

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Villem Madisson 185765IADB

Kinnisvara haldussüsteemi mobiilirakenduse arendamine ettevõttele RunProperty

Bakalaureusetöö

Juhendaja: Kristiina Hakk
Doktor

Tallinn 2022

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: Villem Madisson

16.05.2022

Annotatsioon

Käesoleva bakalaureusetöö eesmärk on luua mobiilirakendus ettevõttele RunProperty. Rakendus on mõeldud kinnisvaramaakleritele, kes puutuvad igapäevaselt töös kokku kliendisuhete ja kinnisvaraobjektide haldusega. Loodav rakendus hakkab pakkuma lisandväärtust RunProperty poolt pakutavale tootele.

Töö käigus lahendatakse probleem, kus maakler saaks kontakte läbi mobiiltelefoni vaadata ja süsteemi lisada ning samuti näha detailseid andmeid kinnisvaraobjektide kohta. Lahendusena luuakse esialgne versioon RunProperty mobiilirakendusest, mida arendatakse edasi hiljem.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 6 peatükki, 13 joonist, 4 tabelit.

Abstract

The aim of this bachelor's thesis is to create a mobile application for company RunProperty. The application is for real estate agents who are in daily contact with customer relations and real estate objects management. The created application will add value to the product offered by RunProperty.

In the course of the work, a problem is solved where the broker can view the contacts via a mobile phone and add them to the system, as well as see detailed information about real estate objects. The solution is to create an initial version of the RunProperty mobile application which will be further developed later.

The thesis is in Estonian and contains text on 29 pages, 6 chapters, 13 figures, 4 tables.

Lühendite ja mõistete sõnastik

App Store	Veebipood IOS telefonile kus on võimalik alla laadida mobiilirakendusi
Figma	Disaini loomiseks mõeldud tarkvara
Google Play	Veebipood Android telefonil kus on võimalik alla laadida mobiilirakendusi
HTTP	<i>Hypertext Transfer Protocol</i> , protokoll teabe edastamiseks arvutivõrkudes
IntelliJ IDEA	Ettevõtte JetBrains'i poolt loodud arenduskeskkond tarkvara arendamiseks
IOS	Apple mobiiliseadmetele mõeldud operatsioonisüsteem
iPadOS	Apple poolt välja töötatud operatsioonisüsteem, mida kasutatakse iPad'idel
JavaScript	Programmeerimiskeel
JSON	<i>Javascript Object Notation</i> , andmevahetusvorming
JWT	<i>JSON Web Token</i> , avatud standard, mida kasutatakse turbetaabe jagamiseks kliendi ja serveri vahel
Kanban tahvel	Vahend töövoos visualiseerimiseks
Kotlin	Avatud lähtekoodiga staatiliselt tüübitud programmeerimiskeel, mis kompileerib Java baitkoodiks ja kasutab Java virtuaalmasinat
MacOS	Apple poolt loodud operatsioonisüsteem, mis on mõeldud Apple poolt loodud arvutitele
SDK	<i>Software development kit</i> , vahendikomplekt rakendusprogrammide kirjutamiseks
Service worker	JavaScript fail, mille tööks on peatada võrgupäringuid, salvestada andmeid vahemällu, tuua andmeid vahemälust ning edastada hüpiksõnumeid
Vanilla JavaScript	Tavaline JavaScript ilma täiendavate teekide või raamistiketa

Sisukord

1 Sissejuhatus	10
2 Taust ja probleem	11
2.1 Ettevõtte RunProperty	11
2.2 Probleemi püstitus	11
2.3 Eesmärk ja metoodika	12
3 Analüüs.....	14
3.1 Nõuded rakendusele	14
3.1.1 Kasutajalood	15
3.1.2 Vastuvõtukriteeriumid	15
3.2 Sarnased rakendused.....	16
3.2.1 Zoho CRM.....	16
3.2.2 Pocket by Rex.....	18
3.2.3 Resolve CRM	19
3.3 Tehnoloogiad	20
3.3.1 Platvormipõhised rakendused.....	20
3.3.2 Hübriidrakendused	21
3.3.2.1 React Native	21
3.3.2.2 Flutter.....	21
3.3.2.3 Xamarin	22
3.3.2.4 Ionic ja Apache Cordova	22
3.3.3 Progressiivsed veebirakendused.....	23
3.3.4 Tehnoloogiate kokkuvõte	24
4 Teostus.....	26
4.1 Arenduse töövahendid	26
4.2 Projekti ülesehitus.....	26
4.3 Rakenduse loomine.....	28
5 Tulemus	30
5.1 Valminud rakendus.....	30
5.2 Vaatlejate tagasiside	36

5.3 Rakenduse tulevik.....	37
6. Kokkuvõte	38
Kasutatud kirjandus	39
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	42
Lisa 2 – Kasutajalood	43
Lisa 3 – Vastuvõtukriteeriumid	45
Lisa 4 – Mobiilitehnoloogiate võrdlus.....	48

Jooniste loetelu

Joonis 1. Zoho CRM koduvaade.	17
Joonis 2. Pocket by Rex koduvaade.	18
Joonis 3. Kausta struktuur.	27
Joonis 4. Projekti vaate kood.....	29
Joonis 5. Sisselogimise vaade.....	31
Joonis 6. Ülevaate vaade	31
Joonis 7. Kontaktide vaade.....	32
Joonis 8. Kontakti loomise vaade.....	32
Joonis 9. Kontakti detail vaade.....	33
Joonis 10. Kuulutuste vaade	33
Joonis 11. Kuulutuse detailvaade	34
Joonis 12. Projektide vaade	34
Joonis 13. Projekti detailvaade	35

Tabelite loetelu

Tabel 1. Funktsionaalsed nõuded.	14
Tabel 2. Kasutajalood	43
Tabel 3. Vastuvõtukriteeriumid	45
Tabel 4. Mobiilitehnoloogiate võrdlus	48

1 Sissejuhatus

Lõputöö eesmärk on luua mobiilirakendus ettevõttele RunProperty OÜ. Loodav rakendus saab olema mõeldud kinnisvaramaakleritele, kes puutuvad oma töös kokku kliendisuhete ja kinnisvaraobjektide haldusega. RunProperty loodud esialgne rakendus on eeskätt mõeldud arvutis kasutamiseks. Mobiilirakendus hakkab pakkuma esialgsele veebirakendusele lisandväärtust, mis parandab ettevõtte konkurentsivõimet.

Mobiilirakendus aitaks koos veebirakendusega paremini tootega turule siseneda. Vajadusest mobiilirakenduse järele on andunud märku RunProperty potentsiaalsed kliendid, kes leiavad, et veebirakendusest jääb üksi väheks. Mobiilirakendus ei hakka täitma kõiki funktsioone mida veebirakendus, küll aga osasid neist. Lisaks on soov juurde lisada mõned funktsioonid mida veebirakenduses pole. Töö tulemuseks saab olema esialgne versioon mobiilirakendusest, mida on plaanis arendada edasi tulevikus.

Lõputöö koosneb neljast sisupeatükist. Teises peatükis antakse ülevaade töö taustast ja probleemist. Kolmandas peatükis räägitakse analüüsist, kus alguses kogutakse kokku töö nõuded, millest kirjutatakse edasi kasutajalood. Veendumaks töö nõuete täituvust koostatakse igale kasutajaloole vastuvõtukriteeriumid. Sellele järgneb sarnaste rakenduste võrdlus. Viimaks tehakse peatükis ülevaade mobiilitehnoloogiast ning valitakse välja sobiv tehnoloogia rakenduse loomiseks. Neljandas peatükis räägib autor töö tulemusest, rääkides täpsemalt arenduse tööriistadest, projekti ülesehitusest ja rakenduse loomisest. Viiendas peatükis hinnatakse saadud tulemust, tutvustatakse täpsemalt valminud rakendust, antakse ülevaade vaatlejate tagasisidest ja pannakse paika arendusprotsessi järgmised eesmärgid.

2 Taust ja probleem

Käesolevas peatükis antakse ülevaade ettevõttest Runproperty, sõnastatakse probleemi püstitus, pannase kirja lõputöö eesmärgid ja valitakse välja sobivad meetodikad töö teostamiseks.

2.1 Ettevõtte RunProperty

Ettevõtte RunProperty on idufirma, mis on tegutsenud alla ühe aasta. Ettevõttes töötab 11 töötajat, kes kõik panustavad igapäevaselt ettevõtte toimimisse. Peamine toode mida luuakse on koostööplatvorm kinnisvaraspetsialistidele, mille eesmärk on kasutajatele pakkuda kliendisuhete haldust ja kinnisvaraobjektide haldust. Alustuseks on ettevõttel soov siseneda Eesti turule, aga ambitsioon on tulevikus hõivata ka osa Euroopa turust. Peamine kliendigrupp on kinnisvarabürood, kus olevad maaklerid vajavad oma tööks abistavat tarkvara, mis kokkuvõttes muudaks nende töö efektiivsemaks. Ettevõtte on väga agiilse lähenemisega, mis väljendub pidevas suhtluses kliendiga, igapäevastes püstijalakoosolekutes ja lühikestes arendustsüklites.

Tootetiim tegeleb igapäevaselt veebirakenduse (edaspidi pearakendus) arendamisega. Eesrakendus on loodud JavaScript raamistikus vue.js, tagarakendus node.js ja andmebaasiks kasutatakse MongoDB'd. Pearakendus on mõeldud kasutamiseks peamiselt veebis ning see otsus tuli ettevõtte poolt toote loomise algfaasis. Rakenduses leidub küll üksikuid vaateid mis on vaadatavad ka mobiilis, kuid peamise töövoogu läbimine on siiski telefonis raskendatud.

2.2 Probleemi püstitus

Lahendatava probleemi saab jaotada kaheks. Esimene osa kujutab endas kontaktide haldust, kus maakler soovib hallata oma klientide kontakte. Teine osa on spetsiifiliselt seotud kinnisvara haldusega. Maakleril on soov näha süsteemi sisestatud kinnisvara objekte ning samuti on soov teha kinnisvara objektidega vajalikke toiminguid. Kõik see peab olema tehtud kasutajale võimalikult mugavaks, kus kõiki toiminguid saab teha kõigest paari nupuvajutusega.

