

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kristiine Pärna 164759IABB

**UUE VEEBILEHE HALDAMISE TARKVARA ANALÜÜS JA
KAVANDAMINE PILETILEVIS**

Bakalaureuse lõputöö

Juhendaja: Dr Gunnar Piho

Tallinn 2020

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: Kristiine Pärna

17.05.2020

Annotatsioon

Antud töö eesmärgiks oli Piletilevis luua Baltic Ticket Holdings grupi jaoks uus firmale kuuluvate veebilehtede haldamise tarkvara Administraatorite paneel, mis oleks kasutajasõbralikum, vähendaks tööülesannete täitmise aega ning oleks paindlikum. Kogu projekti juhtimise ja analüüsiga tegeles töö autor ning tarkvara programmeerimisega arendajad. Uus tarkvara põhineb vanal programmil Adjuster. See tähendab, et uue programmi üldine struktuur ja ülesehitus jäid samasuguseks, kuid muudeti mooduleid, funktsionaalsusi ning kasutajaliidest.

Uue programmi loomine oli vajalik, sest vana on üles ehitatud aegunud arendusmeetoditel ja platvormile, mida enam ei arendata ega toetata. Seetõttu on piiratud vanasse tarkvarasse paljude uute funktsionaalsuste lisamine ning juba olemasolevate kasutamine ja muutmine. Administraatorite paneel põhineb kaasaegsetel arendusmeetoditel ning kasutajaliidese nõuetel. Uue tarkvara aluskujunduseks kasutati Bootstrap raamistikku ning disaini ja stiili lisamisel juba ettevõttele loodud stiiliraamatu eeskujul.

Nii uue kui ka vana programmi peal viis töö autor koos kasutajatega läbi erinevaid teste, et selgitada välja kohti, kus oli tööülesannete täitmine raskendatud või võttis palju aega. Töö autor analüüsis kõik testimise tulemused läbi ning leidis probleeme valmistanud kohtadele parimad võimalikud lahendused.

Töö käigus valmisid (või on arendusprotsess pooleli) mõned järgnevad muudatused: uus programmi disain, kaasajastatud kasutajatesüsteem, tabelite ridade liigutamiseks "lohist ja lase lahti" funktsionaalsus, tööriistavihjete lisamine jne.

Valminud uus tarkvara on esialgu kasutatav vaid testveebilehe ning -andmebaasi peal. Programm on arendatud aga piisavalt kaugele, et ettevõttel on seda võimalik endale sobival hetkel ka koheselt rakendada kasutusversioonil oleva lehe peal.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 31-l leheküljel, viit peatükki, 16 joonist, ühte tabelit.

Abstract

The aim of this work was to create a new website management software Administrators panel in Piletilevi for Baltic Ticket Holdings Group. New program had to be more user-friendly, reduce the time of tasks and be more flexible. The author is the project manager for the whole work.

The new software is based on the old program Adjuster. This means that the general structure of the new program remained the same, but the modules, functionalities and user interface was changed.

The creation of a new program was necessary because the old program has been built with outdated development methods and on a platform that is no longer being developed or supported. That is why it is not possible to add new complicated functionalities and modify the existing. The Administrators panel is based on modern development methods and user interface requirements. The new software is based on the Bootstrap template. Style book was used as an example to create a design to new program. (That style book was created to company during the development of the sales program.)

Users performed various tests (usability tests, manual tests), carried out by author, in both, the new and the old program. This was necessary to identify difficult or time-consuming functionalities or modules. Author analyzed all the test results and found the best possible solutions for all problems and reported it all to developers.

During the work, some of the following changes were made (or the development process is in progress): new program design, updated user system, drag and drop functionality to move table rows, tooltips were added for the difficult functionalities, etc.

Right now it is possible to use the Administrators panel only on test web page and test database. However, the program has been developed so far that it is possible for the company to implement it on the live web page.

The thesis is in Estonian language and contains 31 pages of text, five chapters, 16 figures, one table.

Lühendite ja mõistete sõnastik

<i>API</i>	<i>Application Programming Interface</i> , rakenduse programmeerimise liides
<i>BTH</i>	Baltic Ticket Holding AS
<i>CSM</i>	<i>Client Side Modification</i> , kliendipoolne muutmise võimalus
<i>CSS</i>	<i>Cascading Style Sheets</i> , stiililehe keel
<i>HTML</i>	<i>Hypertext Markup Language</i> , veebibrauseris dokumentide kuvamiseks mõeldud standardne keel
<i>JS</i>	<i>Javascript</i> , laialdaselt kasutusel olev programmeerimiskeel
Kasutusversioon	Inglise keelne vaste antud töös on “live”. Tegemist on versiooniga, mille puhul on lehekülgedele kõigil avalik ligipääs ja kasutamise võimalus ning sellele vastav oma andmebaas.
<i>MySQL</i>	<i>Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL)</i> , avatud lähtekoodiga relatsiooniline andmebaasisüsteem, mis põhineb struktureeritud päringute keelel ja mida kõike toetab Oracle.
<i>PHP4</i>	<i>Hypertext Preprocessor</i> , hüperteksti eelprotssessor. PHP on programmeerimiskeel. Number neli näitab verisooni numbrit.
<i>UML</i>	<i>The Unified Modeling Language</i> , modelleerimiskeel
<i>Widget</i>	Veebilehe komponent. Antud töös on selle all mõeldud Piletilevi ürituste ostuprotsessi, mida on võimalik anda kasutamiseks kolmandatele osapooltele.

Sisukord

1	Sissejuhatus	9
1.1	Üldine taust	9
1.2	Probleem.....	9
1.3	Eesmärk.....	10
1.4	Töö struktuur	11
2	Metoodika.....	12
2.1	Objekti ülevaade.....	12
2.2	Tööriistade ülevaade	13
2.3	Tööprotsessi ülevaade	14
3	Töö tulemused	17
3.1	Kasutajaliides	18
3.2	Funktsionaalsed nõuded	21
3.3	Testimine.....	26
4	Analüüs ja järeldused.....	30
4.1	Kasutajaliides	30
4.2	Funktsionaalsed nõuded	34
4.3	Testimine.....	36
4.4	Võimalused tulevikus.....	38
5	Kokkuvõte	39
	Kasutatud kirjandus	40

Jooniste loetelu

Joonis 1. UML paketi diagramm, mis näitab ettevõttes kasutatava tarkvara struktuuri [3].	13
Joonis 2. Administraatorite paneeli kasutusjuhude diagramm.	17
Joonis 3. Näide radio button valikust uues programmis.	19
Joonis 4. Kasutajavoog bänneri pildi vahetamisest. Vasakul vana programm ja paremal uus.	20
Joonis 5. Kuvatõmmis vanast programmist, kus on võimalik lisada näiteks bännerite pilte.	20
Joonis 6. Kuvatõmmis uuest programmist bännerile pildi lisamise funktsionaalsusest.	20
Joonis 7. Näide tööriistavihjest uues programmis.	21
Joonis 8. Vasakul uues programmis kasutaja parooli kuvamine ja paremal vanas programmis parooli kuvamine.	22
Joonis 9. UML digramm kasutajate olekutest Administraatorite paneelis [3].	24
Joonis 10. Kasutajat tevitav tekst tabeli rea tõstmisel.	24
Joonis 11. Kuvatõmmis vanas programmis andmete muutmisest.	25
Joonis 12. Kuvatõmmis uuest programmist andmete muutmisest.	25
Joonis 13. Kuvatõmmis kasutajale näidatavast märguandest Facebookis.	32
Joonis 14. Kuvatõmmis Adjusterist.	33
Joonis 15. Kaks kuvatõmmist Administraatorite paneelist. Vasakul esialgu rakendatud kujundusest ning paremal parendustega.	33
Joonis 16. Kasutajale esitatav juhend ja võimalused edasiseks toimimiseks programmis. Vasakul mitte kasutajasõbralik näide Juno-st ja paremal kasutajasõbralik näide Windows-ist. [21].	36

Tabelite loetelu

Tabel 1. Testimise tulemused Adjusteris kolme erineva kasutajakogemusega kasutaja peal..... 28

1 Sissejuhatus

Piletilevi kuulub ettevõtte Baltic Ticket Holding AS (BTH) alla (sinna kuuluvad veel ka Lätis Bilesu Serviss SIA, Leedus Bilietai ja Valgevenes Kvitki). Kõik BTH grupi alla kuuluvad firmad pakuvad piletivahendus- ja turundusteenuseid. [1]

1.1 Üldine taust

Antud projekti käigus luuakse BTH grupis kasutusel olevast vanast programmist Adjuster uus ja kasutajamugavam tarkvara Administraatorite paneel. Adjuster ning selle põhjal loodav uus programm kujutab endast BTH grupi alla kuuluvate ettevõtete veebilehtede erinevate osade: reklaamide, API-de (*Application Programming Interface*, rakenduse programmeerimise liides), *widgeti*-te (veebilehe komponent. Antud töös on selle all mõeldud Piletilevi ürituste ostuprotsessi, mida on võimalik anda kasutamiseks kolmandatele osapooltele.), haldamise programmi.

Nii vana kui ka uue tarkvara peamisteks kasutajateks on tarkvara administraator (töö autor), projektijuhid ning turundus- ja müügi-osakond. Vana programmi analüüsimise ning uue programmi loomise ja testimisega tegeletakse Piletilevis. Peale programmi täielikku valmimist testitakse töö autori juhendamisel seda ka teistes riikides ning tehakse nende jaoks vajalikud kohandused. Kogu programm luuakse kõikide riikide kasutajate jaoks inglise keeles, kuid programm peab olema üles ehitatud nii, et jääks ka tulevikus võimalus lisada erinevaid keeli.

1.2 Probleem

Kõik probleemid vana programmi kasutamisel ja arendamisel tulenevad peamiselt sellest, et Adjuster on loodud tuginedes vanadele arendusmeetoditele (näiteks on kasutusel platvormid ja keeled, mida enam ei arendata) ning algselt ei oldud mõeldud ega arvestatud suurte edasiarendustega. Vana tarkvara kasutamine ei vasta hetkel enam täielikult ettevõtte vajadustele. See piirab uute keerukate, kuid lihtsasti kasutatavate funktsionaalsuste loomist, mis omakorda takistab veebilehe kaasajastamist ja mitmekülgsemaid haldamise võimalusi.

Teiseks probleemiks on kasutajamugavus. Hetkel kasutatavat programmi on uuel kasutajal väga keeruline selgeks õppida. Vana programmi struktuur on üles ehitatud keeruliselt (Adjusterisse

lisatud funktsionaalsuste ja moodulite puhul ei ole läbi mõeldud näiteks nende paiknemine struktuuripuu) ning seetõttu vajalike funktsionaalsuste üles leidmine on raske.

Vana tarkvara on küllaltki paindumatu ehk erinäolisemate lahenduste arendamine on peaaegu võimatu. Lisaks ei ole Adjusterile tehtud moodulite refaktoreerimisi, vaid on lisatud kiireid uusi funktsionaalsusi, ilma vanu parandamata. See on viinud selleni, et programm sisaldab paljusid funktsionaalsusi, mis ei tööta või mida enam ei kasutata.

Kolmas probleem on see, et puudub täpne teadmine Adjusteris olevate funktsionaalsuste töötamise ja kasutamise võimaluste kohta, sest pole loodud juhendeid ega dokumentatsioone programmi erinevate osade jaoks. Hetkel on olemas vaid üksikute funktsionaalsuste kirjeldused, mis ei ole täielikult kaasajastatud. Selline olukord on tekkinud vähese testimise ning programmi aeg-ajalt erinevate moodulite töötamise ja kasutamise mitte kontrollimisega.

1.3 Eesmärk

Antud töö eesmärk on luua uus Piletilevi.ee, teatrix.ee, kinkekaart.ee, Bilesuserviss.lt, Bilietai.lv, kvitky.by, vpodarok.by veebilehti haldav tarkvara Administraatorite paneel. Lisaks veebilehtede haldamisele peab tarkvara võimaldama hallata *widget*-eid, maandumislehti ja API-sid ning luua ja kasutada uusi funktsionaalsusi, mis on vajalikud erinevate veebilehtede haldamiseks.

Uus programm peab olema kasutajasõbralik ehk sisaldama vaid lihtsasti kasutatavaid ning kergesti mõistetavaid funktsionaalsusi ja mooduleid. Administraatorite paneeli disain peab olema loodud selliselt, et see lihtsustaks kasutajale erinevate tööks vajalike osade leidmist ja mõistmist. Nõuetele vastavust ning programmi töötamist testitakse põhjalikult võimalikult paljude inimeste poolt (viiakse läbi nii uute kui ka kogunud programmi kasutajatega erinevaid testimisi) töö autori poolt määratud testmeetodeid kasutades.

Uus programm peab olema üles ehitatud kasutades kaasaegseid arendusmeetodeid. Näiteks kasutatakse programmeerimiskeeli, mida arendatakse järjest edasi ja mis on seega jätkusuutlikud, programmi kood ei tohi sisaldada kohti, kus põhifunktsionaalsustesse on kirjutatud konkreetseid väärtusi. Kogu programmi käigus üritatakse lähtuda agiilsest arendusmeetodist ehk näiteks programmeerimine ning testimine toimuvad paralleelselt [2]. Seeläbi on võimalik refaktoreerida vanu mooduleid ja luua uusi vajalikke funktsionaalsusi. Esimesena alustatakse uude programmi

olemasolevate moodulite refaktoreerimise ning ümberkujundamisega. Peale seda hakatakse juurde looma ka uusi funktsionaalsusi.

