

Tallinna Ülikool

Digitehnoloogiate instituut

Koduse sahvriarenduse prototüübi loomine

Bakalaureusetöö

Autor: Patrick Mägi

Juhendaja: Romil Rõbtšenkov

Autor: ,, ,, 2016

Juhendaja: ,, ,, 2016

Instituudi direktor: ,, ,, 2016

Tallinn 2016

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

..... (kuupäev) (autor)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Patrick Mägi (sünnikuupäev: 27.11.1993)

1. annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Koduse sahvriakenduse prototüübi loomine“, mille juhendajaks on Romil Rõbtšenkov, säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis;
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile;
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tallinnas,(autori allkiri ja allkirjastamise
kuupäev)

Sisukord

Sissejuhatus	5
Kirjanduse analüüs	7
1.1 Sarnased rakendused	7
1.2 Arenduskeskkond.....	8
1.3 Versioonihaldus	8
1.4 Veebirakenduste testimine	9
2 Metoodika	10
3 Disain	12
3.1 Persoonad, stsenaariumid ja kasutajalood	12
3.2 Paberprototüübid.....	14
3.3 HTML prototüübid	15
4 Arendus	17
4.1 Platvormi valik.....	17
4.2 Rakenduse ülesehitus	18
5 Testimine.....	20
Kokkuvõte	22
Kasutatud kirjandus	24
Summary.....	25
Lisad	26
Lisa 1 Loodud paberprototüübid	27

Sissejuhatus

Paljudes peredes on siiani säilinud vana harjumus teha hoidiseid ning enamasti tehakse seda iga-aastaselt. Hoidiseid tehakse erinevatest aia- ja metsasaadustest ning seda selle eesmärgiga, et tulevikus oleks hea võtta. Olenevalt inimeste harjumustest või tavadest tehakse hoidiseid osades majapidamistes rohkem ja teistes vähem. Purkidele ja pudelitele pannakse tavaliselt sildid, kus on kirjas tegemise kuupäev ja sisu. Hoidiste arvu kasvades tekib vajadus nende üle arvet pidada. Hetkel pole saadaval hoidiste üle arvepidamiseks muud võimalust kui enda pea umbkaudne mäletamine, paberile üles kirjutamine või parimal juhul arvutisse üles märkimine.

Hoidiste tegemine on siiani populaarne ning seda tõestab artikkel, et kaubanduses täheldatakse hoidiste tegemiseks vajalike kaupade müügi kasvu (Liiva, 2015). Antud informatsiooni põhjal on autoril põhjust järeldada, et inimeste seas on selliseid, kes sooviks rakendust, millega saaks hõlpsasti oma sahvri üle arvet pidada. Seminaritöö tulemusena selgus, et hetkel saadaval olevad rakendused toetavad näiteks veinikeldri või külmkapi sisu haldamist. Hetkel ei ole üheski rakenduses ette nähtud hoidiste üle arvepidamist. Sellest tulenevalt on käesoleva bakalaureuse töö eesmärgiks lihtsa sahvri veebirakenduse prototüübi välja arendamine ja selle dokumenteerimine.

Töö autori arvates on olemas vajadus rakenduse vastu, mille abil saaks sahvrisse hoidiseid lisada, kustutada ja sahvri hetkeseisu kuvada. Seminaritöö käigus analüüsitud sarnastel rakendustel ei olnud tähelepanu pööratud eriti kasutajasõbralikkusele ning samuti oli puudusi vajalikes funktsionaalsustes.

Käesoleva bakalaureusetöö käigus viiakse läbi järgnevad etapid:

- tehakse paberprototüübid;
- luuakse HTML prototüübid;
- lisatakse prototüübile JavaScripti abil funktsionaalsuseid;
- testitakse loodud prototüüpi.

Bakalaureusetöö on jagatud viieks osaks. Esimene osa annab ülevaate läbiviidud kirjanduse analüüsist, kus tuuakse välja sarnased rakendused, arenduskeskkond ja veebirakenduste testimine. Teine osa kirjeldab autori valitud meetodikaid, mille abil viiakse läbi rakenduse arenduse etapp. Töö kolmandas osas räägib autor seminaritöö käigus valminud persoonadest ja

nõuetest ning rakenduse disaini väljatöötamisest, mille käigus loob paber- ja HTML-prototüübid. Neljandas osas kirjeldab autor arendamise käiku ning tekkinud raskusi ja probleeme. Viimane peatükk tutvustab käesoleva töö käigus valminud veebirakenduse prototüübi testimist.

Kirjanduse analüüs

Tehnoloogia areng on andnud käesoleva töö autorile tõuke valmis arendada rakendus, kus kasutaja saab lihtsa vaevaga oma sahvri üle arvet pidada. Käesolevas peatükis esitab töö autor kokkuvõtte seminaritöö käigus valminud nõuete analüüsist. Tutvustatakse sarnaseid rakendusi, kuna kavandatavate funktsionaalsustega rakendusi ei olnud. Samuti antakse ülevaade autori valitud arenduskeskkonnast, versioonihaldusest ja veebirakenduste testimisest.

1.1 Sarnased rakendused

Kõik seminaritöö käigus uuritud rakendused sarnanesid funktsionaalsuste poolest mõningal määral autori kavandatava rakendusega. Näiteks CellarTrackeri¹ rakenduses oli võimalik triipkoodi skanneerida ja keldrit hallata. Sahvirakenduse peamiseks funktsionaalsuseks oleks hoidiste üle arvepidamine väga lihtsal viisil. Sahvri haldamine peab toimuma silmapaistvate nuppudega, mille abil saab kustutada ja sahvrisse hoidiseid juurde lisada.

Sahvirakendus on keskendunud rohkem kõikide hoidiste peale, mida on võimalik sahvris hoiustada, mitte ainult veinide või külmkapi sisu peale. Sahvris saab olema võimalik hoiustada moose, marineeritud kurke, siirupeid, veine, kompotte ja palju muid häid hoidiseid. Funktsioonide poolest sarnaneb autori kavandatav rakendus Kellermeisteri² rakendusele, kuid seal on keskendutud ainult veinidele.

Hoidiste ja veinipudelite üle arve pidamine on sarnased tegevused. Kavandatava rakenduse funktsioonid sarnanevad vaadeldavate rakenduste omadega. Sahvirakenduses ei ole hoidistest pildi tegemise võimalust ega ka triipkoodide skanneerimise võimalust, kuna töö autor leiab, et esialgu prototüübi arendamise osas ei ole need funktsionaalsused vajalikud. Üheks eeldatavaks kasutajagrupiks on eakamad ja seetõttu peab olema rakenduse kasutamine eakamatele sobilik. Hoidiseid peab saama lisada ja kustutada nimekirjast lihtsa nupuvajutusega.