Peamine vajadus just mobiilirakenduse järgi tekib asjaolust, et klient soovib hallata oma kontakte. Tihtipeale saab kinnisvaramaakler kliendi kontaktid arvutist eemal olles, mis ei anna võimlust maakleril andmeid läbi RunProperty veebirakenduse sisestada. Siinkohal tuleb appi mobiiltelefon, kust kaudu saaks maakler kiirelt mobiilirakendust kasutades sisestada peamised kliendi andmed RunProperty süsteemi. Kui kliendi andmed on läbi mobiilirakenduse süsteemi kantud, siis on võimalik kasutajal neid näha ka RunProperty veebirakenduses, kus leidub kontakti haldamiseks juba rohkem funktsioone. Näiteks saab kasutaja veebirakenduses kontakte liigutada pakkumiste *kanban* tahvlil, mis annab hea pildi kliendi staatusest ja huvidest. Sama funktsioon on soov hiljem lisada ka mobiilirakendusse. Lisaks on soov hiljem lisada mobiilirakendusse veelgi funktsionaalsust, mida veebirakenduses pole olnud mõttekas arendada. Näiteks kuvada kasutajale tema kõneajalugu, mis pakuks kasutajale võimalust kontakte mugavamalt sisestada, ilma et ta peaks rakenduse ja kõneajaloo vaadete vahel liikuma. Kogu tulemus teeb kontaktide haldamise lihtsamaks.

Teiseks soovib kasutaja näha erinevaid vaateid, mis on seotud kinnisvara haldusega. Peamine vaade mida rakenduse kasutaja soovib näha on kuulutuse vaade, kus ta saab ülevaate kinnisvaraobjektist. Vaate olemasolu tuleks maaklerile kasuks eriti vahetult enne klindiga kokkusaamist, kui tal tuleb meelde tuletada objekti informatsioon. Lisaks kuulutustele tahab kasutaja näha projekte, selleks et saada ülevaadet süsteemi sisestatud projektidest. Vaadete täpne kirjeldus on kirja pandud kasutajalugude tabelisse (lisa 2, tabel 2).

2.3 Eesmärk ja metoodika

Lõputöö lõppeesmärk on luua esialgne mobiilirakendus, mis läheks hiljem ettevõtte RunProperty klientidele kasutusse. Selleks, et jõuda lõputöö käigus soovitud eesmärkideni valis autor välja sobivad agiilseid metoodikad. Valik langes agiilsete metoodikate kasuks, kuna ka ettevõtte RunProperty kasutab töös agiilseid meetodeid, mis teeb ka lõputöö tegemise lihtsamaks.

Töö algab funktsionaalsete nõuete kogumisest, mis pannakse kirja koos tootetiimiga. Kui töö nõuded on teada, siis pannakse paika kasutajalood, mille kaudu hakatakse rakendust looma. Veendumaks, et nõuded saavad täidetud koostatakse kasutajalugudele vastuvõtukriteeriumid. Järgneb sarnaste rakenduste võrdlus, mille käigus uuritakse rakendusi, mis võiksid sarnaneda loodava rakendusega. Autor võtab uuringu käigus kaasa teadmised, mis lihtsustavad rakenduse

arendamist. Enne veel kui alustatakse rakenduse loomisega, tehakse analüüs mobiilitehnoloogiatest. Analüüsi käigus selgub, milline tehnoloogia rakenduse ehitamiseks kasutusse võetakse. Peale analüüsi lõppu, algab rakenduse arendamine.

Lõputöö käigus luuakse ainult eesrakendus. Tagarakenduse loomise vajadus puudub, kuna see on valminud juba pearakenduse arenduse käigus.

Esialgse rakenduse kasutajaliides valmib autori poolt tema äranägemise järgi. Lõputöö keskel, kui RunProperty disainerid on valmis saanud rakenduse disaini Figma's, ehitab autor kasutajaliidese ümber.

Rakendus peab olema kokkuvõttes kättesaadav nii IOS kui ka Android operatsioonisüsteemidele, kuid lõputöö raames rakendust ei juurutata, kuna rakendus ei valmi lõputöö raames lõplikult. Operatsioonisüsteemide toe vajadust võetakse arvesse tehnoloogiate analüüsis (peatükk 3.3).

3 Analüüs

Järgnevas peatükis loetletakse nõuded, mis peaksid olema täidetud rakenduse minimaalseks toimimiseks, hinnatakse sarnaseid rakendusi ja valitakse välja sobiv tehnoloogia rakenduse loomiseks.

3.1 Nõuded rakendusele

Nõuded annavad ettekujutuse, mida kasutaja rakenduses teha tahab. Nõuded jagunevad funktsionaalseteks ja mittefunktsionaalseteks nõueteks. Funktsionaalsed nõuded kirjeldavad tegevusi, mida süsteem peab olema võimeline tegema. Mittefunktsionaalsed nõuded keskenduvad süsteemi käitumise täpsustamisele [1]. Töö käigus koguti kokku kõik vajaminevad funktsionaalsed nõuded, mis saadi tootetiimiga läbirääkimistel (tabel 1). Nõuetest vormistatakse edasi kasutajalood (peatükk 3.1.1)

Tabel 1. Funktsionaalsed nõuded.

Jrk	Nõue
1	Rakenduses peab olema võimalik näha ühe kontakti vaadet ja vaadet kus on näha mitmeid kontakte.
2	Rakenduses peab olema võimalik lisada kontakti.
3	Rakenduses peab olema võimalik näha ühe kuulutuse vaadet ja vaadet kus on näha mitmeid kuulutusi.
4	Rakenduses peab olema võimalik näha ühe projekti vaadet ja vaadet kus on näha mitmeid projekte.

3.1.1 Kasutajalood

Kasutajalugu on kirjeldus kasutaja vajadusest, mis on kirjutatud kindlas formaadis. Esiteks on määratud kes on kasutaja. See võib tarkvaraarendusel määrata olulist rolli, kuna mõningatel juhtudel saab üks kasutaja teha tegevusi rakenduses ühtemoodi, mõni teine kasutaja aga teistmoodi. Järgmiseks on kirjeldatud soov ehk tegevus, millest järeldub, mida kasutaja teha tahab ning see on ka kasutajaloo kõige olulisem osa. Viimaseks on kirjeldatud tegevuse põhjus, mille teadmine võib toote loomisel kasuks tulla [2].

Kasutajalood sõnastati kogutud funktsionaalsete nõuete põhjal ning kinnitati RunProperty tootetiimi poolt, kuhu kuulub ka lõputöö autor (lisa 2, tabel 2). Kasutajalood on järjestatud vastavalt prioriteedile, mis paneb paika ka töö teostamise järjekorra.

Autor lisab iga kasutajaloo lõppu hinde, mis määratakse planeerimispokkerit mängides. Planeerimispokker on hindamissüsteem mille kaudu hinnatakse tööd, kus peamiseks hindamisfaktoriks on kasutajaloo teostamise keerukus [3]. Ajakulu hindamisel arvesse ei võeta. Hindajateks on üldjuhul terve tootetiim, aga lõputöö raames on ainsaks hindajaks lõputöö autor. Hindamise käigus teeb autor omale selgeks esialgse töökäigu.

3.1.2 Vastuvõtukriteeriumid

Vastuvõtukriteeriumid luuakse stsenaariumite kaudu, kus igale kasutajaloole vastab üks või mitu stsenaariumit. Stsenaarium on üksikasjalik kirjeldus tegevusest, mis saab juhtuda kui kasutaja proovib juhendada kasutajaloost. Vastuvõtukriteeriumid aitavad kasutajalugudest paremini aru saada [4] (lisa 3, tabel 3). Tabelisse on lisaks kirja pandud lihtsustatud kasutajalood, kus on kirjas kasutajaloos olev roll ja kasutaja soov.

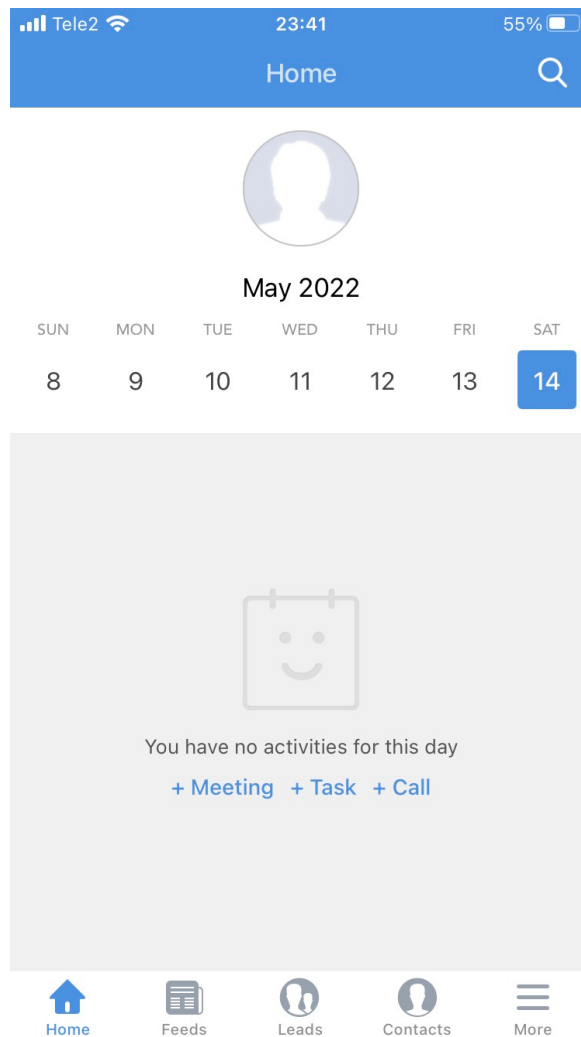
3.2 Sarnased rakendused

Järgnevalt analüüsitakse kolme rakendust, milles leidub sarnaseid jooni loodava rakendusega. Nendeks on: Zoho CRM – Sales & Marketing , Pocket by Rex ja Resolve CRM. Rakendused valiti välja esmase ülevaatus käigus, kust otsiti internetist kliendisuhete juhtimise ja kinnisvara haldamise rakendusi. Analüüsi eesmärk on leida rakenduste head küljed, mida saaks rakendada ka loodavas rakenduses. Vaatluse käigus otsitakse rakendustest peamiselt ainult neid funktsioone, mida soovitakse ka RunProperty mobiilirakendusse luua.