1.4 Töö struktuur

Töö algab vana programmi analüüsiga, mida viib läbi töö autor. Analüüsi kaasab töö autor kõik Adjusterit kasutavad isikud. Analüüsi käigus arutatakse järgnevaid punkte:

- funktsionaalsused, mis on vajalikud ning reaalselt kasutatavad,
- funktsionaalsused, mille kasutamine on keeruline ning valmistab probleeme,
- ettepanekud funktsionaalsuste kasutajasõbralikumaks muutmiseks inimestele, kes kasutavad neid oma töös peaaegu igapäevaselt,
- ettepanekud uute funktsionaalsuste loomiseks.

Kõige selle põhjal koostab töö autor koos arendajatega projektiplaani, kus tuuakse välja kõik töösse minevad moodulid ning määratakse nende loomise järjekord.

Vana programmi analüüs annab võimaluse luua uude programmi parema kasutajaliidese, mis on töö üks eesmärkidest ning millele keskendutakse töö esimeses osas. Lisaks programmi kasutajamugavuse parendamisele annab vana programmi analüüs võimaluse kustutada ebavajalikud funktsionaalsused. Nii on võimalik luua tarkvara, mis sisaldab vaid kasutuses olevaid ja töötavaid mooduleid.

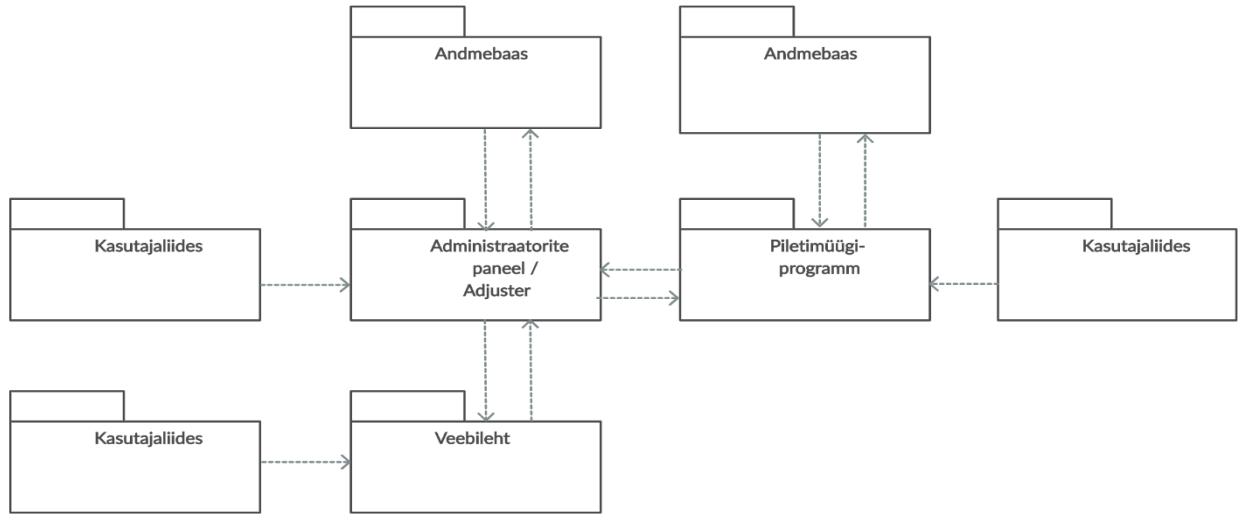
Töö viimaseks osaks jääb kogu uue programmi testimine (töö autor on testide ettevalmistaja ja läbiviija ning valitud isikud Piletilevist on testijad). Esialgsed testimised Administraatorite paneeliga tehakse testveebilehel, millele on külge pandud koopia päris andmebaasist (see saab olema testandmebaas). Seal lehel on võimalik testida algseid funktsionaalsusi ning näha üldist ülesehitust. Testkeskkonnas testides otsitakse vigu, mis on tekkinud uue programmi arenduse käigus või mida varasemalt vanas programmis kasutades ei leitud. Peale vigade parandamist ning programmi viimistlemist, viiakse kõik üle juba kasutuskeskkonnale.

2 Metoodika

2.1 Objekti ülevaade

Attitude Adjuster (või Punane CMS (*Client Side Modification*, kliendipoolne muutmise võimalus)) on arendatud NuAttitude ja/või Mikare poolt umbes aastatel 2003-2004. See on arendatud kasutades PHP4 (*Hypertext Preprocessor*, hüpertexti eelprotsessor. PHP on programmeerimiskeel. Neli tähendab verisooni numbrit.), ning MySQL andmebaasi. Esialgu loodi selle peale Piletilevi portaal aastatel 2007-2008. Aastate jooksul muutusid aga nõudmised funktsionaalsuste osas suuremaks ja keerulisemaks. Näiteks oli vaja lisada ostuprotsesse ning luua andmevahetusi müügiprogrammi sisestatud üritustega. Nii tuligi avalik osa tõsta teistsugusele süsteemile ja seega jäi Adjusterisse vaid veebilehe administreerimine. Seal loodi andmevahetus portaali ning müügiprogrammiga. Samuti lisati Adjusterisse juurde uusi mooduleid, mis olid vajalikud veebilehe haldamiseks.

Hetkel kasutatakse Adjusteris peamiselt järgnevaid moodulitest koosnevaid gruppe: veebilehtede haldus (portaalid, maandumislehed), seaded (kasutajate haldus, kasutajategrupid, veebilehe keskkonna üldhaldus, otsingusõnade määramine, suunavate linkide koostamine, veebipoes olevate klientide arvu määramine), üldmoodulid (programmis vastavalt inglise keeles "*generic modules*"). Sinna alla kuuluvad müügipunktide info, API ja kolmandatele osapooltele jagatava veebipoe haldus), tootekataloog (kontserdid, üritused, toimumiskohad, korraldajad), bännerite süsteem (sisaldab erinevat liiki bännerite haldust).



Joonis 1. UML paketiagramm, mis näitab ettevõttes kasutatava tarkvara struktuuri [3].

2.2 Tööriistade ülevaade

Kogu töö jooksul kasutatakse erinevaid programme, mis aitavad kaasa uue programmi loomisele. Kuna üheks oluliseks punktiks tööprotsessis on programmiga seotud isikutega suhtlemine, siis kasutatakse selleks Skype'i. Seal suhtleb töö autor arendajatega ning Piletilevi töötajatega (kes on programmi peamised kasutajad) nii kirjutades kui ka läbi kõnede.

Lisaks Skype'ile toimub üks osa aruteludest arendajatega läbi Mantis Bug Trackeri, kuhu töö autor või arendaja lisab iga probleemi või muudatuse kohta tööülesande või juhtumi (inglise keeles "task"). See on põhiline koht, kus hoitakse kogu jooksval töö järgi ning Skype on mõeldud pigem kiirete küsimuste ja paranduste aruteluks. Arendajatega koostatakse projekti tööplaani Google Sheets-i, sest nii on võimalik mõlemapoolselt jooksvalt jälgida projekti suuremat üldist käiku. Antud dokumendis jaotus kogu projekt moodulite alusel järgnevatel osadeks:

- moodulid, mida plaanitakse võtta töösse,
- moodulid, millele on koostatud plaan, kuid ei ole veel töösse võetud,
- moodulid, mis on parasjagu töös,
- moodulid, mis on valmis ning vajavad testimist koodi poolelt,
- valmis moodul testimiseks koos lisatud ajakuluga, mis selle tegemiseks läks.

Individaalseteks tööriistadeks on töö autoril järgnevad programmid:

- pilditöötlusprogramm Gimp,
- ekraanipiltide ja -videote tegemise programm Snagit Editor,
- kiirete märkmete tegemiseks Notepad ja Sticky Notes,
- UML diagrammide joonistamiseks Creately,
- suuremahuliste osade ja dokumentide kirjutamiseks Microsoft Word ja Excel.

Koosolekute ja testimise jaoks kasutatakse kontoris asuvat koosolekute nurka. Seal on võimalik kõik koos ümber laua istuda ning ühisele suurele ekraanile vajadusel arvutist pilti jagada.

2.3 Tööprotsessi ülevaade

Kogu tööprotsessi juhtimisega tegeleb töö autor, kes viib vajadusel läbi erinevate osapooltega (arendajate, Piletilevi töötajate ning harvadel juhtudel teiste riikide töötajatega) koosolekuid ja arutelusid. Vajadusel suheldakse ka otse isikuga/isikutega, kellelt on vaja saada tagasisidet või informatsiooni.

Töö algab suhtlusest arendajatega, kellega pannakse koos paika uue programmi jaoks kasutatav raamistik ning valitakse välja aluskujundus. Programmi raamistiku valimisel lähtutakse arendajate soovitudest, ettevõtte eelistustest ning vajalike kujundus- ja funktsionaalsete elementide olemasolust. Uue Administraatorite paneeli jaoks valiti kaasaegne alusraamistik, mis põhineb HTML-i (*Hypertext Markup Language*, veebibrauseris dokumentide kuvamiseks mõeldud standardne keel), CSS-i (*Cascading Style Sheets*, stiililehe keel) ja JS-i (*Javascript*, laialdaselt kasutusel olev programmeerimiskeel) raamistikul Bootstrap [4].

Seejärel tuleb määrata töö autoril, lähtudes ettevõtte soovidest, funktsionaalsed nõuded ning esitada need arendajatele. Nende välja selgitamiseks kutsutakse kokku kõik isikud, kes on programmi kasutamisega seotud ning palutakse neil rääkida, mida nad kasutavad. Kogu saadud info analüüsitakse tarkvara administraatori poolt läbi, kes koostab arendajatele kirjaliku teksti, kus on välja toodud iga mooduli kohta info lähtudes kolmest erinevast otsusest:

1. Funktsionaalsus või moodul on vajalik - tehakse refaktoreering ja integreeritakse Administraatorite paneeli.
2. Funktsionaalsus või moodul ei ole vajalik - arendajad kontrollivad seoseid teiste moodulitega ning kui sellel ei ole seost mõne teise mooduli või funktsionaalsusega, siis

seda ei viida üle uude programmi. Seose korral aga kirjeldatakse see ära ning funktsionaalsus või moodul rakendatakse uude programmi eelmise punkti järgi.

3. Piletilevi töötajatele ei ole funktsionaalsus teada ning vajab uurimist (tundmatud moodulid, mille kasutamise kohta ei omata informatsiooni). Moodulid, mille kohta Piletilevi töötajatel puudub piisav informatsioon, otsivad arendajad programmi koodist võimalikku seletust ja seoseid teiste moodulitega. Arendajad esitavad kokkuvõtte leitud Piletilevi tarkvara administraatorile, kes vaatab selle üle kas üksi või koos teiste töötajatega ning edasi toimitakse vastavalt eelmises punktis seletatule.

Peale esialgseid arutelusid töö autori, Piletilevi töötajate ja arendajate vahel, pannakse paika tööplan, peale mida saavad arendajad hakata programmeerima Administraatorite paneeli. Kogu arendamise käigus tehakse vajadusel täiendavaid koosolekuid ning arutelusid või muudetakse vastavalt vajadusele esialgu planeeritud töid (see on üks osa agiilsest arendusest). Arendajatel on võimalik jooksvalt küsida küsimusi töö autorilt, et täita võimalikult täpselt ettevõtte nõudeid.

Samal ajal, kui arendajad loovad uue programmi jaoks põhja, hakkab töö autor vana programmi peal tegema kasutatavuse teste. See on vajalik selleks, et selgitada detailselt välja kitsaskohti, mida on vaja uue tarkvara jaoks kindlasti parandada. Kõik sellise testimise käigus välja tulevad probleemid paneb töö autor kirja ning raporteerib arendajale, kes vajadusel muudab esialgu planeeritud funktsionaalsuse või mooduli ülesehitust.

Peale Administraatorite paneeli valmimist testandmebaasi ja -veebilehe peal, hakatakse seal nii töö autori kui ka teiste kasutajatega läbi viima testimisi, mis peaksid välja tooma esialgu märkamata jäänud probleeme ning arenduse käigus tekkinud programmi töötamise vigu. Lõpuks, kui kõik on testitud, viiakse programm kasutusversiooni andmebaasile ja leheküljele ning hakatakse lisama soovitud, kuid veel tegemata uusi funktsionaalsusi.

