

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Iris Nael 179067IABB

Darja Molodtsova 179734IABB

Keidi Kaasik 179352IABB

**DUCKEYTAXI: TAKSOTEENUST
OSUTAVA MOBIILIRAKENDUSE
PROTOTÜÜP**

Bakalaureusetöö

Juhendajad: Viljam Puusep
Magister
Gunnar Piho
Doktor

Tallinn 2020

Autorideklaratsioon

Kinnitame, et oleme 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.

Autorid: Iris Nael, Darja Molodtsova, Keidi Kaasik

16.05.2020

Annotatsioon

Antud bakalaureusetöö eesmärgiks on luua funktsionaalne mobiilirakenduse prototüüp, mida saaks kasutada takso tellimiseks või takso teenuse osutamiseks. Probleem, mida rakendus lahendaks on sobiva sõidujagamise mobiilirakenduse puudumine Eesti väikelinnades tegutsevatele taksojuhtidele. Rakenduse realiseerimiseks kasutati Android Studio programmi, Java programmeerimiskeelt ning lokaalset andmebaasi SQLite.

Töö sisaldab potentsiaalsetele klientidele mõeldud küsimustiku analüüsi, juba olemasolevate sarnaste rakenduste analüüsi, ülevaadet tehtud tööst ja selleks kasutatud tööriistadest ning peamisi täiendusi, mida rakenduses oleks vaja veel teha, et valmiks täielikult töötav mobiilirakendus sõidujagamiseks.

Töö tulemusena valminud prototüüp võimaldab kliendil registreerida, sisse logida ja endale meelepärane takso tellida. Juhul on võimalik esitada taotlus juhiks saamiseks, sisse logida ning võtta vastu klientide tellimusi.

Antud lõputöö on valminud meeskondliku projektina, kus iga liige osales mobiilirakenduse loomisel, disainimisel ning dokumentatsiooni kirjutamisel. Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 7 peatükki, 14 joonist, 6 tabelit.

Abstract

DuckeyTaxi: Mobile application prototype for taxi service

The goal of this bachelor's thesis is to create a functional mobile application prototype. This application can be used to order yourself a taxi or to offer taxi services. The main problem that this application attempts to solve is the lack of suitable ridesharing mobile applications for taxi drivers in small Estonian towns. This prototype was developed using Android Studio, Java programming language and local database SQLite.

This thesis contains an analysis of a questionnaire for potential customers, an analysis of similar applications that are already in use, an overview of the work that was done and the tools that were used, and the main things that need to be done in the future to have a fully operational mobile application for ridesharing.

The outcome of this thesis is a prototype that allows the customer to register, log in and order a taxi of their choice. It is possible for the driver to apply in order to become a driver, log in and accept customer orders.

Given thesis has been done as a team project, where every member participated in creating, designing and documenting a mobile application. The thesis is written in Estonian and contains 30 pages of text, 7 chapters, 14 figures, 6 tables.

Lühendite ja mõistete sõnastik

IDE - *Integrated development environment*, e. integreeritud programmeerimiskeskond