3.2.1 Zoho CRM

Ettevõtte Zoho poolt on loodud rakendus nimega Zoho CRM, mis on mõeldud kasutajatele kes soovivad hallata kontakte [5]. Rakendusel on nii veebi- kui ka mobiiliversioon. Veebiversioon on tasuline ja mobiiliversioon on tasuta [6]. Rakenduse mobiiliversioon on kättesaadav nii Google Play's kui ka App Store's. Autor testib rakendust IOS süsteemiga telefonil.

Esmakordsel avamisel soovib rakendus kasutaja sisselogimist. Selle puudumisel saab kasutajatunnuse luua. Kui kasutaja loodud ja rakendusse sisse logitud, avaneb rakenduse koduvaade (joonis 1), kus edasi saab luua üritust, tegevust või kõne. Navigeerimisriba kasutades saab minna uudisvoo, seotud isikute ja kontaktide vaadetesse. “more” nupu vajutamisel avaneb “modules” vaade, kust asetseb uus navigeerimisriba, sisaldades veelgi lehe vaateid. Rakenduses on veel seadete leht kus leidub mitmeid võimalusi rakenduse konfigureerimiseks.



Joonis 1. Zoho CRM koduvaade.

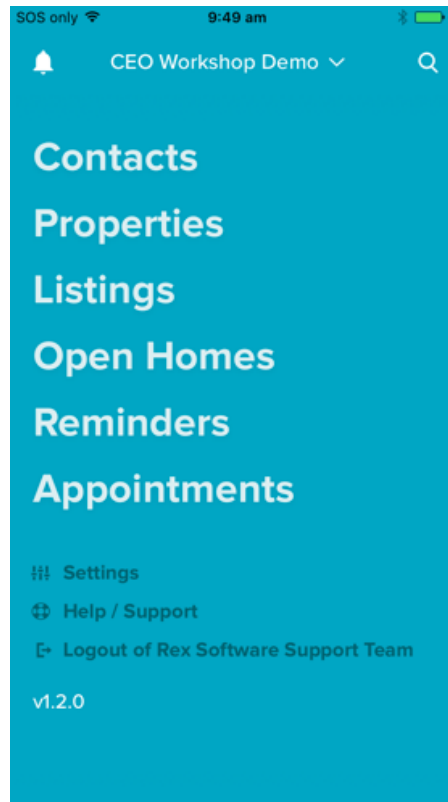
Rakenduse põhjalikumal uurimisel jõudis autor järelduseni, et selle kasutamine on tehtud kasutajale üpris mugavaks. Kõik funktsioonid on kiiresti leitavad, mis on mobiilirakenduste puhul väga oluline. Rakenduses leidub kalendri funktsioon, mis on tulevikus plaanis lisada ka RunProperty mobiilirakendusse. Veel jäävad autorile rakendust testides silma vaadete vahetusega kaasnevad animatsioonid, mis autori arvates on hästi lahendatud. Rakendusest on aga puudu kinnisvara haldamise osa, mis saab olema aga suur osa loodavas rakenduses. Autor leiab, et kui lisada rakendusele juurde kinnisvaraobjektidega seotud funktsioonid ja kaotada ära mõned rakenduses olevad olemasolevad funktsioonid, siis saaks rakenduse, mis sarnaneks loodava rakendusega suhteliselt palju.

3.2.2 Pocket by Rex

Pocket by Rex on ettevõtte Rex Software poolt loodud mobiilirakendus [7]. Ettevõtte tegutseb Austraalias ja Suurbritannias, kus asuvad ka nende peamised kliendid, kelleks on enamjaolt kinnisvara agentuurid. Nende kodulehe põhjal on neil kliente kokku üle 1800. Erinevalt Zoho CRM rakendusest leidub sellel rakendusel kinnisvara spetsiifilisi funktsionaalsusi. Sarnaselt RunProperty ettevõttega on neil olemas ka veebirakendus, kuid autor vaatleb analüüsi käigus ainult nende mobiilirakendust.

Kuna tegemist on tasulise rakendusega, siis rakenduse ülevaatus viiakse läbi selle informatsiooni põhjal, mis on internetist autorile kättesaadav. Peamiseks infoallikaks on Rex'i poolt loodud kasutajatoe leht [8], kust saab rakendusest küllaltki detailse ülevaate.

Rakenduse koduvaade kuvab kasutajale navigeerimisvõimalusi (joonis 2), kus on võimalik liikuda teistesse vaadetes. Teistes vaadetes olles saab navigeerimisvaate avada kas ülevalt vasakult nurgast või libistades sõrme ekraani vasakult servast paremale. Vaadete vahel klikkides on näha, et paljudel vaadetel leidub võimalus kasutada otsingut või filtrit, otsimaks või filtreerimaks lehel kuvatuid objekte.



Joonis 2. Pocket by Rex koduvaade. [30]

Uurimisel selgus, et rakendus sarnaneb oma funktsioonide poolest loodava rakendusega üpris palju. Mõned klassid on teise nimega ja osad objekti väljad on juures ning osad puudu võrreldes RunProperty tootega. Rex'i kasutajaliides on üsna erinev loodava rakenduse disainist, väljendudes näiteks värvides, navigeerimisvaates, erinevate elementide suurustes. Veel märkas autor rakenduse uurimisel ühte suurt puudust. Nimelt on rakendus ühekeelne, mis ka põhjendab asukoha valikuid kuhu ettevõtte oma tootega on sisenenud.

RunProperty veebirakenduse keele valikus on nii inglise kui ka eesti keel ning tulevikus on soov rakendusele keeli veelgi juurde lisada. Täpselt sama teed proovib RunProperty liikuda ka mobiilirakendusega.

Lõppkokkuvõttes leiab autor, et toode tundub igati kinnisvaramaaklerit rahuldav, aga see pole kõige parem valik maaklerile, kes ei pruugi inglise keelt osata.

3.2.3 Resolve CRM

Resolve CRM on Eesti kinnisvara müügitarkvara [9]. Rakendus sarnaneb oma funktsionaalsuste poolest RunProperty pearakendusega. Resolve peamine puudus on vähene võimekus. Andmeid saab sisestada ja kuvada, aga vähe on vaateid kus andmed annaksid kasutajale põhjaliku ülevaate. Paljud meie potentsiaalsed kliendid kasutavad hetkel Resolve toodet. Oleme saanud neilt infot, et Resolve arendustempo on aeglane. Kui on soov tahta mõnda uut funktsiooni või tahta mõnda muudatust rakenduses, siis selle väljaarendamine võtab palju aega. Selle tingib paljuski asjaolu, et rakendust arendab ainult üks inimene ja temagi tegeleb sellega oma põhitöö kõrvalt.

RunProperty pearakendus on võrreldes Resolv'i tootega dünaamilisem. Andmed on omavahel rohkem seotud, mis annab kasutajale parema ülevaate teabest. Näiteks on RunProperty pearakenduses olemas *kanban tahvel*, mis on Resolve rakenduses puudu. Kinnisvarabürood on andnud tagasisidet, et hindavad *kanban tahvli* olemasolu kõrgelt. Samuti puudub Resolve'l mobiilirakendus. RunProperty mobiilirakenduse olemasolul eristuksid ettevõtete vahelised tooted selgemalt ning RunProperty suudaks klientidele tunduvalt suuremat väärtust pakkuda kui Resolve.

3.3 Tehnoloogiad

Järgnevalt uurib autor erinevaid tehnoloogiaid, mille abil saab luua mobiilirakendusi. Tuginedes populaarsusele valis autor analüüsiks kuus tehnoloogiat [10]. Analüüsi lõpus tehakse otsus millise tehnoloogia peale hakatakse rakendust looma.

3.3.1 Platvormipõhised rakendused

Mobiilirakendusi on peamiselt kahte sorti, platvormipõhised ja hübriid tehnoloogiatel põhinevad. Platvormipõhised rakendused on tehtud eraldi kindlatele operatsioonisüsteemidele. Hübriidrakendused on aga seevastu mõeldud töötama mitmel erineval operatsioonisüsteemil. Maailmas on kaks peamist mobiili seadmetele mõeldud operatsioonisüsteemi, IOS ja Android [11] ning mõlemas operatsioonisüsteemis peab ka tulevikus valminud rakendus töötama.

IOS rakendused kirjutatakse Objective –C või Swift'i ning Android rakendused Java või Kotlini programmeerimiskeeltes [12].

Platvormipõhiste rakenduste eelised hübriidrakenduste ees [13]:

- parem jõudlus;
- turvalisus;
- parem kasutajakogemus;
- parem ligipääs mobiilipõhistele funktsionaalsustele;
- arenduse käigus tekib üldjuhul vähem vigu.

Suurimateks nõrkusteks peetakse platvormirakenduste puhul arenduse ajakulu ja rakenduse valmides sellele kuluvat hooldusaega.

Lõplik RunProperty mobiilirakendus peab töötama nii IOS kui ka Android operatsioonisüsteemi peal ehk kirjeldatava tehnoloogia valimine viiks selleni, et rakendusi tuleks luua kaks, mis üldjuhul tähendab omakorda kahekordset arenduse ajakulu võrreldes mõne hübriidtehnoloogiaga.

3.3.2 Hübriidrakendused

Järgnevalt analüüsitakse hübriidtehnoloogiatel põhinevaid lahendusi, milleks on React Native, Flutter, Xamarin ja Ionic koos Apache Cordovaga.

3.3.2.1 React Native

2011 aastal arendas Facebook välja JavaScript raamistiku React, mis on läinud kasutusse maailmas väga laialdaselt [14]. 2015. aastal arendati React'ist välja mobiilirakenduste ehitamiseks mõeldud tehnoloogia React Native, mis on samuti laialt levinud. Sarnaselt teistele vaadeldavatele tehnoloogiatele on tarkvara avatud lähtekoodiga. Avatud lähtekoodiga tehnoloogiad on üldjuhul paremini läbi testitud, mis teeb need töökindlamaks. Internetist leiab palju informatsiooni React Native kohta, mis garanteerib arendustegevuse jätkusuutlikkuse. Suurimaks eeliseks peab autor tehnoloogia juures programmeerimiskeelt, milleks on JavaScript, millega on hästi tuttavad kõik RunProperty arendajad.