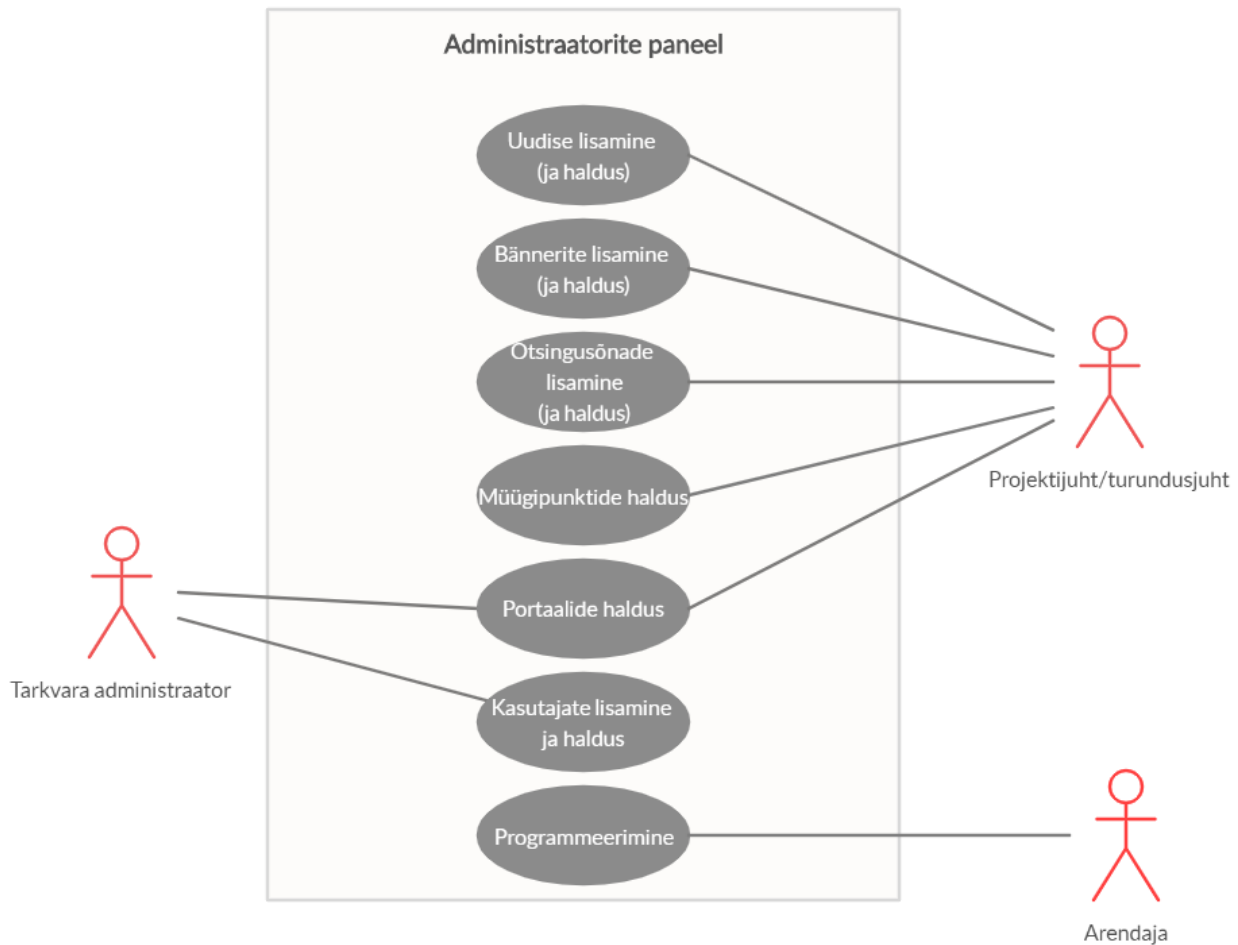
Töö autor vastutab kogu projekti juhtimise eest. See tähendab, et tema ülesanneteks on järgnevad antud töös käsitletud temadega seotud tegevused:

- kogu vana programmi analüüs,
- erinevate koosolekute planeerimine ja läbi viimine Piletilevi töötajatega ning arendajatega,
- arendajatele nõudmiste, lahenduste ning ettepanekute kirjapanek ja esitamine,

- arendajate jooksvatele küsimustele vastamine,
- töö planeerimine ning erinevate töösse minevate osade prioritseerimine,
- kõigi uude programmi loodud disainilahenduste ning funktsionaalsuste analüüs,
- nii uue kui ka vana programmi testide ettevalmistamine, läbiviimine ja analüüs,
- vanas ning uues programmis välja tulnud probleemidele lahenduste välja töötamine,
- administraatorite paneeli kasutajaliidese disaini parendamine.

3 Töö tulemused

Töö tulemusena valmis toimiv uus tarkvara, mis on esialgu kasutatav testandmebaasiga testkeskkonnas. Programm on arendatud aga piisavalt kaugele, et ettevõtte on soovi korral võimalik see viia üle kasutusverioonis veebilehtede ja andmebaasi (inglise keeles kasutatakse vastavalt “live enviroment” ja “live database”. Tegemist on versiooniga, mille kõigile lehekülgedele on avalik ligipääs ja kasutamise võimalus ning sellele vastav oma andmebaas.), peale. Joonisel 1 on kujutatud UML (*The Unified Modeling Language*, modelleerimiskeel) diagrammi Administraatorite paneeli kasutusjuhtudest (välja on toodud vaid need, mida töös käsitletakse) [3]. (Joonisel 1 tähistab töö autorit tarkvara administraator.)



Joonis 2. Administraatorite paneeli kasutusjuhtude diagramm.

3.1 Kasutajaliides

Töö tulemusena valmis kasutajasõbralikum ning parema disainiga kasutajaliides. Programmi kujunduse (värvitoonid, kujunduslikud elemendid, teksti stiil jne) aluseks kasutas töö autor juba varem ettevõtte jaoks loodud stiiliraamatut, mis valmis koos firmale loodud uue kassamüügisüsteemiga. Kogu disaini loomisel jälgis töö autor, et ei tekiks info üleküllust, programm ei muutuks liiga kirjuks, vajalikud nupud, lingid, tekstid jne, oleksid lihtsasti leitavad ja kõik ikoonid ning detailid üheselt mõistetavad. Sellistele punktidele mittevastavates kohtades tegi töö autor vajalikud parandused.

Stiiliraamatus on välja toodud väga täpselt kogu kasutatav kujundus:

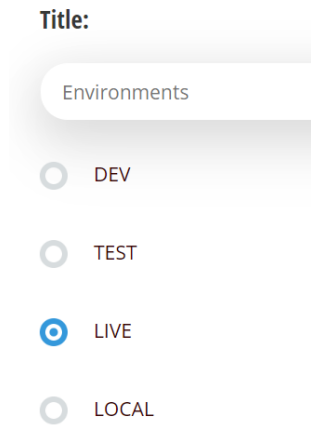
- värvid (Kokku 11 erineva variatsiooniga tooni, mis on jaotatud programmi loomise jaoks neljaks erinevaks osaks: põhitoonid (kolm), rõhutavad toonid (kolm), toetavad (kolm) ja gradient toonid (kaks).),
- teksti fondid,
- tekstide suurused ja neile vastavad paksused (näiteks hinnad kuvatakse paks(bold) ja suuruses 23 px),
- nuppude värv vastavalt funktsionaalsusele ja olekule (näiteks keeldumisnupp vaikeväärtusega on helehall),
- vahelehtede navigeerimise menüü disain,
- andmesisestamise väljad,
- valikud (*radio buttons*),
- lehekülgede numeratsioon,
- tabelite üldine kujundus (ei määra, milliseid või mis kogustes andmeid tablis esitatakse).

Peale esialgse kujunduse integreerimist stiiliraamatu alusel ning lähtuvalt aluskujunduse võimalustest, valmis uue programmi kasutajaliidese disaini esimene versioon. Algset kujundust hakkas seejärel töö autor detailsemalt üle vaatama ning tegi vajalikes kohtades muudatusi, et need sobiksid programmis antud koha sisuga kokku. Näiteks tuli muuta värvitoone (mõnes kohas oli vaja teksti muuta tumedamaks, et see oleks loetavam) või funktsionaalsuste asukohta.

Adjusteris ei olnud võimalik lisada erinevatele väljadele või objektidele piisavalt pikki nimetusi, mis suudaks selle välja funktsionaalsuse piisavalt hästi ära seletada. Töö autor vaatas uues

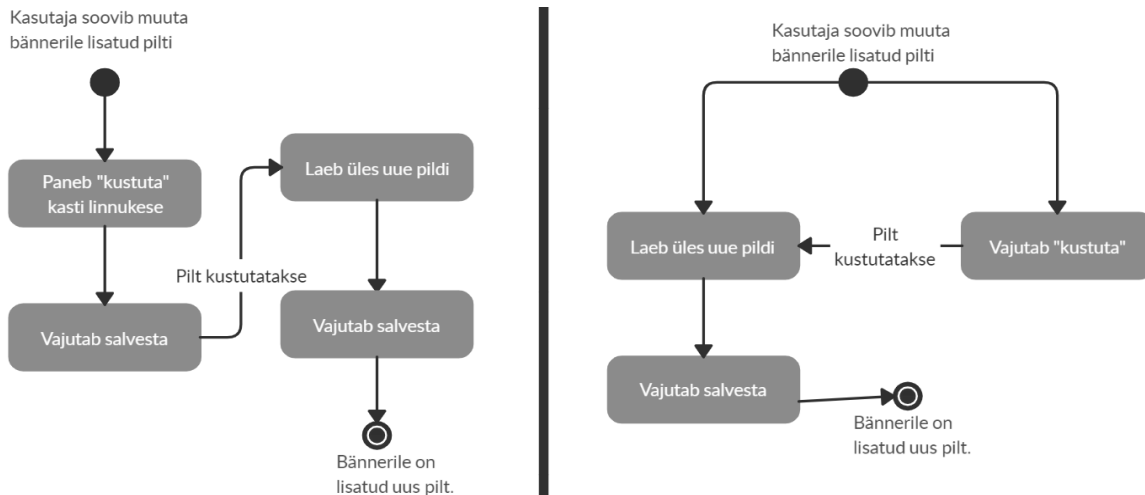
programmis kõik erinevad nimetused üle ning muutis neid vajadusel paremini mõistetavaks. Näiteks nimetati “pealkiri” (inglise keeles “*Title*”) ümber “Pealkiri Administraatorite paneeli jaoks” (inglise keeles “*Title for Admin panel*”).

Adjusteris kasutati valikute tegemisel peamiselt vaid rippmenüüsid. Uues programmis tegi töö autor ettepaneku mõnes kohas vahetada need *radio button* valiku (valik, mille puhul on välja kuvatud kõik vastusevariandid, mille igäühe ees asub ringikujuline valiku kast ning kasutaja peab ja saab neist valida vaid ühe), vastu (joonis 3). Nii tehti ümber näiteks menüüid, milles oli maksimaalselt viis erinevat valikut või kus tundus, et see variant annab kasutajatele valikutest parema ülevaate. [5]



Joonis 3. Näide radio button valikust uues programmis.

Vana programm sisaldas mõnes kohas üleliigseid klikke. Näiteks bänneri pildi vahetamiseks pidi pildi juures vajutama “kustuta” (inglise k. “*delete*”) kasti linnukese ning seejärel vajutama salvesta nuppu. Peale seda oli võimalik pilt uuesti lisada ning taaskord tuli vajutada salvesta. Selline toimimine on paljude kasutajate jaoks segadust tekitav ning võõras. Uues programmis on võimalik vana pilt kas üle laadida või vajutada “kustuta”. Mõlema tegevuse puhul vana pilt koheselt eemaldatakse ning asemele laetakse või on võimalik laadida uus. Kui uus pilt on lisatud, alles siis vajutatakse soovi korral “salvesta” (mitte salvestamise korral taastatakse seis enne kasutaja tehtud muudatusi). Joonisel 4 on bänneri pildi muutmise kasutajavoo diagrammid [6]. Vasakul vanas programmis ning paremal uues programmis. Joonistel 5 ja 6 on pildid lisamise, kustutamise või muutmise võimalused vastavalt vanas ning uues programmis.



Joonis 4. Kasutajavoog bänneri pildi vahetamisest. Vasakul vana programm ja paremal uus.

Hidden:

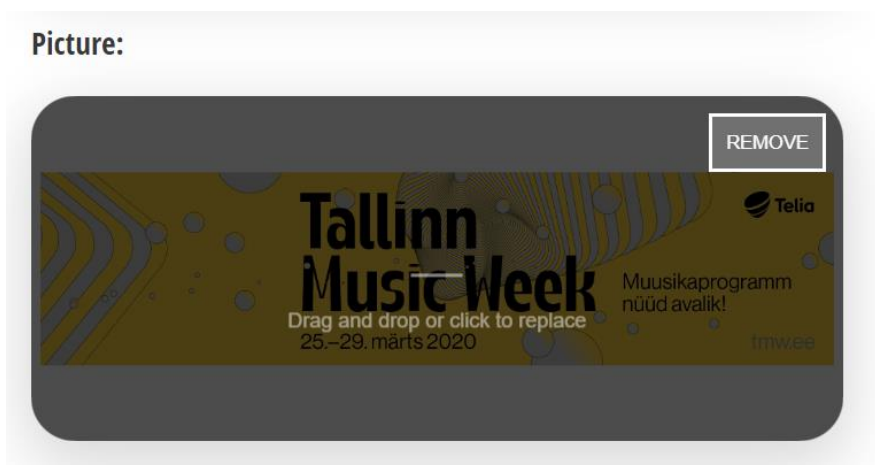
Discount size:

Image format:

Picture: Delete

Mobile picture: Delete

Joonis 5. Kuvatõmmis vanast programmist, kus on võimalik lisada näiteks bännerite pilte.