Sarnaselt Prep & Pantry³ rakendusele on autori arvates vajalik, et sahvri sisu oleks samuti võimalik jagada erinevate kasutajate vahel ja seisu oleks võimalik muuta ka mitmel eri kasutajal. Realiseerimiskuupäeva asemel oleks kasutajal võimalik märkida hoidise tegemise

¹ <https://play.google.com/store/apps/details?id=com.cellartracker.app>

² <https://play.google.com/store/apps/details?id=de.kellermeister.android>

³ <https://play.google.com/store/apps/details?id=com.galalooapps.preppantryandroid>

kuupäeva. Autori loodavas rakenduses oleks rohkem silmas peetud, et rakendus oleks kasutajasõbralik. Eeskujuks ei saa kindlasti võtta My Pantry⁴ rakendust, kuna selle kasutajaliides tundus mitte eriti kasutajasõbralik ja pisut kohmakas.

Analüüsi käigus selgus, et on olemas mitmeid sarnaste funktsioonidega rakendusi, kuid autor leiab, et nendel on puudusi ja neid ei saa lihtsalt ning mugavalt sahvri haldamiseks kasutada. Disaini poolest tundub vaid Prep & Pantry ja Cellar Trackeri rakenduses, et on proovitud tähelepanu pöörata ka kasutajamugavusele. Kellermesteri, Wine Cellar Free⁵ ja My Pantry rakenduste puhul on autori arvates kohmaka ja vananenud disainiga.

1.2 Arenduskeskkond

Arenduskeskkonnaks valis töö autor Atomi⁶, kuhu saab alla laadida erinevaid pistikprogramme. Atomi valis töö autor, kuna sellel on väga lihtsasti kasutatav kasutajaliides, see on tasuta ja paljudele operatsioonisüsteemidele kättesaadav. Antud töö kontekstis kasutab autor arenduseks JsHinti⁷ ja Emmetit⁸. JsHint aitab vigu JavaScripti koodis leida enne kui see veebis kuvatakse, mis muudab vigade leidmise ja parandamise kiiremaks. Emmet aitab töö autoril HTML koodi produktiivsemalt kirjutada. Selle abil saab erinevaid lühendeid kasutades lasta Emmetil kirjutada soovitud koodiread valmis. Eelnevalt loetletud pistikprogrammid aitavad autoril lihtsama vaevaga kirjutada JavaScript ja HTML koodi.

1.3 Versioonihaldus

Versioonihalduseks kasutab töö autor GitHub⁹ veebikeskkonda. GitHub on veebipõhine Git repositooriumi haldamise teenus. Antud töö kontekstis on see failihoidla, mis kasutab Git¹⁰ tehnoloogiat. See pakub koodifailide versioonide kontrolli ja lähtekoodi juhtimise funktsionaalsust. Erinevalt Gitist, mis on rangelt käsurea tööriist, pakub GitHub veebipõhist graafilist kasutajaliidest ja töötab ka mobiilsetel seadmetel. Samuti pakub GitHub juurdepääsu kontrolli ja mitmeid koostöö võimalusi, nagu näiteks vigade jälgimist, märkmete tegemist, ülesannete juhtimist ja vikisid iga loodud projekti kohta. GitHubi on võimalik kasutada üksinda

⁴ <https://play.google.com/store/apps/details?id=com.appdevandroid.framework.pantryfree>

⁵ <https://play.google.com/store/apps/details?id=com.dophan.android.winecellarfree>

⁶ <https://atom.io/>

⁷ <http://jshint.com/>

⁸ <http://emmet.io/>

⁹ www.github.com

¹⁰ <https://git-scm.com/>

arendades ja ka ettevõtte põhiselt ning selle kasutamine on levinud ka avatud lähtekoodiga projektidele, kuna selle abil on võimalik lihtsa vaevaga versioonihaldust läbi viia. Versioonihaldus tähendab seda, et failidesse tehtud muudatustest säilib ajalugu, mida saab hiljem vaadata ja vajadusel tehtud muudatusi tagasi võtta.

1.4 Veebirakenduste testimine

Testimine on tarkvara kvaliteedi järelvalve üks osa, mis on laiem protsess, sisaldades planeeritud ja süstemaatilist lähenemist hindamaks tarkvara kvaliteeti ja vastavust nõutud nõuetele. Saavutamaks testimisel võimalikult häid tulemusi, tuleb testimisega seotud töö planeerida. Eduka testimise läbiviimiseks on vaja koostada testiplaani, mis annab ülevaate testitavatest funktsioonidest. Testiplaan on dokument, mis kirjeldab testimise tegevuse skoobi, meetodid, ressursid ja ajakava. Eelnevalt käsitletud dokument on aluseks eduka testimise läbiviimisel. Edukas testimine on testijate jaoks selline testimine, kus leitakse vigu. Peale testiplaani loomist luuakse testilood, mis aitavad testijal defineerida, mida on vaja testida. Käesolevas töös võetakse loodud kasutajalood ja kasutatakse neid aluseks testilugude loomisel. See tähendab, et kasutajalugudes tekkinud funktsionaalsused on peamised testimise objektid. Kasutajalugudes tekkinud funktsionaalsuseid käsitletakse antud töös kui nõudeid rakendusele, mis tuleb täita ja mille edukat ilma vigateda toimimist peab testimise. Testimist on väga paljusid erinevaid tüüpe, kuid esmase prototüübi valmistamisel on mõistlik kaaluda valge kasti (ingl *whitebox*) testimist ja musta kasti (ingl *blackbox*) testimist. Valge kasti testimine hõlmab endas tarkvara testimist, kus kasutatakse infot testitava objekti sisemise struktuuri ehk programmikoodi kohta. Musta kasti testimine on testimise tüüp, mis käsitleb testitavat objekti musta kastina ehk testide koostamisel vaadeldakse ja analüüsitakse ainult sisend-väljund käitumist. Musta kasti testimise tehnika kasutamisel testitakse funktsioone ilma viitamata tervele rakenduse struktuurile (Board, 2011).

Töö autor viib bakalaureusetöö raames läbi musta kasti testimise silmas pidades rakenduse prototüübi kasutajalugudes kirjeldatud funktsionaalsuseid.

