

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Oleksandra Chmil 142429IABB

**KASUTAJALIIDESE TESTIMISE
AUTOMATISEERIMINE JA SELLE
OTSTARBEKUSE ANALÜÜS MAKSU- JA
TOLLIAMETI RAKENDUSE
„MADALAPALGALISTE TOETUS“ NÄITEL**

Bakalaureusetöö

Juhendaja: Inna Švartsman
MSc
Lektor

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: Oleksandra Chmil

22.05.2017

Annotatsioon

Käesoleva lõputöö eesmärkideks on:

- Luua võimalikult efektiivseid teste, mis kontrollivad kasutajaliidese elementide vastavust Maksu- ja Tolliameti projekti nõuetele ja põhilistele kasutajakogemuse printsiipidele.
- Selgitada välja ja analüüsida, millist testimise meetodikat on kõige otstarbekam kasutada kasutajaliidese testimiseks, juhul kui nõuded kasutajaliidesele on konkreetselt määratud.

Eesmärgid said bakalaureusetöö käigus saavutatud ning lõputöö oodatud tulemus vastab tegelikule tulemusele.

Bakalaureusetöö käigus said kirjeldatud põhilised kasutajakogemuse printsiibid, lähtudes kümnest Jakob Nielsen'i heuristikast. Lisaks said võrreldud kaks automaattestimise vahendit, millest valiti välja *Selenide*'i raamistik. Väljavalitud vahendi abil loodi automaatteste rakenduse kasutajaliidese jaoks. Samuti said väljatoodud ja kirjeldatud erinevad *Selenide*'i pakutavad võimalused testitava rakenduse näitel. Juhul kui testi tulemus ei vastanud oodatud tulemusele, siis analüüsiti tekkinud viga ja tehti vastav ettepanek.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 41 leheküljel, 5 peatükki, 30 joonist, 3 tabelit.

Abstract

Automation of User Interface Testing and its Expediency Analysis Based on Tax and Customs Board's "Support to Low-wage Employees" Application

The aims of this thesis are:

- 1) Automation of user interface testing for Tax and Customs Board's application named "Support to Low-wage Employees" and testing the application's elements in accordance with the Tax and Customs Board's project requirements and main user experience principles.
- 2) Analysis of which method is the most suitable for user interface testing if the project requirements are clearly stated.

The described goals are achieved and the expected result of the thesis matches the real result.

The thesis describes main user experience principles based on the analysis of Jakob Nielsen's 10 heuristics. The author of this thesis compared two main testing methods: manual and automated testing. Based on this comparison, automated user interface testing was chosen.

For automating user interface testing, a library named Selenide was chosen, because in the author's opinion this tool is the most effective for user interface testing when the project requirements are clearly stated. As a result of this thesis, user interface testing of the aforementioned application has been automated and it has been analysed whether the testing results and the project requirements match the chosen Jakob Nielsen's heuristics.

The thesis is in Estonian and contains 41 pages of text, 5 chapters, 30 figures, 3 tables.

Lühendite ja mõistete sõnastik

AJAX	<i>Asynchronous JavaScript And XML</i>
e-	elektrooniline
ICEGOV	<i>International Conference on Theory and Practice of Electronic Governance</i> , rahvusvaheline organisatsioon, mille eesmärk on jagada informatsiooni, mis on seotud uute saavutustega elektroonses valitsuses [10].
IDE	<i>Integrated Development Environment</i>
jQuery	Kiire, väike ja võimalusterohke <i>JavaScript</i> programmeerimiskeele raamistik [30].
Px	<i>Pixel</i> Piksel on väikeste diskreetsete elementide kogu, mis moodustavad kokku pildi [4].
Kasutajakogemus	<i>User experience</i> Isiku taju ja vastused, mis tekivad mingi toote, süsteemi või teenuse kasutamise tulemusel [6].
Kasutajaliides	<i>User interface</i> Ühenduslüli, mis seob kasutajat ja arvutiprogrammi [11].
Kasutatavus	<i>Usability</i> Aste, milles toode võib olla kasutatud kindla kasutaja poolt selleks, et saavutada teatud eesmärged efektiivselt, tulemuslikult ning saavutada rahulolu [5].
Kohalduv disain	<i>Responsive design</i> , disain, mis kohaldub erinevate ekraanisuurustega.
MTA	Maksu- ja Tolliamet
Selenide	<i>Selenium</i> 'i raamistik
Selenium IDE	Tasuta, vaba lähtekoodiga plugin Firefox veebilehitsejale [12].
Silumine	<i>Debugging</i> „Tarkvaraarenduse protsess, mille abil leitakse ja parandatakse arvutiprogrammi või riistvara vigu [28].“
WebDriver	Brauseri automatiseerimisevahend [27].

Sisukord

1 Sissejuhatus	10
1.1 Taust ja probleem	10
1.2 Ülesande püstitus	11
1.3 Metoodika.....	12
2 Tarkvara kasutatavus	14
2.1 Kasutajakogemuse printsiibid lähtudes Jakob Nielsen'i heuristikatest	14
2.2 Tarkvara kasutatavus erinevate ekraanisuurustega.....	18
2.3 Kohalduva disaini põhimõtted.....	19
2.4 Tarkvara kasutatavuse testimine.....	20
2.4.1 Manuaalne ja automatiseeritud tarkvara kasutatavuse testimine.....	21
2.4.2 Manuaalse ja automatiseeritud testimise võrdluse põhjal valiku tegemine...	21
2.5 Automaattestide loomiseks vahendi valimine	22
2.5.1 Automaattestide loomise vahendite võrdlus.....	23
2.5.2 Automaattestide loomise vahendi valik ja põhjendus	24
3 Tarkvara kasutajaliidese testimine.....	26
3.1 Testitava rakenduse kirjeldus testide gruppide loomiseks	26
3.2 Testide gruppide loomine	28
3.2.1 Testide gruppide kirjeldamine arvuti vaate jaoks.....	29
3.2.2 Testimise tulemused ekraanisuurusega 1200x903px	32
3.2.3 Testide gruppide kirjeldamine nutitelefonis vaate jaoks.....	41
3.2.4 Testimise tulemused ekraanisuurusega 599x903px	44
4 Tulemuste analüüs ja ettepanekud	47
4.1 Testitava rakenduse tulemuste analüüs.....	47
4.1.1 Kasutajaliidese testimise automatiseerimise otstarbekus	47
4.1.2 Saadud tulemuste valideeritavus	49
5 Kokkuvõte	50
Kasutatud kirjandus	51

Jooniste loetelu

Joonis 1. Nutitelefonide võrrand [18].....	18
Joonis 2. Kohalduva disaini näide [18].	20
Joonis 3. Rakenduse esimese avanemise vaade.....	27
Joonis 4. Tingimused toetuse taotlemiseks: detailvaade.	28
Joonis 5. Esimese testide grupi testid.	33
Joonis 6. Esimese testide grupi tulemused.	33
Joonis 7. Väljumise nupu stiili testide grupi testid.....	34
Joonis 8. Väljumise nupu testimise tulemused.....	34
Joonis 9. Lingi „Tagasi e-maksuameti/e-tolli lehele” vaikimisi oleku testid.....	35
Joonis 10. „Tagasi e-maksuameti/e-tolli lehele” lingi stiili testid hiire pealasumisel. ...	36
Joonis 11. „Tagasi e-maksuameti/e-tolli lehele” lingi testimise tulemused.	36
Joonis 12. Keelevaliku linkide testimise tulemused.....	37
Joonis 13. „Vaatan lähemalt“ lingi vaikimisi oleku testid.	38
Joonis 14. „Vaatan lähemalt“ lingi testid arvutihiire pealasumisel.	38
Joonis 15. „Vaatan lähemalt” lingi testimise tulemused.	38
Joonis 16. „Kõik kontaktandmed” lingi testimise tulemused.....	39
Joonis 17. „Tagasi e-maksuameti/e-tolli lehele” lingi olemasolu kontrolli test.....	39
Joonis 18. „Tagasi e-maksuameti/e-tolli lehele” lingi kuvamise testimine.....	39
Joonis 19. Kontaktandmete kättesaadavuse testimise tulemused.....	40
Joonis 20. Abiteksti olemasolu test.	40
Joonis 21. Abiteksti olemasolu testimise tulemused.	41
Joonis 22. Rakenduse esmane vaade nutitelefoni.	42
Joonis 23. Väljumise nupp nutitelefoni vaates.	43
Joonis 24. Päise lingid nutitelefoni vaates.....	44
Joonis 25. Esimese testide grupi tulemused mobiilivaates.....	44
Joonis 26. Teise testide grupi tulemused nutitelefoni vaates.....	45
Joonis 27. Kolmanda testide grupi tulemused nutitelefoni vaates.	45
Joonis 28. Kontaktandmete kättesaadavuse testimise tulemused nutitelefoni vaates. ...	45
Joonis 29. Abiteksti olemasolu test nutitelefoni vaates.....	46

Joonis 30. Abiteksti olemasolu testimise tulemused nutitelefonis vaates..... 46

Tabelite loetelu

Tabel 1. Jakob Nielsen'i kümme kasutatavuse heuristikat [24].	15
Tabel 2. Manuaalse ja automatiseeritud testimise võrdlus [1, 9, 12, 14].	22
Tabel 3. <i>Selenium WebDriver</i> ja <i>Selenide</i> võrdlus [26].	23

1 Sissejuhatus

Lõputöö autor on tööl tegelenud Maksu- ja Tolliameti veebilehel asuva rakendusega, mille nimeks on „Madalapalgaliste toetus“. Seetõttu autor on huvitatud töö juures saadud kogemuse sidumisest oma bakalaureusetööga. Rakenduse kasutajaliidest oli enne käesoleva bakalaureusetööga alustamist testitud manuaalselt, mistõttu ei pruugi testimise tulemus olla usaldusväärne. Kasutajaliides on vastavalt definitsioonile ühenduslüli, mis seob kasutajat ja arvutiprogrammi [11]. Lõputöö käigus lahendatakse probleeme, mis on seotud veebidisaini anomaaliatega kasutajaliideses. Probleemi lahendamiseks uuritakse kasutajaliidese testimise automatiseerimise võimalusi. Veebidisaini vead võivad olla ettearvamatud, seega selgitatakse antud töö käigus välja võimalikult palju kombinatsioone rakenduse kasutajaliidese testimiseks.

1.1 Taust ja probleem

Alates veebruarist 2017 on kõikidel Eesti residentidel, kellel on täidetud vajalikud nõudmised, võimalik taotleda toetust „Madalapalgaliste toetus“. Mugava taotluse võimaldamiseks lõi Maksu ja Tolliamet veebipõhise rakenduse, kus igaüks saab tutvuda oma andmetega ja olenevalt nende sobivusest antud toetust taotleda. [16]

Eesti on maailmas tuntud e-riigina ning Juhan Parts, endine majandus- ja kommunikatsiooniminister, on sellel teemal rahvusvahelisel ICEGOV (*International Conference on Theory and Practice of Electronic Governance*) konverentsil oma arvamust avaldanud. Tema sõnul on tänapäeval infotehnoloogia kasutuselevõtt erinevates valdkondades loomulik nähtus. Uute lahenduste väljatöötamise käigus on oluline võtta aluseks vajadused. Eesti on e-riigina tuntud sellepärast, et loodud elektroonilised lahendused on kasutuskõlblikud ja Eestis kasutatakse neid igapäevaselt. [15] Käesoleva bakalaureusetöö autori arvates tähendab kõik ülalkirjeldatu seda, et riigiasutuste veebilehed ja sealhulgas ka rakendus „Madalapalgaliste toetus“ peaksid olema võimalikult kasutajasõbralikud.

Viimaste aastate jooksul on veebidisain oma arengus teinud väga suure sammu. Iga mainekama firma veebileht on uhke disainiga, millega pole tänapäeval sugugi lihtne isegi tavakasutajale muljet avaldada. Kuna veebidisaini tase on niivõrd kõrge, siis peab see olema kvaliteetne ning üha rohkem peab tähelepanu pöörama testimisele. Praegusele põlvkonnale on olulised visuaalsed väärtused, seega müüb tänapäeval hea disainilahendus. Firmad üritavad luua võimalikult kasutajasõbralikku, mugavat, universaalset ja kui vähegi võimalik, siis ka kohalduvat disaini. Hea veebidisaini rohkuse tõttu on kasutaja muutunud väga nõudlikuks, seega tuleb tähelepanu pöörata detailidele. Samuti kasutatakse tänapäeval veebilehti väga erinevates seadmetes, mis teeb selle loomise protsessi väga mahukaks, kuna iga kasutaja peaks saama kasutada sama funktsionaalsust ning lehe välimus peab iga kasutaja jaoks samaks jääma.

Kõige lihtsam viis kasutajaliidese testimiseks on manuaalne testimine, kuid mahukamate projektide puhul võib inimfaktor negatiivselt mõjuda. Lõputöö autori isiklik kogemus näitab, et iga elemendi parameetrite kontrollimine käsitsi võtab palju aega ning isegi kui tingimused saavad testitud, võib disain peale järgmist koodi muutmist katki minna. Seega tuleb samasugust testimist pideval korral, et veenduda disaini korrektsuses ja uute vigade puudumises. Aeg on väga kallis ja oluline ressurss, seega peab autori arvates iga tegevus olema võimalikult efektiivne. Testimise efektiivsuse tagamiseks tuleb seda teha võimalikult vähese vaeva-, ajakulu- ja klikkide arvuga. Suuremal määral on inimfaktorist tulenevaid vigu võimalik vältida võttes kasutusele automaattestimise. Seega tehakse lõputöö käigus manuaalse ja automatiseeritud testimise võrdlused ning valitakse välja sobivaim meetod, lähtudes konkreetse projekti eesmärkidest.

Käesoleva bakalaureusetöö käigus kasutatakse kombineeritud rakenduse testimise metoodikat. Kasutajaliidese teste luuakse, lähtudes projektis määratud nõuetest ja lõputöös kirjeldatud kirjeldatud kasutajakogemuse printsiipidest.

1.2 Ülesande püstitus

Bakalaureusetöö käigus uuritakse kasutajaliidese testimisvõimalusi. Eesmärkideks on:

- Luua võimalikult efektiivseid teste, mis kontrollivad kasutajaliidese elementide vastavust Maksu- ja Tolliameti projekti nõuetele ja põhilistele kasutajakogemuse printsiipidele.
- Selgitada välja ja analüüsida, millist testimise meetoodikat on kõige otstarbekam kasutada kasutajaliidese testimiseks, juhul kui nõuded kasutajaliidesele on konkreetselt määratud.