Joonis 6. Kuvatõmmis uuest programmist bännerile pildi lisamise funktsionaalsusest.

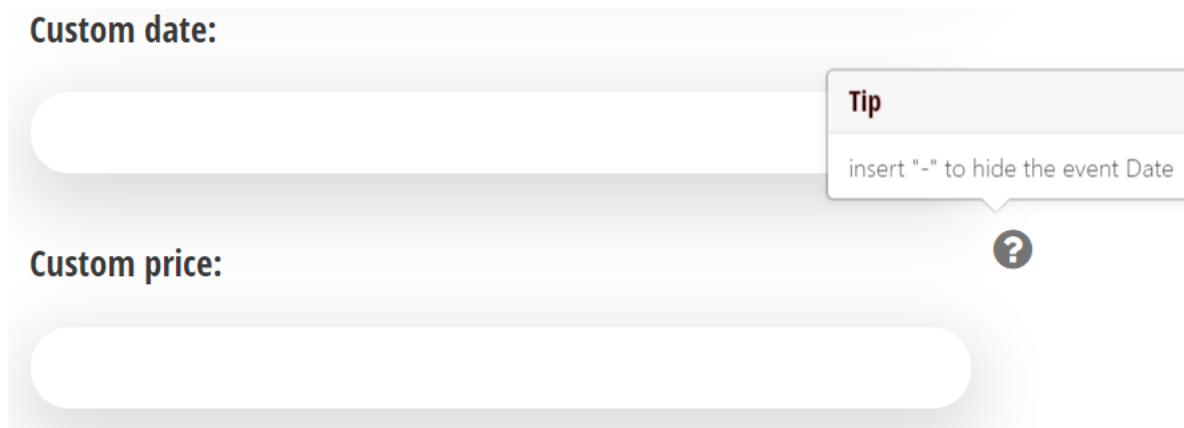
3.2 Funktsionaalsed nõuded

Kõik funktsionaalsused, mis pidid olema ka uues programmis, said refaktoreeritud ning rakendatud. Üleviimise käigus tuli parandada kasutajate jaoks erinevate funktsionaalsuste kasutamist. Selline ümbertegemine ning parendamine andis võimaluse vajalike detailide lisamiseks varem kasutatud moodulitele. Erinevate süsteemide ülesehitust ning toimivust parandati, muutes neid kaasaegsetele nõuetele rohkem vastavaks.

Funktsionaalsuste muutmise jagunes peamiselt kahte osasse: üldised muudatused, mis viiakse sisse kogu programmi ulatuses ning moodulipõhised muudatused, mis tehakse vastava mooduli all ainult selle jaoks.

Funktsionaalseid muudatusi, mis mõjutasid Administraatorite paneelis erinevaid mooduleid, ei tehtud palju, kuid need olid olulised, aitamaks muuta lihtsamaks ja kiiremaks programmi kasutajate tööülesannete täitmist. Tabelitele lisati näiteks “lohist ja lase lahti” (inglise keeles “*drag and drop*”) ridade asukoha muutmise võimalus. Kasutaja tegevus salvestatakse automaatselt peale iga tabelirea liigutamist (ning antakse sellest teada ka kasutajale).

Vana programmi suureks kasutamise probleemiks oli see, et selle kohta puudus ajakohane dokumentatsioon ning spetsialiseerunud inimene, kes teaks kõigist seal olevatest funktsionaalsustest. Nii otsustas töö autor kasutusele võtta kaasaegse tööriistavihje (inglise k. “*tooltip*”), mis lisatakse selgitava tekstiga vajalike ja keerukust tekitavate funktsionaalsuste või väljade kõrvale.



Joonis 7. Näide tööriistavihjest uues programmis.

Funktsionaalsuste parandamise suureks osaks olid kasutajaõigused ja -kontod. Varasemas programmis logis kasutaja sisse eraldi määratud kasutajanimega, mida ta võis ise alati muuta. Nii esitaski töö autor nõude, et uues programmis määratakse kasutajanime asemele töö e-posti aadress, mida kasutaja ise muuta ei saa. Teiseks probleemiks oli, et kõigile kasutajatele määras peadministraator parooli, mis oli parooli kasti kuvatud tavatekstina (kuvatud joonisel 8 paremal pool). Kogu parooli muutmine toimus tavatekstis. Uues programmis esitas tarkvara administraator nõude seda muuta nii, et seal kuvataks kasutaja parool alati varjatud kujul (tähtede, numbrite ja märkide asemel on punktid. Kuvatud joonisel 8 vasakul pool.).

The image shows two side-by-side screenshots of a user profile form. The left screenshot shows a modern form with three input fields: 'Username:' containing 'kristiine', 'First/Last name:' containing 'Kristiine Pärna', and 'Password:' containing a series of dots. The right screenshot shows an older form with a 'User data' section containing five input fields: 'Username:' with 'kristiine', 'Real name:' with 'Kristiine Pärna', 'Password:' with 'Parool123', 'Phone:' which is empty, and 'E-mail' with 'kristiine@piletilevi.ee'. Below this is a 'Groups:' section with a dropdown menu.

Joonis 8. Vasakul uues programmis kasutaja parooli kuvamine ja paremal vanas programmis parooli kuvamine.

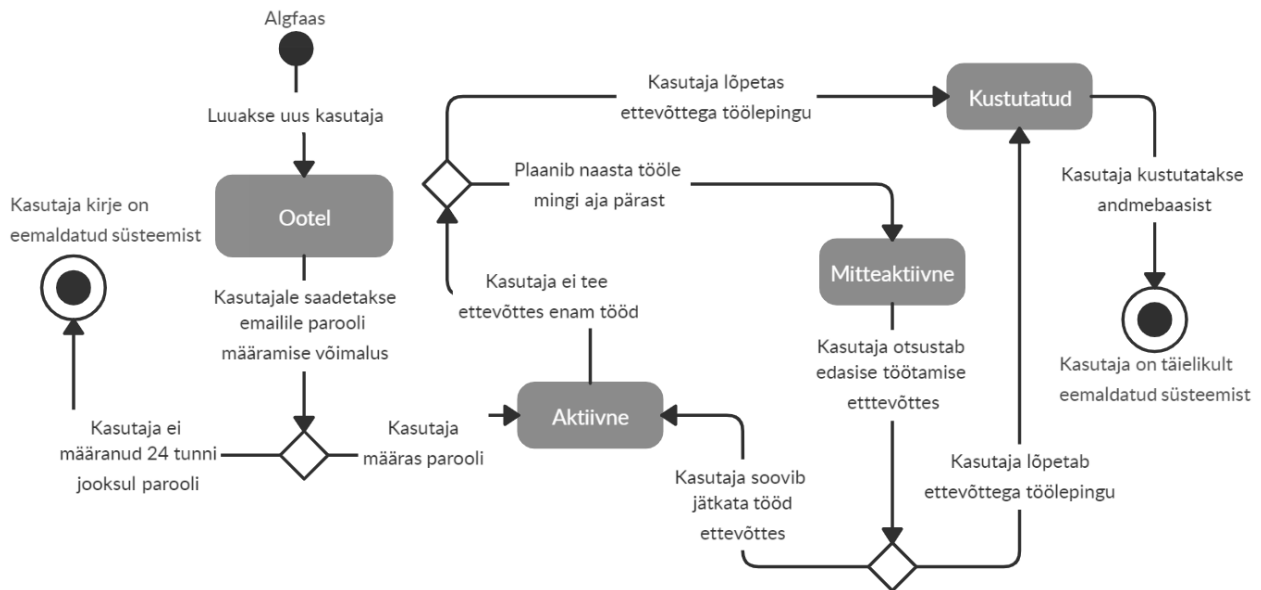
Esiialgu jäi kasutajale parooli andmine veel tarkvara administraatorile, kuid töö autori poolt said arendajatele kirja pandud konkreetset nõuded antud süsteemi muutmiseks. Peale arenduse valmimist peaks toimuma kasutajatele parooli määramine järgnevalt:

1. Kui tegemist on uue loodava kasutajaga, siis määratakse kasutajale tarkvara administraatori poolt kõik korrektsed andmed (mis peavad sisaldama õiget töömeili) peale parooli. Peale uue kasutaja salvestamist saadetakse loodud kasutajale sisestatud töömeilile link, kuhu peale klikkides on tal võimalik määrata endale parool.
2. Kui tegemist on olemasoleva kasutajaga, kes soovib oma parooli uuendada, siis see kasutaja peab enda kontole sisse logima, liikuma enda profiili alla ning sisestama vastavatesse lahtritesse uue parooli ja selle salvestama.
3. Kui tegemist on olemasoleva kasutajaga, kes on unustanud enda parooli, siis see kasutaja annab sellest teada tarkvara administraatorile. Tarkvara administraator liigub programmis parooli unustanud kasutaja alla ning vajutab seal nuppu "lähtesta parool". Peale selle nupu

vajutamist saadetakse kasutaja töö e-posti aadressile link, kuhu klikkides on võimalik endale määrata uus parool.

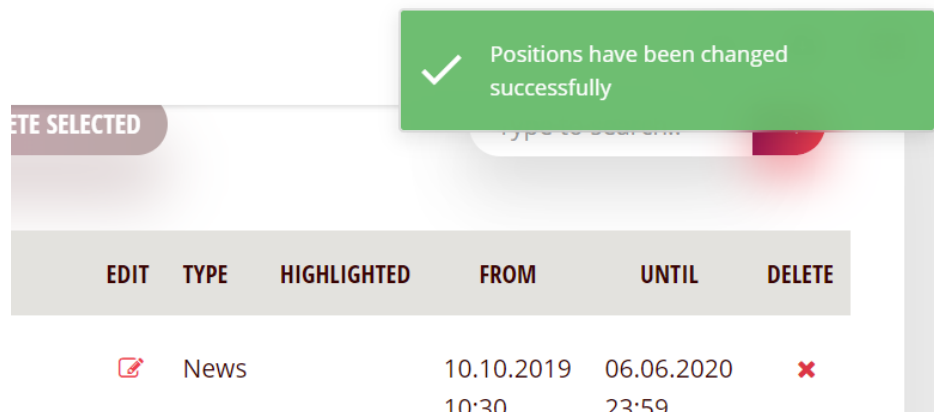
Adjusteris oli kasutajal kaks staatust: aktiivne ja kustutatud. Uude programmi tekib uue kasutajate süsteemi loomisega vajadus kasutajatele juurde luua ka uusi olekuid. Peale uue kasutajatesüsteemi valmimist võivad kasutajal olla Administraatorite paneelis erinevatel hetkedel järgnevad staatused:

1. Ootel (inglise keeles “*pending*”) – kasutaja staatus on ootel, kui ta on loodud, kuid talle ei ole veel määratud parooli. Sellise kasutajaga ei ole võimalik süsteemi sisse logida ning kirje kustutatakse andmebaasist täielikult peale 24 tunni möödumist.
2. Aktiivne – (inglise keeles “*active*”) – kasutaja on aktiivne, kui talle on määratud parool. Seega on antud kasutajaga võimalik programmi sisse logida ning kasutada õigustega määratud mooduleid ja funktsionaalsuseid.
3. Mitteaktiivne (inglise keeles “*inactive*”) – kasutajat on võimalik muuta mitteaktiivseks isiku poolt, kellele on määratud vastavad õigused. Mitteaktiivseks muudetakse kasutajad, kes on lepinguliselt ettevõttes tööl, kuid viibivad ajutiselt pikemaajaliselt töökohast eemal (näiteks lapsepuhkusel olevad inimesed). Sellisel kasutajal ei ole võimalik süsteemi sisse logida. Mitteaktiivset kasutajat on võimalik tagasi aktiivseks muuta.
4. Kustutatud (inglise keeles “*deleted*”) – selline staatus on kasutajal, kes on programmist kustutatud. Kustatud kasutajat ei ole võimalik taas aktiivseks muuta. Tema infot säilitatakse teatud aega andmebaasis ning seejärel kustutatakse täielikult.



Joonis 9. UML digramm kasutajate olekutest Administraatorite paneelis [3].

Adjusteris puudus kasutajate korrektne teavitamine peale erinevaid tegevusi. Näiteks lõi kasutaja uue kausta, artikli, uudise vms, siis teda ei teavitatud peale salvestamist, et uus objekt sai loodud. Nii tegi töö autor ettepanku uude programmi lisada juurde erinevad teavitavad märguanded, mis annaksid kasutajale märku, kui ta on vajutanud näiteks mõnda “salvesta” nuppu, midagi kustutanud või ümber tõstnud.