2 Metoodika

Välja selgitamaks, millised on rakenduse nõuded ning millised funktsionaalsused rakenduses olema peaks, kasutatakse professor Teemu Leinoneni uurimispraktikat teadusuuringutel põhinevat disaini. Teadusuuringutel põhinev disain toob välja rakenduse disainimise protsessi teadusuuringute kaudu. See on tugevalt orienteeritud prototüüpide ehitamise suunas ja rõhutab innovaatiivseid lahendusi, mis uurivad mitmeid erinevaid ideid ja disainilahendusi. Suureks märksõnaks on järjepidev testimine ja uuesti disainimine, et saada jälile kõikidele kasutaja vajadustele ja disainivigadele. Enamus tegevusi toimub dialoogis kasutajaskonnaga, kes hakkab rakendust kasutama (Leinonen, Silfvast, & Toikkanen, 2008).

Kogu disainimisprotsess koosneb neljast etapist (Leinonen et al., 2008):

- taustauuring (ingl *contextual inquiry*);
- osalusdisain (ingl *participatory design*);
- rakenduse disain (ingl *product design*);
- rakenduse prototüüp hüpoteesina (ingl *software prototype as hypothesis*).

Töö autor viis seminaritöö raames läbi esimesed kaks etappi taustauuringu ja osalusdisaini. Etappide läbimisel valmisid persoonad, stsenaariumid ja neid aluseks võttes sai autor kirjeldada kasutajalood. Läbi viidud taustauuringu tulemusena, selgitas autor välja, milline on kontekst ja millised on esmased disaini nõuded. Võimalike kasutajate kaasamisel osalusdisaini etapis sai autor teada kelle jaoks rakendust disainitakse, milline peaks olema prototüübi disain ja milliseid funktsionaalsuseid prototüübi arendamisel lisama. Peale seda sai autor luua rakendusele kasutajalood, mis kirjeldavad loodavale rakendusele vajalikke funktsionaalsuseid. Loodud persoonade, stsenaariumite ja kasutajalugudega on võimalik tutvuda käesoleva töö disaini peatükis.

Bakalaureusetöö raames proovib autor tekkinud funktsionaalsused valmis arendada ja töö käigus viib täide teadusuuringutel põhineva disaini kaks viimast etappi. Nende etappide käigus valmistab autor kavandatavale rakendusele paberprototüübid, et tekiks visioon, milline peaks rakenduse disain realselt välja nägema. Tekkinud prototüüpe saab autor kasutada edasises rakenduse arenduses. Nende abil saab autor rakenduse kasutajaliidest võimalikult lihtsalt disainida. Paberprototüüpimine annab võimaluse kasutajaliidese esmaseks visandamiseks ja loomiseks. Lihtsate paberitükkidega ja kasutajate stsenaariumeid väikeste sammude kaupa

testides saab tõsta oluliselt rakenduse kvaliteeti. Paberprototüüpide abil on võimalik luua kiiresti kasutajaliideseid ja neid täiendada (Snyder, 2003). Peale paberprototüüpide tegemist arendab töö autor veebist vaadatava ja klikitava rakenduse. Autor kasutab veebirakenduse märgendamiseks HTML ja kujundamiseks CSS keelt ning loob veebirakenduse klikitava HTML prototüübi kasutajalugudes kirjeldatud funktsionaalsustega.

3 Disain

Sahvri veebirakenduse disainimise etapis kasutati erinevaid osalusdisaini meetodeid. Autor annab käesoleva peatüki alampeatükkides ülevaate seminaritöö käigus loodud isikute, stenaariumitest ja kasutajalugudest ning disainiprotsessi jaoks olulist informatsiooni andnud paberprototüüpidest. Samuti koostab autor klikitavad HTML prototüübid koos CSS stiilireeglitega, mis annavad rakendusele soovitud väljanägemise.

3.1 Isikud, stenaariumid ja kasutajalood

Disaini esimese sammuna tegi töö autor seminaritöö raames isikud, stenaariumid ja kasutajalood. Isikuna on vahend loodava rakenduse kasutajate näitlikustamiseks, mille abil on võimalik näidata ühe arvatava sihtrühma huve. Kasutajate harjumusi täpsemalt kirjeldades lõi autor iga isikuna kohta välja millised on nende eesmärgid, mida nad selles rakenduses võiksid tahta saavutada ja milline on nende taustainfo (Brown, 2011). Sahvirakenduse prototüübi jaoks loodi kolm isikuna (vt Tabel 1), millest kaks olid sekundaarsed ja üks primaarne.

Tabel 1. Seminaritöö käigus valminud isikud (Mägi, 2016)

Nimi	Tüüp	Tegevusala	Eesmärgid
Silva	Primaarne	Raamatupidaja	Soovib lihtsat viisi, kuidas pidada oma sahvri üle arvet Soovib ülevaadet sahvril leiduvatest hoidistest
Maimu	Sekundaarne	Pensionär	Soovib lihtsat viisi, kuidas hallata oma sahvrit Soovib ülevaadet sahvril olevatest hoidistest Soovib jagada sahvri nimekirja oma perega
Martin	Sekundaarne	Õpilane	Soovib, et saaks ülevaate vanaema sahvril

Võimalikest kasutajatest parema ülevaate saamiseks lõi autor välja lisaks nimele igale isikuna kirjelduse, soo, vanuse, tegevusala ja eesmärgid. Silva on raamatupidaja, kes soovib lihtsat viisi, kuidas pidada oma sahvri üle arvet. Maimu on pensionieas vanadaam, kes soovib ülevaadet sahvril olevatest hoidistest ja hoidiste nimekirja jagada oma perega. Viimane loodud isikuna on õpilane Martin, kes soovib saada ülevaadet oma vanaema sahvril (Mägi, 2016).

Seminaritöö käigus loodi igale isikuna stenaariumid, kus kasutajat kujutatakse ette rakendust kasutamas. Stenaariumite abil sai töö autor kirjeldada konteksti ja tegevusi, mida

rakendust kasutav kasutaja võiks teha. Kirjeldatud lugudega sai autor aimu, milliseid funktsionaalsuseid arendataval rakendusel vaja võiks minna (Rosson & Carroll, 2002).