XML - *Extensible Markup Language*, laiendatud märgistuskeel

SDK - *Software Development Kit*, tarkvaraarenduskomplekt

Java – objekt orienteeritud programmeerimiskeel

API – *Application Programming Interface*, rakendusliides

Messagebox - sõnumi aken

Pop-up - hüpinkaken

Unit - ühik

Android - operatsioonisüsteem

iOS – Apple operatsioonisüsteem

Scrum – raamistik, mis aitab projektide läbiviimisel

Task - ülesanne

Sisukord

1 Sissejuhatus	10
2 Rakenduste võrdlus	12
2.1 Yandex Taxi	12
2.2 Uber	14
2.3 Bolt (Taxify)	15
2.4 Nõuded rakendusele	16
3 Potentsiaalsetele klientidele mõeldud küsimustiku analüüs	19
4 Ülevaade kasutatud tööriistadest	23
4.1 Android Studio	23
4.2 Java	23
4.3 XML	24
4.4 Maps SDK	24
4.5 Espresso UI Testing	24
4.6 SQLite	24
4.7 Trello	25
5 Funktsionaalse prototüübi loomine	26
5.1 Klient	27
5.1.1 Sisselogimine ja kliendiks registreerimine	27
5.1.2 Kliendi peavaade	28
5.2 Taksojuht	29
5.2.1 Sisse logimine ja taksojuhi taotluse saatmine	29
5.2.2 Taksojuhi kaardi vaade	31
5.3 Navigatsiooniriba	32
5.4 Testid	33
6 Analüüs ja järeldused	35
6.1 Küsimustiku analüüsi kokkuvõte	35
6.2 Tehnilised otsused	35
6.3 Andmebaas	36
6.4 Nõuded	37

6.5 Hinnang projekti teostamise kohta	38
6.6 Teostatud tööde logi	39
6.7 Täiendused, mida võiks tulevikus teha.....	39
6.8 Meeskondliku töö analüüs	40
7 Kokkuvõte	41
Kasutatud kirjandus	42
Lisa 1 – Taksode võrdlus tabel	44
Lisa 2 - Küsimustik	45
Lisa 3 - DB Browser For SQLite.....	48
Lisa 4 - Trello	49
Lisa 5 – Taksojuhi ja kliendi tegevusskeem.....	50
Lisa 6 - Klassidiagramm.....	51
Lisa 7 – Kliendi vaade “Telli takso”	52
Lisa 8 – Taksojuhiks taotlemine.....	53
Lisa 8 - jätkub.....	54
Lisa 9 – Administraator vaade - kõik taotlused.....	55
Lisa 10 – Administraator vaade - üksiku taksojuhi taotlus	56
Lisa 11 – Taksojuhi kaardi vaade – marker kaardil koos kliendi andmetega.....	57
Lisa 12 - “Minu sõidud” vaade.....	58
Lisa 13 - “Abi” vaade	59
Lisa 14 – Scrum.....	60
Lisa 15 – Iris isiklik panus.....	61
Minu tegevused projektis.....	61
Eneseanalüüs	62
Lisa 16 - Darja isiklik panus.....	64
Minu tegevused projektis.....	64
Eneseanalüüs	65
Lisa 17 – Keidi isiklik panus	67
Minu tegevused projektis.....	67
Eneseanalüüs	68

Jooniste loetelu

Joonis 1. Küsimus “Kus on teie peamine elukoht?”	19
Joonis 2. Küsimus “Kust olete pärit?	20
Joonis 3. Küsimus “Mis eesmärgil kasutate taksot?”	20
Joonis 4. Küsimus “Kuidas tellite endale taksot?”	21
Joonis 5. Küsimus “Milline võiks olla hea taksosõidu mobiilirakendus?”	22
Joonis 6. DuckeyTaxi rakenduse avanemise vaade.....	26
Joonis 7. Kliendi sisse logimise vaade	27
Joonis 8. Taksojuhti otsimine (vasakul) ja taksojuht on võtnud tellimuse vastu (paremal)	29
Joonis 9. Taksojuhi sisse logimise vaade	30
Joonis 10. Taksojuht on valmis klientide tellimusi vastu võtma.....	31
Joonis 11. Kliendi (vasakul) ja taksojuhi (paremal) navigatsiooniribad	32
Joonis 12. Ühiktestid aadressi lühendamise meetodi jaoks (vasakul) ning pika klikki Espresso UI testid kliendi kaardil (paremal).	34
Joonis 13. Projekti päis GitLab-is.....	39
Joonis 14. DuckeyTaxi logo	68