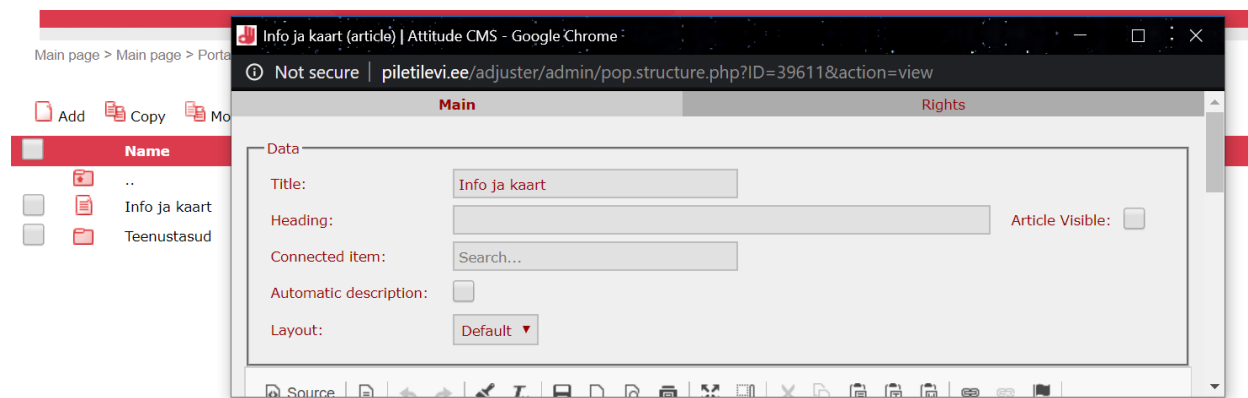


Joonis 10. Kasutajat tevitav tekst tabeli rea tõstmisel.

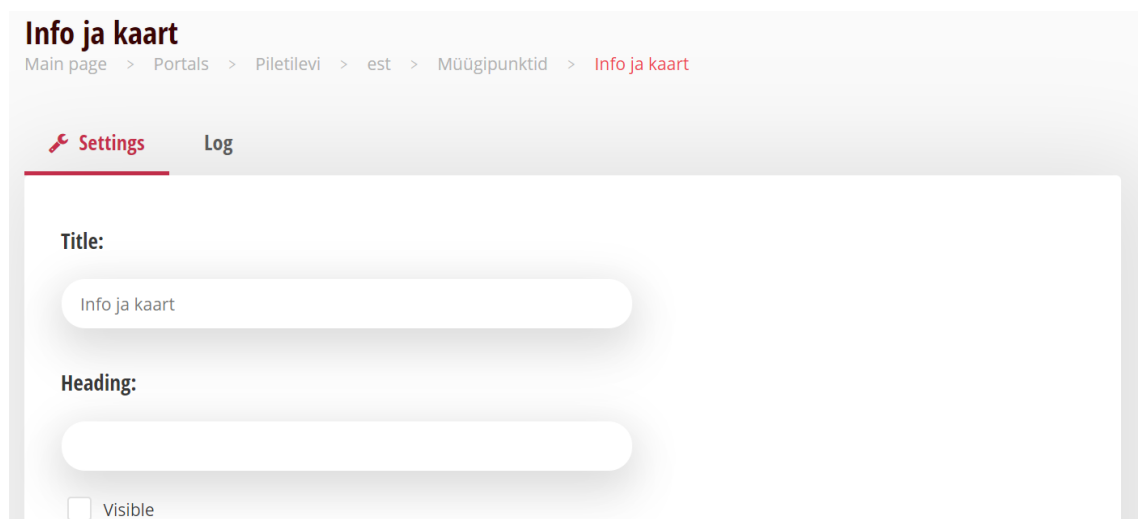
Üks funktsionaalne nõue esitati eraldi seisva mooduli “uudised” jaoks. Varasemalt selleks, et uudis püsiks esimese uudisena tabelis, pidi tema algusaega muutma käsitsi kõige suuremaks (varasemaks). Uues Administraatorite paneelis loodi selle jaoks funktsionaalsus “esiletõstetud” (inglise keeles “*highlighted*”). See võimaldab igale uudisele külge määrata linnukese

“esiletõstetud” ning tõsta ta sellega tabelis ja veebilehel kuvatuna esimeseks, hoolimata tema algusajast. (Kui mitmele erinevale uudisele on ette lisatud linnuke “esiletõstetud”, siis kuvatakse veebilehel ees pool see, mis on tabelis tõstetud üles poole.).

Üheks väga suureks muudatuseks oli loobumine hüpinkas (inglise keeles “*pop-up*”) andmete lisamisest ning muutmisest (joonis 11). Kasutaja jaoks on hüpinkas mahukate andmete muutmine ja vormide täitmine ebamugav. Samuti piirab selline lahendus ka erinevate funktsionaalsuste loomist. Olenevalt arvutist või kasutatavast veebilehitsejast võis pop-upi suurus Adjusteris varieeruda ning tuli ette palju olukordi, kus pidi kõigepealt hakkama sättima akna suurust õigeks enne, kui sai hakata täitma vajalikke tööülesandeid. Uues programmis avaneb iga programmisisene muudatus või lisamine järgmisel vahelehel ning muutub automaatselt vastavalt ekraani suurusele (joonis 12).



Joonis 11. Kuvatõmmis vanas programmis andmete muutmisest.



Joonis 12. Kuvatõmmis uuest programmist andmete muutmisest.

3.3 Testimine

Testimised toimusid mitmel erineval meetodil, et vähendada võimalikult paljude kitsaskohtade ning valesti toimivate funktsionaalsuste tekkimist. Samuti selgitati testimise käigus välja kasutajate jaoks mugavam ning enim nende tööülesannete kiirust toetav kasutajaliidese disain. Testimisel kasutati nii kogenumaid kui ka uusi kasutajaid, kes on vähesel määral või pole peaaegu üldse Adjusteri kasutamisega kokku puutunud. Kogenud kasutajate abil sai enim parandada üldist funktsionaalsust ning uued kasutajad aitasid näha kitsaskohti, mida kaua programmiga töötanud inimene harjumuse tõttu enam tähele ei suuda panna. Kõiki testimisi viib läbi töö autor. Lisaks kasutatavuse testimiste läbiviimisele, teeb töö autor ka suuremahulist Administraatorite paneeli manuaalset testimist (töös keskendutakse kasutatavuse testimise kirjeldamisele ning manuaalset testimist kästiletakse vaid lühikirjeldusena).

Vastavalt testmeetodile viis töö autor läbi testimisi mõlema programmi (nii Adjusteri kui ka Administraatorite paneeli) peal. Enne uue programmi kasutajaliidese loomist viidi esimene testimine läbi vana programmi peal. See aitas üles leida olemasolevaid probleemseid kohti, mida oleks uue programmi loomisel võimalik kohe vältida. Teine testimine viidi aga läbi juba uue programmi peal ja selle eesmärk oli välja tuua kas uusi tekkinud probleeme või varem märkamatuks jäänud mitte kasutajamugavaid kohti.

Üheks testimise meetodiks oli kasutatavuse testimine (inglise keeles *usability testing*). See on üheks peamiseks kasutajakogemuse testmeetodiks. Selle testi läbiviimiseks palutakse kasutajatel läbi viia erinevaid ülesandeid. Ülesannete andja jälgib testitavate tegevust ning küsib nendelt tagasisidet, üritades nii mõista kasutajate käitumist ja arusaamist programmist. Kasutatavuse testimise juures on võimalik keskenduda kahele erinevale variandile olenevalt oodatavast tulemusest: kvalitatiivsed või kvantitatiivsed. Antud töö juures tuli keskenduda mõlemale, kuid rohkem pandi rõhku kvantitatiivsetele tulemustele. Kvantitatiivsete tulemuste mõõtmine aitab peale uue programmi valmimist töö tulemusi paremini analüüsida ehk kas kasutajate tööülesannete täitmise kiirus on paranenud. Kvalitatiivsete tulemuste hindamine aitab toetada kvantitatiivseid ning seeläbi hinnata kas funktsionaalsuste ning kasutajaliidese uus ülesehitus on olnud õige valik. [7]

Testimine viidi läbi koosolekute laua taga, kus istusid töö autor ning testija. Mõlemal oli kaasas enda arvuti. Testija jagas enda ekraanipilti suurele ekraanile, et töö autor saaks jälgida tema

tegevust. Töö autoril olid enda arvutisse ette valmistatud testülesanded. Lisaks arvutile kasutab töö autor ülesannete läbimise aja umbkaudseks määramiseks telefoni taimerit. [8]

Adjusteri ja Administraatorite paneeli jaoks pani töö autor kokku neli erinevat ülesannet, mida ta hakkas testima kolme erineva kasutajakogemusega (mitte kogenud – kogenud) inimese peal:

1. “Kuumad hinnad” bänneri lisamine
2. Müügipunkti Tartu informatsiooni muutmine
3. Müügipunktide teenustasude informatsiooni muutmine
4. Otsingusõna lisamine üritusele

Ülesanded said valitud vastavalt sellele, mis on peamised tööülesanded antud programmis või mis on varasemalt valmistanud erinevatele kasutajatele raskuseid. Testimise käigus jälgiti nende liikumist programmis ning küsiti täpsustavaid küsimusi, miks kasutaja just nii käitus või mida tema arust mingi väli teeb. Tabelis 1 on välja toodud tulemused peale Adjusteris testimist.

Tabel 1. Testimise tulemused Adjusteris kolme erineva kasutajakogemusega kasutaja peal.

	"Kuumad hinnad" bänneri isamine	Müügipunkti Tartu informatsioon muutmine	Müügipunktid teenustasude informatsioon	Ürituse otsingsõna lisamine	
Kogunud programmi kasutaja	<p>Asus loogilises ning lihtsasti leitavas kohas.</p> <p>Ülesande täitmine oli lihtne.</p> <p>Läbimiseks kuluts maksimaalselt viis minutit.</p> <p>Leidis, et ülesande lahendamise aega saaks kindlasti teha kiiremaks ja lihtsamaks, kuid ei osanud selleks teha konkreetseid ettepanekuid.</p>	<p>Paiknemine oli natukene ebaloogilises kohas.</p> <p>Ülesande täitmine oli lihtne.</p> <p>Ülesande täitmiseks kulus neli minutit.</p> <p>Kasutaja suutis õige koha programmis leida kiirel, sest oli harjunud programmi üldise ülesehitusega ning oskas ka otsida natuke vähem</p>	<p>Paiknemine oli raskesti leitavas kohas.</p> <p>Ülesande täitmine oli lihtne.</p> <p>Ülesande läbimiseks kulus umbes 10 minutit.</p> <p>Kasutajal täpsustavaid kommentaare ei olnud.</p>	<p>Funktsionaalsus paiknes loogilises ja lihtsasti leitavas kohas.</p> <p>Ülesande täitmine oli lihtne.</p> <p>Ülesande täitmiseks kulud umbkaudne või oletatav aeg (õige koha leidmine + ülesande täitmine)</p> <p>Kasutaja lisatud kommentaarid.</p>	<p>Kui keeruline oli kasutaja jaoks leida õige koht ülesande täitmiseks?</p> <p>Kui keeruline oli kasutaja jaoks ülesande täitmine, jättes välja õige koha leidmine?</p> <p>Ülesande täitmiseks kulunud umbkaudne või oletatav aeg (õige koha leidmine + ülesande täitmine)</p> <p>Kasutaja lisatud kommentaarid.</p>
Natuke programmi varem kasutanud	<p>Paiknemine oli loogilises kohas ning lihtsasti leitav.</p> <p>Ülesande täitmine oli keeruline, sest ei olnud üheselt mõistetav kõigi väljade funktsionaalsused.</p> <p>Ülesanne on võimalik läbida iseseisvalt, kuid täiesti korrektselt ülesande täitmisele kuluts umbes 20 minutit.</p> <p>Peaks olema üheselt mõistetav, kuidas ning välti töötab. Hetkel on varem mitte ülesannet läbinud kasutaja jaoks ainsaks võimalus nende teada saamiseks kõik üks haaval läbi proovida ja vaadata, mis on tulemus.</p>	<p>Ei paiknenud loogilises kohas ning oli väga raskesti leitav.</p> <p>Ülesande täitmine oli lihtne ja arusaadav.</p> <p>Iseseisvalt läbimisele kulus umbes viis minutit.</p> <p>Kasutajal täpsustavaid kommentaare ei olnud.</p>	<p>Õige koha leidmine oli keeruline.</p> <p>Ülesande täitmine oli lihtne.</p> <p>Umbes 10 minutit.</p> <p>Kasutajal täpsustavaid kommentaare ei olnud.</p>	<p>Kui keeruline oli kasutaja jaoks leida õige koht ülesande täitmiseks?</p> <p>Kui keeruline oli kasutaja jaoks ülesande täitmine, jättes välja õige koha leidmine?</p> <p>Ülesande täitmiseks kulunud umbkaudne või oletatav aeg (õige koha leidmine + ülesande täitmine)</p> <p>Kasutaja lisatud kommentaarid.</p>	<p>Kui keeruline oli kasutaja jaoks leida õige koht ülesande täitmiseks?</p> <p>Kui keeruline oli kasutaja jaoks ülesande täitmine, jättes välja õige koha leidmine?</p> <p>Ülesande täitmiseks kulunud umbkaudne või oletatav aeg (õige koha leidmine + ülesande täitmine)</p> <p>Kasutaja lisatud kommentaarid.</p>
Uus kasutaja	<p>Funktsionaalsus paiknes loogilises kohas ning oli küllaltki lihtsasti leitav.</p> <p>Täidetavate väljade funktsionaalsused olid arusaamad ning raskendasid ülesande korrektset täitmist. Nõudis palju kontrollimist, kuna ei olnud täpselt selge, mida mingi väli kuskil mõjuab.</p> <p>On võimaline läbida iseseisvalt, kuid ajakulu võib ulatuda kuni 30 minutini.</p>	<p>Antud funktsionaalsuse leidmine oli kasutaja jaoks problemaatiline ning arusaamatu. Tema jaoks ei olnud loogiline, et üks osa müügipunktide informatsioonist asub portaali seadistuste all, aga konkreetse müügipunkti info muutmise täiesti</p> <p>Ülesande täitmine oli lihtne.</p>	<p>Antud funktsionaalsuse leidmine võitis aega, sest ei paiknenud loogilises kohas.</p> <p>Ülesande täitmine oli lihtne.</p> <p>Iseseisvalt ülesande läbimiseks võib kuluda kuni 10 minutit.</p> <p>Täpsustavaid kommentaare ei olnud.</p>	<p>Õige koha leidmine oli üsna kerge.</p> <p>Ülesande täitmine oli lihtne, kuid ülesandit natukene ebaloogiline.</p> <p>Ülesande lahendamiseks kulus umbes 10 minutit.</p> <p>Ülesande täitmiseks kulud umbkaudne või oletatav aeg (õige koha leidmine + ülesande täitmine)</p> <p>Kasutaja lisatud kommentaarid.</p>	<p>Kui keeruline oli kasutaja jaoks leida õige koht ülesande täitmiseks?</p> <p>Kui keeruline oli kasutaja jaoks ülesande täitmine, jättes välja õige koha leidmine?</p> <p>Ülesande täitmiseks kulunud umbkaudne või oletatav aeg (õige koha leidmine + ülesande täitmine)</p> <p>Kasutaja lisatud kommentaarid.</p>