Lõputöö oodatavaks tulemuseks on analüüs „Madalapalgaliste toetus“ rakenduse kasutusmugavuse kohta, lähtudes läbiviidud kasutajakogemuse testidest ning kasutajaliidese katvus testidega, mis on loodud lähtudes Jakob Nielsen'i heuristikatest.

1.3 Metoodika

Püstitatud eesmärkide saavutamiseks uurib autor meetodeid ja vahendeid kasutajaliidese testide loomiseks. Testide loomisel rakenduse kasutajaliidese jaoks lähtub bakalaureusetöö autor põhilistest kasutajakogemuse reeglitest. Kasutajakogemuse reeglid valib autor kasutades Jakob Nielsen'i 1995. aastal loodud 10 heuristikat [23]. Lõputöö käigus kasutatakse neid heuristikaid testimisel vastavalt „Madalapalgaliste toetus“ projekti nõuetele. Kuna projekt koosneb vaid ühest vaatest, siis ei ole antud projekti puhul võimalik kontrollida kõigile kasutajakogemuse printsiipidele vastavust. Valitakse välja kindel arv printsiipe, millele vastavust kontrollitakse projekti nõudeid arvestades. Vastavalt sobivale heuristikale luuakse testide grupid. Igale vähegi võimalikule testi grupile luuakse testid.

Lõputöö peatükis 2 võrreldakse kahte põhilist testimismeetodit: manuaalset ja automatiseeritud testimist. Välja tuuakse mõlema meetodi eelised ja lähtudes kriteeriumitest, mis on testitava projekti puhul olulisemad, valitakse välja üks testimismeetod. Kui meetod rakenduse testimiseks saab valitud, kirjeldatakse seda detailsemalt ning vaadeldakse abivahendeid, mida kasutada valitud metoodika puhul.

Lõputöö autor kasutab „Madalapalgaliste toetus“ rakenduses oma isiklikku vaadet (s.t logib sisse oma isikukoodiga), mis võib teiste isikute vaadetest erineda, kuna toetuse määramisel lähtutakse erinevatest tingimustest, mis sõltuvad konkreetsest isikust.

Kui rakenduse testid on loodud, analüüsib töö autor testimise käigus saadud tulemusi. Tehakse otsus, kas saadud tulemused vastavad oodatud tulemustele. Oodatud tulemus tähendab, et testitav kriteerium vastab projekti nõuetele. Kui tulemus on õige ehk vastab projekti dokumentatsioonis kirjeldatud nõuetele, analüüsitakse selle vastavust töös kirjeldatud kasutajakogemuse printsiipidele. Juhul kui tegelik ja oodatud tulemus on erinevad, siis tehakse vastav analüüs tulemuse parandamiseks. Kui selgub, et testi tulemus vastab „Madalapalgaliste toetus“ projekti nõuetele, kuid ei ole korrektne kasutajakogemuse poolest, siis tehakse analüüsi osas vastav ettepanek rakenduse parendamiseks.

Olenevalt sellest, kas rakenduse testimiseks saab valitud manuaalne või automatiseeritud testimine, tehakse bakalaureusetöö tegemise käigus saadud kogemusest lähtuvalt analüüs. Analüüsi osas tehakse kindlaks, milline peab olema testitav tarkvara, projekti nõuded ja ka testija oskused selleks, et kasutada kas üht või teist väljatoodud meetodikat kasutajaliidese kvaliteetseks testimiseks.

2 Tarkvara kasutatavus

Üldises mõistes on kasutatavus (*usability*) kvaliteedi atribuut, mis määrab seda, kui lihtne on midagi kasutada. Mõistele detailsemalt lähenedes hinnatakse kasutatavuse abil seda, kui kiirelt õpivad inimesed midagi kasutama ning kui edukad nad seda kasutades on. Samuti uurib kasutatavus vigade tekkimise tõenäosust ning inimeste arvu, kellele meeldib uuritavat objekti kasutada. [21] Vastavalt ISO (*International Organization for Standardization*) 9241-11 standardile on kasutatavus aste, milles toode võib olla kasutatud kindla isiku poolt selleks, et saavutada teatud eesmäärke efektiivselt, tulemuslikult ning saavutada rahulolu [5].

Tänapäeval on inimestel suured ootused veebilehtedele ja nad muutuvad aina vähem tolerantsemateks halva veebidisaini suhtes [21]. Kasutatavus on omadus, mida paljud tooted omavad, kuid veel rohkem esineb kasutatavuse puudujääke. Õnneks eksisteerib hulk meetodeid, mida on võimalik rakendada disaini kasutatavuse hindamiseks. Need meetodid aitavad tuua välja olulisemaid puudusi, mida tuleks parandada rakenduse disainis selleks, et edukalt tööturul püsida ja areneda. [25]

2.1 Kasutajakogemuse printsiibid lähtudes Jakob Nielsen heuristikatest

Kasutajakogemus on vastavalt ISO 9241-110:2010 standardile (klausel 2.15) isiku tajuga ja vastused, mis tekivad mingi toote, süsteemi või teenuse kasutamise tulemusel [6].

Luues kasutajakogemust, tuleb kõigepealt mõelda sellele, mis teeks tulevaste tarkvara kasutajate elu kasvõi mõnevõrra lihtsamaks ja aegasäästvamaks. Kasutajakogemuse loomisel tuleks alustuseks välja selgitada, millist disaini kasutaja tegelikult vajab. Ülejäänud tuleb juba väga kergelt välja, kuna õnnelik kasutaja on lojaalne kasutaja. [19]

Üheks mitteformaalseks meetodiks kasutajakogemuse analüüsimisel on heuristikate hindamine. Antud meetod seisneb selles, et mingit kokkulepitud arvu kasutajaliidese hindajaid palutakse tutvuda väljatöötatud disainiga ja anda selle kohta tagasisidet. Ideaalne olukord oleks siis, kui kasutajaliidesele hinnangut andes lähtuksid inimesed

tüüpilistest disaini reeglitest. Reaalses elus hindab aga enamik neist heuristikaid oma isiklikust intuitsioonist ja tervest mõistusest lähtudes. [22]

Aastal 1995 tõi Jakob Nielsen välja 10 kasutatavuse printsiipi. Neid printsiipe nimetatakse heuristikateks, sest nad kirjeldavad üldisi kasutusmugavuse reegleid, mitte konkreetseid kasutusmugavuse juhiseid [23]. Jakob Nielseni heuristikad võivad aidata kaasa kasutajamugavuse reeglitest ja printsiipidest paremini arusaamisele ja lõppkasutaja paremini tundmaõppimisele. Järgnevalt loendatakse eelmainitud printsiipe.

Tabel 1. Jakob Nielseni kümme kasutatavuse heuristikat [24].

Heuristika <i>id</i>	Heuristika nimetus
H1	Süsteemi olek on kasutajale alati arusaadav
H2	Seos reaalse maailmaga
H3	Kasutaja kontroll ja vabadus
H4	Kooskõla ja standardid
H5	Vigade ennetamine
H6	Tundmine VS meeldejätmise
H7	Kasutatavuse paindlikkus ja efektiivsus
H8	Disaini esteetika ja minimalism
H9	Vigade äratundmine ja taastus
H10	Abi ja dokumentatsioon

Järgnevalt on iga tabelis olev heuristika detailsemalt lahti kirjeldatud. Heuristika nimetus ja kirjeldus on seotud heuristika *id* kaudu.

H1: Esimese heuristika järgi peab kasutaja aru saama, mis olekus süsteem parasjagu on. Kasutaja peab kergesti aru saama, milline informatsioon, funktsionaalsus või elemendid on antud hetkel kättesaadavad ja millised mitte. Heuristika on täidetud siis, kui on selge, kuhu saab käesolevasest kohast edasi liikuda. Kõik elemendid on hästi eristatavad ja nende eesmärk on kergesti mõistetav. [23]

Antud heuristika alla võib kategoriseerida ka nuppe. Kasutaja peab aru saama, et nupp on nupp ja mis olekus see parasjagu on. Nupul peaks olema vähemalt 3 olekut, et kasutaja saaks aru, et vaatluse all olev element on tõesti nupp. Kasutaja saab halva kogemuse siis, kui ta liigub arvutihiirega nupu peale ja selle olek ei muutu. Juhul kui

nupu stiil ei muutu, siis veendumaks, et vaatluse all olev element on nupp, peab kasutaja selle peale vajutama. Sellist olukorda ei tohiks juhtuda ehk kasutaja ei tohiks nupule vajutada ainult selleks, et veenduda, et see on tõesti nupp. Järgnevalt kirjeldatakse nuppude jaoks erinevaid olekuid. [2]

Esimene olek on vaikimisi olek, millal nupuga ei juhtu midagi. Üks põhiline reegel siin on see, et nupud peavad välja nägema nagu nupud. Kasutaja peab olema võimeline vähese vaevaga nuppe üles leidma. Teiseks olekuks on fokuseeritud olek. Väga heaks praktikaks on muuta nupu stiili siis, kui selle peal asub arvutihiir. Sellisel juhul saab kasutaja kohe aru, et antud element on tõesti nupp ja selles veendumiseks ei pea nupule peale klikkma. Kolmandaks olekuks on elemendi aktiivsus. See tähendab, et nupu peale on vajutatud. Sel juhul omab nupp omakorda uut stiili. [2]

Lähtudes Jakob Nielsen-i heuristikatest, kirjeldas ka Thresa Neil aastal 1999 oma vaatepunkte. Antud osas on oluline, et süsteemi olekud oleksid arusaavad mingi tegevuse täitmisel. Kasutaja peab aru saama, millal tegevus ei ole alustatud, millal tegevus on parasjagu toimumas ja millal ta on lõppenud. [20]

H2: Kasutajaliidese struktuur peab vastama kasutaja eesmärgile ning olema kasutajale arusaadav. Info peab olema loogiliselt paigutatud. Samuti peab kasutajaliidese “keel” olema arusaadav ehk see peaks kasutama inimkeelt. Tuleb järgida reaalse maailma sõnastusi, et saada informatsiooni selle kohta, milline peaks olema loomulik ja loogiline käsk. [23]

H3: Kolmandast heuristikast lähtudes peab kasutajal alati olema võimalus minna tagasi esilehele. Esilehelt saab ligi kõikidele peamistele veebilehe kategooriatele. Tihti peale valivad kasutajad ekslikult funktsioone, mida nad tegelikult valida ei soovinud. Selleks peab kasutajal olema võimalus olukorra parandamiseks ehk peab olema võimalik liikuda eelmisele lehele. [23]

H4: Kasutatavad elemendid, nende paigutus ja nimetused peavad olema terves kasutajaliidesees kooskõlastatud. Kogu veebilehe funktsionaalsus peab olema kõikides enamlevinud veebilehitsejates kasutatav ning ka kasutajaliidese disainis ei tohi esineda suuri erinevusi. Veebilehel asuvate linkide tekstid peavad vastama reaalsele veebilehele, kuhu nad edasi viivad. [23]

H5: Vigade ennetamine on olulisem kui hea ja ilus veateate stiil. Vigu tekitavate olukordade vältimiseks tuleks enne tegevuse kinnitamist tagada kasutajale vajalik informatsioon ning veenduda, et ta tõesti soovib konkreetset tegevust kinnitada. [23]

Veebileht peab sisaldama selliseid elemente nagu juhendid, Korduma Kippuvad Küsimused ja muu abi, kuid veebilehe disain peab olema selline, et kasutajal oleks minimaalne vajadus kirjeldatud elementide kasutamiseks. Kui veebilehel on olemas vormid, siis nad peavad kasutajat informeerima sellest, mida peaks vormi sisestama. Juhul kui mingi tegevus sisaldab palju alamtegevusi, siis selleks peab olema loodud juhend. [13]

H6: Kasutaja ei pea veebilehe kasutamiseks elemente meelde jätma. Veebilehel navigeerumine peab toimuma loomulikult, kasutaja tunde järgi. [23]

H7: Veebilehe põhilised funktsioonid ja menüüpunktid peavad olema kättesaadavad enamike ekraani suurustega. Lehe enimkasutatavate funktsioonide jaoks on loodud kiirklahvid vms. [23]

H8: Rakenduse visuaalne osa peab olema minimalistlik, et kasutajal oleks lihtne iga elemendi rollist aru saada ning veebilehel navigeerida. Selleks, et disain oleks esteetiline, peaks komponentide vahele jääma ruumi, s.t iga komponendi jaoks on oma koht. Samuti peavad veebilehe kõige tähtsamad osad olema esilehelt kättesaadavad. [23]

H9: Kuvatavad veateated peaksid olema kirjeldatud puhtas ja arusaadavas keeles, kus puudub kood. Veateade peab seletama probleemi olemust ja pakkuma tekkinud probleemile konstruktiivset lahendust. [23]

H10: On väga professionaalne, kui süsteemi saab kasutada dokumentatsiooni abita. Sellegipoolest võib kasutajal tekkida abivajadus, mistõttu tuleb tagada abikeskuse ja dokumentatsiooni olemasolu. Abi peab olema kergesti kättesaadav ning võimalikult lihtne ja lühike. [23] Hea komme on abinippide kasutamine, et kasutajal oleks alati lihtne leida vajalikku informatsiooni mingi konkreetse elemendi kohta [20].

2.2 Tarkvara kasutatavus erinevate ekraanisuurustega

Lähtuvalt *W3Schools* veebilehe jaanuaris 2017 läbiviidud uuringust, on ligikaudu 95% *W3Schools*'i kasutajatel arvutiekraani resolutsiooniks 1024x768px või kõrgem. Kõige populaarsem nendest on alates 2012. aastast olnud resolutsioon 1366x768px [31].

Kasutajate jaoks on väga ebameeldivaks kogemuseks näiteks nuppude joonduse puudumine või poolikud tekstid. Kasutajatele meeldiva ja positiivse kogemuse tagamiseks on soovitatav rakendada kasutajaliidese testimist nutiseadmetes, sealhulgas nutitelefonides. Testija jaoks on oluline olla kindel, et testitav rakendus vastab sellistele ootustele nagu värvide stiili ühtne kasutus, ikoonide stiil ja värv ning protsessi indikaatorid veebilehe laadimisel. [17]

Kasutajaliides peab nägema ilus ja korrektne välja võimalikult paljudes seadmetes. Enamasti testitakse veebilehte kõige enamkasutatavate ekraanisuurustega veendumaks, et veebileht on kasutuskõlblik. Järgmisena tuuakse välja standardsed ekraanide resolutsioonid: 640x480px, 800x600px, 1024x768px, 1280x800px, 1366x768px, 1400x900px, 1680x1050px. [17]

Kasutajakogemust, mis on kujunenud rakenduse kasutamisest nutitefonis, kirjeldab hästi alltoodud joonis (Joonis 1).



