suuniste komplekt. Loodud suuniste esitamiseks luuakse kiirelt ja mugavalt kasutatav veebileht.

### **1.3. Töö struktuur**

Käesolev bakalaureusetöö koosneb kolmest peatükist. Esimeses peatükis “Suuniste loomise protsess” selgitatakse kasutatud meetodikat ning kirjeldatakse, kuidas leiti vastused uurimisküsimustele.

Teises peatükis “Loodud suuniste komplekt” esitatakse bakalaureusetöö tulem - kogum nõudeid juurdepääsetavuse tagamiseks nii lauaarvutites kui mobiilseadmetel koos vajalike selgituste ja näidetega.

Viimases peatükis “Loodud suuniste komplekti analüüs” analüüsitakse bakalaureusetöö tulemit kõrvutades seda varem loodud juurdepääsetavuskriteeriumite ja suuniste-komplektidega. Tuuakse välja loodud veebilehe eelised ja puudused võrreldes olemasolevate keskkondadega ning pakutakse välja võimalikud sammud tulevikuks.

## 2. Suuniste loomise protsess

Selles peatükis annab autor ülevaate töö eesmärgi saavutamiseks kasutatud meetodikatest. Lisaks selgitatakse, kuidas ja millised vastused leiti püstitatud uurimisküsimustele.

### 2.1. Intervjuud arendajatega

Selgitamaks välja arendajate soove ja vajadusi seoses juurdepääsetavusnõuetega, viis autor läbi intervjuud kuue kasutajaliidese arendajaga (inglise keeles *front-end developer*), kellest pooled tegelevad ka veebidisainiga.

Intervjuud viidi läbi iga arendajaga individuaalselt. Intervjuu koosnes kaheteistkümnest küsimusest (Lisa 1) ning kestis umbes tund aega. Autor uuris, mida arendajad juurdepääsetavuse tagamisest teavad, kuidas nad selle oma töösse lülitavad ning mida lehtede mobiilseadmetele kohandamisel silmas peavad. Samuti palus autor arendajatel kommenteerida erinevaid keskkondi, kus kirjeldatakse juurdepääsetavuse tagamise suuniseid ning -tehnikaid: WCAG 2.0 ametlik dokumentatsioon [2], *Mobile Web Best Practices* (MWBP) 1.0 ametlik dokumentatsioon [12], WebAIM WCAG 2.0 *Checklist* [13], WebAIM *Section 508 Checklist* [14], WAI *Easy Checks* [15], Wuhcag WCAG 2.0 *Checklist* [16], *The A11Y Project Web Accessibility Checklist* [17], Elsevier *Accessibility Checklist* [18].

Intervjuudest arendajatega sai selgeks, et vajalike suuniste leidmine erinevatelt lehtedelt veebis on keeruline. Ametlikud dokumentatsioonid on usaldusväärsed, kuid raskesti hoomatavad ning tihti aastaid uuendamata. Mitteametlikud suuniste nimekirjad on tihti lihtsamini jälgitavad ja varustatud rohkemate praktiliste näidetega, kuid ei pruugi olla täiesti usaldusväärsed. Kindlasti on vaja suuniste komplekte uute märgistuskeele elementide või tehnoloogiate turule tulekul täiendada ja uuendada.

Vähestes suuniste komplektides on arvestatud kohanduva veebi ning mobiilseadmetega, mis on täna arendajate põhifookuses. Ametlik W3C dokument MWBP [12] on avaldatud aastal 2008, mil turule olid tulnud esimesed puuetundlikud mobiilid. Arendajad usuvad,

et mõned neist suunistest on vananenud või oma tähtsuse kaotanud, sest tehnoloogia on vahepeal palju arenenud. Seega ei juhindu arendajad antud suunistest.

Juurdepäätavuse kontrollimiseks kasutavad arendajad veebipõhiseid validaatoreid, mille raportid on aga tihti raskesti kasutatavad ning annavad vähe infot selle kohta, kuidas tekkinud vigu parandada. Mõned arendajad tõid ka välja, et kasutavad raamistikke, kus elemendid on juba juurdepäätavusega arvestades disainitud. Tavaliselt on aga vaja raamistiku elemente siiski oma projektile vastavalt muuta ning siis võib juhtuda, et juurdepäätavus pole enam tagatud.

Juurdepäätavuse tagamist pärsib ka asjaolu, et projektidesse ei arvestata sellele kuluvat aega, sest tellijad ei näe selles konkreetset ärikasu. Järelikult peab juurdepäätavuse tööriist olema arendajale võimalikult mugav ja aegasäästev. Arendajad pidasid mugavaimaks käsitlusi, kus suunised olid grupeeritud märgistuskeele elementide kaupa. See lihtsustab juurdepäätavuse tagamise lülitamist tööprotsessi, sest iga elementi luues on lihtne leida sellele vastavaid nõudeid. Nõude juures peab olema konkreetset kirjas, kuidas mingit viga vältida või vajalikku elementi kasutada. See säästab otsimisele kuluvat aega ning tagab, et leitakse korrektne lahendus.

Arendajatelt saadud sisendi põhjal peab loodav suuniste komplekt olema veebipõhine ja võimalikult kiirelt kasutatav. Kuna suuniseid on palju, peab lehel saama kasutada otsingut ja filtreid, et jagada info väiksemateks hoomatavamateks osadeks. Info tuleb esmapilgul esitada võimalikult konkreetset ja ülevaatlikult. Detailsem info peab avanema samal lehel, et vältida edasi-tagasi navigeerimist.

## **2.2. Intervjuud erivajadustega inimestega**

Autor viis läbi intervjuud kuue erivajadustega inimesega, et uurida, millised probleemid võivad veebilehti kasutades tekkida arvutis ja millised mobiilseadmel ning kuidas need probleemid üksteisest erinevad. Intervjuudes osales kolm erineva motoorikahäirega, kaks pimedat ning üks nägemispuudega inimene. Antud töö raames ei kaasatud intervjuudesse kurte ja vaegkuuljaid, sest heli on vähestel veebilehtedel peamine infokandja ning üldjuhul kasutavad vaegkuuljad veebi nagu tavalised kasutajad. Intervjuu koosnes neljateistkümnest põhiküsimusest (Lisa 2), mille juurde küsiti lisaküsimusi.

### 2.2.1. Intervjuud liikumispuudega kasutajatega

Esmalt avanes autoril võimalus intervjuuerida noormeest, kes on sündinud Duchenne'i lihasdüstroofiaga ja saab liigutada vaid mõnda sõrme. Mees kasutab igapäevaselt oma suhteliselt vana Android nutitelefoni ning sülearvutit, mis on asetatud laual kõrgemale alusele, silmade kõrgusele. Ta kasutab *trackball*'iga hiirt ning ekraanile kuvatavat klaviatuuri. Intervjuu käigus sai selgeks, et tänased juurdepääsetavusnõuded keskenduvad suuresti inimestele, kes kasutavad navigeerimiseks ainult klaviatuuri, kuid mitte inimestele, kes kasutavad ainult hiirt. Täieliku juurdepääsetavuse tagamiseks oleks aga tarvis, et veebilehega saaks suhelda nii ainult klaviatuuri kui ka ainult hiirega. Näitena tõi mees olukorra, kus e-valimiste keskkond ei võimaldanud kandidaadi numbrit ekraanile kuvatava klaviatuuri abil sisestada. Küsida tuli kaaslaste abi ning välja öelda, kelle poolt hääletada sooviti - info, mis peaks jääma igale inimesele privaatseks.

Mobiili kasutab mees tavaliselt laua peale asetatuna, kodust eemal olles vaid hädaolukorras. Puuetundliku seadme põhiline eelis on mehe jaoks see, et piisab vaid kergest puudutusest ning jõudu pole vaja rakendada. Ebamugavust tekitavad veebilehed, kus puudub menüü ja tuleb ainult alla kerida. Samuti on ebamugav igasugune teksti sisestamine. Kuna ka hiire kasutamine on keeruline, jätavad efektid, mis tekivad hiirega elemendi kohal hõljudes, mehe hätta.

Teisena sai autor võimaluse intervjuuerida reesuskonfliktiga meest, kel puudub võime rääkida, kõndida ning kontrollida oma käsi. Ta on õppinud iseseisvalt programmeerima ning loonud ise kõnesüntesaatori programmi, mida ta suhtlemiseks kasutab. Kuigi mees tegeleb veebiarendusega, ei järgi ta formaalseid juurdepääsetavussuuniseid, sest need on keerulised ja raskesti kasutatavad. Mehel on lisaks lauaarvutile ka iPhone ja iPad, kusjuures kõiki neid kasutab ta jalaga. Puuetundliku ekraani eeliseks peab mees seda, et vajutamiseks pole vaja rakendada jõudu ning kompaktses seadmes saab vaevata vahetada näiteks erinevaid klaviatuure. Raskusi tekitavad veebilehed, mis pole loodud kohanduma mobiilseadmetes ning kus nupud ja tekst on väga väikesed.

Kolmandana intervjueeris autor meest, kellel on käimiskäsi ning kes ei saa juhtida oma käsi. Mees liigutab arvutihiirt lõuaga ning kasutab klaviatuuri suus oleva hambaharjaga. Klahvikombinatsioonide vajutamiseks kasutab mees nakkeklahvide (inglise keeles *sticky keys*) funktsiooni. Tööl olles on mehe käsutuses ka Tobii seade, mis võimaldab kursorit juhtida silmi liigutades. Lingile või nupule saab vajutada sellele

pilguga fokuseerides. Lauale asetatud Android nutitelefoniga kasutab ta nina, keele või puutepulgaga. Põhiline probleem on mehe jaoks see, et ekraanile peab olema väga lähedal. See kurnab silmi ning teeb sisu hoomamise keerulisemaks. Puutetundliku ekraani eeliseks peab mees aga seda, et veebilehel navigeerimine on mugavam ja kiirem kui klaviatuuriga. Ebamugavad on lehed, kus on keeruline navigatsioon ja ebaselged menüüpunktide nimetused. Mees tõi välja ka ärritava situatsiooni, kus veebileht avaneb mobiilis palju piiratumate võimalustega kui arvutis. Mobiilis peaks saama kasutada kõiki funktsionaalsusi, mis arvutis.

### **2.2.2. Intervjuud nägemispuudega kasutajatega**

Esmalt intervjueris autor üht pimedat noormeest, kes kasutab sülearvutit, iPhone'i, Apple Watchi ja iPadi. Nii sülearvutist kui ka mobiilist loeb ekraanilugeja mehele kõrvaklappidesse. Lisaks kannab noormees kaasas väikest punktkirjaseadet juhuks, kui on vaja näiteks arvutist ilma kõrvaklappide ja helita kella vaadata. Mehe mobiili katab spetsiaalne kile, millel mõned kohad on tähistatud kõrgemate punktidega, et ekraanil paremini orienteeruda. Oma iPadi eeliseks peab mees tema kerget kaalu ja mobiilsust. Mugava veebilehe aluseks on mehe sõnul lihtne ja mugav navigatsioon. Menüüpunktid peavad olema sõnastatud nii, et oleks selge, mis nende alt leida võib. Veebilehe struktuur peab olema korrektselt HTML elementidega märgistatud, kuna ekraanilugeja ja klaviatuuriga navigeerimine käib tavaliselt mööda pealkirju või linke. Kasu toovad ka nn ülehüppamislingid, mis on nähtavad ainult ekraanilugejale. Kui veebilehest on tehtud eraldi pimedatele mõeldud tekstiversioon, siis seda mees pigem ei kasutaks, sest selliseid lehti tavaliselt ei uuendata.

Teisena tegi autor intervjuu pimedaga mehega, kes on iseseisvalt ekraanilugejat kasutades õppinud kirjutama HTML koodi. Endaloodud kodulehel [19] kirjutab mees juurde-pääsetavusest ja erinevatest tugitehnoloogiatest. Ta on kirjeldanud nägemispuudega inimeste erisoovide kokkulangevusi teiste kasutajarühmadega ning toonud välja olulisi faktoreid, mis veebilehti neile mugavamaks muudavad. Mees kasutab Android nutitelefoniga, iPadi ja arvutit. Puutetundlikke seadmeid kasutab mees iga päev ning peab nende eeliseks kerget kaalu ja mobiilsust.

Kõige enam probleeme tekitab mehele nutiseadme klaviatuuri kasutamine, kuid kuna mobiilis kirjutamine toimub tavaliselt väljaspool kodu avalikus kohas, ei soovi mees kasutada kõne tekstiks muutmise rakendust. On selge, et teksti sisestamise vajadus tuleks



veebilehtedel viia miinimumini - kasutada lühikesi veebiaadresse ja tekstiväljade asemel alternatiivseid vormivälju. Mees nimetab probleemina ka selle, kui korraka kuvatakse kasutajale liiga palju valikuid ja funktsioone - ekraanilugejat kasutades võib minna kaua aega, et leida üles soovitud nupp või link. Mees märgib ka ära, et probleemid võivad tekkida lehtedel, mis on väga dünaamilised, sest kui sisu laetakse osade kaupa, on ekraanilugejal keeruline lugemisjärge hoida. Mees tõi välja, et kuna ekraanilugeja kasutajad navigeerivad tihti mööda struktuurielemente, on hea, kui lisaks korrektse hierarhiaga pealkirjadele on igal lehel ka *<main>*, *<header>*, *<footer>* ja *<search>* struktuurielemendid.