Teise samade ülesannetega testimise viib töö autor läbi ka Administraatorite paneelis, kuid seda antud töö kirjutamise aja sees teha ei jõutud ning uues programmis sai testitud sarnasel meetodil vaid üksikuid uusi funktsionaalsuseid. Näiteks andis töö autor kasutajale ülesande luua uudiste blokki uus uudis, mis läheb üles homme ning see peab olema lisatud nii, et teda kuvataks alati teise uudisena, isegi siis, kui uudise üleval olemise ajal lisatakse palju teisi uudiseid. See ülesanne andis võimaluse testida kasutajate mõistmist “esiletõstetud” ning tabeli “lohist ja lase lahti” funktsionaalsustest. Testimise tulemusena selgus, et “lohist ja lase lahti” oli kasutajale märkamatu, sest tabelis puudus vastav koht kasutajale selleks märku andmiseks.

Lisaks kasutajatega testimisele viis uues programmis manuaalseid teste läbi ka töö autor. Sellise testimise peamine eesmärk oli erinevate programmivigade leidmine ehk saada teada kas kõik funktsionaalsused töötavad korrektselt. [9]

Vea leidmise korral pani töö autor selle võimalikult täpselt kirja, koos vajalike juhendavate punktidega, kus ja kuidas viga tekkis [10]. Lisaks vea kirjeldamisele pakkus töö autor välja ka võimalikke lahendusi või selgitas võimalikult täpselt, kuidas peale vea parandamist antud funktsionaalsus peaks töötama [10]. Kõik sellised vead raporteeris töö autor arendajale (läbi Mantis Bug Trackeri), kes need parandas. Selle testimise juures oli tarkvara administraatoril eelnevalt välja mõeldud ning kirja pandud juhtumid, kuidas ta soovib antud funktsionaalsust või moodulit testida. Esialgsed testid olid pealiskaudsemad, et näha kas antud funktsionaalsus üldse töötab ning seejärel liiguti alles detailsemale manuaalsele (näiteks kuidas reageerib programm, kui kirjutada vigane sisend mõnda andmevälja või määratakse alguskuupäev suurem kui lõppkuupäev) testimisele.

4 Analüüs ja järeldused

Kogu programmi arendusega ei jõutud esialgu planeeritud kaugusele (kasutusversiooni rakendamiseks), sest vahele tuli palju ootamatuid projekte. Lisaks sellele osutusid teatud moodulite arendused ajakulukamaks, kui seda oli osatud töö autori ja arendajate poolt ette näha. Nii oligi juba projekti testversiooni valmis saamise juures näha, et Administraatorite paneeli kasutusversiooni valmimine lükkub edasi.

Programmi kasutusversioonile üleminek Piletilevis oli planeeritud veeburari 2020, kuid uueks ajaks on töö autor planeerinud juuli 2020. Uua ajahinnangu andmisel lähtus töö autor hetkel uue programmi pooleli olevatest funktsionaalsustest, testimise mahust Administraatorite paneelis ning kõrvaliste projektide ajakulust.

Esialgse projektiplaani paika panemise juures sai antud liiga optimistlik ajahinnang ning ei arvestatud piisavalt paljude kõrvaliste projektide samaaegse tegemisega ning võimaliku suurema arendamise ajakuluga. Järgnevate projektide puhul tuleks kindlasti kaaluda rohkem erinevaid faktoreid, mis võivad kasutusversioonile ülemineku aega pikendada ning keerulisemate funktsionaalsuste puhul rohkem eeltööd teha, et saaks anda optimaalsema ning reaalsema ajahinnangu.

Töö autor arvab, et projekti algusfaasis enda vähese kogemuse tõttu ei olnud ta võimaline omama piisavalt head ülevaadet kogu projekti mahust ning seetõttu sai antud liiga optimistlik ajahinnang.

4.1 Kasutajaliides

Antud programmi ülesehitus keskendub eelkõige kasutajatele. [11]

Stiiliraamatu kujunduse kasutusele võtmiseks oli erinevaid põhjuseid. Üks peamisi põhjuseid oli, et ettevõtte müügiprogramm oli juba varasemalt arendatud samas stiilis ning nii saavad kasutajad hakata uut programmi kasutama juba teatud eelteadmistega. Neil on tekkinud arusaamad, kuidas süsteemis mõni nupp töötada võib või kuidas leida üles oluline informatsioon. Sellise kasutaja jaoks tuttava disaini rakendamine on hea, sest see aitab neil erinevate osade kasutamise uues programmis kiiremini selgeks õppida. Lisaks sellele hoiab antud stiili kasutusele võtmine firmasisest disaini. Kuna ettevõtte müügiprogramm on samuti loodud stiiliraamatu järgi,

siis sellist ülesehitust on paljude kasutajate peal testitud ning see teeb selle rakendamise lihtsamaks ehk on juba teada kuidas kasutaja mingites kohtades toimib ning mis on talle mugav ja mis mitte.

Lisaks juba olemasoleva müügiprogrammi pealt õpitud käitumisele on kasutajatel tekkinud ka teisi veebilehti või programme kasutades teatud ettekujutused ja ootused selles osas, kus mingi element peaks paikenema [8]. Seega tuli välja selgitada, millised on kasutajate varasemad kogemused ning nendest tekkinud ootused uue programmi elementide paiknemise ning disaini osas [12].

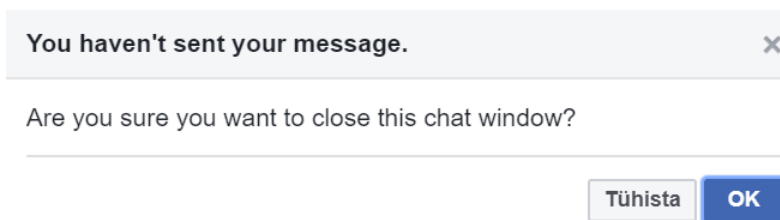
Adjusteris oli kasutusel kolm põhitooni: punane ja valge ning teksti värvina peamiselt must. Uues seevastu on valitud vähem kontrastsed toonid, mis on silmale loetavamad ja tervikuna hallatavamad. Stiiliraamatus kasutatakse erinevate variatsioonidega toone, mis aitab olla varieeruvam funktsionaalsuste disaini osas, muutes tähtsamate asjade rõhutamise lihtsamaks.

Kokku on kasutusel 11 erinevat tooni, millest igale ühele on määratud kindel kasutusala. Selline täpne määratlemine aitab hoida ühtset disaini kogu tarkvara ulatuses ning nii on elemendi tähtsus või funktsionaalsus kõigile alati visuaalselt teada. Näiteks kasutatakse “salvesta” ja “lisa” nuppudel erksamad gradient tooni, et anda kasutajale selgelt märku, et tegemist on just sellist tegevust tegeva nupuga. [13]

Erinevaid veebilehti külastades tekivad inimestel teatud ootused ja arvamused sellest, kuidas näiteks mõni nupp peaks töötama. Seega tuleb järgida kasutajate poolt kasutatavate programmide eeskuju ning disainida erinevad programmi osad ja funktsionaalsused täpselt nii nagu kasutaja seda ootab (põhinedes varasematel kogemustel). Lisaks kasutaja ootustele programmi ülesehituses, tuleb erinevate valikute (näiteks “loobu” või “salvesta”) esitamisel arvestada seda, et iga valikuga esitatakse kasutajale ülesanne, mille ta peab (tavaliselt) täitma. Kui kasutaja on parasjagu hajutatud tähelepanuga, võib tulemuseks olla ekslikult vale nupu (nupp, mida kasutaja ei soovinud vajutada) vajutamine. Seetõttu lisaks üldkasutatavatele kogemustele tuleb iga valiku puhul tähelepanelikult jälgida, et kasutaja oskaks ka vähese süvenemisega vajutada tema soovitud tegevust tegevat nuppu. [14]

Sama loogikat kasutatakse ka uues Administraatorite paneelis. Lisaks on sarnane ülesehitus paljudes tänapäeva programmides ning ka veebilehtedel. Joonisel 13 on toodud näide suurest sotsiaalvõrgustikust Facebook [15]. Joonisel on kasutajatele näidatav märguanne, kus

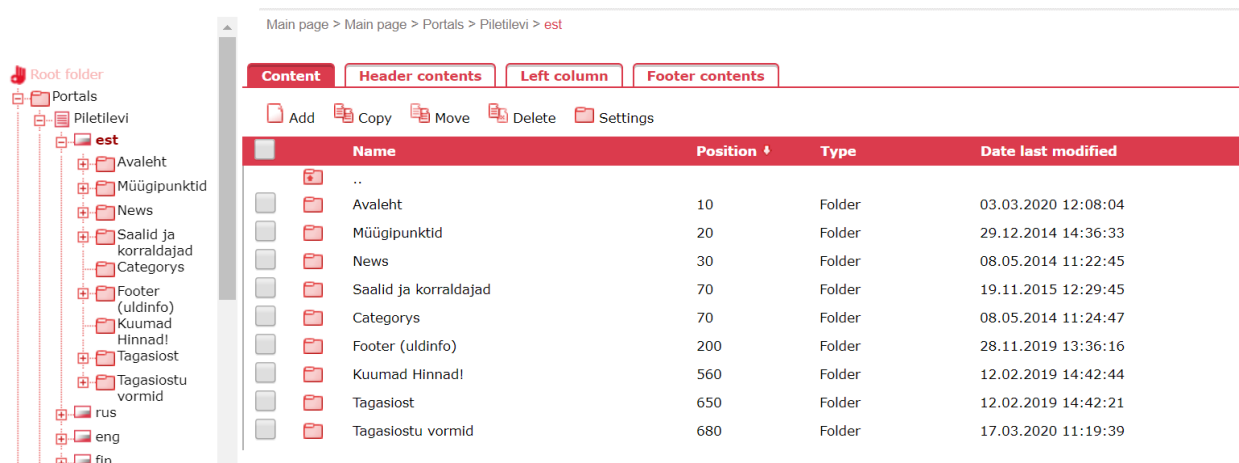
märgatavama tooniga, sinine, on kujutatud “ok” nuppu ning tagasihoidlikuma tooniga, helehall, on kujutatud “tühista” nuppu.



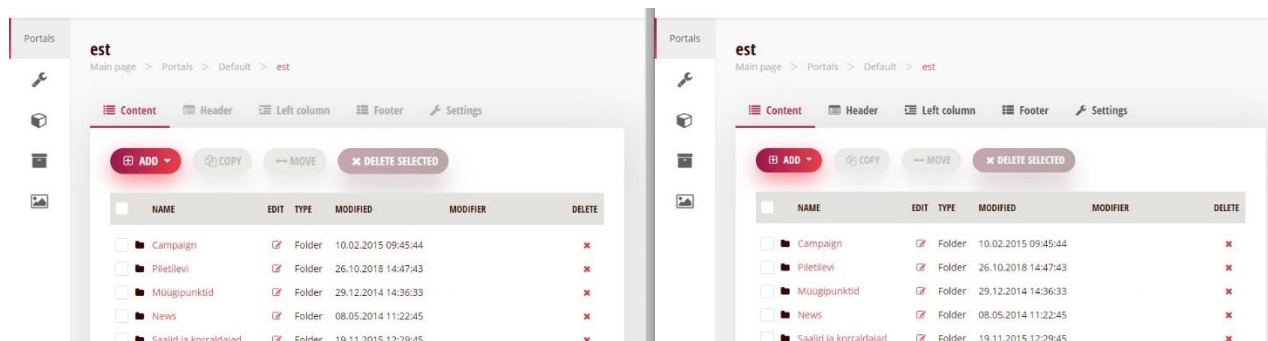
Joonis 13. Kuvatõmmis kasutajale näidatavast märguandest Facebookis.