Esimene stsenaarium kirjeldab koduperenaise Silva kogemust rakendusega, mille abil saab oma sahvri üle arvet pidada. Ühel päeval ta avastab, et sellel rakendusel on kasutajasõbralik kasutajaliides. Tema arvates on oma sahvrisse hoidiste ja nende kohta info lisamine imelihtne. Teine stsenaarium kirjeldab Maimut, kes õpib kasutama nutitelefoniga ja talle tutvustatakse veebirakendust nimega Sahver. Lastelaste abiga oskab ta hoidiseid lisada ja neid sealt ära kustutada. Kolmandas stsenaariumis on juttu Martinist, kellele meeldivad väga oma vanaema tehtud toidud ja hoidised. Ta soovib, et saaks näha vanaema sahvri seisuga reaalselt ja uurib Internetist selliste võimalustega rakenduse kohta. Kasutajasõbraliku kasutajaliidese töttu õpib Martini vanaema lühikese ajaga selgeks kõik rakenduse funktsioonid (Mägi, 2016).

Kasutades loodud stsenaariumeid oli võimalik läbi viia disainisessioonid, kust sai olulist tagasisidet rakenduse disaini loomise ja vajalike funktsionaalsuste osas. Näiteks selgus, et hoidiseid tuleks sorteerida valmistamise kuupäeva, nime, koostise ja asukoha järgi ning eakamatele inimestele mõeldes peaks loodava prototüübi disain olema võimalikult minimalistlik. Saadud infot sisendiks kasutades loodi kasutajalood, mis on lihtsad, selged ja lühikesed kirjeldused kasutajale vajalikest funktsionaalsustest (Cohn, 2004). Käesoleva töö käigus arendatava prototüübi jaoks loodi 22 kasutajalugu (vt Tabel 2).

Tabel 2. Kasutajalood kasutajate rollide järgi (Mägi, 2016)

Roll	Kirjeldatud kasutajalood
Külaline	4
Kasutaja	16
Administraator	2
Kokku	22

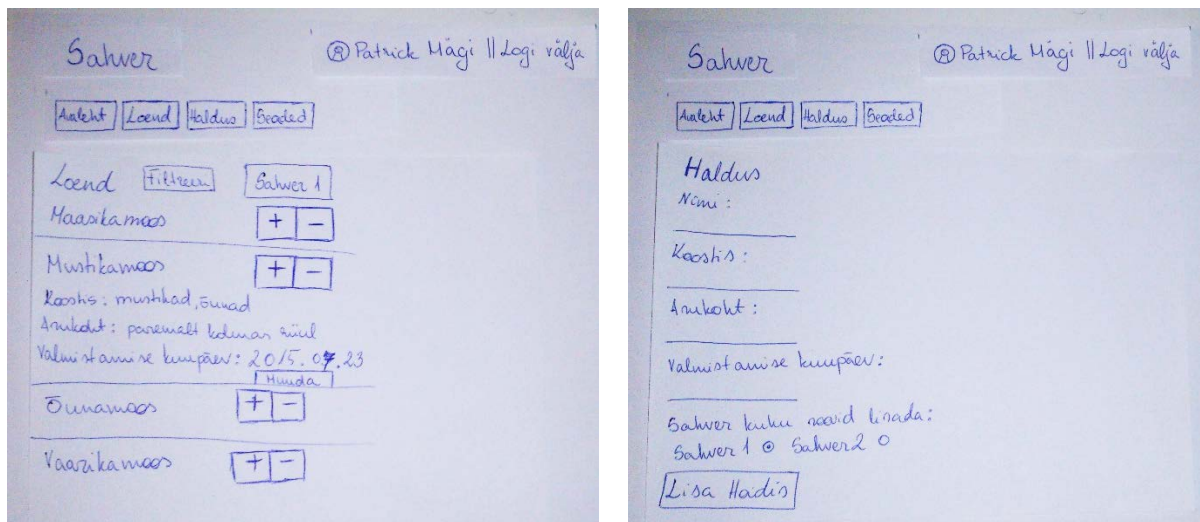
Kasutajalugusid loodi külalisele neli, registreeritud kasutajale 16 ja administraatorile kaks. Näiteks kirjeldati registreeritud kasutaja soovi sisse logida ja hoidiseid lisada. Loodud kasutajalugude abil oli autoril võimalik kirjeldada prototüübi esmased funktsionaalsed nõuded (Mägi, 2016).

3.2 Paberprototüübid

Kasutajalugusid tehes kirjeldas autor kavandatavale veebirakendusele esimesed vajalikud funktsionaalsused. Arvesse võttes ka teisi sarnaste funktsionaalsustega saadaval olevaid rakendusi tegi töö autor paberprototüübid ning nende tegemise tulemusena tekkis autoril ettekujutus, mitu erineva struktuuriga HTML keeles kirjutatud lehte on vaja ja mis seal lehtedel olema peaks. Samuti selgus, et rakendusele on vaja luua logo, et seda kasutada rakenduse päises.

Kokku loodi paberprototüübid üheksa lehe jaoks, mida saab kirjeldada järgmiselt (vt Joonis 1):

1. avaleht – külaline saab näha, mis veebirakendusega tegemist on, rakenduse kirjeldust, idee autorit, versiooni numbrit ja arendaja nime;
2. kasutaja avaleht – kasutaja on sisse loginud ja näeb lihtsat õpetust rakenduse kasutamisest, arendaja nime, versiooni numbrit ja infot rakenduse kohta;
3. loendi vaheleht – kasutaja saab filtreerida sahvri sisu ja hoidiseid lisada ning kustutada;
4. halduse vaheleht – kasutaja saab uusi hoidiseid juurde lisada märkides üles nende nime, koostise, asukoha ja valmistamise kuupäeva;
5. seadete vaheleht – kasutaja saab muuta parooli ja jagada oma loodud sahvrit teiste kasutajatega;
6. registreerimise leht – külalisel on võimalik luua endale kasutaja sisestades oma emaili, kasutajanime ja parooli;
7. avaleht – kasutaja saab sisse logida oma kasutajanime ja parooli sisestades;
8. parooli taastamise leht – kasutajal on võimalik parooli taastada, mille tegemise jaoks peab ta sisestama oma emaili;
9. administraatori leht – rakenduse arendajal on ülevaade loodud sahvrite ja kasutajate arvust.



Joonis 1. Paberprototüübid kasutaja loendi vahelehel ja kasutaja halduse vahelehel

Kõikide loodud paberprototüüpidega on võimalik tutvuda töö lisades (vt Lisa 1).