Mõlemad pimedad intervjuueeritavad ütlesid, et on pidanud sisestama infot vormiväljadele, millel pole nimetusi (inglise keeles *label*) ning mida pole seega võimalik ekraanilugejaga kasutada. Tuli ka välja, et ekraanilugejad ei saa hakkama robotilõksudega (inglise keeles *Completely Automated Public Turing test to tell Computers and Humans Apart* ehk CAPTCHA), kus peab näiteks sisestama pildil kujutatud sõna või tegema linnukese teatud märkeruutu. Ekraanilugeja töötab simuleerides hiireklikke, kuid sellistel vormidel on simuleerimine turvakaalutlustel keelatud. WCAG suunise 1.1.1 edukriteeriumid nõuavad, et robotilõksul oleks selle eesmärgi kirjeldav tekstiline alternatiiv ning et pakutaks ka alternatiivset meetodit enda tuvastamiseks [20]. Samas ei ole nõutud, et alternatiivne meetod oleks kasutatav hiireklikke simuleeriva tugitehnoloogia abil. Ekraanilugejaga pole võimalik simuleerida ka hiire allavajutust ega lahtilaskmist - seega ei tohi kasutada neid tegevusi päästiksündustena.

Kolmandana intervjuueeris autor vaegnägijat, kes ei kasuta ekraanilugejat, kuid suurendab arvutiekraani programmiga *Magnifier*. Naise mobiili sisu on maksimaalselt suurendatud ja lisaks kasutab ta ka tugevat luupi. Kuna mobiili klaviatuuril kirjutamine on keeruline, kasutab naine võimalusel kõne tekstiks muutmise rakendust.

On väga oluline, et veebilehe teksti suurust oleks võimalik veebilehitsejas muuta. Ilmnes aga, et kui lisaks ekraanisuurendusprogrammile kasutatakse ka veebilehel olevat teksti suurendamise funktsiooni, võib osa infost oma suuruse tõttu teiste elementide taha ära kaduda. Veebilehitsejasse sisseehitatud teksti suurendamise funktsioon on töökindlam ning vaegnägijatel on see tavaliselt juba sisse lülitanud. Ekraanisuurendajat kasutava inimese jaoks on ka väga oluline, et elemendid nagu menüü, otsing ja sulgemisnupp

asuksid veebilehel oodatavates kohtades, sest korraga ei mahu ekraanile kogu leht. Navigeerimiseks kasutab naine parema meelega otsingut, sest ka menüüd ei mahu ekraanile ning on tihti mitmetasemelised ja keerulised. Kuigi paljud vaegnägijad peavad veebilehtedele sisseehitatud värvide kontrastsuse muutmise võimalust väga oluliseks [21, lk 53], siis intervjuueeritav seda ei kasuta, sest kollast teksti mustal taustal on tema jaoks raskem lugeda kui musta teksti valgel taustal. Probleemid tekivad halli või helesinise tekstiga valgel taustal ja vastupidi. Raske on lugeda ka kaldkirjas teksti.

#### **2.2.4. Tuvastatud probleemid ja soovitused**

Kõige olulisem on loomulikult, et veebileht kohanduks ka väiksema ekraaniga seadmetele. Veebilehe mobiiliversioonis ei tohi olla vähem funktsionaalsust kui arvutiversioonis. Pimedad kasutajad ei pruugi teada, kui suur on veebilehitseja avatud aken. Pealegi võib kasutaja vabatahtlikult kasutada arvutis veebilehe mobiiliversiooni, sest nendel on enamasti vähem reklaame ja lihtsam navigatsioon. Seega on oluline, et suurel ekraanil oleks võimalik kahandada veebilehitseja aken mobiiliversioonile, kaotamata selle käigus ühtki olulist funktsionaalsust.

Mõned, üldiselt mahukad veebilehed võivad vastata küll formaalsetele juurdepääsetavuskriteeriumitele, kuid olla reaalsuses väga ebamugavad. Näitena tulid intervjuudest välja ministeriumide lehed, kus murekohaks on menüü, sest see on mitmetasemeline ning menüüpunktide pealkirjadest pole lihtne aru saada, mis nende alt leida võib. Menüüpunktide nimed tekitasid probleeme eranditult kõigile intervjuueeritavatele. Menüü peab olema üheselt arusaadav ja maksimaalselt kahetasemeline. Menüü ja muud olulised struktuurielemendid peavad olema korrektses märgistuskeeles esitatud ning asuma lehel oodatavates kohtades. Lehel tuleb kasutada korrektse hierarhiaga pealkirju ning struktuurielemente, mis aitavad navigeerimisel ebavajalikke sektsioone lehel vahele jätta. Lisaks peaksid lehel olema ülehüppamislingid ning lisaks menüüle ka otsinguvõimalus või sisukaart.

Veebileht peab olema kasutatav nii ainult klaviatuuriga kui ka ainult hiirega. Päästiksündmustena ei tohi kasutada hiire allavajutust (sündmus *onmousedown*), lahtilaskmist (sündmus *onmouseup*), topeltklõpsu (sündmus *ondblclick*), kursori peale viimist (sündmus *onmouseover*) ega kursori eemaldamist (sündmus *onmouseout*), sest need on probleemsed nii mootorikahäiretega kui ka pimedatele kasutajatele. Nende asemel tuleks alati kasutada sündmusena tavalist hiireklõpsu (inglise keeles *click*). Kuna

teksti sisestamine tekitas probleeme kõigile intervjueeritavatele, tuleks seda kasutajalt nõuda nii vähe kui võimalik.

### **2.3. Kirjanduslikud allikad**

Autor uuris valikut teadusartikleid ja uurimistöid veebilehtede ja mobiilseadmete juurdepääsetavuse teemadel. Selles peatükis toob autor ülevaate peamistest kasutatud kirjandusallikatest.

WCAG 2.0 ametlik dokumentatsioon [2] kirjeldab 12 juurdepääsetavuse tagamise kriteeriumit, mis on jagatud nelja suurde kategooriasse. Dokument on mõeldud kasutamiseks kahe lisadokumendiga - suuniste olulisust selgitava dokumendiga ning suuniste tehnikate dokumendiga. Käesoleva töö tulemi loomisel on kasutatud kõiki kolme dokumenti. WCAG suunised on jagatud olulisuse järgi kolme tasemesse. Käesolev töö uuris eesmärgi saavutamiseks ainult WCAG A ja AA taseme suunised, sest AAA tase on soovituslik ega ole peaaegu kunagi arendusprojektide kriteeriumiks. Kuigi WCAG suuniseid tunnustatakse kui rahvusvahelist standardit, on leitud, et need keskenduvad liialt tehnilisele olemisele - veebilehele, mitte aga selle kasutajatele ja nende eesmärkidele [22, lk 1]. Selline lähenemine kontrollib küll kriteeriumite täidetust, kuid võib jätta tähelepanu alt välja kasutajakogemuse [22, lk 2].

Power, et al. [23] viisid läbi uuringu, mis tõi välja, et WCAG 2.0 edukriteeriumid katavad vaid 50,4% pimedate kasutajate probleemidest. Uuringus osales 32 pimedat kasutajat, kes testisid 16 veebilehe juurdepääsetavust. Välja tuli probleeme, mida WCAG suunised ei kata: lehel puudub sisu, mida kasutaja ootas; kasutaja leiab sisu sealt, kus ta seda ei oodanud; liiga keeruline arhitektuur; lehel leidub katkiseid linke.

Rootsi juurdepääsetavuse töögrupp Funka leiab samuti, et WCAG nõuded pole täieliku juurdepääsetavuse tagamiseks piisavad ning on avaldanud kogumiku läbitestitud suuniseid mobiilsete kasutajaliideste arendamiseks [24] ja mobiilse navigatsiooni loomiseks [25]. Funka loodud nõuded on kaasatud käesoleva töö tulemi loomisesse.

Rømen ja Svanæs [26] viisid läbi uuringu seitsme puuetega ja kuue tavakasutajaga ning tõi samuti välja mitmeid probleeme, mida WCAG 1.0 ja 2.0 suuniste ning ISO 9241-171:2008 standardi järgimine ei paranda. Näiteks pealkirjad ja lingid peaksid algama alati kõige tähtsama ja sisukama sõnaga, sest nii on need paremini eristatavad nii

vaegnägijatele kui ka ekraanilugejaga kasutajatele. Ebavajalikke ja üksteist kordavaid linke tuleb vältida - näiteks “Loe rohkem” link, mis viib samale lehele kui tekst ise, nõuab lisapingutust ekraanilugejaga kasutajatelt. Ka linkide paigutusele tuleks pöörata tähelepanu - need ei tohi asetseda üksteisele liiga lähedal, aga ka mitte liiga kaugel. Autorid toovad välja sagedase vea, kus nupu suurus ei ühti vajutatava ala suurusega. Motoorikahäiretega kasutajatele tekitavad probleeme dünaamilised *mouse-over* menüüd, mida ei saa paigale lukustada. Rømeni ja Svanæsi ning Power, et al. välja toodud soovitusi on rakendatud käesolevas töös.

Hänni [21] on uurinud nägemis- ja liikumispuudega inimeste interneti kasutamise eesmärke ja harjumusi ning toonud välja mitmeid probleeme ja soovitusi. Näiteks on vaegnägijatele loetamatu rasvases kirjas allajoonitud tekst ja mõne värvi abil esile tõstetud tekst. Oluline on, et veebilehel saaks teksti aktiveerida ning kopeerida, sest teksti sisestamine klaviatuuril on ebamugav nii mobiilis kui ka füüsilisel klaviatuuril. Kui lehele on sisseehitatud teksti suurendamise ja kontrastsuse muutmise võimalused, võiksid need asuda päise keskosas, et see lehe suurendamisel vaateväljast ei kaoks.

Kurniawan ja Zaphiris [27] on loonud 38 suunist veebilehtede loomiseks vanematele inimestele, mis on olulised ka käesolevas töös loodavate suuniste juures, sest vanemad inimesed on vaieldamatult üks juurdepääsetavuse sihtrühmadest. Autorid lõid oma uurimistöö tulemusena 52 suunist, millest sorteerimise ja kategoriseerimise alusel töötati välja 38 suunist. Loodud suuniste vajalikkust hindasid viie palli süsteemis 16 vanemat inimest. Antud suunised kaasati käesoleva töö tulemi valmimisse.

Diaz-Bossini ja Moreno [28] on samuti uurinud mobiilsete kasutajaliideste loomist lähtudes vanemate inimeste vajadustest. Autorid on välja toonud, et kuna mobiilseadmeid kasutatakse tihti vabas õhus, kus valgustingimused võivad piirata ekraanil oleva info nähtavust, ei tohi infot esitada tuginedes vaid värvidele. Lisaks võidakse samal ajal osaleda liikluses, kus müra ja erinevate ohutegurite tõttu võib väikesele ekraanile keskendumine lõppeda õnnetusega. Seega peab info olema esitatud lühidalt ja konkreetselt, et selle saaks kätte pikalt keskendumata. Uurimistöös tuleb välja, et mobiilseadmetel kasutatav piiratud internetimaht võib olla probleemiks, kui lehel on palju pilte ja muud laadimisega vajavat sisu. Autorid ei soovita kasutada animatsioone, kus info teatud aja tagant tekib ja kaob, sest vähenenud võimetega inimestel võib minna

lugemiseks kauem aega. Diaz-Bossini ja Moreno soovitusel võeti arvesse käesoleva töö tulemi loomisel.

*Section 508* [11] nõuded kattuvad enamuses WCAG 2.0 suunistega, kuid vajalikud nõuded, mis ei kattu, lisati käesoleva bakalaureusetöö tulemissse. Näiteks toob *Section 508* välja, et tabelites peaks eraldi ära märkima vertikaalsed ja horisontaalsed päiselahtid.