The Mobile Equation.

Joonis 1. Nutitelefoni võrrand [18].

Hea kasutajakogemuse saamine kujuneb mitmest faktorist: raadiomastist, seadmest endast, seadme operatsioonisüsteemist ja ekraani suurusest. Mida paremini sobivad need komponendid koostöök, seda parem on kasutajakogemus nutitefonis. [18]

Tegelikkuses veedavad inimesed liiga palju aega töölaua taga. Tavaliselt kasutab inimene töölauda siis, kui ta asub kodus, kohvikus, lennukis või kuskil mujal, kus saab istuda ja kulutada mitu minutit selleks, et avada ja laadida oma arvutit. Sel hetkel on

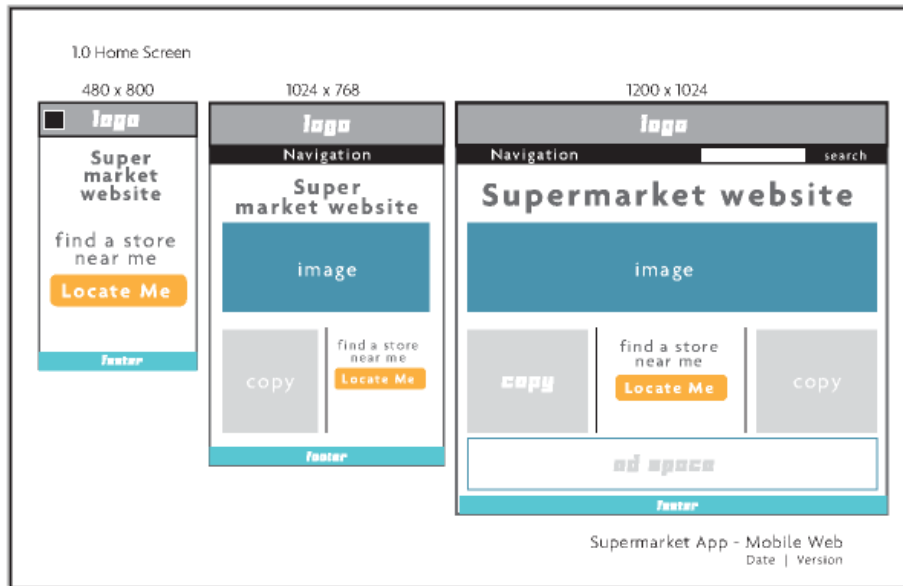
inimene ekraanile keskendunud. Nutitelefone kasutavad inimesed tavaliselt lühema ajaperioodi vältel. Keskmine nutitelefoni kasutamise aeg kestab 17 minutit, töölaua kasutamise aeg aga 39 minutit. Kui inimesed kasutavad oma nutitelefone, on nad keskendunud oma vestlustele, ootavad oma planeeritud kohtumisi jne. Sellel hetkel nad „asuvad” teises keskkonnas. Seega on oluline arvestada taoliste lühiajaliste kogemusega ja luua disaini, mis neid kogemusi mugavamaks muudab. [18]

2.3 Kohalduva disaini põhimõtted

Viimaste aastate jooksul on mobiiliseadmete kasutajate arv suurenenud eksponentsiaalselt. Tänapäeval saab veebilehti kasutada nii arvutiekraanil kui ka telefonis või tahvelarvutis. Nutitelefoni kasutajate arv hakkab tulevikus veelgi rohkem kasvama. [7] Antud faktist järeldub, et mida rohkem digitaalseid seadmeid tööturule jõuab, seda rohkem võimalusi on veebilehtede kasutamiseks. Veebilehti on võimalik vaadata erinevates seadmetes, brauserites ja erinevate operatsioonisüsteemidega.

Aina rohkem võetakse kasutusele erinevaid digitaalseid seadmeid. Selline olukord paneb disainereid leiutama uusi viise selleks, et luua veebilehe disain nutitelefoni vaate jaoks. Aastal 2010 nimetas Ethan Marcotte disaini, mis näeb erinevates seadmetes korrektne välja, kohalduvaks veebidisainiks (*responsive web design*). Selline veebidisain aitab veebilehel kohanduda erinevate ekraani suurustega. Olenevalt sellest, kas ekraan on suurem või väiksem, jääb veebileht kasutatavaks ilma spetsiaalsele domeenile ümberlülitumiseta. [7]

Kohalduva veebidisaini loomisel tuleb tegeleda niinimetatud murdepunktidega. Murdepunkti järgi saab otsustada, millisest ekraani laiuusest alates hakkab veebileht üht või teist disainilahendust kasutama. Järgnev joonis (Joonis 2) illustreerib kohanduva disaini olemust. On näha, et ekraanil, mille resolutsiooniks on 1200x1040px, on rohkem elemente kui ekraanidel resolutsiooniga 1024x786px ja 380x800px. Alljärgneva joonise (Joonis 2) põhiline eesmärk on demonstreerida seost erinevate ekraanilaiuste vahel. [18]



Sample Responsive Design Wireframe Sheet.

Joonis 2. Kohalduva disaini näide [18].

Joonis 2 demonstreerib, et mida väiksema ekraani suurusega on tegemist, seda vähem elemente on lehel vaikumisi nähtavad. Veebilehe elementidel kas muutub kujundus või on nad menüü taga peidus ja on sellest samast menüüst kättesaadavad.

Hea kasutajakogemus töölauda jaoks eeldab, et kõik veebilehe võimalused peavad olema kättesaadavad suure veebilehe kaardina ning mida rohkem võimalusi mahub ühele lehele, seda parem. Nutitelefoniga kasutajakogemus loob aga ühtlase, lineaarse loo. Mida rohkem ekraane kasutaja külastab, seda parem mõju avaldub nutitelefoniga kogemusele. Kõikide võimaluste mahutamine ühele ekraanile asendatakse mobiilis kõigi vajalike osadega. [18]

2.4 Tarkvara kasutatavuse testimine

Rääkides tarkvara kasutatavusest, on oluline mainida sellist mahukat ja aktuaalset tarkvaraarendusprotsessi osa nagu testimine. Tarkvara testimise eesmärgiks on tarkvaras esinevate defektide avastamine. Testimine on tarkvara kvaliteedi mõõdik. Selle käigus mõõdetakse, kui hästi on saavutatud faktorid nagu näiteks korrektsus, usaldusväärsus, kasutusmugavus, taaskasutamise võimalus ja testitavus. [12] Kõik loendatud kriteeriumid mõõdavad tarkvara kasutatavust.

Tarkvara testimine kujutab endast vahendit, mille abil väheneb tarkvaras esinevate vigade ja hoolduste arv. Samuti vähendab testimine üldiseid kulutusi, mida tarkvarale

suunatakse. Testimine on muutunud üheks olulisemaks parameetriks tarkvaraarenduse elutsükliks. [12] Just defektide avastamine varasemas tarkvaraarenduse etapis soodustab selle kõrget kvaliteeti, mille tulemusena paraneb ka kasutajakogemus. Igasuguse testimise rakendamine panustab tarkvara kasutatavusse, kuna testija mängib lõppkasutaja rolli ning avastab vigu ise, enne kui kasutaja on veaga kokku puutunud.

2.4.1 Manuaalne ja automatiseeritud tarkvara kasutatavuse testimine

On olemas kaks põhilist võimalust tarkvara testimiseks: manuaalne ja automatiseeritud testimine. Järgnevalt analüüsitakse mõlemat võimalust ja tulemustest lähtudes valitakse sobivam meetod “Madalapalgaliste toetus” rakenduse testimiseks.

Manuaalset testimist viivad läbi testijad. Nad testivad tarkvara erinevate defektide avastamiseks. Antud juhul on nõutud, et testija mängiks lõppkasutaja rolli ja prooviks kasutada võimalikult palju tarkvara funktsionaalsust veendumaks, et tarkvara käitub õigesti. Testijad järgivad kindlat plaani, kus on välja toodud olulisemad testjuhtumid. Manuaalse testimise põhiliseks puuduseks on suur ajakulu. Samuti ei ole antud protsessi võimalik korduvalt kasutada, puudub kindel stsenaarium. Manuaalse testimise puhul nähakse palju vaeva, kuid mõned vead jäävad tavaliselt siiski katmata. [12]

Manuaalne testimine on aeganõudev protsess, sest vajab inimese sekkumist testimisse [8]. Testimise automatiseerimine lahendab probleeme, mis tavaliselt tekivad manuaalse testimise puhul. Kasutades antud metoodikat käivitab testija koodi ja sellega saab testimine lõpetatuks. Automaattestimise vahendid annavad arendajatele võimaluse vähese vaevaga automatiseerida kogu testimisprotsessi tarkvara loomise käigus. Efektiivne testimine garanteerib tarkvara kõrge kvaliteedi. [12]

2.4.2 Manuaalse ja automatiseeritud testimise võrdluse põhjal valiku tegemine

Võrreldes manuaalse testimisega, loetakse automatiseeritud testimist efektiivsemaks ja usaldusväärsemaks meetodiks tarkvara kvaliteedi ja usaldusväarsuse tõstmisel [9]. Eksisteerib mitmeid arvamusi sellest, kumb testimismeetod on parem: kas manuaalne või automatiseeritud. Selleks, et erinevaid teooriaid kontrollida ja leida kõige sobivam testimismeetod „Madalapalgaliste toetus“ rakenduse jaoks, teeb autor ise uuringu. Kasutades erinevaid allikaid, on tabelis Tabel 2 välja toodud autori arvates olulisemad kriteeriumid, mille järgi tuleb testimismeetodit valida.

Tabel 2. Manuaalse ja automatiseeritud testimise võrdlus [1, 9, 12, 14].

Võrdlemiskriteerium	Manuaalne testimine	Automaattestimine
Võimalik viia läbi palju teste väiksema ajakuluga		+
Testi taaskasutamine		+
Kõrgem usaldusväarsus		+
Efektiivne süsteemi osale, mis on aldis vigadele	+	
Lühiajalised kulud on väiksemad	+	
Suurem tõenäosus kasutaja probleemide leidmiseks	+	
Paindlikkus	+	

Toetudes eelnevale automaat- ja manuaaltestimise võrdlusele, valiti rakenduse testimiseks automaattestid. Kuigi manuaalsel testimisel on palju eeliseid, sobib käesoleva projekti puhul siiski kasutada automatiseeritud testimist. Selle põhjuseks on keerulise funktsionaalsuse puudumine autori rakenduse individuaalses vaates. Projekti nõuetes on kirjeldatud kindlad reeglid, millele rakenduse kasutajaliides peab vastama. Autori eesmärgiks on kontrollida kehtestatud nõuete täitvust ja vastavust lõputöös väljatoodud kasutajakogemuse reeglitele. Kuna teste luuakse rangelt nõuete järgi, siis kirjeldatud eesmärkide täitmise kontrollimiseks sobib automaattestimine paremini kui manuaalne testimine.

2.5 Automaattestide loomiseks vahendi valimine

Lõputöö käigus selgitatakse välja, millist vahendit on kõige otstarbekam kasutada automaattestide loomiseks kasutajaliidese jaoks. Vaatluse alla sai valitud kaks vahendit: *Selenium Webdriver* ja *Selenide*. *Selenium Webdriver* on loodud tarkvara testimise vahendi *Selenium* põhjal. [27] *Selenium IDE* on tasuta, vaba lähtekoodiga plugin Firefox veebilehitsejale. Põhiliselt kasutatakse seda pluginat veebilehtede testimise automatiseerimiseks. [12]

2.5.1 Automaattestide loomise vahendite võrdlus

Aastal 2007 töötas Simon Stewart *Selenium*'i abil välja brauseri automatiseerimisvahendi, mille nimeks sai *WebDriver*. Hetkel on selle nimeks *Selenium WebDriver* ehk *Selenium 2.0*. [27] *Selenium WebDriver* on enimlevinud kasutajaliidese testimise vahend Java programmeerimiskeeles. Antud vahend sobib suurepäraselt kasutajaliidese testimiseks, sellegipoolest oleks seda võimalik paremaks muuta [29].

Selenide raamistik on loodud ettevõtte Codeborne poolt *Selenium WebDriver*'i põhjal. Kasutades *Selenium WebDriver*'it, tundsid firma töötajad puudust testide stabiilsusest ja ilmekusest. Samuti taheti võtta midagi ette selleks, et lahendada ajanihe (*timeout*) ja *AJAX* kasutajatoega seotud probleeme [29]. Nendel põhjustel sai Codeborne'i poolt loodud *Selenium*'i raamistik.

Järgnevalt tuuakse välja *Selenium WebDriver*'i ja *Selenide*'i võrdlus koos koodinäidetega (Tabel 3). Võrdlus on esitatud tabelina selleks, et info oleks visuaalselt väljatoodud ning automaattestide loomiseks vahendi valiku tegemine oleks lihtsam.

Tabel 3. *Selenium WebDriver* ja *Selenide* võrdlus [26].

Tegevus	Selenium WebDriver	Selenide
Brauseri „loomine” ja selle keerukus	Suhteliselt keerukas, võtab umbes viis rida koodi. <pre>„DesiredCapabilities desiredCapabilities = DesiredCapabilities.htmlUnit(); desiredCapabilities.setCapability(HtmlUnitDriver.INVALIDSELECTIONERROR, true); desiredCapabilities.setCapability(HtmlUnitDriver.INVALIDXPathError, false); desiredCapabilities.setJavaScriptEnabled(true); WebDriver driver = new HtmlUnitDriver(desiredCapabilities);“</pre>	Lihtne, kuna koosneb ühest reast ja sisaldab vaid ühte funktsiooni. <pre>„open("/my-application/login");“</pre>
<i>AJAX</i> 'i toetamine	Pikk ja keeruline seadistus. Võtab umbes 12 rida koodi.	Seadistab end ise.
Elementi otsimine teksti järgi	Ei võimalda.	Võimaldab väga lühidalt.