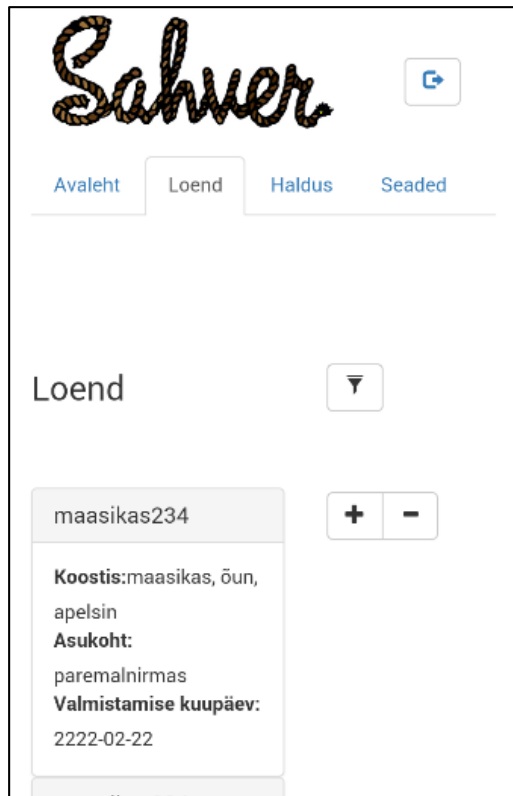
3.3 HTML prototüübid

Paberprototüüpide loomise tulemusel nägi autor kui palju on vaja luua erineva HTML struktuuriga prototüüpe. Paberprototüüpimise tulemusena tekkis üheksa lehte, aga Bootstrap'i abiga sai autor lihtsa vaevaga kasutaja sisselogimise lehe ühendada avalehega, kuhu tegi nupu, millest langeb alla paremale rippmenüü sisselogimise väljadega ning seetõttu valmis kaheksa veebilehe prototüüpi. Loodud lehti kasutades oli autoril CSS keele abiga võimalik määrata soovitud stiilireeglid. Kasutades Bootstrap'i raamistikku sai autor juurde lisada erinevaid elemente, mis teevad rakenduse kasutamise lihtsamaks, kasutajasõbralikumaks ja mobiilsete seadmetega ühilduvaks. JavaScripti keelt kasutades sai autor rakenduse andmeid vahetama Parse¹¹ veebipõhise serveriga. Kirjeldatud veebiprototüüpide loomisel tekkis esimene veebikuvand loodavast rakendusest.

Bootstrap¹² on tasuta raamistik, mille abil saab kiiremini ja kergemini luua kasutaja seadmele reageerivat kujundust. Arendaja saab valida HTML ja CSS abil valmis elemente nagu näiteks nupud, tabelid, vormid ja navigatsiooni ribad. Töö autor valis vajalikud elemendid prototüübile ja kasutas neid struktuuri ja disaini loomisel (vt Joonis 2).

¹¹ www.parse.com

¹² www.bootstrap.com



Joonis 2. Kasutaja halduse vahelehel

HTML prototüüp koos Bootstrap'i raamistikuga erines vähesel määral paberprototüüpidel kujutatust. Töö autor valis nuppude kujundamiseks Bootstrap'i kõige tavalisema stiilivaliku, mille nimi on *default*. Autor disainis rakendusele logo ja kasutas selle kujundamisel nööri imitatsiooni, mille abil on kirjutatud rakenduse nimi Sahver. Nööri imitatsiooni kasuks otsustas autor, kuna tema arvates tekitab see rakenduse logole maalähedase tunde. Kirjeldatud veebiprototüüpide loomisel tekkis esimene veebikuvand loodavast rakendusest, mida kasutades sai autor hakata prototüübile funktsionaalsuseid arendama. Nende arendamisest annab ülevaate järgmine peatükk.

4 Arendus

Toetudes seminaritöö käigus valmistatud nõuete analüüsile ja disaini loomise käigus tekkinud sisendile, arendas töö autor valmis prototüübi, mis võimaldab kasutajal hoidiste haldamiseks vajalikke funktsioone kasutada. Sahvri prototüübi arendamisel võttis autor esmaseks prioriteediks, et oleks olemas kasutajate loomine, hoidiste lisamine, kuvamine ja kasutaja seadete haldamine.

4.1 Platvormi valik

Sahvirakenduse prototüübi arendajaks on käesoleva töö autor, kes on arendamiseks vajalikud teadmised omandanud läbides TLÜ Informaatika digitaalse meedia suuna bakalaureuseõppe. Lähtudes kasutajalugudes välja tulnud funktsionaalsuste vajadustest ja toetudes varasemalt õpitule kasutab autor veebirakenduse arendamiseks HTML, CSS ja JavaScripti keeli.

Mobiilsetel seadmetel töötamise tagamiseks võttis autor kasutusele Bootstrapi raamistiku. CSS keele abil saab autor paika panna loodava veebirakenduse kujunduse. Disaini loomisel võttis autor arvesse, et loodava veebirakenduse kasutamine peab olema lihtne, mugav ja kasutajasõbralik. Vajalike funktsionaalsuste töötamiseks oli autoril vaja luua ühendus veebirakenduse ja andmebaasi vahel. Selleks valis autor Parse nimelise teenuse, mis hõlmab endas ka andmebaasi loomise võimalust. Parse on veebipõhine server, millega on võimalik veebirakenduse kasutajate info ja loodud sisu salvestada. Kirjeldatud serverirakenduse kasutamiseks andmebaasina on võimalik kasutada mitmeid programmeerimiskeeli. Käesoleva töö raames kasutas autor andmebaasi ja veebirakenduse vahelise suhtluse loomiseks JavaScripti keelt.

Failide jagamiseks ja versioonihalduse teostamiseks kasutatakse GitHubi meeskonna poolt loodud GitHub Desktop¹³ nimelist rakendust, mis võimaldab Git repositooriumi haldamisteenust kasutada mugava kasutajaliidesega.