3.3.2.2 Flutter

Flutter on ettevõtte Google poolt loodud tarkvaraarenduskomplekt, mis on avatud lähtekoodiga [15]. Flutter'it kirjutatakse Dart programmeerimiskeeles, mis on erinevalt JavaScript'ist tüübitud, mida peetakse ka tänapäeva programmeerimiskeelte puhul oluliseks. Tarkvarakomplekti kaudu saab luua töölaua-, veebi- ja mobiilirakendusi. Kõige levinumalt kasutatakse tehnoloogiat mobiilirakenduste loomiseks.

Flutter on paljude näitajate poolest sarnane React Native tehnoloogiale. Ometi on Flutter'i kasutatavus kasvavas trendis, aga React Native kasutatavus langevas trendis [10]. Põhjus peitub suuresti selles, et Flutter'it peetakse töökindlamaks [16]. Samuti on Flutter'is töökeskkonna ülesseadistamine React Nativiga võrreldes lihtsam. Veel peetakse Flutter'i puhul oluliseks selle kiirust, mis aitab kaasa just arendamise ajal, kui koodi kompileeritakse.

3.3.2.3 Xamarin

Ettevõtte Microsoft tütarettevõtte poolt loodi 2011. aastal mobiilirakenduse platvorm nimega Xamarin. Xamarin on samuti avatud lähtekoodiga. Selle abil on võimalik arendada nii platvormipõhiseid (Xamarin Native) kui ka hübriidipõhiseid (Xamarin Forms) mobiilirakendusi [17].

Xamarin'i rakendusi luuakse C# programmeerimis keeles, mis on samuti Microsofti poolt loodud [18]. C# on objektorienteeritud ja tüübitud keel. Siiski on tehnoloogia kasutatavus olnud viimaste aastatega langevas trendis [10]. Nõrkuseks peetakse kasutajaliidese loomise ajakulu [19]. Tehnoloogia sobib eriti hästi just siis, kui osa koodibaasist on juba C#'is kirjutatud, mida saaks mobiilirakenduse arendamisel ära kasutada. Kui tehnoloogia peaks osutama valituks, leiab autor, et arendus saab toimuma Xamarin Forms'is, kuna tegemist on hübriidipõhise lahendusega, mis võimaldab jõuda kiiremini valmis rakenduseni.

3.3.2.4 Ionic ja Apache Cordova

Ionic on avatud lähtekoodiga tööriistakomplekt, mille kaudu saab luua hübriid-mobiilirakendusi [20]. Saamaks rakenduse loomisel ligipääsu mobiilipõhiste funktsioonidele, on loodud ka raamistik Apache Cordova. Ionic'ut saab kirjutada Vanilla JavaScript'is, aga levinum viis on kasutada seda koos mõne JavaScript'i raamistikuga. Kõige rohkem kasutust leidnud raamistik on Angular.js [21].

Ionic'u puhul on suureks tugevuseks vähene ajamaht tarkvaraarendusele [22]. Suuresti selle tingivad valmis tehtud kasutajaliideste komponendid, mida saab arendamisel lihtsasti kasutada. Veel on Ionic'ule tehtud palju pistikmooduleid, mis samuti saavad arendamisel kasuks tulla.

Autor leiab, et kui väljavalitud tehnoloogiaks peaks osutama Ionic, siis seda kindlasti koos Apache Cordova'ga ja vue.js JavaScript'i raamistikuga. Apache Cordova kaudu saaks ligipääsu mobiilipõhiste funktsioonidele, mida läheb vaja ka loodavas rakenduses. RunProperty pearakendusest saaks osaliselt üle tuua vue.js'is kirjutatud koodi, mis oleks peamine argument vue.js'i kasuks.

3.3.3 Progressiivsed veebirakendused

Lisaks platvormi- ja hübriidipõhiste mobiilirakenduste on leidumas ka kolmaski tehnoloogia, progressiivsed veebirakendused (*progressive web app*, edaspidi PWA) [26]. PWA on veebirakendus, millele on lisatud juurde mõned mobiilirakendusele omased võimekused [23]. Näiteks saab PWA'd telefoni alla laadida mobiilirakendusena. Erinevalt teistest mobiilirakendustest, kus rakendused laetakse alla rakenduste poest, laetakse PWA alla veebirakenduse lehelt. PWA juurde on võimalik lisada *service worker*, mida kasutades saab rakenduse panna tööle ka võrguühenduseta režiimis.

PWA loomiseks tuleb kasutada JavaScript'i ning sarnaselt Ionic'uga saab sealjuures kasutada ka JavaScript'i raamistikku. Suurimaks eeliseks peab autor PWA juures võimalust kasutada osaliselt pearakenduse koodibaasi, mis vähendaks teostatava töö hulka. RunProperty veebirakendus kasutab vue.js 2 raamistiku, mille peale saab PWA funktsionaalsust lisada suurema vaevata. Tootetiimiga arutleti eelnevalt võimalust muuta pearakendus PWA'ks, kuid jõuti kiirelt järelduseni, et see pole mõttekas, kuna loodav mobiilirakendus tuleb välimuselt suuresti pearakendusest erinev.

Suurimaks nõrkuseks PWA puhul on kehv ligipääs platvormipõhiste funktsionaalsustele. Veel on probleemiks asjaolu, et Apple ei soosi PWA kasutamist, piirates kasutamist erinevate meetmetega. Näiteks IOS ja iPadOS seadmetel saab PWA'd alla laadida ainult Safari veebibrauseris [24]. Kolmanda osapoolte veebibrauserites PWA installerimise võimalus puudub. Kasutades aga MacOS, siis Safari veebibrauser ei paku üldse meetodeid PWA installeerimiseks. Samuti puudub mõnedel IOS versioonidel võimalus kuvada rakenduse teatist.

3.3.4 Tehnoloogiate kokkuvõte

Parema ülevaate saamiseks koostati analüüsitud tehnoloogiatest võrdlustabel (lisa 4, tabel 4), kus on näha iga tehnoloogia tugevused ja nõrkused. Tabel on koostatud tuginedes allikatele [25], [26], [27] ning tabelis leiduvad kriteeriumid on välja valitud olulisuse põhjal.

Autor annab järgnevalt ülevaate mobiilitehnoloogiate võrdlustabelist, kirjeldades igat kriteeriumit eraldi.

- Vajab eraldi arendust IOS ja androidi seadmetele – eraldi arendust vajavad platvormipõhised rakendused ja samuti vajab eraldi arendust Xamarin kui kasutada Xamarin Native. Ülejäänud vaadeldud tehnoloogiad ei vaja mitme rakenduse arendamist.
- Juurdepääs platvormipõhiste funktsionaalsustele – kõikidel lõputöös vaadeldud mobiilitehnoloogiatel on hea ligipääs mobiilipõhiste funktsioonidele. Erandiks on PWA, mis on tingitud sellest, et tehnoloogia on tegelikult struktuurilt veebirakendus.
- Jõudlus – kõige parema jõudluse saab kätte platvormipõhistes mobiilirakendustes. Kõige nõrgem on jõudlus PWA puhul. Nende kahe tehnoloogia vahele jäävad hübriidtehnoloogiad.
- Turvalisus – igal tehnoloogial oma turvanõrkused, kuid kõige vähem leidub neid üldjuhul platvormipõhistes rakendustes.
- Arenduse ajakulu – platvormipõhiste rakenduste puhul on arvestatud sellega, et arendada tuleb kokku mitut rakendust, mis tingib kõrge arenduse ajakulu. Vähe võtab aega Ionic’u ja PWA põhiste mobiilirakenduste ehitus.
- Hoolduskulud – iga rakendus vajab valmimise järgselt hooldust ning selle kulude suurus sõltub suuresti rakenduse mahust ehk mida rohkem on hallatavaid rakendusi seda suuremad on ka hoolduskulud [28].

- Kogemus – varasemalt on lõputöö autor kõigi vaadeldavate tehnoloogiatega vähesel määral kokku puutunud, välja arvatud platvormipõhiste rakendustega.

Sobiva tehnoloogia valimisel võeti arvesse eelnenud analüüsi tulemusi. Valitud tehnoloogiani jõuti välistamise meetodil. Esimesena jäeti valikust välja platvormipõhised lahendused, kuna lahendus nõuab kahe rakenduse väljatöötamist, mis võtab palju aega. Järgmiseks jäi valikust välja PWA, kuna tehnoloogia ei võimalda piisavat head juurdepääsu platvormipõhiste funktsioonidele, mis osutub aga RunProperty mobiilirakenduses oluliseks vajaduseks tulevikus. Töö autori vähese kogemuse tõttu välistati ka Xamarin. Alles oli jäänud kolm tehnoloogiat: Flutter, React Native ja Ionic koos Apache Cordova'ga. Kuna Flutter ja React Native on omavahel üsnagi sarnased, siis otsustas autor kõigepealt välja selgitada nende vahelise soosiku. Autor otsustas valiku lõpuks Flutter'i kasuks, kuigi vaadeldud kriteeriumite poolest olid mõlemad samal tasemel, siis Flutter'it peetakse töökindlamaks [16]. Viimaseks jäi otsustada kas teha mobiilirakendus Flutter'i või Ionic'u peal. Flutter'i kasuks oli tehnoloogia kiirus ja kasvav kasutajaskond [10]. Ionic'u kasuks rääkis asjaolu, et see võimaldaks ära kasutada pearakenduse koodibaasi. Autor hindas, et olemasoleva koodibaasi kasutamine ei annaks suurt eelist, kuna valdav osa rakendusest tuleks ikkagi uuesti kirjutada. Samuti on Flutter'i suure ja kasvava kasutajaskonna aspekt üsnagi veenvaks argumendiks valimaks arenduseks just selle tehnoloogia. Edasi toimus arutelu tootetiimiga, kus jõuti üksmeelselt otsuseni, et mobiilirakendust hakatakse looma Flutter'is.

4 Teostus

Käesolevas peatükis antakse ülevaade arenduse töövahenditest, räägitakse projekti ülesehitusest, kirjeldades projekti kaustsüsteemi struktuuri ning viimaks räägib autor rakenduse loomisest.