Tabelite loetelu

Tabel 1. Yandex taksode tariifid Tallinna linnas.....	13
Tabel 2. Yandex taksode tariifid Tartu linnas	14
Tabel 3. Yandex taksode tariifid Narva linnas	14
Tabel 4. Uber taksode tariifid.....	15
Tabel 5. Bolt taksode tariifid	16
Tabel 6. Duckey Taxi tariifid	17

1 Sissejuhatus

Tänapäeval on üha populaarsemaks muutumas sõidujagamisteenust osutavad mobiilirakendused. Kui varasemalt telliti taksosid peamiselt helistades, siis nüüd on see üle viidud nutiseadmetele, mis võimaldavad kliendil tellida endale takso paari vajutusega. Olemasolevate rakenduste puhul pole oluline isegi, kas kliendil on olemas sularaha, sest rakendust on võimalik ühendada oma pangakontoga, kust raha automaatselt sõidu lõppedes maha läheb.

Enamus rakendused toimivad hästi suuremates linnades, kus on kliente palju, kuid piirkonnad, kus on inimesi vähem või vahemaad suuremad ei ole juba olemasolevad sõidujagamisteenust osutavad takso mobiilirakendused juhisõbralikud. Tihtipeale kliendilt saadud sõidutasu jätab juhi pigem kahjumisse kui kasumisse.

Bakalaureusetöö eesmärgiks on valmis teha rakenduse prototüüp, mille kasutajatel on võimalik tellida endale takso, lisades soovitud pealevõtmis- ja mahapanekukoha, samuti saab klient juurde märkida, mis taksotüüpi soovitakse ja kas on erisoove nagu näiteks takso kasutamine koos lemmikloomaga. Samuti on rakendus kasutatav ka taksojuhtidele, kellel on võimalik võtta vastu tellimusi nähes kliendi infot.

Rakenduse loomiseks kasutati Android Studio platvormi, kus on võimalik üles ehitada mobiilirakendus Android tüüpi telefonidele. Android Studios saab käivitada oma rakendust emulaatoris, mis võimaldab luua mitmeid nutiseadmeid, millel on erinevad ekraanisuurused ja operatsioonisüsteemid. Rakenduse loomiseks kirjutati kood kasutades Java ja XML programmeerimiskeeli. Selleks, et luua võimalikult kliendisõbralik rakendus, uuriti ka potentsiaalsete klientide soove, mis on nende jaoks olulised ühe taksoteenust osutava mobiilirakenduse puhul.

Käesolev töö on jagatud kaheksaks osaks. Esimene ehk antud peatükk selgitab antud töö tausta, ülesande püstitust ning pakub ülevaadet töö peatükkidest. Teises peatükis analüüsitakse sarnaseid olemasolevaid rakendusi ning kirjeldatakse esialgseid nõudeid. Tehtud analüüsi alusel on moodustatud tariifid DuckeyTaxi rakenduse jaoks. Kolmandas peatükis analüüsitakse potentsiaalsetele klientidele esitatud küsimustiku vastuseid, et selgitada välja kasutajatele vajalikud tegurid. Neljandas peatükis on kirjeldatud rakenduse

prototüübi loomiseks kasutatud tööriistad. Viies peatükk kirjeldab funktsionaalse prototüübi loomise protsessi. Kuuendas peatükis analüüsitakse tehtud tehnilisi otsuseid ja rakendusele esitatavaid nõudeid, kirjeldatakse andmebaasi ja selles olevaid tabeleid, hinnatakse projekti teostamist ja kirjeldatakse rakenduse võimalikke tulevasi täiendusi. Lisaks on kuuendas peatükis analüüsitud meeskonnatööd. Seitsmes peatükk on käesoleva bakalaureusetöö kokkuvõte.

2 Rakenduste võrdlus

Käesolevas peatükis on kirjeldatud juba eksisteerivaid taksoteenuste osutamiseks mõeldud mobiilirakendusi. Iga lahendust on analüüsitud nende rakenduste funktsionaalsuse (Lisa 1) ning sõidu hindade põhjal. Analüüsi põhjal kirjeldatakse nõudeid töö raames loodavale mobiilirakendusele.