Bootstrapi raamistikuga pidi autor esmatutvuse tegema, kuna sellega polnud tal kokkupuudet ülikooli ajal olnud. Raamistiku kasutama õppimine ei valmistanud mitte mingeid raskuseid, kuna selle kohta on palju abimaterjali, alustades õpetustest ja lõpetades täpse koodi

¹³desktop.github.com/

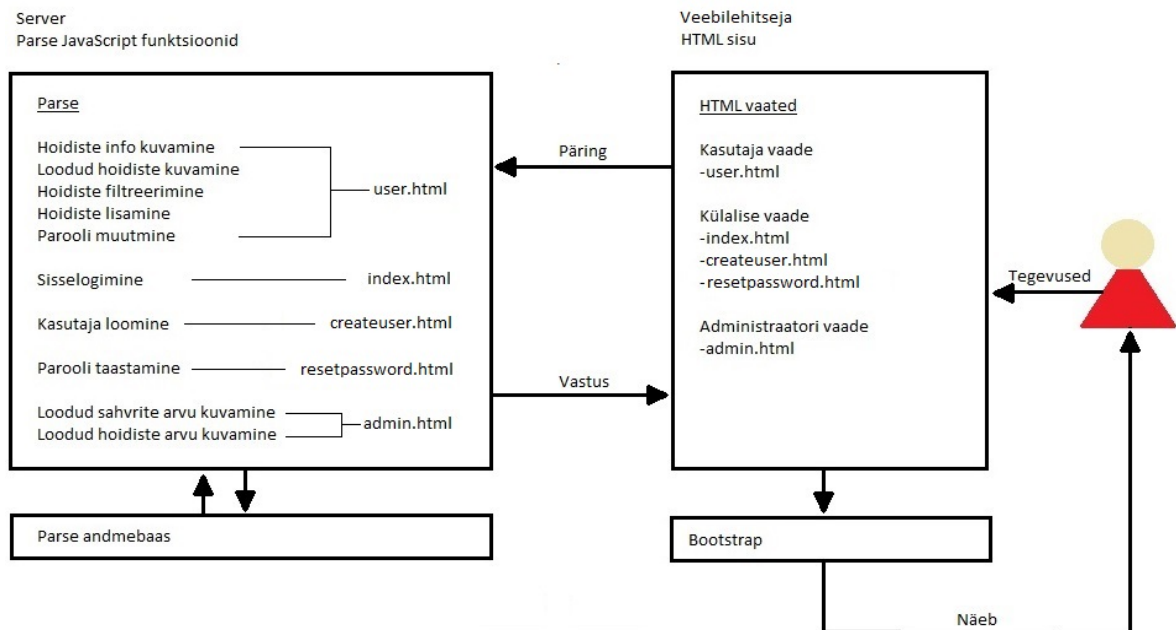
dokumentatsiooniga. Peamiselt kasutas töö autor raamistikuga tutvumiseks W3Schools¹⁴ nimelist veebilehte, kus oli välja toodud väga hea õpetus Bootstrap raamistikuga tutvumiseks.

Autor otsustas veebipõhise rakenduse kasuks, kuna sellega on võimalik kiiremini saada valmis rakenduse prototüüp, mis töötab erinevate operatsioonisüsteemidega. Valiku tegemise kasuks rääkis samuti asjaolu, et töö autorile on enamuse eelnevalt loetletud tööriistadest tuttavad.

4.2 Rakenduse ülesehitus

Sahvri veebirakenduse prototüüp¹⁵ arendati seminaritöö käigus valminud kasutajalugudest tekkinud funktsionaalsustest lähtuvalt ning toetudes koostatud paberprototüüpidele. Rakenduse kasutajaliides loodi kasutades Bootstrap raamistikku. Bootstrapi kasutamiseks on vaja HTML koodiga panna paika vajalikud elemendid, mille struktuuri määravad raamistiku CSS stiilireeglid ja nende edasiarendused. HTML keele abil sai autor määrata struktuuri osad, kasutades „div“ (ingl *division*) elemente. Kirjeldatud veebirakenduse arenduse käigus võeti arvesse eelkõige seda, et rakenduse disain käituks dünaamiliselt ja töötaks hästi mobiilseid seadmeid kasutades.

Sahvirakenduse ülesehituse võib jaotada kaheks: kliendi ja serveri pool (vt Joonis 3).



Joonis 3. Sahvirakenduse ülesehitus: serveri ja kliendi pool

¹⁴ www.w3schools.com/bootstrap

¹⁵ [www.github.com/PatrickMg/Sahver](https://github.com/PatrickMg/Sahver)

Kõikide serveripäringute tegemisel kuvatakse andmed sisselogitud kasutaja põhiselt. Serveri poolel käib suhtlus JavaScriptiga kirjutatud funktsioonide alusel. Kliendipoolse veebirakenduse koodi moodustavad HTML keelega loodud struktuur ja Bootstrapi abil loodud kasutajaliides.

Sahvri veebirakenduse prototüüp on kättesaadav GitHub keskkonnast avatud lähtekoodiga ning avaldatud MIT litsensiga¹⁶. MIT litsentsi valis töö autor, kuna see võimaldab ka teistel arendajatel panustada rakenduse loomisesse kuid kohustab säilitada viide koodi autoriõigusele ja sellega kaasas käiv litsents. Samuti on GitHubis näha veebirakenduse arendamise käik HTML prototüüpidest toimiva veebirakenduse prototüübini. Loodud veebirakenduse prototüüp¹⁷ on avalikult kättesaadav.

¹⁶ <https://opensource.org/licenses/MIT>

¹⁷ www.tlu.ee/~loginz/Sahver

5 Testimine

Rakenduse arendamise käigus viis töö autor läbi ka koheselt testimist ja niimodi selgusid disainivead, mille sai koheselt parandada. Näiteks tekkis autoril viga nupu paigutamisega Bootstrap raamistikus, kus lisamise ja kustutamise nupp läks kaheks erinevaks nupuks, mis asetsevad üksteise peal.

Rakenduse testimise viis töö autor läbi eeldatavatele kasutajatele võimalikult ligilähedaste inimestega. Arenduse viimase etapi käigus jälgis töö autor inimesi prototüüpi kasutamas ja märkis üles tekkinud küsimused, et planeerida veebirakenduse võimalikke edasiarendusi. Autor kaasas testimise protsessi kolm inimest, kes mängisid läbi kasutaja loomise, hoidiste lisamise, kuvamise ja kustutamise (vt Tabel 3). Inimeste kasutamine kasutajatena andis autorile hea ülevaate, milliseid omadusi hindavad rakenduse juures võimalikud rakenduse kasutajad. Autor tegi märkmeid järgides musta kasti testimise põhimõtteid, kus jälgis ainult loodud funktsionaalsuste sisend-väljund käitumist. Samuti andsid testimise protsessis osalenud kasutajad infot loodud rakenduse võimaluste ja puuduste kohta.