Tegevus	Selenium Webdriver	Selenide
Elementide otsimine	Peab detailselt lahti kirjutama. <pre>„WebElement customer = driver.findElement(By.id("customerContainer"));“</pre>	Võimaldab nii jQuery elementide kasutamist kui ka pikema teksti järgi otsimist. Näiteks: <pre>„WebElement customer = \$("#customerContainer"); „</pre>
Silumine (<i>debugging</i>)	Pikk ja keerukas, kuna tuleb välja printida kõiki elemendi atribuute. <pre>„WebElement element = driver.findElement(By.id("customerContainer")); System.out.println("tag: " + element.getTag()); System.out.println("text: " + element.getText()); System.out.println("id: " + element.getAttribute("id")); System.out.println("name: " + element.getAttribute("nam e")); System.out.println("class : " + element.getAttribute("cla ss")); System.out.println("value : " + element.getAttribute("val ue")); System.out.println("visib le: " + element.isDisplayed());“</pre>	Saab teada printides välja ainult ühe <i>id</i> . <pre>„System.out.println(\$("#customerContaine r"));“</pre>

Tabelis kirjeldatud informatsioonist lähtudes valitakse “Madalapalgaliste toetus” rakenduse kasutajaliidese testimiseks sobivam vahend. Vahendi valik ja selle põhjendus on väljatoodud osas 2.5.2.

2.5.2 Automaattestide loomise vahendi valik ja põhjendus

Selenide pakub lihtsa ja elegantse lahenduse selliste tüüpiliste testimisega seotud probleemidega nagu näiteks AJAX, mille poolest erineb *Selenium Webdriver*’ist. Antud

vahendiga on lihtne testimist alustada, sest raamistikku on väga kerge kasutada. Seega pole sellega hakkamasaamiseks vaja uurida mahukaid õpikuid. *Selenide*'iga saab keskenduda rakenduse testimisele ja äri loogika analüüsimisele, mitte süntaksi uurimisele. [29]

Automaattestid, mis on kirjutatud *Selenide*'i abil, saavad ise aru, millal on vaja veebilehitsejat avada ja millal sulgeda. Meetodid, mida *Selenide* pakub, on hästi loetavad ja kergelt arusaavad isegi tavainimesele, kes pole infotehnoloogiaga lähedalt seotud, kuna antud raamistiku meetodite nimetamisel on kasutatud inimkeelt. [3]

Autor peab väga oluliseks ja mugavaks *AJAX*'i toetamist *Selenide*'i poolt, mistõttu ei pea testija ise tegema analüüsi töö ajahinnangute jaoks. Sageli on tähtis teada, mis aja pärast muutub mingi kindla elemendi olek. Vahendite puhul, mis ei toeta *AJAX*'it, peab tesija ise määrama ajavahemikud, mille tagant tegevusi käivitatakse. *Selenide* aga ootab mingi kindla ajavahemiku ise ära, juhul kui tegevus ei ole koheselt täidetud. Vaikimisi ootamise aeg *Selenide*'il on kuni 4 sekundit. [3]

Järgmiseks heaks võimaluseks, mida *Selenide* pakub, on automaatne ekraanipiltide tegemine. Ekraanipiltide tegemist saab seadistada vastavalt sellele, kas pilte soovitakse teha siis, kui test on läbikukutud või läbitud. Võimalik on kasutada ka mõlemat võimalust. [3] Kõige paremini kirjeldab *Selenide*'i raamistikku Andrei Solntsev-i arvamuse avaldamine, et halval tarkvaral ei ole dokumentatsiooni, aga silmapaistvale tarkvarale ei ole dokumentatsiooni vajagi [29].

Uuringu käigus leiti, et käesoleva rakenduse testimiseks on paremaks vahendiks *Selenide*. Koodinäidetest saab järeldada, et *Selenide* võimaldab vähese vaevaga kirjutada kvaliteetseid teste. Mida kiiremini saavad automaattestid tehtud, seda parem, kuna eesmärgiks on teha testimise protsessi võimalikult efektiivseks ja usaldusväärseks.

3 Tarkvara kasutajaliidese testimine

Automaatsete loomise vahend *Selenide* sai valitud lähtudes projekti üldistest nõuetest. See on sobiv vahend selleks, et luua kiiresti lihtsaid ja arusaadavaid teste. Automaatsete kasutamisel tuleb arvestada mitte ainult testide käivitamiseks ja tulemuste saamiseks kulutatud aega, vaid ka aega, mis kulub testijal testi enda loomisele. Kasutades *Selenide* raamistikku, kulub autori arvates automaatsete loomisele ja nende käivitamisele lõppkokkuvõttes vähem aega, kui elementide stiilide kontrollimisele manuaalselt.

Antud lõputöö raames ei looda keerulisi funktsionaalsuse teste, vaid põhiliseks nõudeks on kasutajaliidese elementide disaini vastavus projekti nõuetele, mis on detailsemalt lahti kirjutatud iga testigrupi juures.

3.1 Testitava rakenduse kirjeldus testide gruppide loomiseks

Käesolevas lõputöös testitakse ja analüüsitakse Maksu- ja Tolliameti „Madalapalgaliste toetus“ rakenduse kasutajaliidest. Rakendust on võimalik avada mitmes vaates olenevalt sellest, kas isikul on täidetud tingimused toetuse saamiseks. Lõputöö autor kasutab testimiseks oma isiklikku rakenduse vaadet ning antud juhul on rakenduses üks leht, millest on võimalik edasi liikuda vaid välislinkide kaudu. Ülejäänud rakenduse funktsionaalsus jääb ühele ja samale lehele ning selle URL (*Uniform Resource Locator*) ei muutu.

Selleks, et esitada taotlust toetuse saamiseks, peavad olema täidetud nõutud tingimused. Allkirjeldatud tingimused peavad olema täidetud vähemalt viiel kalendrikuul. [16]

Esimeseks tingimuseks toetuse saamiseks on Eesti residentsus. Teiseks tingimuseks on täisealisus ehk isik peab olema vähemalt 18-aastane. Kolmas tingimus on: ühe või rohkema töölepingu alusel töö- või teenistussuhe kogu kalendrikuu vältel ilma töötamise peatamiseta. Neljandaks tingimuseks on täistööajaga töötamine. Siinkohal on oluline täistööajaga töötamine vähemalt ühe tööandja juures või töötamine mitme tööandja juures kokku täistööajaga. Käesolev tingimus ei ole kehtiv

töövõimetuspensionäride ja vähenenud töövõimega isikute suhtes. Toetuse taotlemiseks peab isiku tulu jääma alla 7817 euro aastas. [16]

Antud toetuse saamisel on ka teisi nüansse, kuid käesoleva lõputöö raames neid ei käsitleta. Selleks, et luua parem ettekujutus rakendusest, esitab autor rakenduse joonise (Joonis 3), milles on kujutatud tema isiklik „Madalalpalgaliste toetus“ rakenduse vaade.

Tagasi e-maksuameti/e-tolli lehele

Kasutaja:
Oleksandra Chmil

In English
Ha русском

Väljun ✕

Madalalpalgaliste toetuse taotluse esitamine

Sellel lehel saate taotleda madalalpalgaliste toetust. Kui kõik toetuse taotlemise tingimused on täidetud ning soovite toetust taotleda, siis lisage allolevale vormile puuduvad andmed ning esitage taotlus. Taotlust saab esitada kuni 30.03.2017. Kui kõik tingimused ei ole täidetud, siis ei ole Teil õigust toetust saada.

Tingimused

Tingimus

- Tuludeklaratsiooni esitamine**
Olete esitanud 2016. aasta füüsilise isiku tuludeklaratsiooni õigeaegselt 30.03.2017.
- 2016. aasta kogutulu ei ole suurem kui 7817 eurot**
Olete 2016. aastal saanud tulu alla kehtestatud piirmäära 7817 eurot.
- Muud tingimused (residentsus, vanus, töö- või teenistussuhte olemasolu, töötamine täistööajaga)**
Tingimused ei ole täidetud.

[Vaatan lähemalt](#)

Joonis 3. Rakenduse esmase avanemise vaade.

Lehe päis sisaldab selliseid elemente nagu:

- Lingid: antud jaotuse hulka kuuluvad lingid nimedega „Tagasi e-maksuameti/e-tolli lehele”, “In English” ja „Ha русском” (“In English” ja „Ha русском” edaspidi keelevaliku lingid)
- Kasutaja nimi ja perenimi koos märgisega
- Väljumise nupp „Väljun”

Rakenduse sisuala sisaldab järgmisi elemente:

- Pealkiri „Madalalpalgaliste toetuse taotluse esitamine”

- Seletav tekst antud lehe kohta, mis annab kasutajale ettekujutuse rakendusest
- Lühikesed juhised toetuse saamiseks, mille juures on erinevate värvidega ikoonid, kus iga ikoon annab aimu tingimuse täitvuse kohta
- Link „Vaatan lähemalt”, mille abil saab avada tabelit, kus on lahti seletatud iga tingimus kuude kaupa (Joonis 4). Tingimuse täitvust iseloomustavad ikoonid, kus roheline ikoon tähendab, et tingimus on täidetud ja punane ikoon tähendab, et tingimus ei ole täidetud.

[Vaatan lähemalt](#)

Need tingimused peavad olema 2016. aasta jooksul vähemalt **kuuel kuul üheaegselt** täidetud.

Tingimus	Ja	Feb	Märts	Apr	Mai	Juuni	Juuli	Aug	Sept	Okt	Nov	Dets
Eesti residentsus ⓘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Kehtiv töö- või teenistussuhe ⓘ	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!
Töötamine täistööajaga ⓘ	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!
Kokku	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!	!!

Joonis 4. Tingimused toetuse taotlemiseks: detailvaade.

- Tabeli sees: tingimuste juures asuvad ikoonid „i” märgiga, mis tähistab informatsiooni. Ikooni peale vajutades avaneb abitekst, mis seletab kasutaja tingimuse detailsemalt lahti
- Tabeli sees: tingimused kokku

Rakenduse jalus sisaldab järgmisi elemente:

- Maksu- ja Tolliameti logo
- Üldinfo, E-tolliteenuste klienditoe ja E-maksuteenuste klienditoe telefoninumbrid
- Link „Kõik kontaktandmed”, mis viib e-MTA lehele, kus on kontaktandmed detailsemalt väljatoodud

3.2 Testide gruppide loomine

Töö autor kasutab rakenduse testimiseks oma isiklikku rakenduse vaadet ning lähtuvalt projekti nõuetest analüüsib, milliseid elemente peab antud veebileht tema vaate puhul

sisaldama. Rakendust testitakse Chrome veebilehitseja viimase versiooniga järgmiste ekraanisuurustega:

- Nutitelefone vaade: 599x903px
- Sülearvuti või monitoriga arvuti vaade: 1200x903px

Olenevalt sellest, kas tingimus toetuse saamiseks on täidetud, kuvatakse kasutajale erinevat teksti. Seletava teksti juures asub ikoon, millel on vastavalt tingimuse täitvusele erinev värv.

Rakenduse detailseks testimiseks loob autor erinevaid testide gruppe, kus kirjeldab, millisele probleemile tahetakse lahendust või vastust saada. Testimise gruppide loomiseks ja rakenduse korrektsuse kontrollimiseks kombineerib autor kahte tüüpi nõudeid: Maksu- ja Tolliameti poolt kehtestatud nõudeid ning kirjeldatud kasutajakogemuse printsiipe. Bakalaureusetöö käigus kontrollitakse elementide stiile vaikimisi olekus ja olekuid, kui elemendi peal asub arvutihiir.

Lõputöö käigus luuakse erinevaid automaattestide tüüpe kasutajaliidese jaoks, demonstreerimaks testide loomise mitmekesisust. Viiakse läbi nelja tüüpi teste:

- Sõnastuse vastavus projekti nõuetele
- Elementide olekute ja stiilide kontroll
- Esilehele tagasiminemise võimaluse kontrollimine
- Kasutaja abi saamise võimaluse kontrollimine

3.2.1 Testide gruppide kirjeldamine arvuti vaate jaoks

Testide grupp 1: Toetuse saamise tingimuste kontrollimine

Esimene testide grupp luuakse lähtudes Jakob Nielsen'i seitsmendast heuristikast. „Madalalpalgaliste toetus“ rakenduse kasutajaliidese põhiliseks funktsiooniks on anda kasutajale infot toetuse saamise võimalusest. Seega võib info kuvamist toetuse tingimuste täitvuse kohta lugeda üheks tähtsamaks kasutajaliidese osaks. Seetõttu kontrollitakse eelkõige just teksti, mis ka vastab konkreetsele tingimusele või mitte. Testide grupi moodustamiseks on toodud välja projekti nõuded.

- 1) Kui tingimus on täidetud, kuvatakse ekraanil roheline staatuse ikoon.

- 2) Kui tingimuse täitmiseks on tarvis täiendavaid andmeid, kuvatakse kollane staatuse ikoon.
- 3) Kui nõutud tingimused toetuse saamiseks ei ole täidetud, siis kuvatakse ekraanil punane staatuse ikoon.

Esimene tingimus toetuse saamiseks: „Tuludeklaratsiooni esitamine“. Esimeseks toetuse saamise tingimuseks on tuludeklaratsiooni andmete esitamine. Selle järgi kontrollitakse, kas isiku sissetulek jääb määratud summa piiridesse, mille alusel oleks võimalik toetust taotleda. Lõputöö autor on aastal 2017 esitanud tuludeklaratsiooni, seega on esimene tingimus toetuse saamiseks täidetud. Veebileht peab sisaldama järgmist lauset: „Olete esitanud 2016. aasta füüsilise isiku tuludeklaratsiooni õigeaegselt“. Samuti peab käesoleva tingimuse juures asuma roheline ikoon.

Teine tingimus toetuse saamiseks: „2016. aasta kogutulu ei ole suurem kui 7817 eurot“. Autor ei ole 2016. aasta jooksul tööl käinud, seega tema aastane sissetulek ei ületa 7817 eurot. Tingimus ei ole täidetud ning veebileht peab sisaldama teksti „Olete 2016. aastal saanud tulu alla kehtestatud piirmäära 7817 eurot“. Antud tingimuse juures peab asuma punane ikoon.

Kolmas tingimus toetuse saamiseks: „Muud tingimused (residentsus, vanus, töö- või teenistussuhte olemasolu, töötamine täistööajaga)“. Lõputöö autor on Eesti resident, ta on vanem kui 18 aastat ning tal on olemas kehtiv töösuhe. Kuna autor ei ole 2016. aasta jooksul täistööajaga töötanud, siis kokkuvõttes on tal antud tingimus toetuse saamiseks täitmata. Arvestades kirjeldatud tingimusi, peab tabel sisaldama järgmist teksti: „Tingimused ei ole täidetud“ ja punast staatuse ikooni.

Testide grupp 2: Elementide olekute ja stiilide kontroll

Antud osas olevate testidega kontrollib autor, kas kasutajale on süsteemis olevate komponentide olekud arusaadavad. Testide läbiviimiseks lähtub autor Jakob Nielsen esimesest heuristikast. Siinkohal valitakse välja elemente, mille olekuid kontrollida.