4.1 Arenduse töövahendid

Autor kasutab Flutter'i rakenduse loomiseks Flutter SDK'd. Rakendust hakatakse looma JetBrains'i poolt loodud IntelliJ arenduskeskkonnas, mis osutus valituks oma töökindluse poolest. Samuti on autoril keskkonnaga eelnev kogemus, mis kiirendab tööprotsessi. IntelliJ arenduskeskkonda installeeritakse veel Flutter ja Dart pistikprogrammid, mis võimaldavad luua Flutter'i projekti. Rakenduste kuvamiseks laetakse alguses alla Android Studio arenduskeskkond, mis on üles ehitatud IntelliJ IDEA tarkvarale. Android Studio võimaldab kasutada Android simulaatorit, mida läheb tarvis rakenduse arendamisel. Apple poolt loodud telefonide testimiseks installeerib autor Xcode, mis on arenduskeskkond, mis võimaldab rakendusi testida IOS operatsioonisüsteemidega telefonidel. Autor otsustab alguses kasutada arendusel Android Studio't, kuna esimesena on eesmärk rakendus välja anda Android telefonidele. Rakenduse arendamise ajal käib autor vahepeal rakendust testimas ka Xcode's, veendumaks et rakendus on mõlemal operatsioonisüsteemil sarnane. Projekti hakatakse kirjutama inglise keeles, mis on tarkvaraprojektide puhul hea tava.

4.2 Projekti ülesehitus

Rakenduse kaustsüsteemi loomine on oluline osa rakenduse arendamisel, mis mõjutab hiljem hooldatavust, skaleeritavust ja kasutajamugavust [29]. Hiljem võib kaustastruktuuri muutmine olla raskendatud, mistõttu on oluline tegeleda sellega arendamise algaasis.

Enamik kirjutatavast koodist koondub lib kausta (joonis 3). Sellest kaustast väljapool asub ainult assets kaust, kus hoitakse rakenduse kirjastiile ja pilte.



Joonis 3. Kausta struktuur.

Lib kausta struktuur on järgnev:

- `business_logic` – Mõeldud hoidma äriloogika koodi. Eesmärk on eraldada disaini ja arvutusi vajavad koodid. See hoiab koodi oluliselt strukturesemalt, mis annab võimaluse hiljem ka rakendust paremini testida.
- `domain` – Hoitakse objektide loomiseks mõeldud klasse. Dart on tüübitud keel ja tüübitud keelte puhul on hea hoida objektide loomiseks mõeldud klasse eraldi kaustas.

- `helpers` – Mõeldud abimeetodite hoidmiseks. Näiteks kuulub kausta sõnumite näitamise kood.
- `providers` – Loodud andmete tarnimiseks. Iga objekti kohta tehakse kausta uus fail, mille eesmärk on hallata selle objekti sisu.
- `screens` – Hoitakse rakenduse kõiki mobiilivaateid. Vaated on grupeeritud omakorda kaustadesse. Näiteks hoitakse `contact` kaustas kontaktide vaadet, kontaktide detailvaadet ja kontakti loomise vaadet.
- `store` – Kaustas hoitakse meetodeid mille ülesanneteks on andmete salvestamine mobiiltelefoni vahemällu ja vahemälust andmete küsimine.
- `styles` – Sisaldab endas rakenduse stiili komponente. Peamiselt on kaustas failid, mida kasutatakse mitmel pool rakenduses.
- `utilities` – Kaustas hoitakse konstantseid muutujaid (värvid, välised võrguaadressid, rakenduse navigeerimise aadressid).
- `widgets` – Kaust sisaldab komponente, mis võivad grupeeruda veel omakorda kaustade vahel.

4.3 Rakenduse loomine

Arendamine algas üldise struktuuri loomisega. Liidestati kirjatüübid, värvid, pildid, navigeerimise aadressid ja loodi salvestamise hoidlad. Järgmiseks hakati täitma kasutajalugusid, mis täideti prioritseerimise järjekorras. Tegevuste käigus tuli alla laadida mitmeid Flutter'i pistikmooduleid:

- `provider` – Mõeldud tarnimaks andmeid erinevate konteinerite vahel.
- `http` – Võimaldab teha HTTP päringuid.
- `flutter_secure_storage` – Turvaline andme hoidla.
- `shared_preferences` – Andmete hoidla, mitte püsivate andmete hoidmiseks.
- `flutter_svg` – Võimaldab rakenduses kuvada `svg` failitüübiga pilte.

Töö jätkus vaadete loomisega, kus kohe alguses hajutati kirjutatud kood võimalikult laiali, selleks et edasine koodibaasi haldamine oleks lihtsam (joonis 4). Arenduse jooksul valmis kokku 49 komponenti.

```
return Scaffold(  
  appBar: const ProjectAppBar(),  
  body: SingleChildScrollView(  
    child: Container(  
      height: size.height,  
      alignment: Alignment.center,  
      child: const Padding(  
        padding: EdgeInsets.symmetric(horizontal: 16),  
        child: ProjectContent(),  
      ) // Padding  
    ), // Container  
  ), // SingleChildScrollView  
  bottomNavigationBar: BottomNavBar(currentIndex: 3),  
); // Scaffold
```

Joonis 4. Projekti vaate kood.

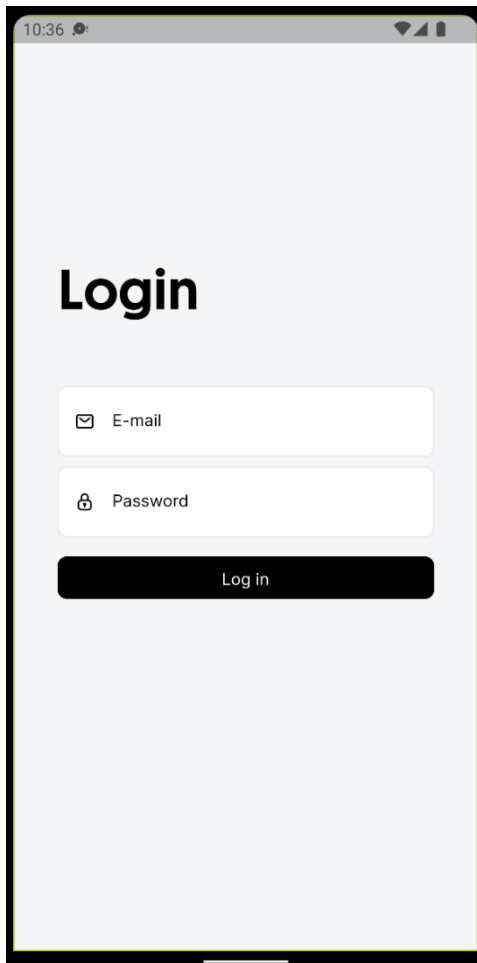
Rakenduse algne kasutajaliides loodi autori enda poolt. Lõputöö keskel said RunProperty disainerid valmis rakenduse disaini, peale mida uuendati rakenduse kasutajaliidest. Arenduse käigus polnud vaja suuri muudatusi teha, kuna autor oli arvestanud, et disain muutub, jättes sellega paljude komponentide väljaarendamise poolikuks.

5 Tulemus

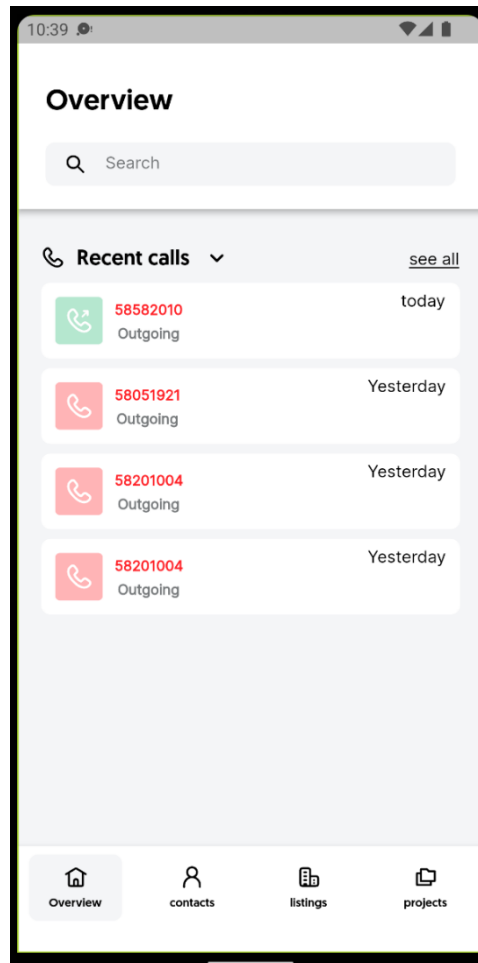
Käesolevas peatükis kirjeldatakse lõputöö käigus valminud lahendust, antakse ülevaade laekunud tagasisidest ja räägitakse järgmise arendusetapi eesmärkidest.

5.1 Valminud rakendus

Rakenduse esmakordsel avamisel kuvatakse kasutajale logimise vaade (joonis 5). Kui kasutaja sisselogimine ebaõnnestub kuvatakse kasutajale veateadet. Peale sisselogimist või juhul kui kasutaja on juba ühe korra rakendusse sisse loginud, kuvatakse kasutajale ülevaate vaadet (joonis 6). Sisse logides salvestatakse mälli serverist saadud kasutaja JWT (JSON Web Token) ning juhul kui JWT on rakenduse taas avades aegunud, siis kuvatakse kasutajale sisselogimise vaadet. Ülevaate lehelt on võimalik näha ühe sektsioonina telefoni kõneajalugu, kuid loodud lahendus töötab hetkel ainult Android telefonide peal. Kõneajaloo näitamist lõputöö raames eesmärgiks ei võetud ning see tehti vaid rakenduse edasise arenduse eesmärgil. Ülevaate lehele on tulemas tulevikus veelgi sektsioone, mida kasutaja soovib rakenduse avades eeskätt näha. Näiteks on plaan lehele tuua veel ülesannete ja viimati lisatud kuulutuste sektsioonid.

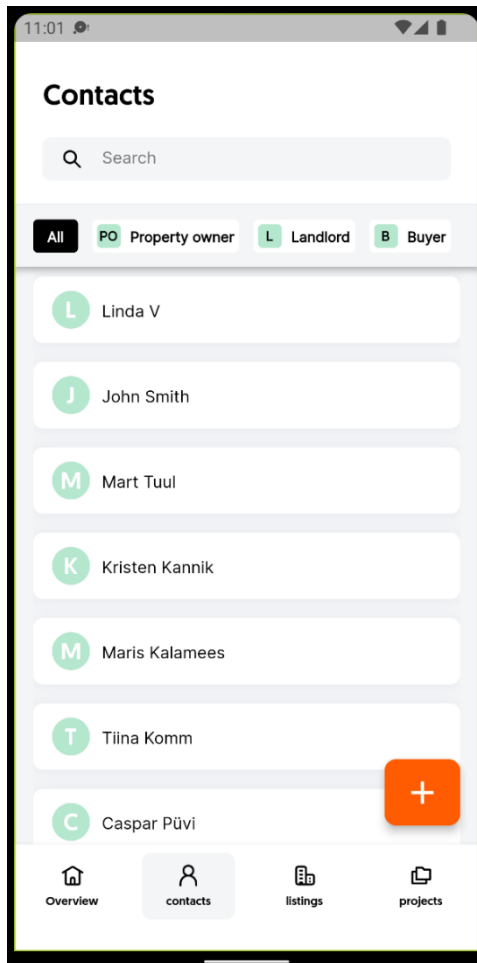


Joonis 5. Sisselogimise vaade.