Käesoleva töö jaoks on autorid analüüsinud hetkel kolme kõige populaarsemat Eesti turul tegutsevat rakendust: Yandex Taxi, Uber ja Bolt.

2.1 Yandex Taxi

2018 aastal tuli Eestis kasutusele Venemaal sõidujagamisteenust pakkuva ettevõtte Yandex Taxi mobiilirakendus [1]. Rakendusse on võimalik sisse logida kasutades e-maili või mobiiltelefoni numbrit. Sõitu tellides saab märkida, kelle jaoks takso tellitakse ning kui sõit tellitakse kellelegi teisele saadetakse SMS sõiduki infoga sõitu soovivale isikule edasi [2].

Võimalik on lisada mitu peatust ja valida erinevate teenuseklasside vahel (säästuklass, mugavusklass, *premium*). Lisaks on valikus ka "kiireim", mis tähendab, et mitme teenuseklassi hulgast valitakse lähim auto. Samuti on rakenduses võimalus märkida, et soovitakse sõitu lapsega ning on vaja kasutada turvatooli, uue takso tüübina on lisandunud Yandex Taxi rakendusse ka pakivedu, mille eesmärk on aidata klientidel kiirelt pakke saata [2].

Rakenduses on võimalik luua pere- ja ärikontod, näha enda sõitude ajalugu, hallata makseviise, lisada lemmik aadresse, kasutada sooduskuponge, muuta rakenduse seadeid. Samuti on rakenduses menüüs olemas valik "ohutus", kus on võimalik ühe nupuvajutusega saada ühendust kiirabi ja politseiga, usaldusväärse kontaktiga või anda märku hädaolukorrast [2].

Yandex Taxi mobiilirakendusest saab tellida taksot kolme piirkonda – Tallinn, Ida-Virumaa (Narva, Jõhvi, Kohtla-Järve) ja Tartu. Igas linnas on erinevaid takso tüüpe, kõige rohkem neist Tallinnas [2].

Tallinnas on võimalik leida endale taksot kiirelt ning igapäevale sobiva hinnaga (Tabel 1). Kõige odavam takso tüüp on “Säästuklass”, mille sõidu alustustasuks on 1,80 eurot ja kilomeetri hind varieerub vahemikus 0,26 eur/km kuni 0,80 eur/km. Minuti hind säästuklassil varieerub 0,13 eur/min kuni 0,15 eurt/min. Varieeruvus sõltub sõidetud kilomeetrite arvust ning kas sõidetud on linnas või äärelinnas. Premium klassi takso sõidu alustustasu on 3,70 eurot ning kilomeetri hind varieerub 0,55 eur/km kuni 1,20 eur/km ja minuti hind varieerub 0,18 eur/min kuni 0,28 eur/min [3].

Yandex	Säästuklass	Mugavusklass	Lapsed	Premium	Pakivedu
Alustustasu (eurot)	1,80	2,50	3	3,70	1,80
Ooteaeg (min)	3	3	5	3	3
Makstud ooteaeg (eur/min)	0,15	0,18	0,15	0,18	0,15
Seejärel linnas (eur/km)	0,26	0,35	0,3	0,55	0,26
Seejärel linnas (eur/min)	0,13	0,2	0,15	0,28	0,13
Linnas üle 8 km (eur/km)	0,55	0,55	0,55	0,60	0,55
Linnas üle 8 km (eur/min)	0,13	0,2	0,15	0,28	0,13
Seejärel äärelinnas (eur/km)	0,55	0,65	0,55	1	0,55
Äärelinnas üle 8 km (eur/km)	0,80	0,80	0,80	1,20	0,80
Seejärel äärelinnas (eur/min)	0,15	0,20	0,15	0,28	0,15
Ootamine sõidu ajal (eur)	0,15	0,20	0,15	0,28	0,15
Lapse turvaiste (eurot)	-	-	0	-	-
Uksest ukseni (eurot)	-	-	-	-	1

Tabel 1. Yandex taksode tariifid Tallinna linnas

Tartu linnas on võimalik tellida vaid ühte klassi taksot - säästu (Tabel 2). Alustustasu 2 eurot, kilomeetri hind linnas 0,54 eur/km ja äärelinnas 1,08 eur/km. Ooteaeg sõiduajal on 0,1 eurot [4].