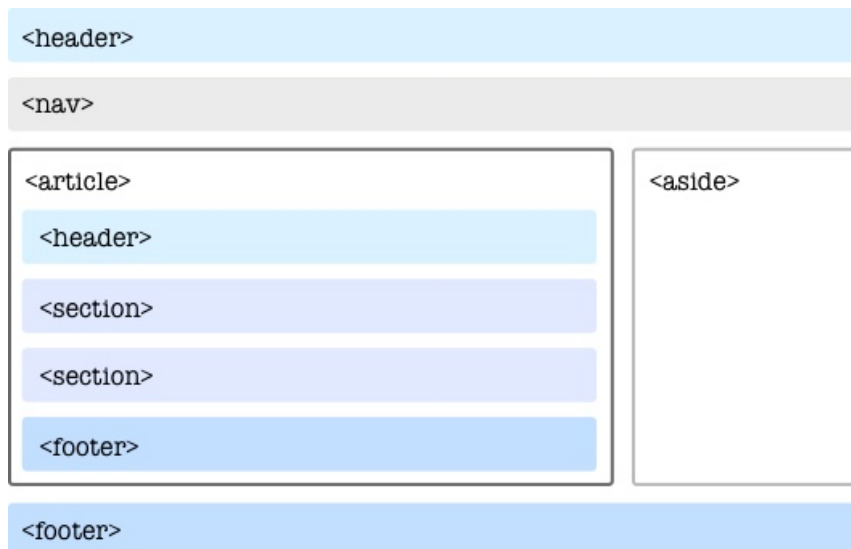
Käesoleva töö tulemi loomisel on kasutatud ka soovitusi WAI loodud juurdepääsetavuse käsiraamatust [29], mis annab nõuandeid ja näiteid lehe struktuuri, menüüde, piltide, tabelite, vormide ja karussell-galeriide juurdepääsetavuse parendamiseks. Icoonide juurdepääsetavuse kirjeldamisel on kasutatud laialdaselt kasutatava Font Awesome ikoonide juurdepääsetavuse juhiseid [30]. Autor kasutas ka näiteid ja soovitusi isikutelt, kes on süvitsi uurinud mõnda juurdepääsetavuse aspekti. Näiteks Wild [31] on kirjutanud palju linkide juurdepääsetavusest. Panchang [32] on uurinud, kuidas anda linkidele pealkirju, mida loeksid välja kõik ekraanilugejad. Snook [33] on muuhulgas kirjutanud sellest, kuidas peita sisu visuaalselt nii, et ekraanilugejad seda siiski ette loeksid. Häid ja halbu näiteid kogus autor näiteks Gustafsoni värvipimedatega arvestava disaini artiklist [34] ja Google *Material Design* veebilehelt [35].

### 3. Loodud suuniste komplekt

Käesolevas bakalaureusetöös toetuti suuniste loomisel arendajatelt ja erivajadustega inimestelt saadud infole ning peatükis 2.3 kirjeldatud allikatele. Olemasolevatest ametlikest ja mitteametlikest suuniste komplektidest, teadustöödest, veebilehtedelt ja intervjuudest saadud mõtted grupeeriti kaardisorteerimismeetodil. Grupeeritud suunised sõnastati ümber võimalikult täpselt ja konkreetselt ning varustati koodinäidetega. Suunised jaotati viieteistkümnesse kategooriasse: üldised põhimõtted, lehe andmed, navigatsioon, pealkirjad, sisu paigutus, tekst, lingid, värvid, pildid, ikoonid, heli ja video, vormid, veateated, nupud, tabelid. Mahukamad kategooriad jagati väiksemateks osadeks. Lisa 3 esitab valiku loodud suuniseid illustreerivatest headest ja halbade näidetest.

#### 3.1. Üldised põhimõtted

1. Lehe aadress tuleb hoida võimalikult lühikesena, sest mobiilseadmetel veebilehitsejasse aadressi trükkimine on ebamugav. Lühike aadress, näiteks “firma.ee” peab suunama lehele “http://www.firma.ee/index.html” ja “firma.ee/leht” lehele “http://www.firma.ee/leht.html”. Lehe aadress peab avama alati sama temaatikaga lehe, olenemata seadmest.
2. Veebileht peab olema kasutatav puuetundlikes mobiilseadmetes. Soovitav on disainida leht kohanduvalt (*responsive*) ja esmalt mobiilile (*mobile first*). Lehte peab olema võimalik mobiilis suurendada – ei tohiks kasutada `<meta>` märgendi `user-scalable="no"` atribuuti.
3. Soovitav on kasutada HTML5 struktuurielemente nagu joonisel 1: `<article>` suure iseseisva sisuosa jaoks, `<section>` sektsioonide eristamiseks, `<header>` päise esitamiseks, `<main>` põhisisu märkimiseks, `<footer>` jaluse märkimiseks, `<nav>` menüü jaoks, `<aside>` külgriba jaoks, `<address>` kontaktinfo (soovitavalt e-posti aadressi) jaoks. Elementidele tuleks lisada pealkiri `aria-label="..."` atribuudiga või kirjutada vastava pealkirja id `aria-labelledby="..."` atribuuti.
4. Vältida tuleks raame nagu `<frameset>` või `<iframe>`, sest paljud mobiilseadmed ei toeta neid ning nad on üldiselt vananenud ja problemaatilised. Kui siiski kasutatakse raame, tuleb neile anda kirjeldavad `title="..."` atribuudid.



Joonis 1. HTML struktuurielementide kasutamise näide.

- Kujundus tuleb luua kasutades laadilehti (*stylesheets*), mitte *style="..."* atribuute. Sisu peab olema kättesaadav ja loogilises järjekorras ka laadilehti kasutamata.
- Peidetud elemendile tuleb anda CSS-is nii *display: none* kui ka *visibility: hidden* stiilid nagu joonisel 2, et ükski ekraanilugeja ei näeks peidetud elementi.

```
.hidden {
    display:none;
    visibility:hidden;
}
```

Joonis 2. CSS-is objekti peitmine.

- Elementide suurused tuleb määrata relatiivsetes ühikutes nagu *rem* ja *em*. Eranditeks on veerised, raamid ja vooderdus (*margin*, *border* ja *padding*) ning pildi suurused, mis võivad olla määratud pikslites.
- Lehel peab olema võimalik teha kõiki tegevusi (välja arvatud nt joonistamine) kasutades ainult hiirt ja ka ainult klaviatuuri (*tab* klahvi või nooleklahve). Lehel ei tohi leiduda klaviatuurilõkse ehk olukordi, kus klaviatuuriga mingi funktsiooni juurde minna on võimalik, kuid sealt ära enam mitte.
- Kasutajalt ei tohi nõuda hõljutamist (*hover*), lohistamist (*drag*) ega libistamist (*slide*), vaid peab lubama ka tavalist hiireklõpsu. Lohistamise asemel peab lubama kopeerimist ja kleepimist.
- Päästiksündmustena tuleb vältida hiire allavajutust (*onmousedown*), lahtilaskmist (*onmouseup*), topeltklõpsu (*ondblclick*), kursori peale viimist (*onmouseover*) ja

kursori eemaldamist (*onmouseout*), sest need pole juurdepääsetavad klaviatuuri ja tugitehnoloogiatega.

11. Ettevaatlik tuleb olla robotilõksude kasutamises, sest CAPTCHA pildid on ekraanilugejaga loetamatud ning nupud ja libistamisribad ei luba tihti turvakaalutlustel hiireklõpsu simuleerimist olles seega samuti ekraanilugejaga kasutatavad. Parem variant on robotilõksuna kasutada lihtsaid, kuid vastamisel inimlikku loogikat vajavaid küsimusi, näiteks “Kas päike on kollane või sinine?”.
12. Kerimist peaks lubama vaid ühes suunas (ülevalt-alla või vasakult-paremale). Erandiks on näiteks galeriid.
13. Leht ei tohi ennast automaatselt värskendada, sest see võib kustutada kasutaja pooleli jäänud töö ning kasutada piiratud mobiilse interneti mahtu.
14. Kui lehel on ajalimiit (lühem kui 20 tundi), peab kasutajat sellest hoiatama, lubama limiidi välja lülitamist või pikendamist (selleks peab andma vähemalt 20 sekundit aega).
15. Pole soovitatav kasutada animatsioone, kus info tekib ja kaob, sest kasutaja ei pruugi jõuda piiratud ajaga infot kätte saada. Kui veebilehe sisu laetakse osade kaupa, võib olla ekraanilugejatel keeruline lugemisjärge hoida.
16. Vältida tuleks hüplikaknaid ja modaale. Kui neid siiski kasutatakse, peab olema selgelt arusaadav, kuidas neid sulgeda.
17. Kui võimalik, vältida kaasapandud objekte (nt pistikprogramme) ja skripte, sest need ei pruugi olla toetatud ning võivad pikendada lehe laadimisaega. Kui leht ei tööta ilma *JavaScript*ita, tuleb kasutajat sellest teavitada.
18. Miski veebilehel ei tohi miski sähvida rohkem kui 3 korda sekundis, sest see võib tekitada haigushooge. Sähvimist peab saama peatada ja välja lülitada.
19. Elementi fokuseerides ei tohi juhtuda midagi ettearvamatut – fookus ei tohi minna mõnele teisele elemendile, vorm ei tohi ennast ise ära saata jne.
20. Märgistuskeeles kirjutatud kood peab valideerima formaalseid validaatoreid (näiteks W3 validaator) kasutades.

### 3.2. Lehe andmed

1. Lehe keel peab olema märgitud ISO 639-1 koodiga `<html>` elemendi *lang*=“...” atribuudis, näiteks `<html lang="et">`.



2. Igal lehel peal olema informatiivne ja lehte kirjeldav tiitel `<title>` märgendina nagu joonisel 3.
3. Kui lehel on teateid või vigu, võiks need kirjutada tiitli algusesse. Nt: “1 uus sõnum | Foorum”.
4. Mõistlik on alustada tiitlit alamlehe nimega. Nt: “Kontakt | Firmanimi OÜ” mitte “Firmanimi OÜ | Kontakt”, sest kui vahelehed on kitsad, pole viimasel juhul võimalik alamlehti üksteisest eristada.

```
<head>
  <title>
    Kontakt - Firmanimi OÜ
  </title>
</head>
```

Joonis 3. HTML lehe tiitel.

### 3.3. Navigatsioon

1. Navigatsioon peab olema lihtne ja ühtne üle terve veebilehe.
2. Tihedamini külastatavad lehed tuleb teha kiiresti kättesaadavaks, kuid harvem kasutatavate lehtedeni jõudmiseks võib minna rohkem aega.
3. Navigatsioon peab lehele minnes kohe näha olema. Soovitatav on kasutada menüüd lehe ülaservas või vasakul. Mobiilsetel lehtedel võib kasutada lisaks ka kuni kuute kerimisel kaasaliikuvat kiirvalikut lehe allservas.
4. Menüü, otsing, keelevalik ja sulgemisnupp peavad olema veebilehel nähtaval ja oodatavas kohtas. Mobiilis peaks olema lisaks menüü ikoonile ka kiri “Menüü”.
5. Navigeerimiseks peab olema vähemalt kaks võimalust: lisaks menüüle näiteks otsing, sisukaart (mis töötab ilma JavaScriptita) või jäljerada (*breadcrumbs*), mis on esitatud joonisel 4.
6. Otsingukasti juures peab olema selgelt arusaadav, kuidas otsingut käivitada - kas vajutada mingit nuppu või *enter* kahvi.
7. Otsing peab leidma vasteid ka lähedastele otsingusõnadele. Näiteks vaste “nina-kõrva-kurguarst” peab leidma ka kasutades otsingusõna “kurgu-nina-kõrvaarst”.
8. Juurdepääsetavaim viis navigatsioon esitamiseks on `<nav role=“navigation” aria-label=“...”>` märgendiga, mille sees on `<ul>` märgendiga esitatud nimekiri. Joonisel 4 näidatud alammenüüga elemendil on `aria-haspopup=“true”` atribuut.

```

<nav aria-label="Asud siin:">
  <a href="#">Avaleht</a>
  <a href="#">Raamatud</a>
  <span aria-label="Avatud leht:">Looduse lood</span>
</nav>

```

Joonis 4. HTML järjerada.

```

<nav role="navigation" aria-label="Menüü">
  <ul>
    <li><a href="#">Avaleht</a></li>
    <li>
      <a href="#" aria-haspopup="true">Portfoolio</a>
      <ul>
        <li><a href="#">Maal</a></li>
        <li><a href="#">Foto</a></li>
      </ul>
    </li>
    <li><a href="#">Kontakt</a></li>
  </ul>
</nav>

```

Joonis 5. Kahetasemeline HTML menüü.

9. Menüüpunktid peavad olema sõnastatud nii, et oleks selge, mis nende all leidub.
10. Vältida tuleks mitmetasemelisi menüüsid – jääda ühe või kahe taseme juurde.
11. Menüü peab avanema sellele vajutades. Ei tohi nõuda kasutajalt hõljutamist (*hover*) või libistamist (*slide*). Menüü ei tohi iseenesest mingi aja järel sulguda.
12. Soovitav on lisada mugavaid ülehüppamislinke, näiteks “Jäta ... vahele”. Lehe kõige esimene element peab olema “Liigu põhisisu juurde” nagu joonisel 6, mis teeb tabuleerivate kasutajate jaoks infoni jõudmise mugavamaks.

```

<a href="#sisu">Liigu põhisisu juurde</a>
...
<main id="sisu">

```

Joonis 6. HTML põhisisuni liikumise link.

### 3.4. Pealkirjad

1. Pealkirjad peavad olema visuaalselt hästi eristatavad - suured ja selged.
2. Pealkirjad tuleb eristada märgenditega `<h1>` kuni `<h6>`, mitte ainult kujundusega. Pealkirjadena `<h1>` kuni `<h6>` ei tohi esitada teksti, mis pole tegelikult pealkiri.