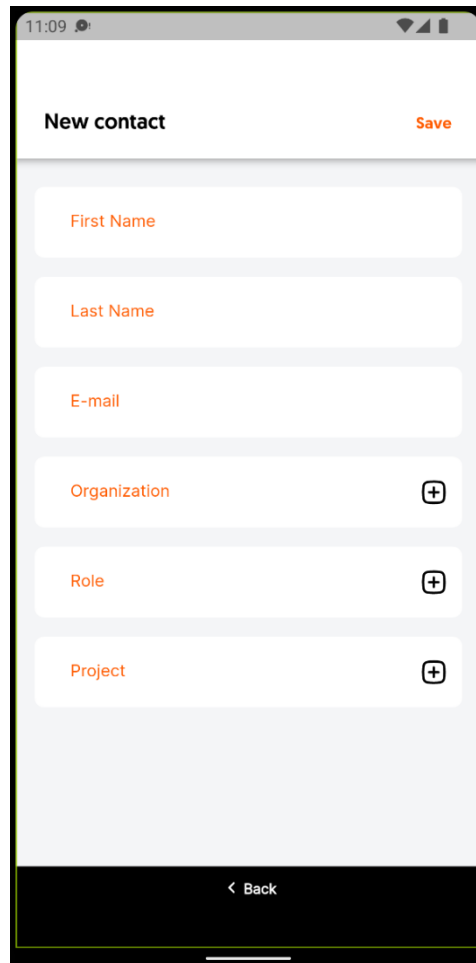


Joonis 6. Ülevaate vaade.

Mobiilirakenduse allosas paikneb navigeerimisriba, mille kaudu on võimalik liikuda rakenduse teistesse vaadetesse. Kontaktide vaates olles on näha all paremal nurgas pluss nappu, mille kaudu viib rakendus kasutaja kontakti loomise vaatesse (joonis 7, joonis 8). Kontakti luues valideeritakse kasutaja sisestatud eesnimi, mille käigus kontrollitakse, et sisestatud tulemus ei oleks täitmata. Eesnime täitmata jätmisel kuvatakse kasutajale veateadet.

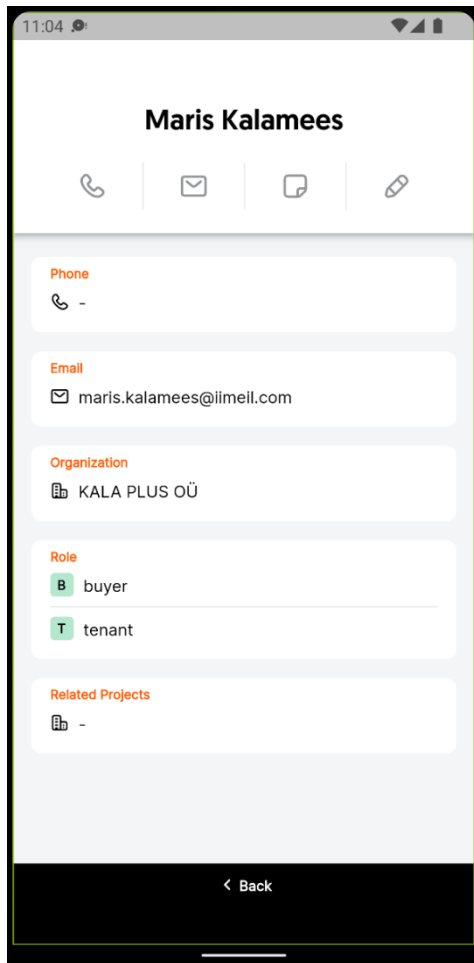


Joonis 7. Kontaktide vaade.

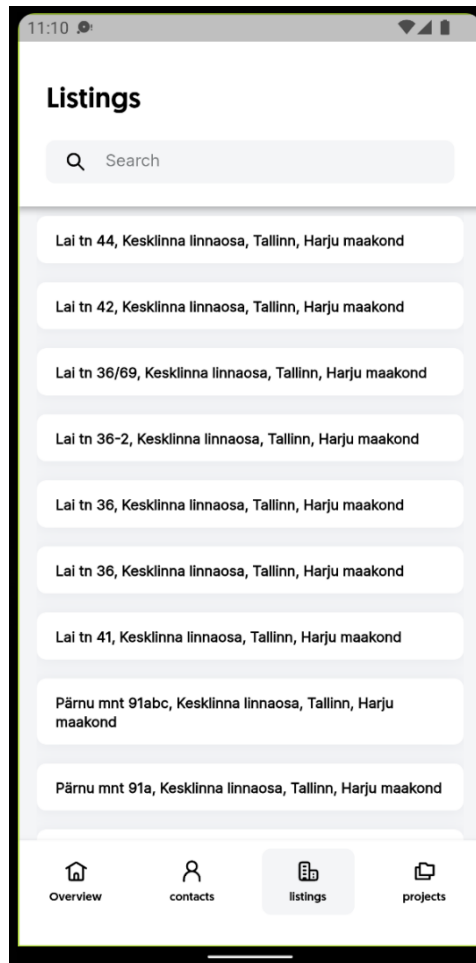


Joonis 8. Kontakti loomise vaade.

Korrektse kontakti vormi täitmise korral suunatakse kasutaja peale kontakti salvestamist kontakti detailvaate lehele, kus kasutaja näeb kontakti ülevaatlike andmeid (joonis 9). Vaate ülaosas on veel näha tegevusriba, mille funktsionaalsus on plaanis tööle saada tulevikus. Navigeerimisriba kolmas valik viib kasutaja on kuulutuste vaatesse, kus on näha kasutaja kõiki sisestatud kuulutusi aadressi järgi (joonis 10).

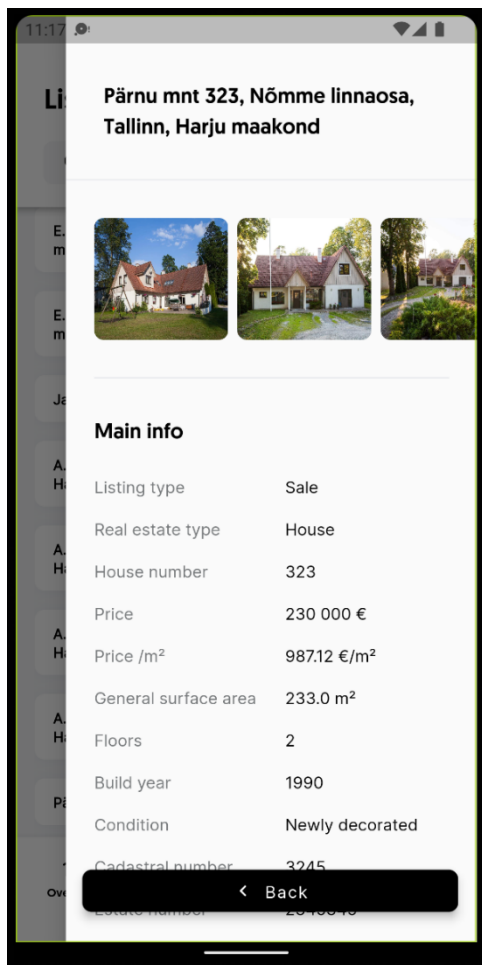


Joonis 9. Kontakti detail vaade.

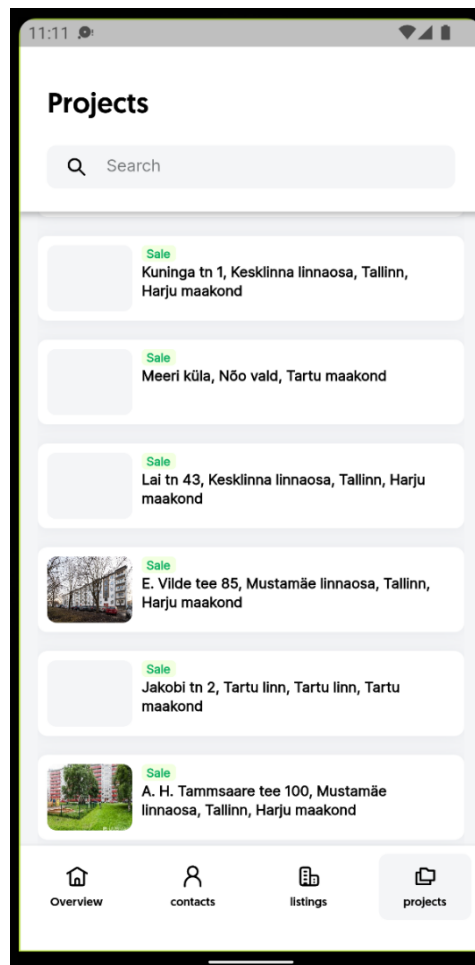


Joonis 10. Kuulutuste vaade.

Olles kuulutuste lehel ja klikkides kuulutuse peale avaneb küljepaneel kust on näha kuulutuse täpsemaid detailandmeid (joonis 11). Selleks et kasutaja saaks kiirema ülevaate kinnisvaraobjektist, siis kuvatakse kasutajale ainult neid väljasid, mis on eelnevalt täidetud. Paneeli saab panna kinni libistades paneeli sõrmega vasakult paremale või vajutades paneeli all osas paiknevale tagasi nupule. Projektide vaatel olles kuvatakse kasutajale kõiki sisestatud projekte, väljaarvatud need, millel on juures projekti lõpetatud märgis (joonis 12).