Yandex	Säästuklass
Alustustasu (eurot)	2
Ooteaeg (min)	3
Makstud ooteaeg (eur/min)	0,1
Seejärel linnas (eur/km)	0,54
Seejärel äärelinnas (eur/km)	1,08
Ootamine sõidu ajal (eur)	0,1

Tabel 2. Yandex taksode tariifid Tartu linnas

Ida-Virumaal (eelkõige Narvas) on samuti kättesaadav üksainus säästuklass (Tabel 3), mille sõidu alustus maksab 2,30 eurot ning kilomeetri hind linnas on 0,34 eur/km ja äärelinnas 0,50 eur/km. Minuti hind kilomeetri kohta linnas on 0,12 eur/min ja äärelinnas 0,15 eur/min [5].

Yandex	Säästuklass
Alustustasu (eurot)	2,3
Ooteaeg (min)	3
Makstud ooteaeg (eur/min)	0,12
Seejärel linnas (eur/km)	0,34
Seejärel linnas (eur/min)	0,12
Hind äärelinnas (eur/km)	0,50
Hind äärelinnas (eur/min)	0,15
Ootamine sõidu ajal (eur)	0,15

Tabel 3. Yandex taksode tariifid Narva linnas

2.2 Uber

2015 aastast võeti Eestis kasutusele sõidujagamisteenust pakkuva San Francisco ettevõtte Uber Technologies rakendus [6]. Antud rakendus lubab kliendil sisselogimist läbi Facebooki ja Google kontode ning lisaks neile on võimalik sisselogimiseks kasutada mobiiltelefoni numbrit. Rakenduses saab tellida taksot ette, kuid ei garanteerita, et takso tuleb. Taksot tellides saab klient lisada sõidule mitu peatust ning valida taskukohane sõiduk või tippklassi sõiduk. Menüüs on võimalik kliendil näha varasemaid sõite, hallata makse meetodeid, kasutada sooduskuponge ning muuta konto seadeid (lisada lemmik aadressi, andmeid hallata) [7].

Uberil on Eestis kasutusel kaks takso klassi – UberX ja UberSelect (Tabel 4). UberX on säästlikum takso versioon, kus sõidu alustustasuks on 1,19 eurot ning kilomeetri hind

kuni 13-nda kilomeetrini on 0,35 eurot ning edaspidi 0,62 eurot ja minuti hind püsivalt 0,12 eurot. Minimaalseks sõidutasuks ning ka tühistamistasuks on 2,5 eurot [8].

UberSelect on premium klassi takso. Baashind on 2,5 eurot ning hind kilomeetri kohta on 0,42 eurot ja minuti kohta 0,14 eurot. Minimaalse sõidutasu ja tühistamistasu on mõlemad 4 eurot [9].

Uber	UberX	UberSelect
Baashind	1,19	2,50
Hind km kohta (eur)	0,35	0,42
Hind minuti kohta (eur)	0,12	0,14
Minimaalne sõidutasu (eur)	2,50	4
Tühistamistasu (eur)	2,50	4
Hind km kohta alates tellimuse 13 km-st (eur)	0,62	-