3. Pealkiri peab olema lühike, kuid sisukas ning kirjeldama sellele pealkirjale järgnevat sisu. Pealkiri peab algama kõige tähtsama ja sisukama sõnaga.
4. Igal lehel peaks olema vähemalt üks `<h1>`.
5. Pealkirjad peavad järgnema üksteisele hierarhiliselt korrektses järjekorras nagu joonisel 7. Järjekorra korrektsust saab kontrollida näiteks WAVE tööriistaga.

```
<h1>Esimese taseme pealkiri</h1>
<h2>Teise taseme pealkiri</h2>
<h3>Kolmanda taseme pealkiri</h3>
<h3>Kolmanda taseme pealkiri</h3>
<h2>Teise taseme pealkiri</h2>
<h2>Teise taseme pealkiri</h2>
```

Joonis 7. HTML pealkirjade järjestuse näide.

### 3.5. Sisu paigutus

1. Kasutaja peab lehelt leidma selle ja ainult selle sisu, mida ta sealt ootas.
2. Prioriteetne info peaks asuma lehe keskel ning olema visuaalselt esile tõstetud.
3. Mobiilseadme kasutajad peaksid eeskätt pääsema ligi kõige vajalikumale informatsioonile, kuid kogu lehe sisu peaks olema siiski kättesaadav. Väikesel ekraanil peab olema olemas sama sisu ja funktsionaalsus, mis suurel ekraanil.
4. Lehe sisu tuleb jagada mõistlike suurustega alamlehtedeks, et näidata kasutajale vaid seda, mille järgi ta on tulnud. Liiga pikad lehed võtavad kaua aega laadimiseks, liiga lühikesed lehed nõuavad kasutajalt edasi-tagasi navigeerimist.
5. Sisu kordamist ühe lehe ulatuses tuleb vältida, sest see võib olla ekraanilugejaga kasutajale olla segadusseajav ja tüütu.
6. Tabuleerimisjärjekord peab ühtima sisu loogilise järjekorraga: kasutaja võib olla sunnitud navigeerima klaviatuuriga ning vales järjekorras sisu kaotab oma mõtte.

### 3.6. Tekst

1. Kasutada tuleb lihtsat ja selget keelt, sest veebi kasutavad erineva kultuurilise ja haridusliku taustaga inimesed.
2. Laused peaksid olema lühikesed ja konkreetsed, sest näiteks mobiiliekraanile ei mahu palju teksti ning mobiilseadmele keskendumine võib mõnes olukorras kasutaja ohtu seada.

3. Märkide (näiteks “-“) asemel on soovituslik kasutada sõnu (näiteks “kuni”), sest tugitehnoloogiad võivad neid valesti lugeda (näiteks “miinus”).
4. Lühendite kasutamisel tuleb nende seletus esimesel korral välja kirjutada.
5. Ühes reas tohib olla kuni 90 tähemärki, mobiilis kuni 70 tähemärki.
6. Tekst peab olema joondatud vasakule.
7. Soovituslik on kasutada seriifideta fonti, näiteks Helvetica või Arial.
8. Kaldkirjas ja paksu allajoonitud kirja tuleb vältida, sest neid on raske lugeda.
9. Soovituslik on vältida sõnades läbivaid suurtähti, sest selliseid sõnu on raskem lugeda. Kui aga läbivaid suurtähti siiski kasutatakse, tuleb seda teha CSS-is *text-transform: uppercase* reegli abil, sest *caps-lock*’iga kirjutatud sõnu loevad ekraanilugejad täht haaval mitte tervikuna.
10. Sisutekst peab olema minimaalselt 14px suur, soovitatav suurus on 16px.
11. Kui lehele on sisseehitatud teksti suurendamise võimalus, peaks see asuma päise keskosas, et see lehe suurendamisel ekraanilt ära ei kaoks.
12. Teksti suurused tuleb määrata *rem* ühikutes nagu joonisel 8, et veebilehitseja seadetes teksti suurust muutes muutuks vastavalt ka veebilehe teksti suurused. Teksti (v.a subtiitreid) peab saama suurendada 200% ilma, et sisu ekraanilt kaoks või kattuks.
13. Reavahe tuleb määrata ilma ühikuta nagu joonisel 8. Soovituslik universaalne reavahe on 1.5.
14. Lõikude vahe tuleks määrata *rem* ühikutes nagu joonisel 8. Soovituslik universaalne lõikude vahe on võrdne teksti suurusega.

```
p {  
  font-size: 1rem;  
  line-height: 1.5;  
  margin-top: 1rem;  
}
```

Joonis 8. Lõiguvahe, reavahe ja teksti suurus CSS-is.

15. Tekst ei tohiks lehel liikuda (olla animeeritud), sest seda võib olla raske lugeda.
16. Teksti peab olema võimalik korrektses järjekorras aktiveerida ning kopeerida.
17. Teksti asemel ei tohi kasutada pilti, millel on tekst, sest seda ei saa kopeerida ega lugeda ekraanilugejaga. Erandiks on logod, millele tuleb lihtsalt lisada *alt= "... "*.

18. Kui osa tekstist on märgistatud täрни või mõne muu sümboliga, peab sümboli seletus asuma enne teksti. Seletuses peab sisalduma “[sümbol] tähistab ...”.
19. Kui osa tekstist on lehe üldisest keelest erinevas keeles, peab see olema märgitud näiteks ISO 639-1 koodiga *lang*=“...” atribuudis nagu joonisel 9.

```
<span lang="en">Hello</span>
```

Joonis 9. Võõrkeelse fraasi keel on HTML-is ära märgitud.

### 3.7. Lingid

1. Lingid peavad olema muust tekstist visuaalselt eristatavad, näiteks alla joonitud.
2. Küllastatud ja küllastamata lingid peaksid olema visuaalselt eristatavad.
3. Lingitekstist peab saama selgelt aru, mis sellele vajutades juhtub. Lingitekst peaks algama kõige tähtsama ja sisukama sõnaga. Lingitekst ei tohiks olla lihtsalt “Kliki siia” või “Rohkem” ega lihtsalt URL, sest ekraanilugeja loeb selle ette täht haaval. Lingitekst ei tohiks sisaldada sõna “link”, sest ekraanilugejad ütlevad, et tegu on lingiga ning nägijate jaoks peavad lingid olema tekstist visuaalselt eristatavad.
4. Sama lingitekstiga lingid ei tohi viia erinevatele lehtedele.
5. Ebavajalikke ja kordavaid linke tuleb vältida - näiteks pilt ja “Loe rohkem” tekst, mis viivad samale lehele kui link ise, tuleb panna kõik ühte *<a>* märgendisse.
6. Lehel ei tohi olla katkiseid linke, mis ei vii kasutajat õigesse kohta.
7. Kui linke on väga palju, võiks need grupeerida nimekirja *<ul>* või *<ol>* märgendiga. Lingid ei tohi paikneda üksteisele liiga lähedal ega ka liiga kaugel.

#### 3.7.1. Link, mis laeb alla faili

1. Teavitus, et algab faili allalaadimine, peab olema lingi tekstis näha. Ära tuleks märkida ka faili suurus ning formaat nagu joonisel 10, et aeglase või piiratud mahuga interneti kasutajad saaksid vältida aja- ja rahakulu.

```
<a href="doc.pdf">Lae alla dokumendi fail (1MB PDF)</a>
```

Joonis 10. Faili allalaadimist alustav HTML link koos selgitusega.

### 3.7.2. Link, mis avaneb uues aknas

1. Lingid ei tohiks ette hoiatamata avaneda uues aknas või uuel vahelehel. Kasutaja ei pruugi aru saada, et on sattunud uuele lehele ning võib proovida tulutult veebilehitseja “Tagasi” nuppu kasutada. Mobiilseadmes pole tavaliselt näha, mitu akent on avatud ning nende vahel navigeerimine on ebamugav.
2. Joonisel 11 on lisatud lingile informatiivne ikoon, peidetud see ekraanilugejate eest atribuudiga *aria-hidden="true"* ning lisatud sama infot edasi andev tekstiline kirjeldus, mida visuaalselt pole näha, aga mida ekraanilugejad saavad lugeda. Ikoon tuleb alati paigutada peale linki, et lugemisjärjekord oleks loogiline.

```
<a href="..." target="_blank">
  Company
  <span class="invisible">Link avaneb uuel lehel</span>
  <i aria-hidden="true" class="icon-external"></i>
</a>

.invisible{
  position: absolute !important;
  height: 1px;
  width: 1px;
  overflow: hidden;
  clip: rect(1px 1px 1px 1px); //Internet Explorer 6 ja 7
  clip: rect(1px, 1px, 1px, 1px);
}
```

Joonis 11. Uuel lehel avanev HTML link koos ikooni ja ekraanilugejatele nähtava selgitusega.

3. Joonisel 12 on lisainformatsioon lingi atribuudis *aria-label="..."* ning lingi tekst dubleeritud atribuudis *title="..."*, et kõik ekraanilugejad loeksid välja kogu vajaliku informatsiooni.

```
<a href="..." target="_blank" title="Company" aria-label="Link
avaneb uuel lehel">
  Company
  <i aria-hidden="true" class="icon-external"></i>
</a>
```

Joonis 12. Uuel lehel avanev HTML link koos ikooniga.

### 3.7.3. Link, mis viib sama lehe teise kohta

1. Soovitav on kasutada ülehüppamislingi tekstis nt “Liigu sisuni: ...” nagu joonisel 13 või lisada lisainformatsioon atribuudis *aria-label*=“...” ning dubleerida lingi tekst atribuudis *title*=“...”.

```
<a href="#pt2">Liigu sisuni: Peatükk 2</a>
```

Joonis 13. HTML ülehüppamislink.

### 3.8. Värvid

1. Elemendid ja nende olekud ei tohi olla eristatud ainult värviga. Info, mida antakse edasi värvidega, peab olema kättesaadav ka ilma värvideta.
2. Aktiivne element (näiteks menüüpunkt, mis viitab lehele, kus hetkel asutakse) peab olema selgelt eristatav ka värve nägemata.
3. Fookuses olev element peab olema selgelt visuaalselt eristatav, et klaviatuuriga navigeerijatel oleks selge, mille juures nad lehel asuvad.
4. Esiplaani (teksti või kujutise) ja tagaplaani (tausta) värvid peavad olema piisavalt erinevad, et nende kontrastsus oleks minimaalselt: kui font on kuni 18pt ja *regular* - 4,5:1 ning kui font on vähemalt 18pt ja *regular* või vähemalt 14pt ja *bold* - 3:1.
5. Kui lehele on sisseehitatud kontrastsuse muutmise võimalus, peaks see asuma päise keskosas, et see lehe suurendamisel ekraanilt ära ei kaoks.

### 3.9. Pildid

1. Igal pildil peab olema märgitud tekstiline alternatiiv *alt*=“...” atribuudiga nagu joonisel 14, mis peab kandma edasi sama informatsiooni kui pilt ise. Tekstiline alternatiiv ei tohi olla lihtsalt “Pilt” või “Tabel”. Tekstilisse alternatiivi pole vaja kirjutada “Pildi peal on...”, sest ekraanilugeja ütleb, et tegu on pildiga.
2. Pildile võiks lisada allkirja *figcaption* märgendiga nagu näidatud joonisel 14. Pildi tekstiline alternatiiv ei tohi korrata pildi allkirja, vaid andma lisainfot.

```
<figure>  
    
  <figcaption>TTÜ - Sinu elustiil</figcaption>  
</figure>
```

Joonis 14. HTML pilt tekstilise alternatiivi ning allkirjaga.

3. Vältida tuleb hüperpilte (*image map*), mille eri osadele vajutades avanevad erinevad lingid.
4. Kui kasutatakse animeeritud galeriid (nt karusell), peab kasutajal olema võimalik animatsiooni peatada. Galerii peab olema kasutatav klaviatuuri abil. Aktiivne galerii element peab olema eristatav ka tugitehnoloogiatega – sellele võib lisada visuaalselt peidetud teksti `<div aria-live="polite">Esimene pilt kuuest</div>`.

### 3.9.1. Pilt, mis ei kannu edasi vajalikku infot

1. Kui tegu on illustreeriva pildiga, mida informatsiooni kätte saamiseks näha pole vaja, tuleb lisada pildile tühi `alt=""` atribuut, sest siis jätavad ekraanilugejad selle pildi vahele. Kui `alt` atribuuti ei lisata, loeb ekraanilugeja ette pildi faili nime.
2. Illustreerivate piltide puhul tuleks kasutada CSS-i `background`, `:before` või `:after` reegleid HTML-i `<img>` märgendite asemel.

### 3.9.2. Taustapilt

1. Taustapiltide kasutamisel tuleb kindlaks teha, et tekst on pildi pealt loetav. Sisu peab loetavaks jääma ka taustapildi eemaldamisel.
2. Taustapildi fail peab olema võimalikult väikesemahuline, et lehe laadimisaeg oleks võimalikult lühike.

## 3.10. Ikoonid

1. Ikoonid peavad olema lihtsad, selged, äratuntavad ja üheselt mõistetavad.
2. Kui ikoon on pelgalt illustreeriv ega kannu endas lisainformatsiooni, tuleks see ekraanilugejate eest peita atribuudiga `aria-hidden="true"` nagu joonisel 15. Antud atribuut ei mõjuta elemendi visuaalset nähtavust, vaid võimaldab ekraanilugejatel antud elemendi vahele jätta.

```
<a href="...">
  <i class="icon-help" aria-hidden="true"></i>Help
</a>
```

Joonis 15. Illustreeriv HTML ikoon on ekraanilugejate eest peidetud.