Lisaks aitab juba tuttavate värvide ja toonide kasutamine kasutajas tekitada turvatunnet, et selle programmi kasutamine on usaldusväärne, sest kasutaja on ka varasemalt endale juba tuttavates programmides näinud sarnast või samasugust kasutust. [13]

Uue stiili rakendamine aitas uuesti kasutusele võtta funktsionaalsusi, mis olid nende keerulise ülesehituse tõttu muutunud kasutuskõlbatuteks, sest kasutajatel puudusid vajalikud teadmised ja oskused. Adjusteris on veebilehe komponentide lisamiseks eraldatud vaheaknad, kus on võimalik muuta infot vastavalt päises, sisus, vasakpoolses menüüs ja jaluses. Vanas programmis olid need kehvasti nähtavad ning seetõttu ei osanud inimesed neile tähelepanu pöörata ja seetõttu neid ei kasutanud. Nii tuligi uue administraatorite paneeli jaoks muuta seda kasutajasõbralikumaks. Esialgu sai sellele loodud algne disainilahendus, kuid see ei olnud jätkuvalt piisavalt kasutajate tegevusi toetav ning tuli veel muudatusi teha: muuta toone (tumedamad toonid on paremini nähtavad) ja elementide suuruseid ning seeläbi tekitada visuaalselt teistsugused suhted erinevate programmis kuvatavate osade vahel. Joonisel 14 on välja toodud kuvatõmmis vanast programmist. Joonisel 15 on välja toodud kuvatõmmised uuest programmist, kus vasakul on esimene versioon rakendatud disainist ja paremal hiljem parendatud variant.



Joonis 14. Kuvatõmmis Adjusterist.



Joonis 15. Kaks kuvatõmmist Administraatorite paneelist. Vasakul esialgu rakendatud kujundusest ning paremal parendustega.

Hea kasutajaliidese ülesehituse puhul on väga oluline jälgida kasutajatele ette antavaid valikuid (kuidas need on sõnastatud, kui palju neid on, kuidas need on kuvatud.) Seda on oluline jälgida, sest iga kord, kui kasutajale esitatakse valik, on ta olukorras, kus ta peab sooritama mingi otsuse. Kriitiliseks muutuvad need otsused siis, kui kasutaja ei ole huvitatud asja süvenema. [16]

Radio button kasutamine aitab kasutajatel koheselt näha ja lugeda kõiki vastusevariante kohas, kus peab kindlasti valiku sooritama. Nii saab kindel olla, et kasutajal ei jää olulises kohas tähtis valik tegemata. Kui kõik vastused on korraga välja kuvatud, siis on kasutajal lihtsam neid hallata ning võrrelda kõiki olemasolevaid variante. Seega suudab ta omakorda langetada ka kiiremini otsuse, hoides sellega kokku enda töö tegemise aega. Samuti nõuab *radio button* valik vaid ühte vajutust, kuid iga rippmenüü valik kahte ning suuremas mahus ülesannete täitmise juures võtab see kokkuvõtvalt rohkem aega. Rippmenüüdeks jäid valikud, kus oli üle viie vastusevariandi ning seega nende välja kuvamine oleks võtnud ära palju ruumi. Samuti üle viie vastusevariandiga *radio button* valik muudaks kasutajatele vastuste haldamise ning sealt õige leidmise keeruliseks. [17]

4.2 Funktsionaalsed nõuded

Töö käigus valminud uued funktsionaalsused ning juba varem olemas olnud funktsionaalsuste parendamine oli vajalik, sest Adjusterit pole suuremas mahus uuendatud umbes 10 aastat. See tähendab, et seal puudub ka kaasaegsete mudelite järgimine. Vana programm piiras uute funktsionaalsuste loomist ning olemasolevate muutmist, sest see oli loodud praeguseks juba aegunud struktuurile, millega ei olnud paindlikke lahendusi võimalik luua. Seetõttu jälgiti uue raamistiku valimisel eelkõige tema paindlikkust funktsionaalsuste loomisel.

Koos uue programmi loomisega oli oluliseks nõudeks ka välja töötada uuendatud ja turvalisem kasutajate süsteem, kus on kasutajate isiklik info paremini kaitstud ning ettevõtte pahatahtlike inimeste eest rohkem kindlustatud. Adjusteris oli võimalik igal ühel ükskõik kellele määrata parooli ja seda tavatekstina. Uude programmi aga loodi süsteem, kus on vaid kasutajal endal võimalik määrata enda kontole parool ja see on alati kuvatud varjatud kujul. Kogu kasutajate süsteemi loomisel lähtuti tänapäeval kasutusel olevatest meetoditest. Näiteks juba kasutaja loomisel saab kasutaja endale ise parooli määrata mitte seda ei määrata kellegi teise poolt. Nii saab alati olla kindel, et keegi teine ei tea kasutajale määratud parooli (kui ta seda ei kirjutanud avalikult, keegi ei rünnanud ta arvutit, see pole kuskile üles kirjutatud või muudel erandlikel juhtudel). Samuti saab parooli vahetada vaid see kasutaja, kellele see kuulub. Nii välditakse pahatahtlike isikute ründamist läbi programmi ettevõtte poole, kellegi teise andmeid kasutades.

Kasutajatele lisati paremaks kasutajate haldamiseks juurde staatuseid. Varasemalt sai kasutaja olla kas aktiivne või konto kustutati. Ettevõttes on aga alati kasutajaid, kes on näiteks lapsepuhkusel või mõnel muul põhjusel ei tee ettevõttes aktiivselt tööd. Nii loodigi selliste kasutajate jaoks staatus “mitte aktiivne” (inglise keeles “*inactive*”). Selliselt on kasutaja konto programmis alati olemas, kuid kasutajal ei ole võimalik sinna ajutiselt sisse logida. Nii säilivad programmis inimese andmed ja hiljem, kui kasutaja taas hakkab aktiivselt ettevõttes töötama, määratakse talle tagasi staatus “aktiivne”. Selline kahe staatuse vahel vahetamine aitab kaitsta paremini ettevõtte andmeid.

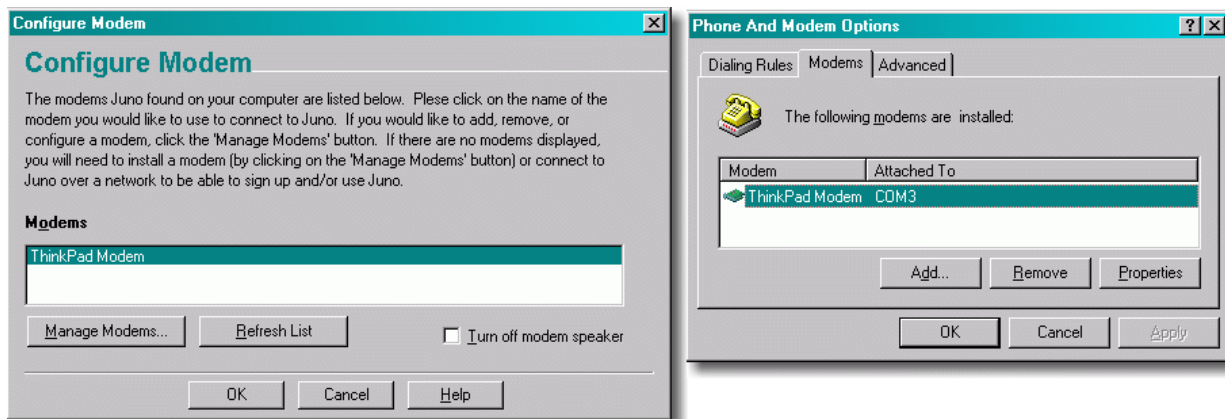
Uue programm jaoks tehti ümber tabelite ridade liigutamine. Vanas programmis määrati elemendid järjestusse vastavalt nendele antud kohanumbritele (üleval väiksem, all suurem). Kui oli vajadus tabelis asju ringi tõsta, siis tuli selleks muuta eraldi avanevas hüpinkaknas tema kohanumbrit vastavalt selliseks, kus on soov tabelis rida paigutada (selleks pidid eelnevalt täpselt järgi vaatama numbri, mida talle määrata ning selle meelde jätma).

Nii loodigi uude programmi “lohist ja lase lahti” tabeliridade ümber tõstmise lahendus. See aitab elementide paigutamisel vähendada kasutaja poolt tehtavaid samme ning ülesande täitmiseks kuluvat aega. Lisaks tekitab selliselt elementide ümber paigutamine kasutajates tunde, et nad justkui liigutaksid midagi füüsilist ning seega omavad tegevuse üle täielikku kontrolli [18].

Adjusteris puudusid kasutajate teavitamiseks erinevad sõnumid. Näiteks, kui kasutaja oli loonud uue objekti või teinud kuskil mõne muudatuse, siis peale salvestamist ei antud talle märku, kas tema tegevus toimus või mitte. Kasutajad aga vajavad sellist informatsiooni, sest on harjunud saama ka lisaks programmide kasutamisele, muudelt objektidelt reaalses maailmas tagasisidet. Näiteks lifti nupule vajutades muutub selle värv. Selline tagasiside andmine aitab neis tekitada kindlust, et nende tegevus tõesti toimus või vastupidiselt, anda märku, et midagi läks valesti. Samuti aitab selline teavitamine vältida erinevate lihtsamate vigade tekkimist ning usaldada rohkem programmi. [19]

Lisaks läbi kasutajate tegevuse kasutajatele teavitavate sõnumite näitamisele, lisati programmi ka tööriistavihjed. See parandab kasutajakogemust kogu programmi kasutamisel, sest annab kasutajale võimaluse kohe abi saada. Samuti aitavad tööriistavihjed kasutajal uusi funktsionaalsusi kiiremini selgeks õppida. Tööriistavihjed on peidus küsimärgiga tähistatud ikooni all ning seetõttu lihtsasti leitavad. Töö autor valib tööriistavihjete kuvamiste kohad ning koostab neile vastava sisu, jälgides, et juhendid oleksid võimalikult lühikesed, kuid informatiivsed. [20]

Nii teavitavate sõnumite kui ka tööriistavihjete puhul on väga oluline jälgida seal esitatavat infot. Kogu abistav info peab olema kirja pandud võimalikult lühidalt ja selgelt, sest kasutajad tihtipeale ei süvene pikkadesse tekstidesse piisavalt. See võib viia selleni, et nad jäävad ilma olulisest infost. [21] Joonisel 16 vasakul on toodud näide sellest, kuidas ei ole soovitatav kasutajale esitada olulist juhendavat infot ning paremal on toodud kasutajate jaoks lihtsamini ja kiiremini hallatavad võimalikud tegevused.



Joonis 16. Kasutajale esitatav juhend ja võimalused edasiseks toimimiseks programmis. Vasakul mitte kasutajasõbralik näide Juno-st ja paremal kasutajasõbralik näide Windows-ist. [21]

4.3 Testimine

Kasutatavuse testimine on antud projekti juures suure tähtsusega. See testimise meetod aitab mõista, kuidas programm töötab ehk kuidas tulevad sealsed kasutajad toime erinevate funktsionaalsuste kasutamisega – kas neile on kõik mõistetav või on palju segadust ja küsimusi tekitavaid kohti. Selle testimismeetodi käigus vana programmi peal tulid välja erinevad probleemid, millega võivad kasutajad kokku puutuda erinevate tööülesannete lahendamise juures ning nii oli neid uue programmi jaoks kohe võimalik paremini lahendada.

Kasutatavuse testimise juures keskenduti peamiselt kvantitatiivsetele tulemustele, sest tegemist on lõppkasutajate igapäevase töövahendiga ning mida vähem aega kulutatakse iga tööülesande täitmise peale, seda rohkem asju on võimalik antud programmis ühe päeva jooksul ära teha.

Peale testide tegemist ja kogu info kirja panemist (esitatud tabelis 1.) tuli kogu info töö autoril läbi analüüsida ning teha vajalikud järeldused. Neljast testitavast ülesandest kahe paiknemine programmis oli kasutajate jaoks loogiline. Müügipunktide teenustasude informatsiooni paremaks leidmiseks otsustas töö autor muuta struktuuripuus erinevate kaustade nimetusi, et need oleksid selgemalt märku andvad selle kohta, mis sealt kaustast leiab.

Konkreetses müügipunkti informatsiooni muutmise on aga eraldiseisev moodul ning seetõttu ei ole seda võimalik liigutada kokku müügipunktide teenustasude infoga portaali all. Probleemi lahendamiseks pakkus töö autor välja lihtsa lahenduse: müügipunktide teenustasude info bloki juurde võiks lisada teksti, kus on kirjas “Konkreetses müügipunkti info muutmise asub

üldmoodulite müügipunktide mooduli all.”. Nii on kasutajal võimalik jõuda kiirelt õigesse kohta, kui ta on ekslikult jõudnud teenustasuda informatsiooni muutmise alla.