Tabel 4. Uber taksode tariifid

2.3 Bolt (Taxify)

2013 aastal asutati Eesti ettevõtte Bolt (endise nimega Taxify), mille põhitegevus on sõidujagamine [10]. Rakendus lubab sisselogimist läbi Facebooki või kasutades mobiiltelefoni numbrit. Sõitu tellides on võimalik lisada mitu peatust, valida mitme sõiduki vahel (bolt, elektriauto, *basic*, *premium*, *XL*, ehk auto mis mahutab üle nelja inimese) lisaks on võimalus valida sõiduk, kus on olemas turvaiste lapsele või abistav sõiduk, kus juht saab klienti vajadusel abistada. Menüüs saab klient hallata makseid, näha oma sõitude ajalugu, kasutada sooduskupongi, saada abi varasemate sõitudega seoses. Rakenduses on võimalik luua ka oma äriprofiil [11]. Bolt mobiilirakendus teenindab kümmet Eesti linna: Haapsalu, Jõhvi, Kohtla-Järve, Kuressaare, Narva, Pärnu, Rakvere, Tallinn, Tartu ja Viljandi [12].

Vastavalt igale klassile ja linnale on takso hinnad (Tabel 5) erinevad. Rakveres Bolt tüüpi takso sõidu alustustasuks on 2,5 eurot ja *Comfort* tüübil 4 eurot. Hinnad kilomeetri kohta on vastavalt 0,35 eurot kilomeeter Boltil ja 0,69 eurot *Comfort*-il.

Tallinnas on Bolt alustustasu 2,50 ja Premiumil 4 eurot. Kilomeetri hinnad 0,39 eurot ja 0,60 eurot kilomeeter ning minut maksab 0,14 ja 0,30 eurot. Elektriauto valides taksona on alustustasu 1,19 eurot ning kilomeetri hind 0,29 eurot ja minuti hind 0,14 eurot [11].

BOLT	Bolt (Rakvere)	Comfort (Rakvere)	Bolt	Basic	Electric	Premium	XL	Child Seat	Assist
Miini- mum- hind (eur)	2,50	4	2,50	2,50	2,50	4	3,50	3	5
Alustus- tasu (eur)	1,50	2,95	1,19	0,99	1,19	3	3	2,19	4,19
Hind km kohta	0,35	0,69	0,39	0,35	0,29	0,60	0,60	0,39	0,39
Hind min kohta	0,13	0,20	0,14	0,10	0,14	0,30	0,20	0,14	0,14
Ooteae- g (eur)	0,13	0,20	0,14	0,10	0,14	0,30	0,20	0,14	0,14
Tühis- tamis- tasu (eur)	2,50	4	2,50	2,50	2,50	4	3,50	3	3

Tabel 5. Bolt taksode tariifid

2.4 Nõuded rakendusele

Analüüsidest teiste mobiilirakenduste taksode tüüpe ja hindu jõudsid töö autorid otsusele võtta rakenduses kasutusele kolm peamist takso tüüpi – *normal* (tava), *business* (*premium*) ja elektriauto. Juhtidel on võimalus registreerides märkida juurde, kas nende taksosse mahub rohkem kui 4 reisijat, kas nad on nõus sõidutama oma autos ka looma (kass, koer või mõni teine väiksem lemmikloom), kas nad saavad pakkuda sõitu ka kliendile, kellel oleks vaja lastele mõeldud turvaistet ning kas juht on valmis vajadusel klienti abistama. Rakendus tuleks esialgu kasutusele väike linnades.

Kliendid peavad saama rakenduses teha järgnevaid toiminguid:

- Sisse ja välja logimine
- Takso tellimine vastavalt oma vajadusele ning selle tühistamine
- Profiili nägemine ja muutmine
- Sõitude ajaloo nägemine

Juhid peavad saama rakenduses teha järgnevaid toiminguid:

- Sisse ja välja logimine

- Klientide tellimuste nägemine, nende vastuvõtmine või tühistamine
- Profiili nägemine ja muutmine

DuckeyTaxi taksode hinnad vastavalt klassidele (Tabel 6):