3. Kui ikoon annab edasi tähtsat informatsiooni, mida tekstiliselt kirja ei taheta panna, tuleks see ekraanilugejate eest peita atribuudiga `aria-hidden="true"` ning



lisada sama infot edasi andev tekstiline kirjeldus, mida visuaalselt pole näha, aga mida ekraanilugejad saavad lugeda. See tehnika on näidatud joonisel 16.

4. Icoonile võib lisada ka atribuudi *title*=“...” nagu joonisel 16, mis annab nägijatele lisainformatsiooni, kui kursoriga elemendi peale liikuda.
5. Kui ikooniga on tähistatud linki, tuleks lingile lisada atribuut *aria-label*=“...” nagu joonisel 17. Antud atribuut võimaldab ekraanilugejatel kasutajale lisainformatsiooni lugeda.
6. Icoonile võib lisada ka atribuudi *title*=“...” nagu joonisel 17, mis annab nägijatele lisainformatsiooni, kui kursoriga elemendi peale liikuda.

```
<i class="icon-phone" aria-hidden="true" title="Phone number"></i>  
<span class="invisible">Phone number</span>
```

```
.invisible{  
  position: absolute !important;  
  height: 1px;  
  width: 1px;  
  overflow: hidden;  
  clip: rect(1px, 1px, 1px, 1px);  
}
```

Joonis 16. HTML ikoon koos CSS-is peidetud kirjeldusega.

```
<a class="button" href="..." aria-label="Vaata ostukorvi">  
  <i class="icon-bag" aria-hidden="true" title="Ostukorv"></i>  
</a>
```

Joonis 17. HTML link, mille sisuks on ikoon.

### 3.11. Heli ja video

1. Heli, videot ja animatsiooni (mis kestab kauem kui 3 sekundit) peab saama kinni panna ning heli tugevust muuta.
2. Veebilehel olevale helile ja videole peab lisama kirjeldava *title*=“...” atribuudi.
3. Eellindistatud videofailile tuleb lisada sünkroonsed tiitrid ja helikirjeldused (näiteks [rahulik muusika]), kui kogu edastatav info pole lehelt teisiti kättesaadav.

4. Reaalajas edastatavale videofailile tuleb lisada sünkroonsed tiitrid ja helikirjeldused (näiteks [rahulik muusika]), kui kogu edastatav info pole lehelt teisiti kättesaadav.
5. Eellindistatud heli- ja videofailidele tuleb lisada tekstiline sisukirjeldus, mis asub või millele viidatakse faili vahetus läheduses. Joonisel 18 on videole lisatud tekstiline kirjeldus `<track>` märgendiga `<video>` märgendi sees. Ilma helita videole võib tekstilise alternatiivi asemel lisada sama sisu edasi andva helifaili.
6. Eellindistatud videofailile tuleb lisada heliline sisukirjeldus, sest selle kuulamine on nägemispuudega kasutajatele mugavam.

```
<video controls>  
  <source src="video.mp4" srclang="et" type="video/mp4">  
  <track src="description.vtt" kind="descriptions" srclang="et"  
    label="Video kirjeldus">  
</video>
```

Joonis 18. HTML video koos tekstilise kirjeldusega.

### 3.12. Vormid

1. Vajalike vajutuste hulk tuleb hoida minimaalsena. Sisestuskastide asemel võiks kasutada valikukaste, raadionuppe või märkeruute, sest need on mugavamad, eriti mobiilseadmel.
2. Igal vormiväljal peab olema täpselt üks pealkirja `<label>`, mis on temaga seotud `id="..."` ja `for="..."` atribuutide kaudu nagu näidatud joonisel 19. Pealkiri peab visuaalselt paiknema vormivälja suhtes nii, et nende kokkukuuluvus oleks selge.

```
<label for="nimi">Nimi:</label>  
<input type="text" name="nimi" id="nimi">
```

Joonis 19. HTML sisestuskast koos pealkirjaga.

3. Kokkukuuluvad vormiväljad nagu raadionupud, märkeruudud ja sisestuskastid tuleb grupeerida `<fieldset>` elemendi sisse ning anda grupile üldine pealkiri `<legend>` elemendi abil nagu näidatud joonisel 20.

```

<fieldset>
  <legend>Millist kontaktviisi eelistate?</legend>
  <input type="radio" name="contact" id="email" value="email">
  <label for="email">E-post</label>
  <input type="radio" name="contact" id="phone" value="phone">
  <label for="phone">Telefon</label>
</fieldset>

```

Joonis 20. Grupeeritud ja ühise pealkirjaga HTML raadionupud.

4. Kasutajalt ei tohiks nõuda sisestatud andmete meelde jätmist, vaid tuleks näidata tema eelmisi valikuid ja varem sisestatud andmeid.
5. Kui vorm on jaotatud sammudeks, mis asuvad mitmel lehel, tuleks kirjutada lehe *<title>* märgendisse, mitmenda sammu juures ollakse.
6. Vormielementi fokuseerides või klikkides ei tohi juhtuda midagi ettearvamatut ega muutuda lehe kontekst.
7. Kui vormi ära saatmisel on õiguslikud või finantsilised tagajärjed, peab kasutajal olema võimalik sisestatud info üle vaadata ning vajadusel saatmine tagasi võtta.

### 3.12.1. Sisestuskastid

1. Soovitav on kasutada võimalusel sisestuskasti tüüpe *search*, *email*, *url*, *number*, *tel*, *range*, *date* või *time* nagu joonisel 21, et valideerida sisestatud infot automaatselt ning vältida vigu. Mobiilseadmes kuvatakse sisestuskasti tüübi olemasolul ka vastav klaviatuur, mis teeb sisestamise mugavamaks. Sisestuskasti tüübid pole töö kirjutamise hetkel veel kõigis veebilehitsejates toetatud.

```

<label for="vahemik">Määra maksimaalne hind: </label>
<input type="range" name="vahemik" id="vahemik" min="0"
max="100" step="10">

```

Joonis 21. HTML sisestuskast vahemiku, ekstremaalsete väärtuste ning sammuga.

2. Kui info tuleb sisestada mingil kindlal kujul, tuleb seda kuju kirjeldada sisestuskastiga seotud *<label>* märgendis nagu joonisel 22. Soovitud kuju võib kirjutada ka sisestuskasti *placeholder="..."* atribuuti, kuid see ei tohi asendada *<label>* märgendit. *Placeholder="..."* atribuuti ei loe välja paljud ekraanilugejad ning enamus veebilehitsejaid kuvavad seda madala kontrastsusega värvides, mida vaegnägijad ei pruugi näha.

3. Sisendile võib seada piiranguid *min*="...", *max*="...", *maxlength*="..." ja *steps*="..." atribuutidega. Veelgi täpsema piirangu saab seada mustrikontrolli ehk *pattern*="..." atribuudiga nagu joonisel 22, mis võimaldab kontrollida kasutaja sisendi vastavust regulaaravaldisele ning vältida vigu.

```
<label for="aeg">Sisesta aeg (tt:mm)</label>  
<input type="text" id="aeg" pattern="[0-9]{2}(:)[0-9]{2}">
```

Joonis 22. HTML sisestuskast koos mustrikontrolliga.

4. Parim võimalus on paigutada *<label>* elemendid sisestuskastide kohale, sest selline paigutus töötab hästi nii arvutis kui ka mobiilis.
5. Kui sisestuskasti funktsioon on visuaalselt piisavalt arusaadav, et ei vaja *<label>* elementi nagu joonisel 23, tuleb kasutada *aria-label*="..." atribuuti nagu joonisel 24, et selgitust saaksid lugeda ainult ekraanilugejad.



Joonis 23. Otsingunupul olev kiri selgitab sisestuskasti funktsiooni.

```
<input type="text" name="otsi" aria-label="Sisest märksõnad,  
mida soovid otsida">  
<button type="submit">Otsi</button>
```

Joonis 24. HTML juurdepäätav otsingukast.

### 3.12.2. Raadionupud

1. Kokkukuuluvatele raadionuppudele tuleb anda ühine *name*="..." atribuut nagu näidatud joonisel 25, et korraga saaks valida ainult ühe raadionupu.

```
<input type="radio" name="contact" id="email" value="email">  
<label for="email">E-post</label>  
<input type="radio" name="contact" id="phone" value="phone">  
<label for="phone">Telefon</label>
```

Joonis 25. HTML raadionuppudel on ühine nimi.

### 3.12.3. Valikukastid

1. Kasutajale ei tohiks näidata korraga liiga palju valikuid ja võimalusi, sest see võib olla segadusseajav ja frustreriv. Valikud tuleks jagada väiksemateks gruppideks. Pikast valikute nimekirjast võiks pakkuda otsimisvõimalust.
2. Kokkukuuluvad valikud võib grupeerida `<optgroup>` märgendi abil nagu joonisel 26, et soovitud valiku leidmine oleks kiirem.

```
<select>
  <optgroup label="Gluteenivabad toidud">
    <option value="kinoa">Kinoa juurviljadega</option>
    <option value="salat">Kevadine salat</option>
  </optgroup>
  <optgroup label="Vegan toidud">
    <option value="burger">Kikerherne burger</option>
    <option value="supp">Porgandisupp</option>
  </optgroup>
</select>
```

Joonis 26. HTML valikukasti kokkukuuluvad valikud on grupeeritud.

### 3.12.4. Kohustuslikud väljad

1. Kohustuslikud väljad peavad olema esitatud `<input required>` märgendiga. Soovituslik on lisada ka atribuut `aria-required="true"`, sest kõik tugitehnoloogiad ei pruugi arvestada `required` atribuuti.
2. Kohustuslikud väljad peavad olema visuaalselt eristatavad mittekohustuslikest väljadest, kuid mitte ainult värvi abil.
3. Kui kohustuslikku välja tähistatakse tärniga, peab sellekohane selgitus asuma enne esimest vormivälja. Näiteks `*` tähistab kohustuslikku välja.

### 3.13. Veateated

1. Kasutajat tuleb teavitada, kui tekkis viga ning ka siis, kui kõik õnnestus.
2. Vigaselt täidetud väljad või täitmata jäetud kohustuslikud väljad peavad olema visuaalselt eristatavad. Vigaseid välju ei tohi eristada ainult värviga.
3. Kasulik on veateadet kuvada välja juures kohe, kui fookus sellelt ära läheb (kasutaja järgmisele väljale liigub). Veateatele võiks lisada `aria-live="assertive"`

atribuudi nagu joonisel 27, sest siis katkestab ekraanilugeja oma töö ning loeb koheselt ette veateate.

```
<input type="text" aria-describedby="message">  
<span id="message" aria-live="assertive">Sisesta parool!</span>
```

Joonis 27. Veateate kuvamine HTML-is.

4. Veateatest peab olema selgelt arusaadav, miks viga tekkis ja kuidas seda parandada. Peab olema kirjas, kas probleem on ajutine või püsiv ning kas kasutaja saab probleemi ise lahendada või tegeleb sellega teenusepakkuja (sellisel juhul peaks lisama teenusepakkuja või kasutajatoe kontakti).
5. Veateade tuleb kuvada võimalikult lähedal kohale, kus viga tekkis.
6. Kui veateade kuvatakse eraldi lehel, peab seal olema vähemalt üks järgmistest: “Tagasi” link eelmisele lehele (käitub nagu veebilehitseja “Tagasi” nupp), “Proovi uuesti” link (käitub nagu veebilehitseja “Värskenda” nupp) või link avalehele saamiseks.

### 3.14. Nupud

1. Nupud peavad kogu lehe ulatuses olema sarnased ja äratuntavad.
2. Nupud peavad olema vähemalt 40px suurused ja piisavalt suurte vahedega, et neid oleks puutetundlikul ekraanil mugav vajutada.
3. Mobiilivaates peaksid nupud asuma võimalikult ekraani keskel.
4. Nupud tuleb esitada `<button>` või `<input type="button">` märgendiga. Mitte kasutada `<a role="button">` märgendit nupu esitamiseks, sest ekraanilugejaga kasutaja võib nupu vajutamiseks kasutada *space* klahvi, mis linki aga ei aktiveeri. Nuppu ei tohi esitada `<div>` või `<span>` märgendiga, sest tugitehnoloogiad ei pruugi selliseid nuppe ära tunda.
5. Nupp peab olema vajutatav kogu oma ulatuses, mitte ainult nupul olev tekst.
6. Nupul peavad olema selgelt visuaalselt eristatavad fokuseeritud ja aktiivne olek näiteks nagu joonisel 28.
7. Kui nupu sees on teksti asemel ikoon, pilt või üksik täht nagu joonisel 29, tuleb nupule anda atribuut `aria-label="..."`, et nupu funktsioon oleks ekraanilugejat kasutavale inimesele mõistetav.

```

button:focus {
  border: 2px dashed black;
}
button:active {
  border: 2px solid black;
}

```

Joonis 28. Nupu fokuseeritud ja aktiivne olek on eristatud CSS-iga.

```
<button aria-label="Sulge aken">x</button>
```

Joonis 29. Juurdepääsetav HTML sulgemisnupp.