Käesoleva veebilehe puhul saab autor automaatselt kontrollida, kas veebilehel olevad nupud ja lingid omavad iseloomustavaid olekuid. Nuppudel kontrollitakse kahte olekut: default ehk nupu vaikimisi stiil hetkel, mil sellega ei teostata mingeid tegevusi ja hover ehk nupu peal asub arvutihir.

Väljumise nupp. Esimesena kontrollib autor väljumise nupu vaikumisi värvi, milleks on rgba(255, 155, 155, 1). Seejärel luuakse automaattest, mis kontrollib, kas elemendi stiil muutub siis, kui kasutaja liigub selle peale arvutihiirega. Antud olekus on tellija poolt määratud nupu tausta värv: rgba(255, 155, 155, 1) ja nupu piirjoone värv: rgba(184, 68, 68, 1). Nupu sees oleva teksti värviks on rgba(255, 255, 555, 1) ehk valge.

Lingid. Järgmisteks elementideks valis autor lingid. Linkide suuruseks peab olema 14px. Kõikide linkide puhul on määratud reegel, et kui nende peal asub arvutihiir, siis nad ei tohi olla allajoonitud. Vaikumisi olekus peab linkide all olema tõmmatud joon.

Projekti „Madalapalgaliste toetus” nõuetes on päises asuvate linkide ehk lingi „Tagasi e-maksuameti/e-tolli lehele“ ning keelevaliku linkide („In English”, „На русском”) vaikumisi värviks määratud: rgba(93, 96, 113, 1). Kui lingi peal asub arvutihiir, siis selle värviks on rgba(0, 131, 186, 1). Linkide „Vaatan lähemalt“ ja „Kõik kontaktid“ värviks kõikides olekutes on rgba(255, 128, 0, 1).

Testide grupp 3: Esilehele tagasiminemise võimaluse kontrollimine

Antud osa sisaldab ainult ühe testi, millega kontrollitakse, kas kasutajal on alati võimalus minna tagasi esilehele. Rakenduses on esilehele tagasiminemiseks ettenähtud link tekstiga „Tagasi e-maksuameti/e-tolli lehele“. Kirjutatakse automaattest, mis kontrollib selle lingi kättesaadavust veebilehel.

Testide grupp 4: Abi saamise võimaluse kontrollimine

Selles osas loodud testid aitavad kontrollida, kas veebilehel on olemas veebielement selleks, et kasutajal oleks võimalik esitada oma muret või leida oma küsimusele vastust.

Abi saamiseks on projektis ettenähtud kolm telefoninumbrit ning üks link, mis suunab kasutajat e-MTA lehele, kus on saadaval kõik e-MTA kontaktandmed. Samuti on veebilehel tabelis tingimuse juures olemas ikoon, mille peale klikkides peab avanema abiinfo, kus on tingimus detailsemalt lahti seletatud.

Kontaktandmed: Telefoninumber ja üksuse nimi on veebilehel kuvatud tavalise tekstina. Tähtis on, et need andmed ei läheks kaduma. Seega luuakse automaattestid, mis kontrollivad nende andmete olemasolu. Kontaktandmete lingi puhul on samuti tähtis selle olemasolu veebilehel.

Abitekst: Juhul kui veebilehel on olemas seletatava tekstita ikoon, siis selle juures peab olema abitekst. Käesolevas projektis kasutatakse ikoone, mille sees on „i” täht, mis tähistab sõna „Informatsioon“. Sellise ikooni peale klikkides peab avanema abitekst. Abiteksti abil saab kasutaja aru, mida selle ikooni juures asuv tekst täpsemalt tähendab. Loodavate testidega kontrollitakse, kas ikooni peale vajutades avaneb abitekst. Antud nõuet võib samuti käsitleda kui üht Jakob Nielsen'i heuristikat.

Vastavalt projekti nõuetele, peab tingimusele „Eesti residentsus” vastama abitekst: „Tingimus on täidetud, kui olete olnud Eesti resident kogu kalendrikuu jooksul. Kui need andmed erinevad Teile teadaolevaist, siis pöörduge palun Maksu- ja Tolliameti poole residentsuse andmete muutmiseks.”.

Tingimus “Kehtiv töö- või teenistussuhe” peab omama abiteksti: „Tingimus on täidetud, kui Teie töö- või teenistussuhe on kehtinud kogu kalendrikuu ja töötamine ei ole peatunud Teie lapsehoolduspuhkuse, aja- ja asendusteenistuse või tasustamata puhkuse tõttu. Kui need andmed erinevad Teile teadaolevaist, siis pöörduge palun oma tööandja poole palvega andmed töötamise registris parandada.”.

Viimasele tingimusele ehk “Töötamine täistööajaga” peab vastama abitekst: „Tingimus on täidetud, kui olete töötanud täistööajaga. Tingimus on täidetud ka siis, kui vähenenud töövõimega töötaja eest maksab sotsiaalmaksu riik. Kui need andmed erinevad Teile teadaolevaist, siis pöörduge palun oma tööandja poole palvega andmed deklaratsioonil TSD parandada.”

3.2.2 Testimise tulemused ekraanisuurusega 1200x903px

Automaattestide loomise planeerimise käigus üritab autor luua võimalikult väikeseid teste. Et automaattesti tulemus oleks arusaadav, üritakse luua teste, mille tulemus vastab vaid ühele sisendile. Seega loob autor ühe veebilehe elemendi testimiseks mitu väiksemat automaattesti, mis kontrollivad elemendi vastavust nõuetele ühekaupa.

Käesolevas osas tuuakse välja automaattestid, mis on loodud lähtuvalt peatükis 3.2 kirjeldatud testimise gruppidest. Samuti analüüsib autor peatükis testimise käigus saadud tulemusi. Läbiviidud testide tulemusi kirjeldavad pildid, mille autor lisab koodijuppide alla. Kui testi nimetuse juures olev ikoon on rohelist värvi, tähendab see

seda, et testi tulemus on positiivne ehk test läks edukalt läbi. Kui aga ikoon on kollakas, siis test on läbikukkunud.

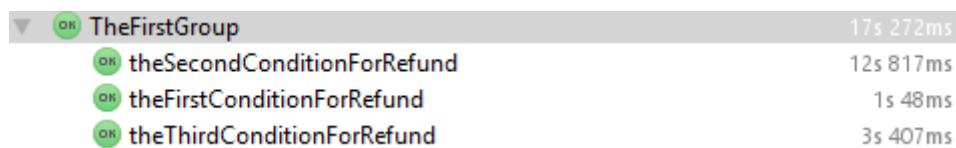
Juhul kui testi nimetuses sisaldub sõna „*Hover*“, tähendab see seda, et testiga kontrollitakse elemendi olekut siis, kui selle peal asub arvutihiir. Kui testi nimetus sisaldab sõna „*Default*“, siis kontrollitakse testiga elemendi vaikimisi olekut. Kõikidele testidele üritatakse anda lihtsad inglisekeelsed nimed, mis oleksid ka infotehnoloogiaga mitteseotud inimesele arusaadavad. Kui testi nimetuses sisaldub „*TextDecoration*“, siis tähendab see, et testitakse elemendi allajoonitust.

Testide grupp 1: Automaattestiga kontrolliti tekstide olemasolu ja korrektsust vastavalt sellele, millised tingimused toetuse saamiseks on autoril täidetud. Autor lõi automaatse, mille abil kontrollitakse, kas rakenduses kuvatav tekst vastab projekti nõuetes määratud tekstile. Vajalikke elemente otsitakse antud juhul teksti järgi ning seda demonstreerib joonis 5.

```
@Test
public void theFirstConditionForRefund() {
    $(getElement(new Selectors.ByText("Olete esitanud 2016. aasta
    füüsilise isiku tuludeklaratsiooni õigeaegselt
    30.03.2017."))).shouldBe(visible);
}
@Test
public void theSecondConditionForRefund() {
    $(getElement(new Selectors.ByText("Olete 2016. aastal saanud tulu
    alla kehtestatud piirmäära 7817 eurot."))).shouldBe(visible);
}
@Test
public void theThirdConditionForRefund() {
    $(getElement(new Selectors.ByText("Tingimused ei ole
    täidetud."))).shouldBe(visible);
}
```

Joonis 5. Esimese testide grupi testid.

Testide tulemusi kirjeldab järgmine joonis (Joonis 6).



OK	TheFirstGroup	17s 272ms
OK	theSecondConditionForRefund	12s 817ms
OK	theFirstConditionForRefund	1s 48ms
OK	theThirdConditionForRefund	3s 407ms

Joonis 6. Esimese testide grupi tulemused.

Sai loodud kolm testi, mis kontrollivad kolme tingimuste teksti. Testi tulemustest on näha, et antud projekti osa vastab projekti nõuetele ja seal asuv tekst on korrektne.

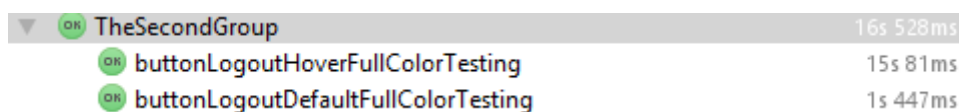
Testide grupp 2: Selles osas olevaid teste loodi lähtudes Nielsen'i heuristikatest ja projektis määratud nõuetest. Veebilehel kuvatavate komponentide olekute kontrollimiseks lõi autor teste, mis kontrollivad, kas mingi konkreetse tegevuse rakendamisega elemendile tema stiil muutub. Kui elemendi stiil muutub vastavalt projektis kirjeldatud nõuetele, siis test loetakse õnnestunuks.

“Väljun” nupp. Kõigepealt kirjutatakse automaattest, mille abil kontrollitakse, kas nupu vaikimisi stiil vastab dokumentatsioonis olevatele nõudmistele. Järgmiste testidega tehakse selgeks, kas nupu olekute stiilid on rakenduses korrektselt realiseeritud. Kuna nupp kasutab klassi nimega „*btn_logout*”, siis demonstreeritakse käesolevas osas *Selenide*-i võimalust kasutada elementide klasse (Joonis 7).

```
@Test
public void buttonLogoutDefaultFullColorTesting() {
    assert ($(".btn_logout").getCssValue("border-color").equals("rgb(255,
115, 115)"));
    assert ($(".btn_logout").getCssValue("background-
color").equals("rgba(255, 115, 115, 1)"));
    assert ($(".btn_logout").getCssValue("color").equals("rgba(255, 255,
255, 1)"));
}
@Test
public void buttonLogoutHoverFullColorTesting() {
    $(".btn_logout").hover();
    assert ($(".btn_logout").getCssValue("border-color").equals("rgb(184,
68, 68)"));
    assert ($(".btn_logout").getCssValue("background-
color").equals("rgba(255, 115, 115, 1)"));
    assert ($(".btn_logout").getCssValue("color").equals("rgba(255, 255,
255, 1)"));
}
```

Joonis 7. Väljumise nupu stiili testide grupi testid.

Testide tulemused on järgmised (Joonis 8):



OK	TheSecondGroup	16s 528ms
OK	buttonLogoutHoverFullColorTesting	15s 81ms
OK	buttonLogoutDefaultFullColorTesting	1s 447ms

Joonis 8. Väljumise nupu testimise tulemused.

Läbiviidud testide tulemused demonstreerivad, et liikudes antud nupu peale arvutihiirega, muutub selle stiil. Väljumise nupp vastab seega nii projekti nõuetele kui ka Jakob Nielsen'i esimesele heuristikale.

Link „Tagasi e-maksuameti/e-tolli lehele“ ja keelevaliku lingid. Nimetatud linkide puhul kasutatakse samu stiile, seega erinevad nende jaoks kirjutatud automaatedid vaid nimede poolest, mille järgi elemente lehel otsitakse.

Käesoleva testidega (Joonis 9) kontrollib autor lingi „Tagasi e-maksuameti/e-tolli lehele“ stiili vastavust nõuetele vaikimisi olekus.

```
@Test
public void returnLinkDefaultColorTest() {
    assert ($("#navbarResponsive").find(By.linkText("Tagasi e-
maksuameti/e-tolli lehele"))).getCssValue("color").equals("rgba(93,
96, 113, 1)");
}
@Test
public void returnLinkDefaultTextDecorationTest() {
    assert ($("#navbarResponsive").find(By.linkText("Tagasi e-
maksuameti/e-tolli lehele"))).getCssValue("text-decoration-
line").equals("underline");
}
@Test
public void returnLinkDefaultSizeTest() {
    assert ($("#navbarResponsive").find(By.linkText("Tagasi e-
maksuameti/e-tolli lehele"))).getCssValue("font-
size").equals("14px");
}
```

Joonis 9. Lingi „Tagasi e-maksuameti/e-tolli lehele“ vaikimisi oleku testid.

Alljärgnevate testidega (Joonis 10) kontrollib autor lingi „Tagasi e-maksuameti/e-tolli lehele“ stiili vastavust nõuetele siis, kui selle peal asub arvutihiir. Selleks on loodud kolm testi, mis kontrollivad kas lingi värv ja suurus on korrektne. Samuti on oodatud tulemuseks, et link ei ole käesoleva oleku puhul allajoonitud.

```

@Test
public void returnLinkHoverColorTest() {
    $("#navbarResponsive").find(By.linkText("Tagasi e-maksuameti/e-
tollli lehele")).hover();
    assert($("#navbarResponsive").find(By.linkText("Tagasi e-
maksuameti/e-tollli lehele")).getCssValue("color").equals("rgba(0, 131,
186, 1)"));
}
@Test
public void returnLinkHoverTextDecorationTest() {
    $("#navbarResponsive").find(By.linkText("Tagasi e-maksuameti/e-
tollli lehele")).hover();
    assert($("#navbarResponsive").find(By.linkText("Tagasi e-
maksuameti/e-tollli lehele")).getCssValue("text-decoration-
line").equals("none"));
}
@Test
public void returnLinkHoverSizeTest() {
    $("#navbarResponsive").find(By.linkText("Tagasi e-maksuameti/e-
tollli lehele")).hover();
    assert($("#navbarResponsive").find(By.linkText("Tagasi e-
maksuameti/e-tollli lehele")).getCssValue("font-size").equals("14px"));
}

```

Joonis 10. „Tagasi e-maksuameti/e-tollli lehele” lingi stiili testid hiire pealasumisel.

Joonis 11 illustreerib „Tagasi e-maksuameti/e-tollli lehele” lingi testide tulemusi.