- *Normal* - sõidu alustustasu 2 eurot. Rakvere linna 30km raadiuses hind kilomeetri kohta 0,75 eurot ja minuti kohta 0,12 eurot. Raadiusest väljaspool 0,85 eur/km ja 0,14 eur/min. Minimaalne sõidu hind peab olema vähemalt 3,5 eurot. Tühistamise korral peab klient maksma 3,5 eurot.
- *Business/Premium* - sõidu alustuseks on 2,7 eurot ja miinimum sõidu hinnaks määratud 4,2 eurot. Raadiuse sees kilomeetri hind on 0,95 eurot ja minuti hind 0,16 eurot ning raadiusest väljaspool 1,05 eur/km ja 0,18 eur/min. Takso tühistamisel tuleb maksta 4.2 eurot.
- Elektriauto - sõidu baashinnaks on 2,5 eurot, miinimumtasuks 3,2 eurot ja sõidu tühistushinnaks on samuti 3,2 eurot. Raadiuses kilomeetri hind on 0,85 eurot ja raadiusest väljas 0,95 eurot. Minuti hinnaks 0,14 eurot raadiuses ja väljas 0,17 eur/min.

DuckeyTaxi	NORMAL	BUSINESS	ELEKTRIAUTO
Sõidu alustus (eur)	2	2,70	2,50
Miinimum tasu (eur)	3,50	4,20	3,20
Hind km kohta (eur/km) [Rakvere linna raadiuses 30km]	0,75	0,95	0,85
Hind min kohta (eur/min) [Rakvere linna raadiuses 30km]	0,12	0,16	0,14
Hind km kohta (eur/km) [Väljaspool raadiust]	0,85	1,05	0,95
Hind min kohta (eur/min) [Väljaspool raadiust]	0,14	0,18	0,17
Ooteaeg (eur)	0,12	0,16	0,14
Tühistamine (eur)	3,50	4,20	3,20

Tabel 6. Duckey Taxi tariifid

Enamik hindu on siiski kõrgemad kui teistel eksisteerivatel takso rakendustel. Väikelinnades, nagu näiteks Rakvere, ei ole enamikel taksojuhtidel kasutusel juba

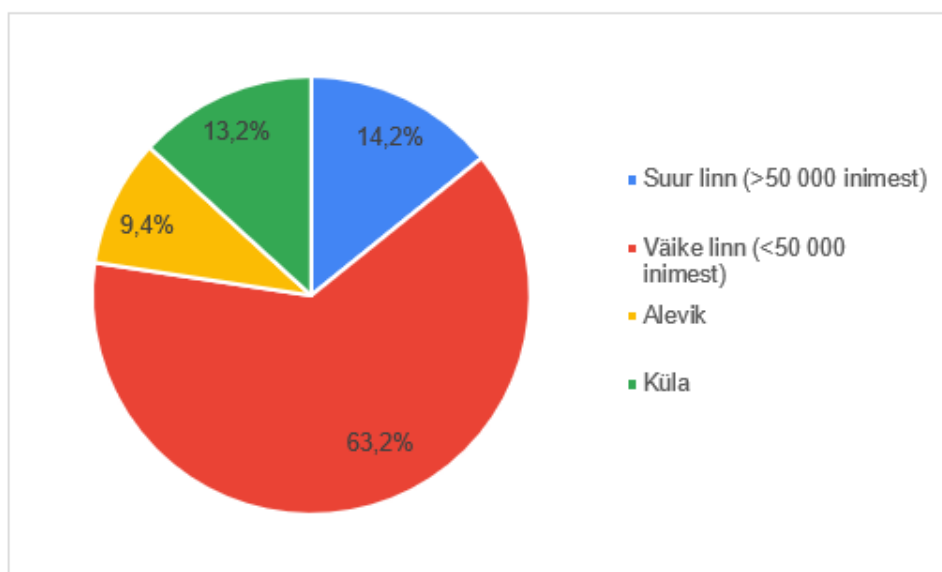
olemasolevad mobiilirakendused nagu näiteks Bolt ja Uber, sest nende rakenduste tariifidega jäävad juhid pigem kahjumisse kui kasumisse.