### 3.15. Tabelid

1. Tabeleid ei tohi kasutada lehe sisu paigutamiseks ja kujundamiseks.
2. Tabel tuleb esitada kasutades `<table>`, `<thead>` (päiselahtrite grupeerimiseks), `<tbody>` (sisuosa eraldamiseks), `<tfoot>` (kokkuvõtivate lahtrite eraldamiseks), `<tr>`, `<td>`, `<th>` märgendeid nagu joonisel 30.
3. Tabelil peab olema päis. Horisontaalse päise lahtrid tuleb märkida `<th scope="col">` ja vertikaalse päise lahtrid `<th scope="row">` nagu joonisel 30.
4. Kui tabelil on nii horisontaalne kui ka vertikaalne päis, peaks lahtritele andma atribuudi `headers="..."`, milles on selle lahtriga seotud päiselahtrite id-d.
5. Tabelile tuleks anda pealkiri `<caption>` märgendiga.
6. Keerulisele tabelile tuleks lisada pikem kirjeldus `aria-describedby="..."` atribuudiga nagu joonisel 30.

```

<p id="kirjeldus">Keerulise tabeli kirjeldus</p>
<table aria-describedby="kirjeldus">
  <caption>Tabeli pealkiri</caption>
  <thead>
    <tr>
      <th scope="col">Päis</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Lahter</td>
    </tr>
  </tbody>
</table>

```

Joonis 30. HTML tabel kirjelduse, pealkirja, päise ja kehandiga.

### 3.16. Loodud suuniste esitus veebilehena

Intervjuudest saadud sisendi põhjal loodi suuniste esitamiseks HTML veebileht, mis on nähtav aadressil <https://tinyurl.com/l1g49xo>. Suunised on lehel grupeeritud kategooriate järgi akordionitesse, mida saab avada ning sulgeda. See võimaldab keskenduda korraga väiksemale hulgal informatsioonile ning teeb selle hoomatavamaks. Lehel saab navigeerida ka kasutades sisukataloogi, mis püsib alati lehe vasakus servas ja annab ülevaate lehel olevast infost. Suuniseid saab filtreerida tüübi – WCAG A tase, AA tase, mobiilne suunis või soovituslik suunis – ning sihtgrupi järgi – arendajale, disainerile või sisuloojale. Vaikimisi on valitud kõik tüübid ja sihtgrupid. Eemaldades filtri eest märke, kuvatakse vähem suuniseid ning arendaja saab keskenduda vaid olulisele. Konkreetse suunise leidmiseks saab kasutada otsingut, mis leiab märksõnale vastava suunise ka suletud akordionist.

Akordionite sisu laetakse *JavaScript*-iga lehele eraldi failist, mis lihtsustab veebilehe haldaja jaoks suuniste muutmist ja lisamist. Iga suunise järel asuvad sellega seotud koodinäited, pildid reaalistest headest ja halbade näidetest ning mõnel puhul ka viide sobivale kontrolltööriistale.

Joonisel 31 on esitatud loodud veebilehe avakuva ja joonisel 32 avatud akordioni vaade.

The screenshot shows a web page titled "Juurdepäasetavussuunised" with a last updated date of 22.05.2017. On the left, there is a "FILTREERI:" sidebar with two sections: "TÜÜP:" and "SIHTGRUPP:". Under "TÜÜP:", there are checkboxes for "Soovitatav", "WCAG A", "WCAG AA", and "Mobiil", all of which are checked. Under "SIHTGRUPP:", there are checkboxes for "Disainerile", "Arendajale", and "Sisuloojale", all of which are checked. Below the sidebar is a "LIIGU SISUNI:" section with a list of links: "Üldised põhimõtted", "Lehe andmed", "Navigatsioon", "Pealkirjad", "Sisu paigutus", "Tekst", "Lingid", "Värvid", "Pildid", "Ikoonid", "Heli ja video", "Vormid", "Veateated", "Nupud", and "Tabelid". On the right, there is a search bar with the text "Sisesta otsingusõna ja vajuta Enter" and a "Sulge kõik" button. Below the search bar is a list of accordion items, each with a title and an "Ava" button with a downward arrow: "Üldised põhimõtted", "Lehe andmed", "Navigatsioon", "Pealkirjad", "Sisu paigutus", "Tekst", "Lingid", "Värvid", and "Pildid".

Joonis 31. Loodud veebilehe avakuva ekraanitõmmis.



FILTREER:

Sisesta otsingusõna ja vajuta Enter Sulge kõik

**TÜÜP:**  
 Soovitatav  
 WCAG A  
 WCAG AA  
 Mobiil

**SIHTGRUPP:**  
 Disainerile  
 Arendajale  
 Sisuloojale

LIIGU SISUNI:

- [Üldised põhimõtted >](#)
- [Lehe andmed >](#)
- [Navigatsioon >](#)
- [Pealkirjad >](#)
- [Sisu paigutus >](#)
- [Tekst >](#)
- [Lingid >](#)
- [Värvid >](#)
- [Pildid >](#)
- [Ikoonid >](#)
- [Heli ja video >](#)
- [Vormid >](#)
- [Veateated >](#)
- [Nupud >](#)
- [Tabelid >](#)

**Üldised põhimõtted** Ava ▾

**Lehe andmed** Sulge ▲

- Lehe keel peab olema märgitud ISO 639-1 koodiga **<html>** elemendi **lang="..."** atribuudis, näiteks **<html lang="et">**.
- Igal lehel peal olema informatiivne ja lehte kirjeldav tiitel **<title>** märgendina.
 

```
<head>
  <title>
    Kontakt - Firmanimi OÜ
  </title>
</head>
```
- Mõistlik on alustada tiitlit alamlehe nimega. Nt: **"Kontakt | Firmanimi OÜ"** mitte "Firmanimi OÜ | Kontakt", sest kui vahelehed on kitsad, pole viimasel juhul võimalik alamlehti üksteisest eristada.

✔

Teater | Linn... Inimesed | Li... Festival | Lin... +

Vahelehed algavad alamlehe nimega

✘

Directo - De... Directo - Par... Directo - Be... +

Kõik vahelehed algavad sama sõnaga

Joonis 32. Loodud veebilehe avatud akordioniga ekraanitõmmis.

Loodud veebilehte valideeriti kahe *front-end* arendajaga ning saadud kommentaaride põhjal tehti lehel mõningaid korrekture. Järgnevas peatükis tuuakse lisaks soovitusi veebilehe edasiseks arendamiseks.

## 4. Loodud suuniste komplekti analüüs

Juurdepääsetavuskriteeriumite olemasolu ei taga seda, et arendajad neid alati järgiksid. Nõuded peavad olema hoomatavad ja kasutatavad ning esitatud mugavalt ja lihtsalt. Käesoleva bakalaureusetöö tulemina loodi praktiline abivahend juurdepääsetavate veebilehtede loomiseks – kogum suuniseid ametlikest dokumentatsioonidest, kirjandusest ja erivajadustega inimestelt, koondatult mugavale veebilehele. Selles peatükis kirjeldatakse mõnede olemasolevate suunisekomplektide ja nende veebipõhiste esituste tugevaid ja nõrku külgi ning võrreldakse neid loodud abivahendiga. Lisaks tuuakse välja olulisi samme suuniste komplekti edasi arendamiseks.

### 4.1. Võrdlus olemasolevate suuniste komplektidega

WCAG 2.0 ametlik dokumentatsioon [2] on normatiivne dokument, mis oma suure mahu ning tehnilise sõnastuse tõttu võib olla raskesti hoomatav ja kasutatav. See on mõeldud kasutamiseks koos WCAG juurdepääsetavussuuniseid seletava dokumendiga [36] ning WCAG ametlikke juurdepääsetavuse tagamise tehnikaid kirjeldava dokumendiga [21]. Ametlik dokumentatsioon on tõlgitud eesti keelde, seletuste ja tehnikate dokumente pole aga tänaseni tõlgitud ning kõik lingid eestikeelsest dokumendist viivad inglisekeelse seletuse või tehnikani. Käesoleva bakalaureusetöö raames valminud suuniste komplekt on täielikult eestikeelne. WCAG tehnikate dokument on varustatud filtrite ja *tag*'ide süsteemiga, mis küll lihtsustavad sisu kasutamist, kuid on siiski piisavalt keerulised, et kasutaja võib kaotada motivatsiooni ning vajalikku suunist mitte leida. Selle bakalaureusetöö tulem võimaldab filtreerida suuniseid tüübi ja sihtrühma järgi. Vajaduse tekkimisel tuleks tulevikus luua lehele uusi filtreid.

MKM loodud WCAG 2.0 suuniste leht [37] on eestikeelne, kokkuvõtlik ja varustatud korralikult lahti seletatud näidetega. Kasutatud on vahelehti ja akordione, kuid lehel peab siiski üles-alla kerima ja navigeerima. Lehe tegeliku sisu jaoks on navigatsiooni kõrval jäetud üsna vähe ruumi. Lisaks leidub lehel soovitusi, mis autori uuringute kohaselt pole tegelikult juurdepääsetavuse tagamiseks parimad - näiteks vormi pealkirjade asemel kohatäite tekstide (inglise keeles *placeholder*) kasutamine, mida ei loe ekraanilugejad ning mis võivad olla halvasti nähtavad ka vaegnägijatele. Käesoleva töö tulemis on

suunised samuti komplekteeritud akordionitesse ning varustatud heade ja halbade näidetega. Silmas on peetud ka informatiivsele sisule piisava ruumi jätmist.

Wuhcag *checklist* [16] esitab WCAG 2.0 suunised hoomatavamal kujul. Iga nõude juures on paarisõnaline selgitus, kuid näidete, tehnikate ja pikema selgituse nägemiseks tuleb minna uuele lehele, mis tähendab, et kasutaja peab pidevalt edasi-tagasi navigeerima. Eraldatud on A, AA ja AAA taseme suunised. Antud töö tulemis on erinevate tasemete suunised koos, kuid neid on võimalik taseme järgi filtreerida. Näited ja tehnikad leiab iga suunise juurest ning kasutajal pole vaja lehelt lahkuda.

WebAIMi loodud WCAG 2.0 *checklist* [13] esitab suunised tabeli kujul, kus iga suunise juurest viib link WCAG 2.0 ametliku dokumentatsiooni lehele. WebAIMi Section 508 *checklist* [14] esitab nõuded samuti tabelina. Iga nõude juures on head ja halvad näited, kuid need on esitatud väga sarnaselt. Käesoleva töö tulemis on head ja halvad näited selgelt eristatavad. WebAIMi kasutatav esitus on lühike ja ülevaatlik, kuid ei tundu oma nipsisõnalisuse tõttu usaldusväärne. Selle töö tulemis on suuniseid kirjeldatud mahukamalt, kuid siiski võimalikult täpselt ja konkreetselt. Tulevikus võiks usaldusväärse tagamiseks lisada suunistele viiteid ametlikele dokumentatsioonidele ja teadusartiklitele.

*The ALLY Project Web Accessibility Checklist* [17] toob välja Section 508 nõuded. Nõuded on esitatud paari sõnaga, enamiku nõuete juures on koodinäited. Esitus on väga konkreetne, kuid selgitusi ja meetodeid nõuete täitmiseks pole. Iga nõude ees on märkeruut, mis võimaldab mugavalt ära märkida, millised nõuded on täidetud. Intervjuudest arendajatega tuli välja, et täidetud nõuete ära märkimisest võib tõesti abi olla – nii ei kao järg käest ära. Käesoleva töö raames loodud suuniste esitamisel on samuti kasutatud märkeruute.

Elsevier *Accessibility Checklist* [18] sisaldab nii WCAG 2.0 kui ka Section 508 nõudeid. Sisu saab filtreerida tasemete ja *tag*'ide abil. Nõuded on esitatud akordionis, mis on mugav ja ülevaatlik, kuid millele on filtreerimissüsteemi ja menüü kõrval jäetud veidi vähe ruumi. Lisaks on lehel kõrvuti kaks paralleelselt keritavat paneeli, mis võib olla segadust tekitav. Bakalaureusetöö tulemusena loodud keskkond on ülesehituselt sarnane Elsevieri lehega, kuid keritav on ainult üks lehe pool, navigeerida saab sisukaardi abil, kasutada saab otsingut ning kõiki akordione korraga sulgeda.

*Mobile Web Best Practices* [12] esitab W3C parimad praktikad mobiilse veebi loomiseks. Igale praktikale on antud koodnimi ja paarilauseiline selgitus. Mõned praktikad on varustatud ka koodinäidetega. Käesoleva töö tulemi põhiline eelis kõigi olemasolevate suuniste komplektide ees on see, et lauaarvutile ja mobiilseadmele mõeldud suunised asuvad ühes dokumendis. Põhimõtteliselt ei looda ühtegi lehte tänapäeval ainult lauaarvutitel kasutamiseks ning veebi loomisel on oluline algusest peale arvestada ka mobiilse veebi juurdepääsetavusega. Loodud abivahendi teine eelis on see, et suunised on grupeeritud enamasti HTML elementide järgi. Nii saab arendaja lülitada juurdepääsetavuse tagamise oma töösse juba lehe struktuuri loomisel ning kerge vaevaga leida suunised kindlate elementide loomiseks.