Kõige keerulisemaks ülesandeks osutus kasutajate jaoks “kuumad hinnad” bänneri lisamine. Mitte kogenud ning vähe kogenud kasutajatele valmistas enim probleeme erinevate väljade täitmine. Nad kirjeldasid, et ei ole üheselt mõistetav, millist informatsiooni andmeväljadesse sisestama peaks ning ülesande korrektseks täitmiseks tuleks lihtsalt proovida sisestada suvalisi oletatavalt sobivaid andmeid väljadesse ja vaadata, mis tulemust see annab. Kogenud kasutaja jaoks oli ülesande täitmine lihtne, kuid mainis, et on siiski mõned väljad, mida ta ei ole kunagi kasutanud ning seega ka ei tea, mida need teevad. Kokkuvõtvalt järeldub sellest see, et selle ülesande täitmine korrektselt paari minuti jooksul on võimalik vaid siis, kui kasutajale on eelnevalt õpetatud, kuidas nende väljade täitmine oleks korrektne. Uue programmi jaoks tuli see probleem lahendada. Töö autor alustas väljade ümber nimetamisega ning nendele tööriistavihjete lisamisega (need mõtleb välja, paneb kirja ja esitab arendajatele töö autor). Peale uute nimetuste ning tööriistavihjete lisamist viiakse kasutajatega läbi uued testimised ning võrreldakse, kas tulemused muutusid paremaks. See tähendab, et olenemata kasutajakogemusest peab kasutaja olema suuteline ülesannet läbima iseseisvalt umbes sama ajaga nagu kogenud kasutaja.

Üritusele otsingusõna lisamise ülesande täitmisega suuri probleemseid kohti välja ei tulnud. Kasutajad lisasid vaid kommentaari, et uue kirje lisamine võiks asuda tabelis üleval pool mitte all. Samuti peale “lisa” nupu vajutamist võiks tegevus automaatselt salvestuda (ei peaks vajutama veel lisaks “salvesta” nuppu). Mõlemad eelpool nimetatud ettepanekud pani töö autor kirja, esitas arendajatele ning need said rakendatud Administraatorite paneeli.

Uues programmis läbi kasutatavuse testide testiti näiteks “lohist ja lase lahti” funktsionaalsusest aru saamist. Selleks andis töö autor neile konkreetse ülesande (kirjeldatud punktis 3.3), et näha kas kasutajad oskavad iseseisvalt sellist ridade ümber tõstmise funktsionaalsust kasutada. Testimise käigus selguski, et kuigi uus lahendus peaks kasutajate ülesande täitmist lihtsustama, siis tegelikkuses ei saanud nad aru, et selline uus lahendus oli lisatud. Probleemi lahendamiseks tuli mõelda midagi lisaks, et anda kasutajatele paremini mõista, et selline lahendus on loodud. Üheks lahenduseks pakkus töö autor kursori visuaalset muutmist, kui kasutaja on liikunud tabeli real kohta, kus oleks võimalik hakata rida ümber tõstma [22]. Teise lahendusena pakkus ta sinna

juurde, et tabeli rea algusesse võiks lisada vastava üheselt mõistetava ikooni, mis annaks kasutajale märku, et tabelil on “lohist ja lase lahti” funktsionaalsus [22].

4.4 Võimalused tulevikus

Uus administraatorite paneel on üles ehitatud põhimõttega, et sinna oleks võimalik vähese ajakuluga juurde arendada uusi paindlikke funktsionaalsusi. Erinevatele vana programmi kasutajatele anti võimalus tuua välja uusi ideid, mida võiks Administraatorite paneeli juurde lisada ehk millest oleks abi nende töö automatiseerimisel ning käsitsi tehtavate tööülesannete vähendamisel.

Üheks suureks sooviks avaldas müügi- ja turundusosakond bannerite kalendri loomise. See oleks selline tabel, kuhu on märgitud kõigi korraldajate poolt soovitud reklaamide info (pildid, kuupäevad, banneri liik, maksumused ja muud sellist). Hetkel luuakse käsitsi erinevaid tabeleid, kuid see aitaks luua automaatse, lihtsasti hallatava ning loetava süsteemi koos kuvatava infoga.

Teine ning väiksemamahulisem soov oli tõsta veebilehe ostuprotsessis olevate inimeste arvu statistika näitamine uude programmi. Lisaks ületõstmisele ka selle info näitamist täiustada ehk lisada juurde kogutavate andmete põhjal erinevaid visuaale. See aitab tuvastada erinevaid pudelikaelu. Hetkel asub väga lihtsa ehitusega antud funktsionaalsus eraldi veebiaadressil ning puudub võimalus detailsema info nägemiseks. See tähendab, et hetkel on võimalik näha vaid üldisi numbreid selle kohta, mitu inimest hetkelisel ajahetkel ostuprotsessis kokku on.

Programmi on vaikumisi tehtud ning jäetud võimalus ka teiste keelte loomiseks peale inglise keele. Edasiarendusena saaks luua eraldi tõlkemooduli, kus kasutajad saavad ise lisada tõlkeid erinevate keelte jaoks. Seda siis kas teatud osade või kogu programmi ulatuses.

5 Kokkuvõte

Töö eesmärk oli valmis saada Piletilevi AS-i jaoks kasutatav uus veebilehtede haldamise tarkvara, mis oleks kasutajamugavam, sisaldaks refaktoreeritud mooduleid ning võimaldaks luua uusi paindlikke funktsionaalsusi.

Programm arendati nii kaugele, et seda on võimalik kasutada testveebilehega, mille külge on rakendatud testandmebaas. Administraatorite paneeli tehti uuendatud kujul kõik moodulid, mis olid olemas Adjusteris ja mille üle tõstmise kohta oli esitatud nõudmine. Lisaks olemasolevate moodulite parandamisele, lisati ning saadeti arendamisse ka uusi funktsionaalsusi, mis on vajalikud tööülesannete täitmiseks väiksema ajakuluga.

Vana programmi üheks miinuseks oli sinna loodud kasutajaliides, mis oli kasutajate jaoks segadust tekitav ning ei olnud kooskõlas kaasaegsete nõuetega. Uue programmi disaini eeskujuks võeti juba ettevõttes kasutusel olev (ning kirja pandud) aluskujundus, mis aitab Adjusteri kasutajatel uude programmi üle minemisel seda kiiremisti selgeks õppida, sest ettevõttesiseselt on juba müügiprogramm loodud sama kujunduse ja sarnase ülesehitusega. Parima disainilahenduse, põhinedes stiiliraamatul, aitas välja töötada töö autor.

Töö autor viis kasutajatega nii uue kui ka vana programmi peal läbi kasutatavuse testimist. Adjusteris oli nii võimalik üles leida kohad, mis valmistasid kasutajatele raskusi või aeglustasid tööülesannete täitmist. Uues programmis testimine aitas välja tuua kohti, mis ei olnud tehtud veel kasutajate jaoks mugavaimaks ning kus oli vaja muuta uut lisatud disaini. Kõik tehtud testimised analüüsis töö autor läbi ning välja tulnud probleemidele leidis parimad võimalikud lahendused.

Uue programmiga ei jõutud nii kaugele, kui esialgu selle aja jooksul oli planeeritud ehk rakendada kasutusversiooni andmebaasile ja veebilehtedele. Administraatorite paneeliga aga jõuti piisavalt kaugele, et ettevõttel on võimalik avaldada soovi see tõsta kasutusversiooni veebilehtede peal kasutamiseks. Töö autor määras töö kasutusversioonile ülemineku esimeseks parimaks võimalikuks ajaks juuli 2020.

Kasutatud kirjandus

- [1] Piletilevi AS, „Piletilevi,“ Piletilevi As, 2020. [Võrgumaterjal]. Available: <https://www.piletilevi.ee/est/uldinfo/firmast/tutvustus/>. [Kasutatud 14 mai 2020].
- [2] Wikipedia, „Wikipedia,“ mai 2020. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Agile_software_development. [Kasutatud 14 mai 2020].
- [3] R. Miller, „Embarcadero,“ Embarcadero Technologies, Incv, 2003. [Võrgumaterjal]. Available: <http://edn.embarcadero.com/article/31863>. [Kasutatud 10 mai 2020].
- [4] Creation Studio, „Envato market,“ 23 juuli 2015. [Võrgumaterjal]. Available: <https://getbootstrapadmin.com/remark/material/base/forms/file-uploads.html>. [Kasutatud 4 mai 2020].
- [5] C. Holst, „Baymard institute,“ 13 november 2018. [Võrgumaterjal]. Available: <https://baymard.com/blog/drop-down-usability>. [Kasutatud aprill 2020].
- [6] C. Browne, „Career Foundry,“ 4 november 2019. [Võrgumaterjal]. Available: <https://careerfoundry.com/en/blog/ux-design/how-to-create-a-user-flow/>. [Kasutatud 10 mai 2020].
- [7] User Interviews Inc., „User Interviews,“ User Interviews Inc., [Võrgumaterjal]. Available: <https://www.userinterviews.com/ux-research-field-guide-chapter/qualitative-usability-testing>. [Kasutatud 12 mai 2020].
- [8] C. M. Barnum, „Google Books,“ 2011. [Võrgumaterjal]. Available: https://books.google.ee/books?hl=en&lr=&id=tzX3J81MAAMC&oi=fnd&pg=PP1&dq=usability+testing&ots=uV64tC_x8j&sig=7zDV5JXAp4vy3u5k_wSoWr2WLuE&redir_esc=y#v=onepage&q&f=false. [Kasutatud 20 aprill 2020].
- [9] J. Bartlett, „TestLodge,“ 9 mai 2016. [Võrgumaterjal]. Available: <https://blog.testlodge.com/what-is-manual-testing/>. [Kasutatud 11 mai 2020].
- [10] MuseScore BVBA, „Musescore,“ MuseScore BVBA, märts 2020. [Võrgumaterjal]. Available: <https://musescore.org/en/handbook/developers-handbook/getting-started/how-write-good-bug-report-step-step-instructions>. [Kasutatud 7 mai 2020].
- [11] "Wikipedia," 20 märts 2020. [Online]. Available: https://en.wikipedia.org/wiki/User-centered_design. [Accessed aprill 2020].
- [12] C. J. M. W. S. M. Debbie Stone, „Google Books,“ 2005. [Võrgumaterjal]. Available: <https://books.google.ee/books?id=VvSoyqPBPbMC&pg=PA197&lpg=PA197&dq=ui+desi>

gn+concentrates+on+users&source=bl&ots=d7Q-NWoTQ6&sig=ACfU3U0GHxLhxpJe_w4knr_n2pSWENS4gQ&hl=en&sa=X&ved=2ahUKEwjPr_vDuO_oAhUmpIsKHbvcCMEQ6AEwFXoECA0QKA#v=onepage&q=ui%20design%20co. [Kasutatud 2 mai 2020].

- [13] S. P. Adnerson, „Alistapart“, 21 aprill 2009. [Võrgumaterjal]. Available: <https://alistapart.com/article/indefenseofeyecandy/>. [Kasutatud veebruar-aprill 2020].
- [14] J. Spolsky, „Joel on software“, 24 Oktoober 2001. [Võrgumaterjal]. Available: <https://www.joelonsoftware.com/2001/10/24/user-interface-design-for-programmers/>. [Kasutatud 3 mai 2020].
- [15] Facebook Inc., „Facebook“, Facebook Inc., veebruar 2004. [Võrgumaterjal]. Available: <https://www.facebook.com/>. [Kasutatud 4 mai 2020].
- [16] J. Spolsky, „Joel On Software“, 12 aprill 2000. [Võrgumaterjal]. Available: <https://www.joelonsoftware.com/2000/04/12/choices/>. [Kasutatud 12 mai 2020].
- [17] S. Minhas, „Meedium“, 6 mai 2018. [Võrgumaterjal]. Available: <https://blog.prototypr.io/7-rules-of-using-radio-buttons-vs-drop-down-menus-fddf50d312d1>.
- [18] A. Reardon, „Rethinking drag and drop“, 16 August 2017. [Võrgumaterjal]. Available: <https://medium.com/@alexandereardon/rethinking-drag-and-drop-d9f5770b4e6b>.
- [19] A. Harley, „n N/g Nielsen Norman Group“, 26 Juuli 2019. [Võrgumaterjal]. Available: <https://www.nngroup.com/videos/usability-heuristic-system-status/>.
- [20] A. Joyce, „N N/g Nielsen Norman Group“, Nielsen Norman Group, 27 jaanuar 2019. [Võrgumaterjal]. Available: <https://www.nngroup.com/articles/tooltip-guidelines/>. [Kasutatud 11 mai 2020].
- [21] J. Spolsky, „Joel on software“, 26 aprill 2000. [Võrgumaterjal]. Available: <https://www.joelonsoftware.com/2000/04/26/designing-for-people-who-have-better-things-to-do-with-their-lives/>. [Kasutatud 12 mai 2020].
- [22] P. Laubheimer, „N N/g Nielsen Norman Group“, 23 veebruar 2020. [Võrgumaterjal]. Available: <https://www.nngroup.com/articles/drag-drop/>.