Tabel 3. Testimise ülevaade (testija, kestvus, märkused ja kommentaarid)

Nr	Testija	Testimise kestvus	Märkused ja kommentaarid
1	Koduperenaine	17 min	Sahvri sisu on võimalik hallata lihtsa vaevaga Peale hoidise info sisestamist võiks väljad tühjaks minna, et oleks mugavam uusi hoidiseid lisada
2	Pensionär	26 min	Sahvri haldamiseks vajalikud funktsionaalsused on olemas Kas tehtud muudatused õnnestusid või ebaõnnestusid?
3	Õpilane	10 min	Sahvri üle on võimalik arvet pidada loodud funktsionaalsuste abil

Käesoleva töö raames loodud veebirakenduse prototüübi testimise käigus selgusid puudused, mida oleks vaja edasiarendusega muuta või täiendada. Prototüübi testimise käigus sai selgeks, et loodud funktsionaalsused võimaldavad hallata sahvri sisu ja on kasutajate jaoks esmatähtsad. Kõik testimise protsessi kaasatud kasutajad tõid välja, et peale vormide täitmist võiks kirja pandud info peale sisestamist väljadelt ära kaduda, et oleks mugavam uusi hoidiseid lisada. Loodud prototüüp vajab autori arvates erinevaid juurdearendusi, millest kriitilisem on kasutajamugavuse parendamine, näiteks tuleks kasutajale anda informatsiooni tehtud muudatuste õnnestumise või ebaõnnestumise kohta. Üheks suureks edasiarenduseks oleks

kindlasti isikliku domeeni omamine, kuna see tundub rakenduse uutele kasutajatele usaldusväärsem ning aitab kaasa ka rakenduse reklaamimisele.

Võimalike edasiarenduste üles märkimiseks kasutas autor GitHubi Issues¹⁸ keskkonda, et loodud prototüüpi oleks võimalik autoril või teistel huvilistel edasi arendada. Samuti lisati sinna prototüübi arendamisel tekkinud puuduste loetelu, millele tuleks tähelepanu pöörata.

¹⁸ <https://github.com/PatrickMg/Sahver/issues>

Kokkuvõte

Käesoleva bakalaureusetöö lahendatavaks probleemiks oli lihtsa sahvri haldamist võimaldava rakenduse puudumine. Kirjeldatud probleemi lahendamiseks viis autor esmalt seminaritöö käigus läbi nõuete analüüsi etapi, kus selgus, et vajadus loodud rakenduse vastu on olemas. Põhiline on see, et nõuete analüüsis sai autor kirjeldada võimalikud kasutajad, nende eesmärgid ja kasutajalugude abil rakendusele vajalikud funktsionaalsused. Arendamise etapi moodustasid prototüübi disaini väljatöötamine ja selle testimine kasutajatega, mis tõi välja loodud prototüübi võimalused ja puudused.

Eesmärgi täitmiseks uuris autor esmajoonel sarnaste rakenduste puuduseid ja võimalusi, seadis paika loodava prototüübi disaininõuded ja kasutas tekkinud nõudeid sisendiks arendamisel. Sahvirakenduse disainiprotsessi jooksul kasutas autor mitmeid osalusdisaini meetodeid. Nimelt loodi seminaritöö läbiviidud nõuete analüüsi käigus prototüübile persoonad, kellele loodi tarkvara kasutusstsenariumid, mida testiti loodud persoonadele võimalikult ligilähedaste kasutajatega. Järgmise sammuna loodi kasutajalood, mis andsid sisendi funktsionaalsuste kirjeldamiseks. Peale kasutajalugude loomist koostas töö autor paber- ja HTML-prototüübid, disaini ning arendas rakenduse prototüübi. Funktsionaalsuste lisamisel võttis autor tekkinud kasutajalugudest saadud nõuded ja arendas veebirakenduse prototüübi esimese versiooni, millele saab tulevikus juurde lisada veel täiendavaid funktsionaalsuseid. Töö käigus testiti loodud rakenduse prototüüpi, et välja selgitada võimalikke võimalusi ja puudusi.

Lähtudes bakalaureusetöö käigus läbi viidud uurimisest ja loodud prototüübist, saab autor välja tuua disainitud ning arendatud hoidiste üle arvepidamise rakenduse Sahvri tähtsamad võimalused ja puudused:

- loodud prototüübi funktsionaalsused võimaldavad kasutajatel hallata enda sahvri sisu;
- suuremat rõhku tuleb pöörata kasutajamugavusele;
- hoidise salvestamisel võiks kirja pandud info väljadelt ära kaduda, et oleks lihtsam uusi hoidiseid lisada;
- vormide salvestamisel ei ole selgelt aru saada, kas muudatuste salvestamine õnnestus või mitte.

Edasise uurimissuunana näeb töö autor tekkinud informatsiooni põhjal kasutajamugavuse veelgi paremaks muutmist. Autor lisas edasiarenduse soovitusel rakenduse projekti GitHubi

Issues keskkonda, et hiljem oleks võimalik töö autoril või teistel huvilistel arendusprotsessi jätkata.

Kasutatud kirjandus

Board, I., S., T., Q., (2011). *Foundation Level Syllabus: Certified Tester*

Brown, D. M. (2010). *Communicating Design: Developing Web Site Documentation for Design and Planning*. Berkeley: New Riders.

Cohn, M. (2004). *User Stories Applied For Agile Software Development*. Boston, MA: Addison Wesley

Leinonen, T., Toikkanen, T., & Silfvast, K. (2008). *Software as Hypothesis: Research-Based Design Methodology*. Helsinki: Media Lab, University of Art and Design.

Liiva, S. (2015, 5. august). *Hoidiste tegemine on sel suvel erilise hoo sisse saanud* (artikkel).
Loetud aadressil: <http://tarbija24.postimees.ee/3284329/hoidiste-tegemine-on-sel-suvel-erilise-hoo-sisse-saanud>

Mägi, P. (2016). *Koduse sahvrihalduse rakenduse nõuete analüüs*. (seminaritöö). Loetud aadressil:

http://www.cs.tlu.ee/teemaderegister/get_file.php?id=432&name=Koduse%20sahvrihalduse%20rakenduse%20n%C3%B5uete%20anal%C3%BC%C3%BCs_Patrick%20M%C3%A4gi.pdf

Rosson, M. B., Carroll, J. M. (2002). *Scenario-Based Design*. Blacksburg: Virginia Tech.

Snyder, C. (2003). *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. San Fransisco: Morgan Kaufmann.

Summary

Creating the Prototype of Household Pantry Management Application

Bachelor Thesis

This bachelor thesis focuses on designing and developing a software prototype, that enables users to create and manage their pantry.