Bakalaureusetöö kirjutamise ajal on W3C välja töötamas WCAG versiooni 2.1 [38], kus võetakse arvesse ka mobiilseadmeid. Tegu on poolelioleva projektiga, sammuna WCAG versiooni 3.0 poole. 19. aprilli seisuga on lisatud järgmised suunised: sisu peab saama kaotusteta suurendada 400% (tase A); graafikale, mis annab edasi olulist informatsiooni, on kehtestatud minimaalsed kontrastinõuded (tase AA); kontekstimuutusi ja vahelesegamisi peab saama lihtsasti peatada (tase AA); tahtmatut aktiveerimist tuleb ära hoida, näiteks kasutades ainult hiirekliki (*onclick*) sündmuseid (tase A). Kuna antud suunised pole veel lõplikult kinnitatud, ei ole neid lisatud antud töö tulemisse.

## 4.2. Soovitused

Veebiarendajad on motiveeritud juurdepääsetavussuuniseid järgima, kui nad saavad aru, kelle jaoks nad seda teevad. Kus võimalik, on autor sõnastanud suunised nii, et oleks arusaadav, miks ja kellele need olulised on. Tulevikus võiks aga lisada suunistele viiteid näiteks videotele, kus on näha, kuidas erinevad erivajadustega inimesed veebi kasutavad ning kuidas seda nende jaoks lihtsamaks teha.

Tehnoloogia areneb pidevalt ning juurdepääsetavusnõuded võivad selle käigus muutuda. Seetõttu on oluline, et loodud tööriista hooldataks pidevalt ning uuendataks vastavalt vajadusele. Viimaste uuenduste sisseviimise kuupäev peab olema lehel alati märgitud, et tagada usaldusväärsus ning ajakohasus. Nagu toodi välja eelmises alapeatükis, võib vajaduse tekkimisel lisada lehele ka täiendavaid filtreid. Filtrite ja otsingu funktsionaalsust võiks tulevikus samuti täiendada. Näiteks valides filtritest vaid WCAG AA taseme, tuleks tegelikult näidata ka A taseme suuniseid, sest AA taseme saavutamine

tähendab, et täidetud on nii A kui ka AA taseme edukriteeriumid. Otsingukasti juurest võiks tulevikus saada liikuda järjest kõigi otsingutulemuste juurde, et soovitud suunise leidmine oleks veelgi kiirem ja lihtsam. Usaldusväarsuse suurendamiseks võiks lisada suunistele ka viiteid erinevatele allikatele.

Käesoleva bakalaureusetöö raames loodud veebileht on mõeldud kasutamiseks eelkõige lauaarvutites, kuid nende osatähtsuse kestva langemise tõttu veebitarbimisvahenditena on kindlasti oluline, et tulevikus kohanduks veebileht ka tahvelarvutitele ja mobiilidele. Lisaks on autori arvates oluline, et juurdepääsetavussuuniseid esitav veebileht ise vastaks kõigile nendele suunistele.

## 5. Kokkuvõte

Mobiilseadmeid kasutatakse veebi tarbimiseks üha rohkem ning seda ka erivajadustega kasutajate hulgas. Seega on veebilehtede loomisel väga oluline tagada nende juurdepääsetavus ka mobiilseadmetel. Olemasolevad juurdepääsetavussuuniste komplektid keskenduvad kas ainult mobiilseadmetele või ei arvesta neid üldse. Pealegi ei kiputa juurdepääsetavussuuniseid järgima, sest need on enamasti keerulised ja raskesti hoomatavad.

Bakalaureusetöö “Suunised tagamaks veebi sisu juurdepääsetavust puuetundlikel mobiilseadmetel” eesmärgiks oli välja töötada erivajadustega inimeste ja arendajate vajadustega arvestavate suuniste komplekt, mis tagaks veebi sisu juurdepääsetavuse ka puuetundlikel mobiilseadmetel. Töö laiemaks eesmärgiks võib nimetada juurdepääsetavussuuniste järgimise suurendamist veebiarendajate seas, et luua seeläbi paremini kasutatavaid veebilehti.

Eesmärgi saavutamiseks püstitati sissejuhatuses kolm uurimisküsimust, millele töö käigus vastused leiti. Selgitamaks välja, millised suunised tagaksid piisava juurdepääsetavuse puuetundlikel seadmetel ning milliste tugitehnoloogiatega peaksid need arvestama, uuriti erinevaid juurdepääsetavussuuniste komplekte ja puuetega inimeste veebikasutust puudutavaid artikleid ning viidi läbi intervjuud kuue erivajadustega inimesega. Saamaks teada, milliseid suuniseid sooviksid veebiarendajad ning kuidas need peaksid esitatud olema, viidi läbi intervjuud kuue kasutajaliidese arendajaga.

Intervjuude ning kirjanduslike allikate põhjal loodi 140 suunist, mis jagati viieteistkümnesse kategooriasse, et arendajatel oleks suunistes kergem orienteeruda. Suuniste esitamiseks loodi veebileht, mis võimaldab minimaalse vaevaga leida vajalikke suuniseid ning neid mõista. Loodud veebilehte valideeriti kahe *front-end* arendajaga ning tehti vajalikke korrekture. Saadud tagasiside põhjal võib väita, et bakalaureusetöö eesmärk on saavutatud ja mugav keskkond juurdepääsetavussuuniste kasutamiseks loodud.

## Kasutatud kirjandus

1. Introduction to Web Accessibility. — *Web Accessibility Initiative*. [WWW] <https://www.w3.org/WAI/intro/accessibility.php> (03.03.2017)
2. Web Content Accessibility Guidelines 2.0. — *World Wide Web Consortium*. [WWW] <https://www.w3.org/TR/WCAG20/> (15.03.2017)
3. Section 508 Law. [WWW] <https://www.section508.gov/section508-laws> (26.03.2017)
4. Guide d'Accompagnement. — *Les Documents de Référence du S.l. de l'État*. [WWW] <https://references.modernisation.gouv.fr/rgaa-accessibilite/guide-accompagnement-RGAA.html> (26.03.2017)
5. Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz. — *Bundesministerium der Justiz und für Verbraucherschutz*. [WWW] [http://www.gesetze-im-internet.de/bitv\\_2\\_0/\\_1.html](http://www.gesetze-im-internet.de/bitv_2_0/_1.html) (26.03.2017)
6. Rogers, M. Government accessibility standards and WCAG 2. — *PowerMapper*, 2016. [WWW] <https://www.powermapper.com/blog/government-accessibility-standards/> (26.03.2017)
7. Veebide koosvõime raamistik 1.0. — *Majandus- ja Kommunikatsiooni ministeerium*, 2012. [WWW] [https://www.mkm.ee/sites/default/files/veebide\\_raamistik.pdf](https://www.mkm.ee/sites/default/files/veebide_raamistik.pdf) (15.03.2017)
8. Avaliku sektori veebilehtede vastavus WCAG 2.0 nõuetele 2015. aastal: Uuringu aruanne. — *Majandus- ja Kommunikatsiooni ministeerium*, 2015. [WWW] [https://www.mkm.ee/sites/default/files/wcag\\_aruanne\\_2015.pdf](https://www.mkm.ee/sites/default/files/wcag_aruanne_2015.pdf) (15.03.2017)
9. Mobile and tablet internet usage exceeds desktop for first time worldwide. — *StatCounter*, 2016. [WWW] <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide> (10.03.2017)
10. Schade, A. Responsive Web Design (RWD) and User Experience. — *Nielsen Norman Group*, 2014. [WWW] <https://www.nngroup.com/articles/responsive-web-design-definition/> (20.03.2017)
11. Section 508 Standards for Electronic and Information Technology. — *United States Access Board*, 2000. [WWW] <https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards> (12.04.2017)
12. Mobile Web Best Practices 1.0. — *Web Accessibility Initiative*. 2008. [WWW] <https://www.w3.org/TR/mobile-bp/> (14.03.2017)

13. WebAIM's WCAG 2.0 Checklist for HTML documents. — *WebAIM*. [WWW]  
<http://webaim.org/standards/wcag/checklist> (14.03.2017)
14. Section 508 Checklist. — *WebAIM*. [WWW]  
<http://webaim.org/standards/508/checklist> (14.03.2017)
15. Easy Checks - A First Review of Web Accessibility. — *Web Accessibility Initiative*. [WWW] <https://www.w3.org/WAI/eval/preliminary.html> (14.03.2017)
16. McGrath, L. WCAG 2.0 Checklist. — *Wuhcag*. [WWW]  
<https://www.wuhcag.com/wcag-checklist/> (14.03.2017)
17. Web Accessibility Checklist. — *The ALLY Project*. [WWW]  
<http://a11yproject.com/checklist.html> (14.03.2017)
18. Accessibility Checklist. — *Elsevier*. [WWW]  
[http://romeo.elsevier.com/accessibility\\_checklist/](http://romeo.elsevier.com/accessibility_checklist/)(14.03.2017)
19. Rääp, A. Nägemispuudega inimestele ligipäasetavad kodulehed. [WWW]  
<http://rtr.planet.ee/ligipaasetavad.html> (10.03.2017)
20. How to Meet WCAG 2.0. — *Web Accessibility Initiative*. [WWW]  
<https://www.w3.org/WAI/WCAG20/quickref/> (06.04.2017)
21. Hänni, K. Internetikasutuse ligipäasustrateegiad ja kasutamispriktikad nägemis- ja liikumispuudega inimeste näitel: magistritöö. Tartu Ülikool, Tartu, 2016.
22. Cooper, M., Sloan, D., Kelly, B., Lewthwaite, S. A challenge to web accessibility metrics and guidelines: putting people and processes first. — *Proceedings of the 9th International Cross-Disciplinary Conference on Web Accessibility, W4A '12*, April 16-18, 2012. [Online] <http://opus.bath.ac.uk/29190/2/w4a-2012-p23-final.pdf> (12.03.2017)
23. Power, C., Freire, A., Petrie, H., Swallow, D. Guidelines are only half of the story: accessibility problems encountered by blind users on the web. — *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*, 2012. ACM, New York, NY, USA, 433-442. [Online]  
[https://www.cs.york.ac.uk/hci/publications/001/guidelines\\_are\\_only\\_half\\_of\\_the\\_story.pdf](https://www.cs.york.ac.uk/hci/publications/001/guidelines_are_only_half_of_the_story.pdf) (12.03.2017)
24. Guidelines for the development of accessible mobile interfaces. Stockholm: Funka Nu AB. [WWW]  
[http://www.funka.com/contentassets/5f2704e1efa942a29f9bbb20facea4d1/guidelines\\_for\\_the\\_development\\_of\\_accessible\\_mobile\\_interfaces.pdf](http://www.funka.com/contentassets/5f2704e1efa942a29f9bbb20facea4d1/guidelines_for_the_development_of_accessible_mobile_interfaces.pdf) (11.03.2017)
25. Mobile Navigation Guidelines. Stockholm: Funka Nu AB, 2014. [WWW]  
<http://www.funka.com/contentassets/5f2704e1efa942a29f9bbb20facea4d1/mobile-navigation-guidelines-funka-2014.pdf> (11.03.2017)
26. Rømen, D., Svanæs, D. Validating WCAG versions 1.0 and 2.0 through usability testing with disabled users. — *Universal Access in the Information Society*, 2012, 11 (4), 375-385. [Online] Springer (12.03.2017)



27. Kurniawan, S., Zaphiris, P. Research-Derived Web Design Guidelines for Older People. — *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, 2005. ACM, New York, NY, USA, 129-135. [Online] ACM Digital Library (13.03.2017)
28. Diaz-Bossini, J.-M., Moreno, L. Accessibility to mobile interfaces for older people. — *Procedia Computer Science*, 2014, 27, 57-66. [Online] [http://ac.els-cdn.com/S1877050914000106/1-s2.0-S1877050914000106-main.pdf?\\_tid=c167a298-2ce5-11e7-bc34-00000aacb361&acdnat=1493475276\\_1cd8e60d11472725c5d95136f5a54469](http://ac.els-cdn.com/S1877050914000106/1-s2.0-S1877050914000106-main.pdf?_tid=c167a298-2ce5-11e7-bc34-00000aacb361&acdnat=1493475276_1cd8e60d11472725c5d95136f5a54469) (15.03.2017)
29. Web Accessibility Tutorials. — *Web Accessibility Initiative*, 2017. [WWW] <https://www.w3.org/WAI/tutorials/> (18.03.2017)
30. Using Font Awesome with Accessibility in mind. *Font Awesome*. [WWW] <http://fontawesome.io/accessibility> (18.03.2017)
31. Wild, G. Links and accessibility. — *AccessibilityOz*, 2014. [WWW] <https://www.accessibilityoz.com/2014/02/links-and-accessibility/> (21.03.2017)
32. Panchang, S. Text Links: Best Practices for Screen Readers. *The Deque Blog*. 2014. [WWW] <https://www.deque.com/blog/text-links-practices-screen-readers/> (20.03.2017)
33. Snook, J. Hiding Content for Accessibility. [WWW] [https://snook.ca/archives/html\\_and\\_css/hiding-content-for-accessibility](https://snook.ca/archives/html_and_css/hiding-content-for-accessibility) (02.04.2017)
34. Gustafson, A. Understanding Color Blindness: A Guide to Accessible Design. — *CruX Collaborative*, 2015. [WWW] <http://www.cruxcollaborative.com/understanding-color-blindness-guide-to-accessible-design/> (28.04.2017)
35. Material Design Components: Bottom Navigation. — *Google*. [WWW] <https://material.io/guidelines/components/bottom-navigation.html> (01.05.2017)
36. Understanding WCAG 2.0. — *World Wide Web Consortium*. <https://www.w3.org/TR/UNDERSTANDING-WCAG20/Overview.html> (06.04.2017)
37. WCAG 2.0 rakendusjuhised. — *Majandus- ja Kommunikatsiooni ministeerium*. [WWW] <https://www.mkm.ee/et/wcag-20-rakendusjuhised> (15.04.2017)
38. Web Content Accessibility Guidelines 2.1 Working Draft. — *World Wide Web Consortium*. [WWW] <https://www.w3.org/TR/WCAG21/> (10.05.2017)