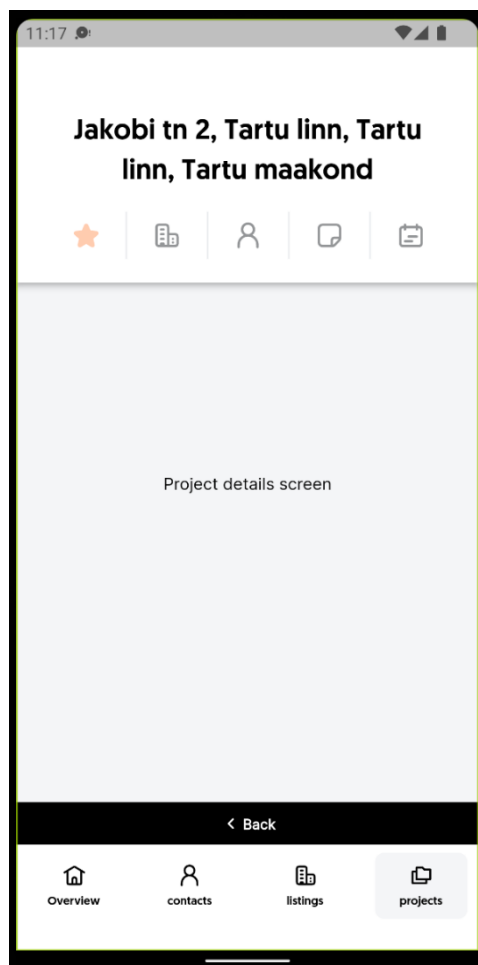


Joonis 11. Kuulutuse detailvaade.



Joonis 12. Projektide vaade.

Projekti peale vajutades kuvatakse projekti detailvaadet, kus hetkel näidatakse kasutajale projekti nime ja tegevusriba (joonis 13). Vaate sisuosa täiendatakse tulevikus. Kõik lehed on tehtud keritavaks, ehk kui lehe sisu ei peaks ära mahtuma, siis saab sõrmega vaadet alla kerida, peale mida on näha lehe ülejäänud sisu. Samuti on ekraanil laadimisrõngas igakord kui lehe andmeid kuvamiseks serverist alla laetakse.



Joonis 13. Projekti detailvaade.

5.2 Vaatlejate tagasiside

Valminud rakenduse hindamiseks lasi autor rakendust proovida kahel inimesel (edaspidi vaatlejad), kes polnud RunProperty toote arendamisega seotud või sellega tuttavad. Vaatlejalatel puudusid teadmised tarkavara arendusest. Vaatlus viidi läbi lõputöö autori arvutis oleva Android simulaatori kaudu.

Vaatlejatele valmistas kohe alguses muljet rakenduse disain, mis nende arvates nägi välja väga hea. Mainiti et vaated näevad välja väga lihtsad, millest tulenevalt saab aru tervest rakendusest väga kiirelt ning kasutajatele polegi vaja instruktaaži või juhendit teha. Tulevikus küll rakendus suureneb, mistõttu võib muutuda ka rakenduse kasutamine kasutajatele keerukamaks, kuid seda proovitakse kindlasti vältida.

Vaatlejaid proovisid testimisel vajutada tegevusnuppe, mis asetsevad kontakti ja projekti detailvaadetes (joonis 7 ja joonis 12), kuid said õige pea aru, et need ei tee veel midagi. Tegevusnuppude funktsionaalsus on plaanis lisada arenduse järgmises etapis, kuid lõputöö käigus valmis juba tegevusnuppude disain.

Järgmiseks panid vaatlejad tähele, et lehed laevad aeglaselt. Laadimine võtab aega keskmiselt 1.5 sekundit esmakordsel vaatele navigeerimisel. Aeglane laadimine on tingitud tagarakenduse ülesehitusest. Nimelt saadab tagarakendus mobiilirakendusele päringute tulemusi aeglaselt. Kuigi kiiruse tõstmiseks võiks näiteks tagarakendust optimeerida, on ettevõttel plaanis mobiilirakendusele luua hoopis uus tagarakendus. Lehe laadimisest annab kasutajale märku laadimISRõngas. Siinkohal leiab lõputöö autor, et laadimISRõnga lisamine rakendusele oli hea otsus, selle puudumisel võib kasutajatele jääda mulje, et leht on hangunud.

Peale rakenduse testimist, leidsid vaatlejad, et rakendus on heas algstaadiumis ning nad usuvad, et rakendusele töötab tulla helge tulevik. Vaatlejad andsid autorile teada, et on valmis rakendust testima ka tulevikus kui toodet on edasi arendatud.

5.3 Rakenduse tulevik

Lõputöö käigus valmis üks osa planeeritud tootest. Enne rakenduse kliendile näitamist tuleb rakendust täiendada. Selleks pani autor kirja järgmise arendusetapi eesmärgid:

- Rakendusele lisada juurde keele vahetamise võimalus.
- Rakendusele lisada juurde kalendrivaade.
- Rakendusele rakendada tegevusnuppude funktsionaalsus.
- Rakendusele lisada juurdepääs telefoni kõneajaloole nii Android kui ka IOS telefonil ja kuvada rakenduses kasutajale tema kõneajalugu.
- Rakenduse lisamine Google Play poodi.

Lõputöö algfaasis oli ettevõttel suur soov alustada mobiilirakenduse arendamisega, kuid õige pea tulid esiplaanile uued eesmärgid, mis jätsid mobiilirakenduse arendamise tahaplaanile. Projekt on plaanis taas ette võtta mai teises pooles, kui ettevõtte esiplaanil olevad eesmärgid on täidetud. Projektiga jätkates on plaanis sellesse kaasata ka rohkem arendusressurssi. Lõputöös valminud rakenduse koodi osa valmis ainult autori jõul, kasutades selleks töövälisest aega.

6. Kokkuvõte

Diplomitöö eesmärk oli arendada välja ettevõttele RunProperty kinnisvara haldussüsteemi mobiilirakendus. Ettevõtte pakub klientidele kinnisvara müügi- ja haldusplatvormi ning peamiselt on toode suunatud kinnisvaramaakleritele. Eelnevalt oli ettevõttel olemas ainult veebirakendus, kuid soov oli teha ka turule sisenemiseks mobiilirakendus. Lõputöö käigus tuli valmis saada esialgne versioon rakendusest, mille peal saab hakata hiljem toodet edasi arendama.

Lõputöö eesmärgi saavutamiseks alustati funktsionaalsete nõuete kogumisega, kirja pandi esmatähtsad kasutaja vajadused. Nõuetest vormistati edasi kasutajalood. Veendumaks et kasutajalood saavad arenduse käigus täidetud, koostati kasutajalugudele vastuvõtukriteeriumid. Järgnes olemasolevate sarnaste rakenduste võrdlus, millest autor sai inspiratsiooni arendatava rakenduse loomiseks. Seejärel võrreldi erinevaid mobiilitehnoloogiaid ning valiti välja Flutter, mille peale hakati rakendust looma. Töös kirjeldati ka töö teostamiseks kasutatud vahendeid ning projekti kaustsüsteemi. Ettevõtte konfidentsiaalsuse nõudest lähtuvalt ei käsitletud väga süvitsi rakenduse loomist, kuid ülevaatlikul kujul oli võimalik sellest töös lugeda. Viimaks anti ülevaade töö tulemusest, räägiti millisenä rakendus valmis ja kirjeldati iga rakenduse lehevaadet. Vaatlejate tagasisidest selgus, et loodud rakendusel esines küll mõningaid puudujääke, kuid kogupilt tundus hea. Kuna rakendusega on plaanis tegeleda ka tulevikus siis loetleti viimaks ülesse ka järgmise arendusetapi eesmärgid.

Lõputöö jooksul valmis mobiilirakendusest esialgne versioon, mis täitis lõputöö alguses püstitatud eesmärgid. Loodud tulemusega on rahul nii lõputöö autor kui ka ülejäänud RunProperty tootetiimi liikmed. Autori hinnangul kujunes töö keerukamaks osaks tarkvaraarenduse alguse osa, kus autoril läks planeeritust rohkem aega Flutter'i tundmaõppimiseks.

Kasutatud kirjandus

- [1] „Tarkvara arendusnõuded,” [Võrgumaterjal]. URL: <https://web.htk.tlu.ee/digitaru/testimine/chapter/tarkvara-arendusnouded/>. [Kasutatud 23.04.2022].
- [2] „Story,” 01.07.2021. [Võrgumaterjal]. URL: <https://www.scaledagileframework.com/story/>. [Kasutatud 12.03.2022].
- [3] M. N. Adams, „Brief overview of planning poker,” 10.09.2021. [Võrgumaterjal]. URL: <https://www.atlassian.com/blog/platform/a-brief-overview-of-planning-poker>. [Kasutatud 12.03.2022].
- [4] „What Is User Story And Acceptance Criteria (Example),” 07.04.2022. [Võrgumaterjal]. URL: <https://www.softwaretestinghelp.com/user-story-acceptance-criteria/>. [Kasutatud 23.03.2022].
- [5] „Zoho CRM,” [Võrgumaterjal]. URL: <https://www.zoho.com/crm/>. [Kasutatud 23.04.2022].
- [6] „Zoho CRM mobiile,” [Võrgumaterjal]. URL: <https://www.zoho.com/crm/mobile-app.html?src=mble-hub>. [Kasutatud 19.03.2022].
- [7] „About Rex,” [Võrgumaterjal]. URL: <https://www.rexsoftware.com/about-us>. [Kasutatud 19.03.2022].
- [8] „Rex help Center,” [Võrgumaterjal]. URL: <https://support.rexsoftware.com/>. [Kasutatud 19.03.2022].
- [9] „Kinnisvarafirma müügitarkvara,” [Võrgumaterjal]. URL: <https://www.resolve.ee/>. [Kasutatud 20.03.2022].
- [10] L. S. Vailshery „Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021,” 21.02.2022. [Võrgumaterjal]. URL: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>. [Kasutatud 13.03.2022].
- [11] F. Laricchia, „Market share of mobile operating systems worldwide 2012-2022,” 07.02.2022. [Võrgumaterjal]. URL: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>. [Kasutatud 23.04.2022].