In order to reach the main goal, following research was made:

- literature analysis, where the author covers similar applications, development platform and mobile application testing;
- description of research methodology;
- creation of personas, scenarios, user stories, paperprototypes and HTML prototypes;
- prototype development;
- testing of the developed prototype.

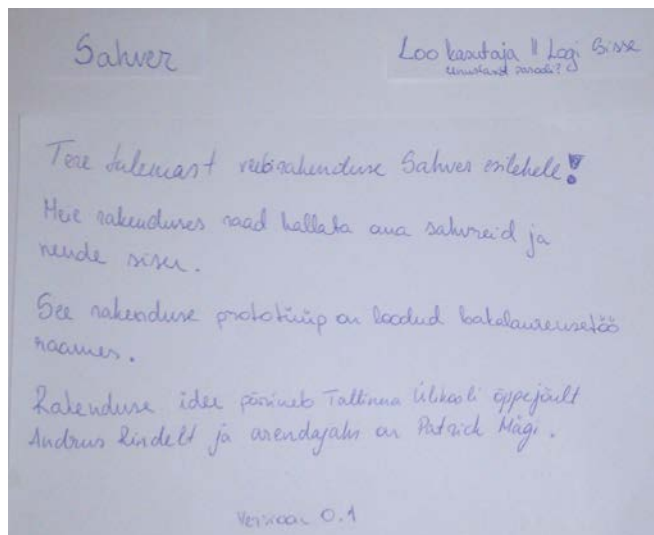
The software prototype was designed using multiple participatory design methods. Which included creating personas and scenarios, writing user stories, creating paper and HTML prototypes. During the design process the author also conducted design sessions with expected users. The author tested the prototype by himself during the development process and later with expected users. Testing the prototype with expected users arose future development issues, which are pointed out in GitHub Issues.

Future research should focus on adressing application usability and solving current issues.

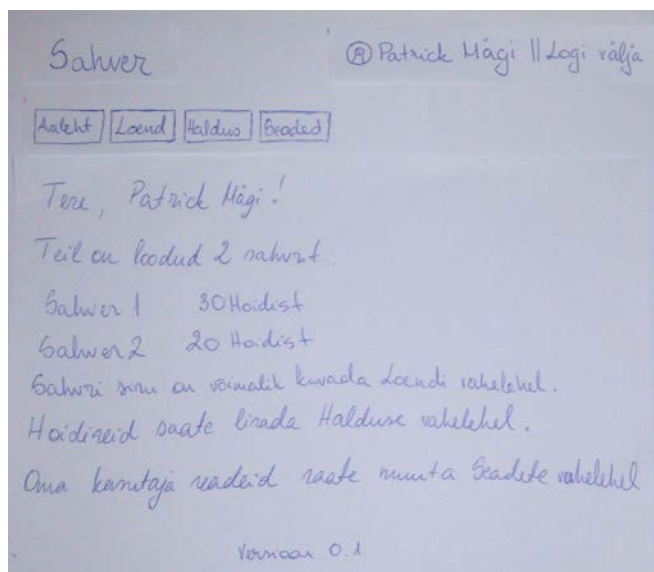
Lisad

Lisa 1 Loodud paberprototüübid

Külastaja avalehel



Kasutaja avalehel



Kasutaja loendi vahelehel

Sahver @ Patrick Hägi // Logi välja

Aoleht Loend Haldus Branded

Loend Filtreeri Sahver 1

Haarikammas + -

Mustikammas + -

Koostis: mustikad, õunad
Arukoht: põrselalt kolmas rüütl
Valmistamise kuupäev: 2015.07.23

Õunammas + -

Vaarikammas + -

Kasutaja halduse vahelehel

Sahver @ Patrick Hägi // Logi välja

Aoleht Loend Haldus Branded

Haldus

Nimi: _____

Koostis: _____

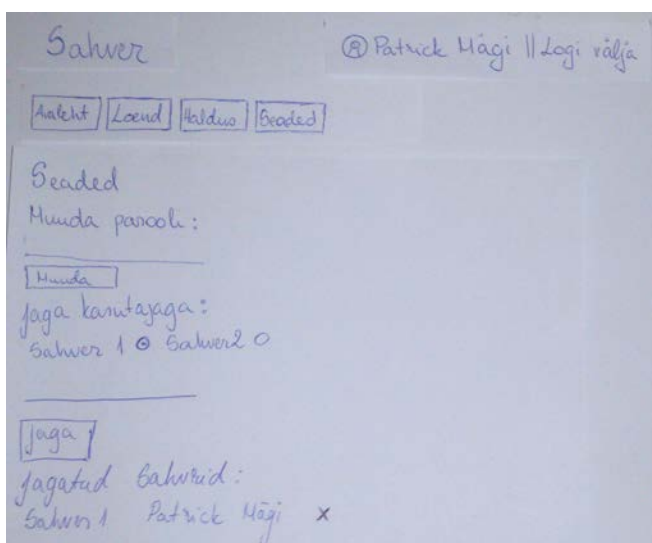
Arukoht: _____

Valmistamise kuupäev: _____

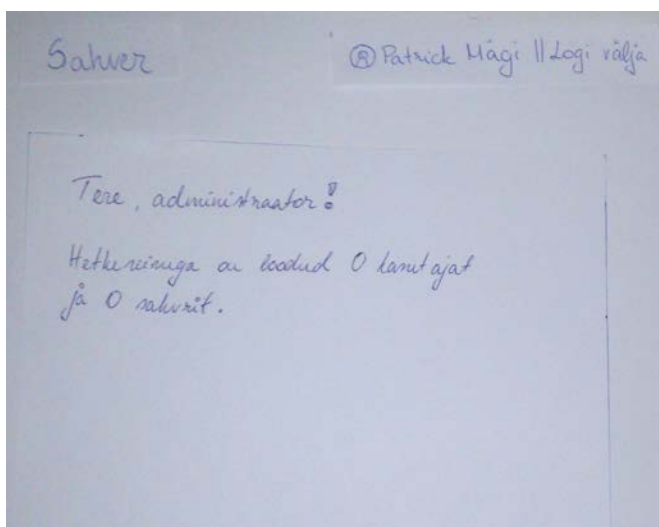
Sahver kuhu saavid lisada:
Sahver 1 @ Sahver 2 o

Lisa Haldis

Kasutaja seadete vahelehel



Administraator avalehel



Kasutaja loomine

Sahver

E-mail

Kasutajanimi

Parool

Parool uuesti

Loo kasutaja

Sisselogimine

Sahver

Kasutajanimi

Parool

Sisene

Parooli taastamine