TheSecondGroup		27s 338ms
OK	returnLinkDefaultSizeTest	12s 108ms
OK	returnLinkHoverColorTest	4s 176ms
OK	returnLinkDefaultTextDecorationTest	3s 985ms
OK	returnLinkHoverSizeTest	1s 29ms
!	returnLinkHoverTextDecorationTest	4s 997ms
OK	returnLinkDefaultColorTest	1s 43ms

Joonis 11. „Tagasi e-maksuameti/e-tollli lehele” lingi testimise tulemused.

Keelevaliku linkidel peab olema sama stiil nagu lingil “Tagasi e-maksuameti/e-tollli lehele”. Seega erinevad keelevaliku linkide “In English” ja „Ha русском” testid vaikimisi ja hover oleku jaoks „Tagasi e-maksuameti/e-tollli lehele” lingist vaid otsitava teksti poolest. Seega on töösse välja toodud ainult läbiviidud testide tulemused (Joonis 12).

! TheSecondGroup		29s 301ms
!	inRussianLinkHoverTextDecorationTest	10s 559ms
OK	inRussianLinkDefaultTextDecorationTest	1s 151ms
OK	inRussianLinkHoverColorTest	1s 29ms
OK	inRussianLinkHoverSizeTest	967ms
OK	inRussianLinkDefaultColorTest	4s 897ms
OK	inEnglishLinkDefaultColorTest	906ms
OK	inEnglishLinkDefaultTextDecorationTest	3s 640ms
OK	inEnglishLinkHoverSizeTest	1s 40ms
OK	inEnglishLinkDefaultSizeTest	1s 385ms
OK	inEnglishLinkHoverColorTest	1s 171ms
OK	inRussianLinkDefaultSizeTest	936ms
!	inEnglishLinkHoverTextDecorationTest	1s 620ms

Joonis 12. Keelevaliku linkide testimise tulemused.

Automaattestide tulemused (Joonis 11 ja Joonis 12) näitavad, et päise lingid ei vasta 100-protsendiliselt projekti nõuetele. Linkide värvid ja suurus on realiseeritud õigesti, kuid testi tulemustest järeldeb, et kõikide ülaltestitud linkide puhul ei ole täidetud nõue, et arvutihiire asumisel lingi peal peab see olema allajoonimata. Veebilehe päise linkide all on kõikides olekutes allatõmmatud joon, mis ei vasta projekti nõuetele. Kuigi linkide stiil ei vasta projekti nõuetele, võib siiski järeldada, et Jakob Nielsen'i teine heuristika on täidetud, kuna kahes olekus on linkide stiil värvidega eristatav.

„Vaatan lähemalt“ ja „Kõik kontaktandmed“ lingid. „Vaatan lähemalt” lingi testid on väga sarnased ülalkirjeldatud ehk päise linkide testidele. „Vaatan lähemalt” link peab erinema lehe päises olevatest linkidest värvi poolest. Kuna vaatluse all oleval lingil on olemas unikaalne id, mille nimeks on „*more_info*”, siis demonstreeritakse *Selenide*-i pakutavat võimalust otsida objekti id kaudu, kasutades selleks jQuery elementide nimetusi.

Kõigepealt testib autor lingi vastavust projekti nõuetele vaikimisi olekus ning testimiseks kasutatava koodiga saab tutvuda alljärgneval joonisel (Joonis 13).

```

@Test
public void goToDetailsLinkDefaultColorTest() {
    assert($("#more_info").getCssValue("color").equals("rgba(255, 128,
0, 1)"));
}
@Test
public void goToDetailsLinkDefaultTextDecorationTest() {
    assert($("#more_info").getCssValue("text-decoration-
line").equals("underline"));
}
@Test
public void goToDetailsLinkDefaultSizeTest() {
    assert($("#more_info").getCssValue("font-size").equals("14px"));
}

```

Joonis 13. „Vaatan lähemalt“ lingi vaikimisi oleku testid.

Järgmisena tuuakse välja testid (Joonis 14), mis on loodud lingi oleku jaoks, kui selle peab asub arvutihir.

```

@Test
public void goToDetailsLinkHoverColorTest() {
    $("#more_info").hover();
    assert($("#more_info").getCssValue("color").equals("rgba(255, 128,
0, 1)"));
}
@Test
public void goToDetailsLinkHoverTextDecorationTest() {
    $("#more_info").hover();
    assert($("#more_info").getCssValue("text-decoration-
line").equals("none"));
}
@Test
public void goToDetailsLinkHoverSizeTest() {
    $("#more_info").hover();
    assert($("#more_info").getCssValue("font-size").equals("14px"));
}

```

Joonis 14. „Vaatan lähemalt“ lingi testid arvutihire pealasumisel.

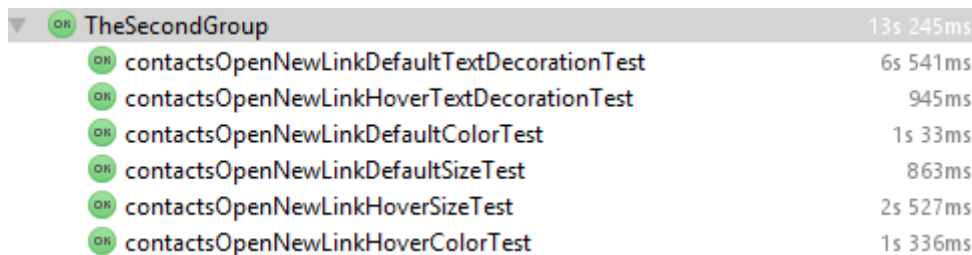
Tutvuda „Vaatan lähemalt“ lingi stiili testimise tulemustega saab alljärgneval joonisel (Joonis 15).

Testi nimi	Tulemus
TheSecondGroup	17s 931ms
goToDetailsLinkDefaultTextDecorationTest	10s 360ms
goToDetailsLinkDefaultSizeTest	849ms
goToDetailsLinkHoverTextDecorationTest	1s
goToDetailsLinkHoverSizeTest	838ms
goToDetailsLinkDefaultColorTest	1s 259ms
goToDetailsLinkHoverColorTest	3s 625ms

Joonis 15. „Vaatan lähemalt“ lingi testimise tulemused.

„Vaatan lähemalt“ lingi testimise tulemustest võib järeldada, et antud veebilehe element on korrektselt realiseeritud ning vastab „Madalapalgaliste toetus“ projektis määratud nõuetele.

„Kõik kontaktandmed” lingi automaatsete kood on peaaegu sama nagu „Vaatan lähemalt” lingil, erinedes vaid lingi ülesleidmise viisis. Seega koodinäidet välja ei tooda, vaid näidatakse järgneval joonisel (Joonis 16) ainult testide tulemusi.



Test Name	Duration
contactsOpenNewLinkDefaultTextDecorationTest	6s 541ms
contactsOpenNewLinkHoverTextDecorationTest	945ms
contactsOpenNewLinkDefaultColorTest	1s 33ms
contactsOpenNewLinkDefaultSizeTest	863ms
contactsOpenNewLinkHoverSizeTest	2s 527ms
contactsOpenNewLinkHoverColorTest	1s 336ms

Joonis 16. „Kõik kontaktandmed” lingi testimise tulemused.

Joonisel 15 ja 16 väljatoodud tulemustest saab järeldada, et „Vaatan lähemalt“ ja „Kõik kontaktandmed“ lingid vastavad projektis määratud nõuetele ja ka Jakob Nielsen esimesele heuristikale.

Testide grupp 3: Kuna rakenduse käesolev vaade on enamasti staatiline ehk URL ei muutu, siis pole antud testi läbiviimiseks vajalik antud funktsionaalsust korduvalt testida. Kontrollimiseks piisab ühest automaatsest (Joonis 17).

```
@Test
public void ifReturnLinkIsVisible() {
    $("#navbarResponsive").find(By.linkText("Tagasi e-maksuameti/e-
    tolli lehele")).shouldBe(visible);
}
```

Joonis 17. „Tagasi e-maksuameti/e-tolli lehele” lingi olemasolu kontrolli test.

Alljärgnev joonis (Joonis 18) demonstreerib saadud tulemuse korrektsust.



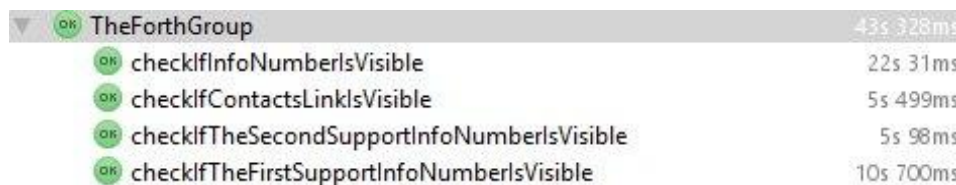
Test Name	Duration
ifReturnLinkIsVisible	22s 747ms

Joonis 18. „Tagasi e-maksuameti/e-tolli lehele” lingi kuvamise testimine.

Läbiviidud test on lihtne ja lühike, kuid väga oluline, kuna sellega saab kontrollida veebilehe vastavust Jakob Nielsen kolmandale heuristikale. Kasutaja peab teadma, et tal on alati võimalik minna avalehele. Testi tulemusest järeldub, et test on edukalt läbitud ehk nõue on täidetud ja vastab Nielsen kolmandale heuristikale.

Testide grupp 4: Automaattestide abil saadakse teada, kas kasutaja jaoks on abi ja konsultatsiooni saamiseks ettenähtud andmed, link ja abitekstid veebilehel kuvatud. Kuna testide loomisel kasutatakse sama süntaksit nagu esimeses testide gruppis, siis lingi ja telefoninumbrite jaoks koodinäiteid eraldi välja ei tooda.

Telefoninumbriid ja kontaktandmetega link abi saamiseks. Testide abil kontrollitakse, kas vajalikud telefoninumbriid koos seletava tekstiga ja kontaktandmetega link on lehel kuvatud. Järgnevalt (Joonis 19) on toodud välja testi tulemused.



Test Name	Execution Time
checkIfInfoNumbersVisible	22s 31ms
checkIfContactsLinksVisible	5s 499ms
checkIfTheSecondSupportInfoNumbersVisible	5s 98ms
checkIfTheFirstSupportInfoNumbersVisible	10s 700ms

Joonis 19. Kontaktandmete kättesaadavuse testimise tulemused.

Testi tulemus on positiivne. Testide tulemustest järeldub, et nõutud Maksu- ja Tolliameti kontaktandmed on veebilehel kättesaadavad.

Abitekst. Abiteksti ülesleidmiseks kasutati keerulisemat koodi, kui teiste elementide ülesleidmise jaoks. Seega tuuakse välja üks test tingimuse „Eesti residentsus” kohta. Testi oodatud tulemus on, et „i” ikooni peale klikkides avaneb täiendav informatsioon konkreetse tingimuse kohta. Testi kood on välja toodud alljärgneval joonisel (Joonis 20).

```
@Test
public void residencyOfEstoniaTooltipIsAvailable() {
    $("#more_details_table").find(byText("Eesti
residentsus")).closest("th").find("a").click();
    $(".tooltip-inner").shouldHave(text("Tingimus on täidetud, kui
olete olnud Eesti resident kogu kalendrikuu jooksul. Kui need andmed
erinevad Teile teadaolevaist, siis pöörduge palun Maksu- ja Tolliameti
poole residentsuse andmete muutmiseks."));
}
```

Joonis 20. Abiteksti olemasolu test.

Samu, kuid teiste abitekstide sõnastustega teste loodi ülejäänud kahe erineva abiteksti kontrollimiseks. Tulemustega tutvumiseks on väljatoodud joonis (Joonis 21).

OK TheForthGroup		21s 177ms
OK	checkIfContactsLinksAvailable	10s 336ms
OK	residencyOfEstoniaTooltipsAvailable	1s 325ms
OK	checkIfInfoNumbersVisible	955ms
OK	contactsOpenNewLinksVisible	917ms
OK	checkIfTheSecondSupportInfoNumbersVisible	878ms
OK	workRelationshipTooltipsAvailable	1s 58ms
OK	checkIfTheFirstSupportInfoNumbersVisible	4s 563ms
OK	fullWorkloadTooltipsAvailable	1s 145ms

Joonis 21. Abiteksti olemasolu testimise tulemused.

Testide läbiviimisel selgus, et nõutud elemendid on veebilehel nähtavad, seega on vaatluse all olev projekti nõue täidetud. Iga teksti juures kuvatakse just sellele vastavat abiteksti.

Testides rakendust ekraanisuurusega 1200x903px sai välja selgitatud, et projekti nõuded on vähemalt neljast Jakob Nielsen'i heuristikast lähtudes korrektselt määratud ja „Madalalgaliste toetus“ rakendus kvalifitseerub kasutajasõbralike veebilehtede alla. Kõik nõuded kasutajaliidese disainile ei saanud täidetud vastavalt sellele, kuidas on dokumentatsioonis kirjeldatud. Vaatamata sellele saab praegust realisatsiooni pidada kasutajasõbralikuks, kuna projekti kasutajaliides vastab suuremal määral kirjeldatud kasutajakogemuse printsiipidele.

3.2.3 Testide gruppide kirjeldamine nutitelefone vaate jaoks

Jakob Nielsen-i seitsmendast heuristikast järeldub, et põhilised veebilehe elemendid peavad olema kättesaadavad enamike ekraanisuurustega [23]. Antud peatükis antakse ülevaade projekti nõuetest nutitelefone vaate jaoks. Autori selgituste paremaks mõistmiseks on toodud välja rakenduse vaade nutitelefonis (Joonis 22). Joonisel on kasutajaliidese tähtsamate osade juurde, mida on töös edaspidi mainitud, lisatud sulgudes numbrid 1, 2 ja 3.

(1)
(2)

Madalalpalgaliste toetuse taotluse esitamine

Sellel lehel saate taotleda madalalpalgaliste toetust. Kui kõik toetuse taotlemise tingimused on täidetud ning soovite toetust taotleda, siis lisage allolevale vormile puuduvad andmed ning esitage taotlus. Taotlust saab esitada kuni 30.03.2017. Kui kõik tingimused ei ole täidetud, siis ei ole Teil õigust toetust saada.

Tingimused

Tingimus:

Tuludeklaratsiooni esitamine

Olete esitanud 2016. aasta füüsilise isiku tuludeklaratsiooni õigeaegselt 30.03.2017.

2016. aasta kogutulu ei ole suurem kui 7817 eurot

Olete 2016. aastal saanud tulu alla kehtestatud piirmäära 7817 eurot.