- [12] „Native Apps, Web Apps or Hybrid Apps What’s the Difference?,” [Võrgumaterjal]. URL: <https://www.mobiloud.com/blog/native-web-or-hybrid-apps>. [Kasutatud 23.04.2022].
- [13] „5 Key Benefits of Native Mobile App Development,” [Võrgumaterjal]. URL: <https://clearbridgemobile.com/benefits-of-native-mobile-app-development/>. [Kasutatud 20.03.2022].
- [14] L. S. Vailshery, „Most used web frameworks among developers worldwide, as of 2021,” 23.02.2022. [Võrgumaterjal]. URL: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>. [Kasutatud 20.03.2022].
- [15] „What is Flutter,” [Võrgumaterjal]. URL: <https://docs.flutter.dev/resources/faq#what-is-flutter>. [Kasutatud 20.03.2022].
- [16] j. Holmgren, „Flutter is better than React Native*,” 15.03.2022. [Võrgumaterjal]. URL: <https://shift.infinite.red/flutter-is-better-than-react-native-fed10c92a768>. [Kasutatud 21.03.2022].
- [17] „What is xamarin,” 16.12.2021. [Võrgumaterjal]. URL: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>. [Kasutatud 21.03.2022].
- [18] „A tour of C# language,” 18.03.2022. [Võrgumaterjal]. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>. [Kasutatud 21.03.2022].
- [19] N. Kamienski, „React Native vs Xamarin pros and cons (2020 edition),” 25.08.2020. [Võrgumaterjal]. URL: <https://pagepro.co/blog/react-native-vs-xamarin-pros-and-cons/>. [Kasutatud 21.03.2022].
- [20] „6 Pros and 3 Const of Ionic Development,” [Võrgumaterjal]. URL: <https://www.ukad-group.com/blog/6-pros-and-3-cons-of-ionic-development/>. [Kasutatud 23.03.2022].
- [21] C. Griffith, „What is Apache Cordova?,” [Võrgumaterjal]. URL: <https://ionic.io/resources/articles/what-is-apache-cordova>. [Kasutatud 23.03.2022].
- [22] „Difference between Ionic and Cordova,” [Võrgumaterjal]. URL: <https://www.javatpoint.com/ionic-vs-cordova>. [Kasutatud 23.03.2022].
- [23] E. Toonen, „What is a progressive web app (PWA)? Why would you want one?,” 06.05.2020. [Võrgumaterjal]. URL: <https://yoast.com/what-is-a-progressive-web-app-pwa/>. [Kasutatud 24.03.2022].
- [24] M. Muchmore, „PWAs Turn Websites Into Apps: Here’s How,” 06.10.2021. [Võrgumaterjal]. URL: <https://www.pcmag.com/how-to/how-to-use-progressive-web-apps>, [kasutatud 24.03.2022].

- [25] A. Kumar, „Comparing Different Types of Mobile Application Development,” [Võrgumaterjal]. URL: <https://blog.vsoftconsulting.com/blog/comparing-different-types-of-mobile-application-development>, [kasutatud 26.03.2022].
- [26] „Native, hübriid või PWA mobiiliäpp - milline on sinu ettevõtte jaoks õige?,” [Võrgumaterjal]. URL: <https://www.uptime.ee/native-hubriid-voi-pwa-milline-rakendus-on-sinu-ettevotte-jaoks-oige/>. [Kasutatud 13.03.2022].
- [27] E. Dziadosz, „Mobile Technologies Comparison,” 10.08.2021. [Võrgumaterjal]. URL: <https://www.netguru.com/blog/mobile-technologies-comparison>. [Kasutatud 27.03.2022].
- [28] M. Georgiou, „Cost of Mobile App Maintenance in 2022 and Why It’s Needed,” 15.12.2021. [Võrgumaterjal]. URL: <https://imagination.net/blog/importance-mobile-app-maintenance-cost/>. [Kasutatud 26.03.2022].
- [29] R. Silva, „Scalable folder structure for Flutter applications,” 20.12.2021. [Võrgumaterjal]. URL: <https://medium.com/flutter-community/scalable-folder-structure-for-flutter-applications-183746bdc320>. [Kasutatud 13.04.2022].
- [30] Rex CRM team, „Feature release: Pocket – your app open homes,” 28.06.2017. [Võrgumaterjal]. URL: <https://www.rexsoftware.com/articles/feature-release-pocket-open-homes>. [Kasutatud 19.03.2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Villem Madisson

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Kinnisvara haldussüsteemi mobiilirakenduse arendamine ettevõttele RunProperty”, mille juhendajaks on Kristiina Hakk
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

16.05.2022

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Kasutajalood

Tabel 2. Kasutajalood.

Jrk nr	Roll	Soovin ma ...	Selleks, et ...	Planeerimispokkeri hinne
1	Maaklerina	oma RunProperty kasutajasse sisse logida	saada ligipääs RunProperty keskkonnas olevale informatsioonile	2
2	Maaklerina	näha kõigi kontaktide esmast infot nimekirjana	saada kiire ülevaade olemasolevatest kontaktidest	2
3	Maaklerina	näha ühe kontakti vaadet	saada kontaktist detailsemat ülevaadet	2
4	Maaklerina	luua kontakti	näha hiljem kliendi andmeid	2

Jrk nr	Roll	Soovin ma ...	Selleks, et ...	Planeerimispokkeri hinne
5	Maaklerina	näha kõigi kuulutuste esmast infot nimekirjana	saada kiire ülevaate olemasolevatest kuulutustest	2
6	Maaklerina	näha ühe kuulutuse vaadet	saada kuulutusest detailsemat ülevaadet	2
7	Maaklerina	näha kõigi projektide esmast infot nimekirjana	saada kiire ülevaate olemasolevatest projektidest	2
8	Maaklerina	näha ühe projekti vaadet	saada projektist detailsemat ülevaadet	2
9	Maaklerina	näha märget kui leht laeb	veenduda, et leht laeb	1

Lisa 3 – Vastuvõtukriteeriumid

Tabel 3. Vastuvõtukriteeriumid.

Jrk nr	Kasutajalugu	Stsenaariumi nr	Stsenaarium
1	Maaklerina soovin ma RunProperty kasutajasse sisse logida.	1	Kasutaja avab rakenduse esimest korda, peale mida näeb ta sisse logimise vormi ning selle korrektsel täitmisel logitakse kasutaja sisse ja näidatakse ülevaate lehte.
		2	Kasutaja avab rakenduse teist korda. Kasutajale kuvatakse koheselt ülevaate lehte.
		3	Kasutaja ebaõnnestub sisselogimisel ja ta jääb sisse logimise lehele, kus kuvatakse talle veateadet.
2	Maaklerina soovin ma näha kõigi kontaktide esmast infot nimekirjana.	1	Kasutaja avab rakendusest kontaktide nimekirja vaate, kust ta näeb iga kontakti eesnime ja perekonna nime.
		2	Kasutaja avab rakendusest kontaktide nimekirja vaate. Kontaktide puudumisel kuvatakse vastav teade kasutajale.

Jrk nr	Kasutajalugu	Stsenaariumi nr	Stsenaarium
3	Maaklerina soovin ma näha ühe kontakti vaadet.	1	Kasutajale kuvatakse kontakti detail vaates kontakti eesnime, perekonnanime, telefoni numbrit, e-maili, organisatsiooni, rolli ja seotuid projekte.
4	Maaklerina soovin ma luua kontakti.	1	Kasutaja läheb kontakti loomise lehele, kust avaneb võimalus täita kontakti väljasid(eesnimi, perekonnanimi, e-mail, organisatsioon, roll, seotud projekt). Kasutaja täidab väljad korrektselt ja vajutab salvesta nuppu. Kasutajale kuvatakse järgmiseks kontakti detail vaadet.
		2	Kasutaja läheb kontakti loomise lehele. Kasutaja unustab lisada kontakti eesnime, mis on vormi kohustuslik väli. Kasutaja vajutab salvesta nuppu, peale mida kuvatakse teade, mis annab märku, et eesnime väli on jäänud täitmata.
5	Maaklerina soovin ma näha kõigi kuulutuste esmast infot nimekirjana.	1	Kasutaja avab rakendusest kuulutuste nimekirja vaate, kust ta näeb iga kuulutuse aadressi.
		2	Kasutaja avab rakendusest kuulutuste nimekirja vaate. Kuulutuste puudumisel kuvatakse vastav teade kasutajale.

Jrk nr	Kasutajalugu	Stsenaariumi nr	Stsenaarium
6	Maaklerina soovin ma näha ühe kuulutuse vaadet.	1	Kasutaja vajutab kuulutuste vaates kuulutuse peale, peale mida avaneb külgpaneel, kus kuvatakse kuulutuse andmeid (aadress, pildid, loomise kuupäev, kuulutuse liik, kinnisvaratüüp, maja number, hind, hind ruutmeetri kohta, pindala, seisukord, korteri number).
7	Maaklerina soovin ma näha kõigi projektide esmast infot nimekirjana.	1	Kasutaja avab rakendusest projektide nimekirja vaate, kust ta näeb iga projekti pilti, nime ja projekti tüüpi.
		2	Kasutaja avab rakendusest kuulutuste nimekirja vaate. Projektide puudumisel kuvatakse vastav teade kasutajale .
8	Maaklerina soovin ma näha ühe projekti vaadet.	1	Kasutaja vajutab projektide vaates projekti peale, peale mida kuvatakse kasutajale projekti detailvaate lehte. Lehel kuvatakse projekti nime.
9	Maaklerina soovin ma näha märget kui leht laeb.	1	Kasutaja navigeerib vaatele, mis teostab andmete pärimist. Kasutajale kuvatakse ekraanil senikaua laadimisrõngast kuni andmeid ekraanile kuvatakse.

Lisa 4 – Mobiilitehnoloogiate võrdlus

Tabel 4. Mobiilitehnoloogiate võrdlus.

Kriteerium / Tehnoloogia	Platvormi põhised rakendused	React Native	Flutter	Xamarin	Ionic ja Apache Cordova	PWA
Vajab eraldi arendust IOS ja androidi seadmetele	+	-	-	+/-	-	-
Juurdepääs platvormipõhistele funktsionaalsustele	Hea	Hea	Hea	Hea	Hea	Kehv
Jõudlus	Hea	Keskmine	Keskmine	Keskmine	Keskmine	Kehv
Turvalisus	Kõrge	Keskmine	Keskmine	Keskmine	Keskmine	Keskmine
Arenduse ajakulu	Kõrge	Keskmine	Keskmine	Keskmine	Madal	Madal
Hoolduskulud	Kõrge	Madal	Madal	Madal	Madal	Madal
Kogemus	Puudub	Vähene	Vähene	Vähene	Vähene	Vähene