## Lisa 1: küsimustik arendajatele

1. Kas tavaliselt on Sinu projektides kehtestatud nõuded juurdepääsetavusele?
2. Kui nõutud pole, siis kas järgid juurdepääsetavussuuniseid?
3. Millistest juurdepääsetavussuuniste komplektidest oled Sa kuulnud?
4. Milliseid suuniseid, keskkondi ja kontrollitööriistu kasutad?
5. Kuidas lülitad juurdepääsetavuse tagamise oma töösse? Mida teed teistmoodi kui projektis, kus pole vaja juurdepääsetavusega arvestada?
6. Mida teed selleks, et ka mobiilne versioon veebilehest oleks juurdepääsetav?
7. Kuidas kontrollid oma koodi vastavust juurdepääsetavusnõuetele?
8. Kuidas hindad juurdepääsetavuse tagamise keerukust? Millest see tuleneb?
9. Millised probleemid on tekkinud juurdepääsetavuse tagamisel?
10. Kas kasutad *aria*-atribuute nagu näiteks *aria-hidden*, *aria-label*?
11. Millist tööriista kasutad veebilehel oleva värvi koodi teada saamiseks ja selle kontrastsuse kontrollimiseks?
12. Palun kommenteeri järgmiste veebilehtede arusaadavust, hoomatavust ja kasutusmugavust:
  1. WCAG 2.0 ametlik dokumentatsioon <https://www.w3.org/TR/WCAG20/>
  2. *Mobile Web Best Practices* (MWBP) 1.0 ametlik dokumentatsioon <https://www.w3.org/TR/mobile-bp/>
  3. WebAIM WCAG 2.0 *Checklist* <http://webaim.org/standards/wcag/checklist>
  4. *WebAIM Section 508 Checklist* <http://webaim.org/standards/508/checklist>
  5. *WAI Easy Checks* <https://www.w3.org/WAI/eval/preliminary.html>
  6. *Wuhcag WCAG 2.0 Checklist* <https://www.wuhcag.com/wcag-checklist/>
  7. *The A11Y Project Web Accessibility Checklist* <http://a11yproject.com/checklist.html>
  8. *Elsevier Accessibility Checklist* [http://romeo.elsevier.com/accessibility\\_checklist/](http://romeo.elsevier.com/accessibility_checklist/)

## **Lisa 2: küsimustik erivajadustega inimestele**

1. Milliste seadmete ja abivahenditega Sa internetti kasutad?
2. Milliseid puuetundlikke seadmeid Sa kasutad või oled kasutanud?
3. Kui tihti Sa puuetundlikke seadmeid kasutad?
4. Mille jaoks Sa neid kasutad?
5. Millised on puuetundliku seadme eelised füüsilise klaviatuuriga seadme ees Sinu jaoks?
6. Milliseid raskusi on ette tulnud puuetundlikku seadet kasutades?
7. Millised on kõige ebamugavamad tegevused puuetundlikul seadmel?
8. Kas oled pidanud valima teistsuguse puuetundliku seadme, et seda oleks lihtsam kasutada?
9. Milliseid rakendusi ja veebilehti Sa oma puuetundlikul seadmel kasutad? Kui sageli?
10. Millised on Sinu arvates parimad ja mugavaimad veebilehed ja rakendused, mida kasutad? Mis teeb need mugavaks?
11. Milliseid negatiivseid kogemusi on ette tulnud veebilehtede või rakenduste kasutamisel?
12. Nimeta mõni keskkond, mida on keeruline kasutada? Mis teeb selle keeruliseks?
13. Kui kõik oleks võimalik, siis mida sooviksid muuta oma puuetundlikus seadmes või veebilehtedes, mida selle seadmega kasutad?
14. Kas oled pidanud puuetundlikku seadet kasutades kelleltki abi küsima? Mille jaoks tema abi vajasid

## Lisa 3: head ja halvad näited

Üldised põhimõtted: suunis nr 11



(a)

(b)

Joonis 33. (a) halb näide: kasutajat sunnitakse lohistama, (b) hea näide: kasutajalt küsitakse loogiline küsimus.

Lehe andmed: suunis nr 3



(a)

(b)

Joonis 34. (a) halb näide: kõik vahelehed algavad sama sõnaga, (b) hea näide: vahelehed algavad alamlehe nimega.

Navigatsioon: suunis nr 4



(a)

(b)

Joonis 35. (a) halb näide: menüü ikooni võib pidada logo osaks, (b) hea näide: lisaks ikoonile on olemas ka tekst.

## Navigatsioon: suunis nr 6

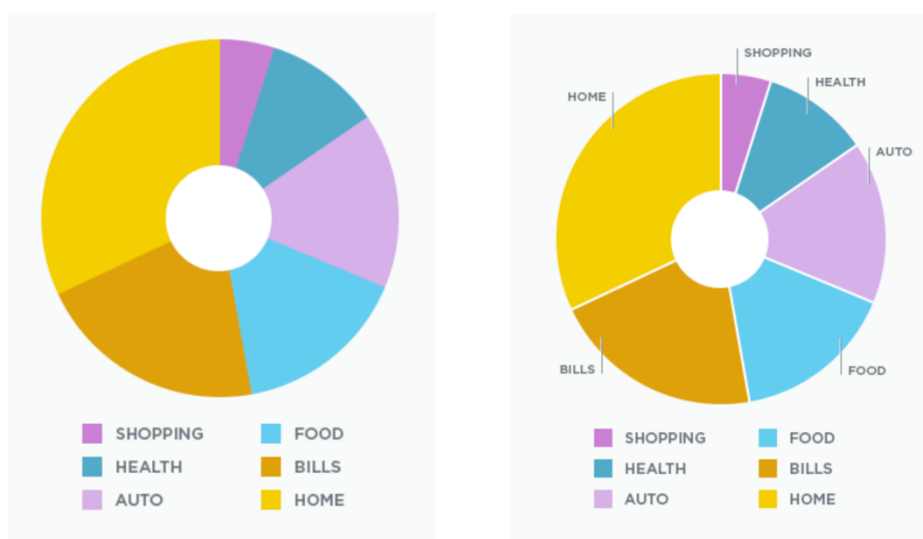


(a)

(b)

Joonis 36. (a) halb näide: ei ole selge, kuidas otsingut käivitada, (b) hea näide: on selge, et vajutada tuleb nuppu.

## Värvid: suunis nr 1



(a)

(b)

Joonis 37. (a) halb näide: graafiku sektorid on eristatud vaid värviga, (b) hea näide: graafiku sektoritel on lisaks värvile juures ka sildid.

## Värvid: suunis nr 4

Posted on March 9, 2017 by  
Resources | 5 con

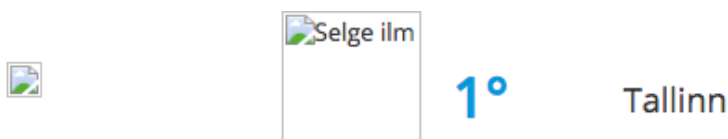
How to Accept  
Payments on  
WordPress V

(a)

(b)

Joonis 38. (a) halb näide: liiga madal kontrastsus 2.21:1, (b) hea näide: kõrge kontrastsus 14.79:1.

### Pildid: suunis nr 1



(a)

(b)

Joonis 39. (a) halb näide: tekstiline alternatiiv puudub ja info läheb kaduma, (b) hea näide: tekstiline alternatiiv on olemas ning kannab endas sama infot, mis pilt.

### Ikoonid: suunis nr 1



(a)

(b)

Joonis 40. (a) halb näide: menüüikoon ei ava menüüd, vaid lisafiltrid, (b) hea näide: tööriistaikoon avab lisafiltrid.

### Vormid: suunis nr 3

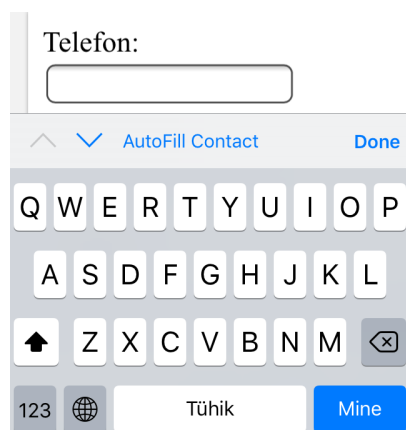
<p><b>Kauba saaja aadress:</b> <input type="text"/></p> <p><b>Kauba saaja postiindeks:</b> <input type="text"/></p> <p><b>Kauba saaja linn:</b> <input type="text"/></p> <p><b>Arve saaja aadress:</b> <input type="text"/></p> <p><b>Arve saaja postiindeks:</b> <input type="text"/></p> <p><b>Arve saaja linn:</b> <input type="text"/></p>	<p><b>Kauba saaja aadress:</b> Aadress: <input type="text"/></p> <p>Postiindeks: <input type="text"/></p> <p>Linn: <input type="text"/></p> <p><b>Arve saaja aadress:</b> Aadress: <input type="text"/></p> <p>Postiindeks: <input type="text"/></p> <p>Linn: <input type="text"/></p>
--	--

(a)

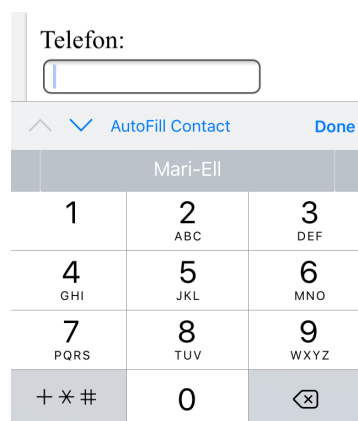
(b)

Joonis 41. (a) halb näide: pealkirjades on palju sõnakordust, mis on visuaalselt segadusseajav, (b) hea näide: kokkukuuluvad väljad on grupeeritud ja ühise pealkirja all.

## Sisestuskastid: suunis nr 1



(a)



(b)

Joonis 42. (a) halb näide: telefonivälja tüübiks on määratud *text* ning mobiilseadmes kuvatakse vaikimisi klaviatuur, kus numbrite sisestamiseks peab tegema lisaliigutuse ning esmalt on näha tähteklahvid, mida antud välja puhul vaja ei ole. (b) hea näide: telefonivälja tüübiks on määratud *tel* ning mobiilseadmes kuvatakse numbriklaviatuur, mis on loodud just telefoninumbrate sisestamiseks. Lisaks pakutakse otseteed oma telefoninumbri sisestamiseks.

## Kohustuslikud väljad: suunis nr 3

Legal last name\*

example of persistent helper text

\* indicates required field

(a)

\* indicates required fields

**Database type \***

MySQL, MariaDB, or equivalent

The type of database your Drupal data will be stored in. \ supports a single database type, so it has been automati

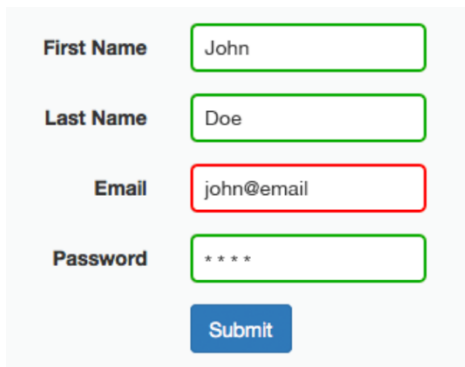
**Database name \***

The name of the database your Drupal data will be stored before Drupal can be installed.

(b)


Joonis 43. (a) halb näide: selgitus täрни kohta asub peale välju, (b) hea näide: selgitus täрни kohta asub enne välju.

Veateated: suunis nr 2



Form (a) shows a registration form with four fields: First Name (John), Last Name (Doe), Email (john@email), and Password (\*\*\*\*). The First Name, Last Name, and Password fields have green borders, while the Email field has a red border, indicating a validation error.

(a)



Form (b) shows the same registration form as (a), but with green checkmarks next to the First Name, Last Name, and Password fields, and a red X next to the Email field, indicating that the form is now valid.

(b)

Joonis 44. (a) halb näide: vigane väli on eristatud vaid värviga, (b) hea näide: vigane väli on lisaks värvile eristatud ka ikooniga.

Nupud: suunis nr 5



(a)



(b)

Joonis 45. (a) halb näide: ainult tekst on klikitav, (b) hea näide: kogu nupu pind on klikitav.