Muud tingimused (residentsus, vanus, töö- või teenistussuhte olemasolu, töötamine täistööajaga)

Tingimused ei ole täidetud.

Vaatan lähemalt

Tabeli vaatamiseks pöörake telefon küliliasesse. Need tingimused peavad olema 2016. aasta jooksul vähemalt **kuuel kuul üheaegselt** täidetud.

Tingimus	Jaan	Veeb	Märts	Apr	Mai	Juuni
(3) Eesti residentsus	✓	✓	✓	✓	✓	✓
(3) Kehtiv töö- või teenistussuhe	!!	!!	!!	!!	!!	!!
(3) Töötamine täistööajaga	!!	!!	!!	!!	!!	!!
Kokku	!!	!!	!!	!!	!!	!!



MAKSU- JA TOLLIAMET



880 0810 Üldinfo
 880 0814 E-tolliteenuste klienditugi
 880 0815 E-maksuteenuste klienditugi

[Kõik kontaktandmed](#), t²

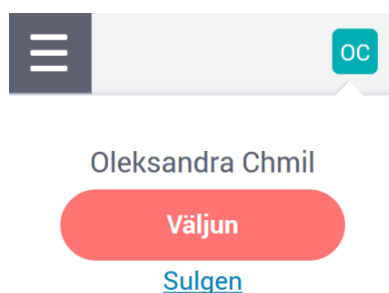
Joonis 22. Rakenduse esmane vaade nutitefonis.

Käesoleva töö raames kontrollitakse ka elementide kättesaadavust ekraanisuurusega 599x903px, mis vastab nutitelefoniga vaatele. Järgnevalt kirjeldab autor 1200x903px ekraanisuurusele loodud testide gruppide käitumist nutitelefonile vastava ekraani suurusega.

Testide grupp 1 ei erine nutitelefoniga vaates töölaua vaatest, kuna vajalikud tekstid peavad samuti olema esilehel koheselt nähtavad.

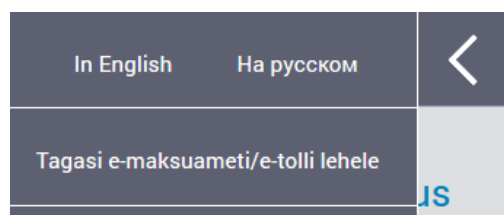
Testide grupp 2 erineb suuremast ekraani vaatest selle poolest, et väljumise nupp ei ole nutiteleskrinnis vaikimisi kuvatud. Täisnime asemel tekivad lehe päisesse kasutaja nime initsiaalid (Joonis 22 (2)), mille peale klikkides avaneb aken, kus omakorda asub väljumise nupp (Joonis 23). Samuti muutub telefoni vaates väljumise nupu suurus, kuid selle töö raames nupu suurust ei kontrollita. Nutiteleskrinnis ei testita nupu stiili siis, kui selle peal asub arvutihiir, kuna antud juhul eeldatakse, et kasutajal arvutihiir puudub.

Järgneval joonisel (Joonis 23) on kujutatud aken, mis avaneb nutiteleskrinnis vaates siis, kui kasutaja on oma initsiaalide peale klikkinud. Avanenud akna kaudu on kasutajal võimalik näha oma täisnime, logida oma kontost välja kasutades „Väljun“ nuppu ja sulgeda aken „Sulgen“ lingi kaudu.



Joonis 23. Väljumise nupp nutiteleskrinnis vaates.

„Tagasi e-maksuameti/e-tolli lehele” link ja keelevaliku lingid ei ole nutiteleskrinnis vaikimisi kuvatud. Tavalise päise asemel kuvatakse kasutajale burgermenüü (Joonis 22 (1)). Päise lingid avanevad alles siis, kui burgermenüü peale on klikitud (Joonis 24). Linkide värv nutiteleskrinnis vaates on valge: rgba(255, 255, 255, 1). Samuti kontrollitakse burgermenüü taustavärvi, kuna selle peal asuvad päise lingid, mis peavad olema võrreldes taustavärviga kontrastsed. Avanenud menüü taustavärviks peab olema rgba(93, 96, 113, 1). Järgmine pilt illustreerib kirjeldatud menüüid ja selles asuvaid linke (Joonis 24).



Joonis 24. Päise lingid nutitelefonis vaates.

Testide grupp 3 erineb suurema ekaaniga testimisest vaid selle poolest, et “Tagasi” lingi kuvamiseks tuleb kõigepealt burgermenüü peale klikkida. Tingimuste sõnastused on nähtavad joonisel 22. Iga nõutud teksti juures on joonisel tehtud märgis: „(3)“.

Testide grupp 4: nutitefonis peab samuti olema saadav abiinfo. Abiinfo saamiseks tuleb vajutada teksti peale, mille all on punktiirjoon. Tekstide juures on joonisel 22 märgitud: „(3)“. Antud juhul tähendabki punktiirjoon abiteksti. Samuti peab nagu monitoriga arvutigi puhul olema lehe jaluses kuvatud kõik nõutud kontaktandmed.

3.2.4 Testimise tulemused ekraanisuurusega 599x903px

Käesolavas peatükis testitakse rakendust ekraanisuurusega, mis vastab nutitelefonis vaatele ehk antud juhul ekraani resolutsioonile 599x903px. Siin viiakse läbi kõik testid, mis ka eelmise ekraanisuuruse puhul. Kuna kõik elemendid ei mahu ühele lehele, siis mobiilivaate puhul muudetakse lehe stiili. Vaatamata sellele peavad kõik elemendid, mis olid töölaual puhul kuvatud, olema ka nutitefonis kergesti kättesaadavad.

Nagu kujutatud joonisel 2, ei pruugi nutitelefonis vaates elemendid olla kuvatud samamoodi nagu suuremate ekraaniresolutsioonide puhul [18]. “Madalalpalgaliste toetus” rakenduses on kasutatud sama põhimõtet.

Testide grupp 1 peab mobiilis olema samasugune nagu töölaual vaates. Kõik sõnastused jäävad samaks. Tulemused on välja toodud joonisel 25.

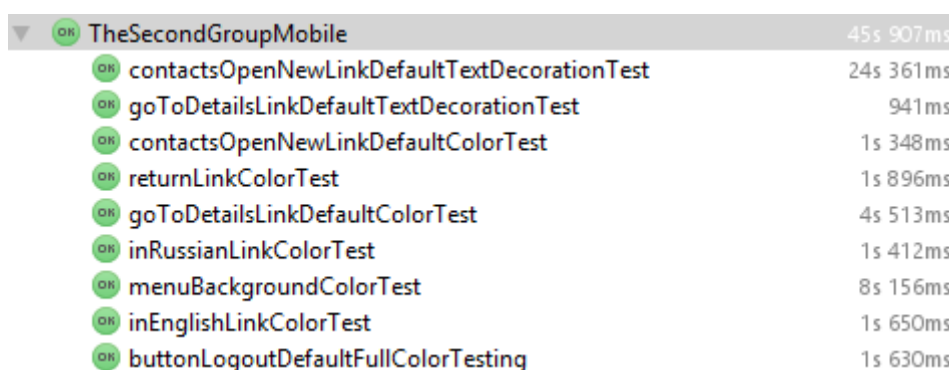
OK	TheFirstGroupMobile	17s 722ms
OK	theSecondConditionForRefund	12s 242ms
OK	theFirstConditionForRefund	4s 39ms
OK	theThirdConditionForRefund	1s 441ms

Joonis 25. Esimese testide grupi tulemused mobiilivaates.

Kõik nõutud sõnastused on nutitefonis nähtavad ning korrektselt realiseeritud.

Testide grupp 2 erineb selle poolest, et väljumise nupp ei ole nutitefonis vaikimisi kuvatud. Täisnime asemel tekivad lehe päisesse kasutaja nime initsiaalid, mille peale klikkides avaneb aken, kus omakorda asub väljumise nupp.

„Tagasi e-maksuameti/e-tolli lehele” ja keelevaliku lingid ei ole vaikimisi kuvatud, vaid avanevad siis, kui burgermenüü peale on klikitud. Kuna nutitelefoni vaates kontrollitakse ainult vaikimisi elementide värve, on kõik antud grupi testide tulemused toodud koos välja järgneval joonisel (Joonis 26).



Test Name	Duration	Execution Time
TheSecondGroupMobile	45s	907ms
contactsOpenNewLinkDefaultTextDecorationTest	24s	361ms
goToDetailsLinkDefaultTextDecorationTest		941ms
contactsOpenNewLinkDefaultColorTest	1s	348ms
returnLinkColorTest	1s	896ms
goToDetailsLinkDefaultColorTest	4s	513ms
inRussianLinkColorTest	1s	412ms
menuBackgroundColorTest	8s	156ms
inEnglishLinkColorTest	1s	650ms
buttonLogoutDefaultFullColorTesting	1s	630ms

Joonis 26. Teise testide grupi tulemused nutitelefoni vaates.

Tulemustest järeldub elementide stiilide korrektsus.

Testide grupp 3 koosneb ainult ühest testist. Koodinäidet eraldi välja ei tooda, kuna test nutitelefoni vaates ei erine oluliselt suurema ekraaniga testist. Tulemused on illustreeritud alljärgneval joonisel (Joonis 27).

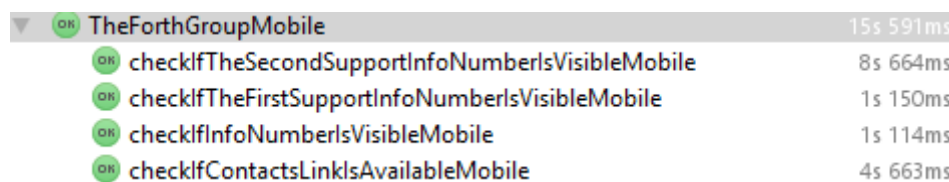


Test Name	Duration	Execution Time
TheThirdGroupMobile	11s	273ms
returnLinkColorTest	11s	273ms

Joonis 27. Kolmanda testide grupi tulemused nutitelefoni vaates.

Läbiviidud testi tulemus demonstreerib, et veebileht võimaldab e-maksuameti ja e-tolli lehele minekut. Järelikult on Jakob Nielsen'i heuristika täidetud.

Testide grupp 4. Nutitelefoni puhul kontrollitakse, kas abi saamiseks vajalikud lingid on lehe jaluses kuvatud. Antud nõue ei erine töölaua vaatest, seega koodinäidet eraldi välja ei tooda. Tulemusi illustreerib järgnev joonis (Joonis 28).



Test Name	Duration	Execution Time
TheForthGroupMobile	15s	591ms
checkIfTheSecondSupportInfoNumberIsVisibleMobile	8s	664ms
checkIfTheFirstSupportInfoNumberIsVisibleMobile	1s	150ms
checkIfInfoNumberIsVisibleMobile	1s	114ms
checkIfContactsLinksAvailableMobile	4s	663ms

Joonis 28. Kontaktandmete kättesaadavuse testimise tulemused nutitelefoni vaates.

Testimise tulemustest on näha, et kõik nõutud andmed on lehe jaluses kuvatud. See tähendab, et testitav nõue on ka nutitelefonil täidetud.

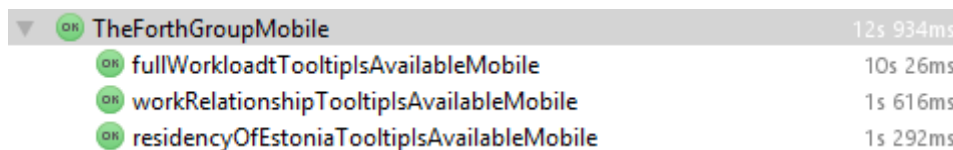
Abiinfo saamiseks nutitefonis tuleb kasutajal klikkida teksti peale, mille all on punktiirjoon. Selle peale klikkides kuvatakse kasutajale abiinfo. Abiinfo tekst väiksema ekraani puhul ei erine töölaua omast.

Küll aga erineb nutitelefonis vaates abiinfo testimise loogika töölaua omast, kuna HTML (*HyperText Markup Language*) koodi struktuur on teistsugune. Selleks, et tuua veel üht näidet *Selenide*'i kasutamisest keerulisemate eesmärkide jaoks, on välja toodud testi kood tingimuse „Eesti residentsus” sees oleva abiteksti kontrollimiseks. Kirjeldatud näide on kuvatud joonisel 29.

```
@Test
public void residencyOfEstoniaTooltipIsAvailableMobile() {
    $("#more_info").click();
    $("#more_details_table").findAll("a.hidden-md-
up").filterBy(Condition.text("Eesti residentsus")).first().click();
    $(".tooltip-inner").shouldHave(text("Tingimus on täidetud, kui
olete olnud Eesti resident kogu kalendrikuu jooksul. Kui need andmed
erinevad Teile teadaolevaist, siis pöörduge palun Maksu- ja Tolliameti
poole residentsuse andmete muutmiseks."));
}
```

Joonis 29. Abiteksti olemasolu test nutitelefonis vaates.

Abiteksti olemasolu testimise tulemustega saab tutvuda järgneval pildil (Joonis 30).



OK	TheForthGroupMobile	12s 934ms
OK	fullWorkloadtTooltipsAvailableMobile	10s 26ms
OK	workRelationshipTooltipsAvailableMobile	1s 616ms
OK	residencyOfEstoniaTooltipsAvailableMobile	1s 292ms

Joonis 30. Abiteksti olemasolu testimise tulemused nutitelefonis vaates.

Teksti peale klikkides kuvatakse veebilehe kasutajale vastav abiinfo, nagu on ka projektis nõutud. Läbiviidud testide tulemused demonstreerivad nende tulemuste korrektsust.

Käesolevas peatükis said analüüsitud „Madalapalgaliste toetus“ projekti nõuded koos Jakob Nielsen-i heuristikatega, millest lähtuvalt on realiseeritud kasutajaliidese automaattestimine. Kasutajaliidese testimise tulemuste detailsemat analüüsi ja järeldusi tehakse järgmises peatükis (Peatükk 4).

4 Tulemuste analüüs ja ettepanekud

Uuringu käigus selgus, et projektis määratud nõuded vastavad põhilistele kasutajakogemuse printsiipidele. Lähtudes Jakob Nielsen'i heuristikatest, võib järeldada, et projektis määratud nõuded vastavad tema esimesele heuristikale. Nõuded elementide olekutele on tehtud sellistena, et kasutajal oleks mugav ja meeldiv süsteemi kasutada. Elementide erinevatele olekutele said määratud vastavad stiilid.

Projekti testimise käigus selgus, et realiseeritud disain ei vasta 100% projektis määratud nõuetele, kuid enamik neist vastab siiski väljavalitud Jakob Nielsen'i heuristikatele. Vaatamata sellele, et kõikide linkide stiilid ei vasta täielikult projektis määratud stiilidele, on nende olekud alati arusaadavad, kuna stiilid muutuvad vastavalt staatusele.

4.1 Testitava rakenduse tulemuste analüüs

Peatükis 2 kirjeldatud kasutajakogemuse reeglitest lähtudes vastab „Madalalpalgaliste toetus“ rakendus vähemalt neljale neist. Siiski ei vasta kasutajaliidese disain kõikidele projektis kirjeldatud nõuetele. Analüüsid projektis määratud nõudeid võib järeldada, et nad on püstitatud korrektselt, lähtudes põhilistest kasutajamugavuse ja kasutajakogemuse printsiipidest. Kuna autori isiklik rakenduse vaade koosnes vaid ühest lehest, siis ei olnud võimalik kontrollida kõikidele Jakob Nielsen'i heuristikatele vastavust ning samasugust testimist tuleks usaldusväärse tulemuse jaoks korrata teiste võimalike vaadetega. Testimine on tarkvaraarenduse elutsüklis üks kõige olulisematest etappidest, seega peab autor testimise protsessi kvaliteetse tarkvara loomisel kohustuslikuks. Nagu sai lõputöös analüüsitud, on tarkvara testimiseks olemas kaks põhilist meetodit: manuaalne ja automatiseeritud testimine. Olenevalt konkreetsetest eesmärkidest, tuleb valida kas üks või teine. Osas 4.1.1 on välja toodud autori analüüsimeetodikate valimiseks.

4.1.1 Kasutajaliidese testimise automatiseerimise otstarbekus

Uuringu ja analüüsi käigus sai rakenduse testimiseks valitud automaattestimine. Valik oli tehtud enne rakendust testimise hakkamist. Kuna testimine on hetkel lõppenud, siis analüüsitakse automatiseeritud testimise eelisi ja kasutamise otstarbekust kasutajaliidese jaoks nüüd autori isiklikust kogemusest lähtuvalt.

Automaattestimist on mugav kasutada juhul, kui projektis on nõuded kontreetselt määratud. Antud kriteerium väldib segadust testide loomise ja hooldamise etappides. „Madalalpalgaliste toetus“ projektis on nõuded selgelt määratud, mistõttu pole testide loomise käigus tekkinud segadust ega arusaamatusi oodatud tulemuste kohta.

Autori eesmärgiks oli teha kasutajaliidese testimine võimalikult efektiivseks. Testimine on efektiivne siis, kui sellele on kulutatud võimalikult vähe aega ja vaeva, säilitades seejuures testimise kvaliteedi. Kirjeldatud olukorda saab automatiseeritud testimise abil saavutada siis, kui testijal on olemas programmeerimisoskused. Juhul kui testimine on automatiseeritud, siis kulub testide käivitamisele ja läbiviimisele võrreldes manuaalse testimisega oluliselt vähem aega. Automatiseerimise protsess ise on suhteliselt ajakulukas juhul, kui testijal puuduvad vastavad teadmised ja oskused.

Järgmiseks oluliseks kriteeriumiks selleks, et kasutajaliidese testimise automatiseerimine oleks manuaalse testimisega võrreldes efektiivsem, on puhas kood. Mida korrektsem ja puhtam on HTML/CSS (*HyperText Markup Language/Cascading Style Sheets*) kood, seda sujuvamalt edeneb testimine. „Madalalpalgaliste toetus” rakenduses edenes kõige kiiremini nende elementide testimine, millel oli olemas korrektne unikaalne id või midagi, mille järgi oleks võimalik elementi kergesti identifitseerida. Ideaalne olukord oleks juhul, kui iga kasutajaliidese element omaks unikaalset id-d, kuid reaalsuses esineb seda väga harva. Kõige raskemaks osaks oli autori jaoks abitekstide leidmine, kuna nende puhul oli kõige raskem leida sobivat loogikat, mille järgi neid elemente otsida.

Automatiseeritud testimine on hea lahendus projektidele, kus on palju tähelepanu pööratud tööprotsessile. Selliste projektide hulka kuuluvad näiteks need, mis kasutavad agiilseid tarkvaraarendusmetoodikaid. Kuna sellistes metoodikates on täpsed ajahinnangud ja tehtud tööde ülevaade väga olulised, siis automaattestimine aitab sellistes projektides hinnangute analüüsile kaasa.

Üldjuhul on autori arvates kasutajaliidese testimise automatiseerimine väga kasulik ja efektiivne, kuid antud meetod ei pruugi kõikide projektide jaoks sobida. Enne valiku tegemist tuleb veenduda, et konkreetse projekti puhul on automatiseerimine otstarbekas. Selles veendumiseks soovib autor lähtuda eelkõige kolmest ülalkirjeldatud

kriteeriumist. Samuti on kasuks osas 2.4.2 välja toodud Tabel 2, kus on lahti kirjeldatud manuaalse- ja automaattestimise eelised.

4.1.2 Saadud tulemuste valideeritavus

Manuaalse kasutajaliidese testimise puhul ei ole võimalik valideerida tulemusi nii, et tehtud töö hinnangu määramatus oleks minimaalne. Automatiseeritud testimist kasutades on võimalik saadud tulemusi suhteliselt täpselt hinnata. Käesoleva bakalaureusetöö puhul on kasutatud automaattestimist. Automaattestide loomise vahendiks oli valitud *Selenide*'i raamistik, millega sai teste luua suhteliselt kiirelt ja mugavalt. Iga käivitatud automaattesti juures on kirjeldatud tulemus koos testi läbiviimisele kulunud ajaga. Aeg on kõige parem viis testija töö tulemuste valideerimiseks. Manuaalse testimise puhul pole võimalik täpselt kontrollida, kui palju aega kulus ühe või teise elemendi kontrollimiseks.

Lõputöö käigus realiseeritud automaattestid teevad testimise efektiivseks, kuna testija poolt tehtud klikkide arv väheneb oluliselt. Kui automaattestid on loodud, siis on ainsateks testijalt nõutud tegevusteks programmikoodi käivitamine ja testide tulemuste analüüs. Kõigest ülalkirjeldatust saab järeldada, et tehtud töö muutis testimise olulisemalt efektiivsemaks ja usaldusväärsemaks, võimaldades anda täpseid ajahinnanguid töö organiseerimiseks ja tulemuste analüüsimiseks.

5 Kokkuvõte

Bakalaureusetöö eesmärkideks oli: luua võimalikult efektiivseid teste, mis kontrollivad kasutajaliidese elementide vastavust Maksu- ja Tolliameti projekti nõuetele ja põhilistele kasutajakogemuse printsiipidele; selgitada välja ja analüüsida, millist testimise meetoodikat on kõige otstarbekam kasutada kasutajaliidese testimiseks, juhul kui nõuded kasutajaliidesele on konkreetselt määratud.

Kirjeldatud eesmärgid said lõputöö käigus saavutatud ning lõputöö oodatud tulemus vastab tegelikule tulemusele. Eesmärkide täitmise teekond on järgnevalt detailsemalt lahti seletatud. Lõputöös said kirjeldatud põhilised kasutajakogemuse printsiibid. Osa kirjeldatud printsiipidest rakendati testimisel, lähtudes projektis püstitatud nõuetest. Nimelt leiti projekti nõuetes ja kirjeldatud kasutajakogemuse printsiipides kattuvusi, mille põhjal otsustati, millisele kriteeriumile vastavust soovitakse testida. Vastavalt sellele said moodustatud testide grupid. Samuti said kirjeldatud manuaalse ja automatiseeritud testimise eelised, mille põhjal valis autor rakenduse testimiseks automaattestimise. Lähtudes automaattestimise vahendite võrdlusest, valis autor Selenide-i raamistiku, kuna selle abil saab võimalikult väikese ajakuluga ja vaevaga luua efektiivsed kasutajaliidese teste.

Valitud Jakob Nielseni heuristikatest lähtudes loodi *Selenide*'i raamistiku abil kasutajaliidese automaattestid. Automaattestid on loodud vastavalt „Madalalpalgaliste toetus“ projekti nõuetele ning testide käivitamise järel on tehtud analüüs iga testi grupi kohta. Läbiviidud testide ja analüüsi tulemustest sai järeldatud, et enamik testitavaid kriteeriume vastavad Jakob Nielseni heuristikatele, kuigi kõik elemendid ei olnud realiseeritud vastavalt projektis määratud nõuetele.

Automaattestide abil said testitud elemendid, mille vastavust põhilistele kasutajakogemuse printsiipidele oli soovitud kontrollida. Samuti suurendas autori arvates automatiseeritud kasutajaliidese testimine saadud tulemuste usaldusväarsust ning andis võimaluse töö paremaks organiseerimiseks. Tehtud töö efektiivsust saab tänu automaattestidele mõõta ja hinnata. Kõigest ülalkirjeldatust järeldub, et püstitatud eesmärgid said saavutatud ning bakalaureusetöö käigus saadud tulemus vastab oodatud lõputöö tulemusele.

Kasutatud kirjandus

- [1] Automated vs. Manual Testing: The Pros and Cons of Each. [WWW]
<http://www.base36.com/2013/03/automated-vs-manual-testing-the-pros-and-cons-of-each/>
(29.04.2017)
- [2] Babich, N., Button UX Design: Best Practices, Types and States. [WWW]
<https://uxplanet.org/button-ux-design-best-practices-types-and-states-647cf4ae0fc6>
(07.04.2017)
- [3] Codeborne. Selenide and Selenium comparison. Why is not Selenium WebDriver enough?
[WWW] <http://selenide.org/documentation/selenide-vs-selenium.html> (12.04.2017)
- [4] Definition of pixel. [WWW] <https://www.merriam-webster.com/dictionary/pixel>
(03.04.2017)
- [5] Ergonomic requirements for office work with visual display terminals (VDTs) – Part 10:
Guidance on usability. The international organization for standardization ISO W. 9241-11:
Switzerland, 1998.
- [6] Ergonomics of human system interaction-Part 210: Human-centred design for interactive
systems. International Standardization Organization ISO DIS 9241-210:2010. Switzerland,
2009.
- [7] Firdaus, T. Responvise Web Design by Example: Beginner's Guide. 1st ed. Birmingham:
Packt Publishing, 2013.
- [8] Gojare S., Joshi R., Gaigaware D. Analysis and Design of Selenium WebDriver Automation
Testing Framework.– *Elsevier*, 2015, 341-346. [Online] ScienceDirect (03.05.2017)
- [9] Guan, D. Manual to Automated Testing: doktoritöö. Victoria University of Wellington,
Wellington, 2014.
- [10] International Conference on Theory and Practice of Electronic Governance. [WWW]
https://en.wikipedia.org/wiki/International_Conference_on_Theory_and_Practice_of_Electronic_Governance (17.05.2017)
- [11] Kasutajaliides. [WWW] <https://et.wikipedia.org/wiki/Kasutajaliides> (04.04.2017)
- [12] Kaur, H., Gupta, G. Comparative Study of Automated Testing Tools: Selenium, Quick Test
Professional and Testcomplete. – *Journal of Engineering Research and Application*, 2013,
2248-9622. [Online] Semantic Scholar (29.04.2017)
- [13] Laubheimer, P. Preventing User Errors: Avoiding Unconscious Slips. [WWW]
<https://www.nngroup.com/articles/slips/> (03.04.2017)
- [14] Leitner, A., Ciupa, I., Meyer, B., Howard, M. Reconciling Manual and Automated Testing:
The AutoTest Experience. – *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii
International Conference on*, 261-261. [Online] IEEE (29.04.2017)
- [15] Majandus- ja Kommunikatsiooniministeerium. Parts: e-riik ei ole vaid virtuaalne maailm.
[WWW] <https://www.mkm.ee/et/uudised/parts-e-riik-ei-ole-void-virtuaalne-maailm>
(15.05.2017)

- [16] Maksu- ja Tolliamet. Madalalpalgaliste töötajate toetus 2017. aastal. [WWW]
<https://www.emta.ee/et/eraklient/tulu-deklareerimine/madalalpalgaliste-toetus-2017-aastal>
 (08.04.2017)
- [17] Mobile Testing – UI. [WWW]
https://www.tutorialspoint.com/mobile_testing/mobile_testing_ui.htm (01.04.2017)
- [18] Mobile User Experience : Patterns to Make Sense of it All / A. Mendoza. Waltham: Elsevier Science, 2013. [Online] ebrary (17.04.2017)
- [19] Moule, J. Killer UX Design: Create User Experiences to Wow Your Visitors. 1st ed. Victoria: SitePoint, 2012.
- [20] Neil, T. 6 Tips for a Great Flex UX: Part 5. [WWW] <http://designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5> (07.04.2017)
- [21] Nielsen, J., Loranger, H. Prioritizing Web Usability. 1st ed. Berkeley: 2006.
- [22] Nielsen, J., Molich R. Heuristic evaluation of user interfaces. – *Proceedings of the SIGCHI conference on Human factors in computing systems.* – ACM, 1990, 249-256. [Online]. Academia. (11.04.2017)
- [23] Nielsen, J. 10 Usability Heuristics for User Interface Design. [WWW]
<https://www.nngroup.com/articles/ten-usability-heuristics/> (03.04.2017)
- [24] Quiñones, D., Rusu C. How to develop usability heuristics: A systematic literature review. – *Computer Standards & Interfaces.* 2017, 53, 89-122. [Online]. ScienceDirect (18.05.2017)
- [25] Rubin, J., Chisnell, D. Handbook Of Usability Testing: How to Plan, Design and Conduct Effective Tests. 2nd ed. Indianapolis: Wiley Publishing, 2008.
- [26] Selenide vs Selenium. [WWW] <https://github.com/codeborne/selenide/wiki/Selenide-vs-Selenium> (21.04.2017)
- [27] Selenium. [WWW] [https://en.wikipedia.org/wiki/Selenium_\(software\)](https://en.wikipedia.org/wiki/Selenium_(software)) (20.04.2017)
- [28] Silumine. [WWW] <https://et.wikipedia.org/wiki/Silumine> (08.05.2017)
- [29] Solntsev, A., Codeborne. Selenide - Concise UI Tests in Java. [WWW]
<http://www.methodsandtools.com/tools/selenide.php> (03.04.2017)
- [30] What is jQuery? [WWW] <https://jquery.com/> (15.05.2017)
- [31] W3Schools. Browser Display Statistics. [WWW]
https://www.w3schools.com/browsers/browsers_display.asp (03.04.2017)