Võrreldes Tallinna ja Tartuga ei ole väikelinnades pidevat klientide sõidutamist ehk väikelinna taksojuhid viivad oma kliendi lõpp-punkti ning sõidavad tagasi kohta, kus liikleb rohkem rahvast, mis võib jääda mitme kilomeetri kaugusele eelmise kliendi lõpp-punktist. Tallinnas näiteks on taksojuhil võimalik kliendi lõpp-punkti viimist võtta sealt samast piirkonnast uus klient ja see protsess pidevalt kordub – taksojuht ei kaota oma raha, sest tal on pidevalt kliente. Väikelinnas võib aga uus klient jääda mitme kilomeetri kaugusele. Klient maksab juhile küll oma taksosõidu kinni, kuid sellest rahast tuleb juhil maksta osalus rakendusele (10-20%) ning katta kütuse kulu kliendi juurde ning tagasi sõidu eest, mis võib lõpuks olla suurem kui kliendilt saadud tasu.

3 Potentsiaalsetele klientidele mõeldud küsimustiku analüüs

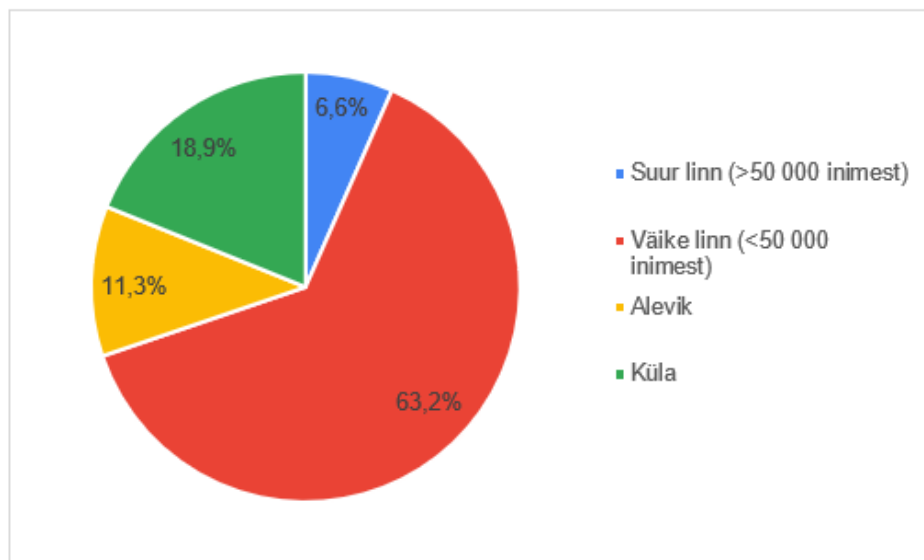
Üks osa antud tööst oli potentsiaalsetele klientidele mõeldud küsimustik (Lisa 2), mille eesmärgiks oli saada arvamusi, milline oleks hea taksotellimise mobiilirakendusest. Küsimustik loodi Google Forms keskkonnas ning koosnes 13 küsimusest, mis olid koostatud nii, et vastaja vastustest sõltus järgnev küsimus. Analüüsi käigus tuuakse välja olulisemad tulemused. Sihtgrupiks oli valitud peamiselt Rakvere ümbruses elavad inimesed, kuna linnas ja selle ümbruses olevad külad ja alevikud ei kasutata aktiivselt takso tellimiseks mobiilirakendusi vaid pigem tellitakse takso helistades.

Vastanute seast 63,2% märkis oma peamiseks elukohaks väiksema linna, kus on alalisi elanikke alla 50 000. 9,4% vastanutest elas alevikus, 13,2% külas ning 14,2% elavad peamiselt suuremas linnas, kus alalisi elanikke üle 50 000 (Joonis 1).



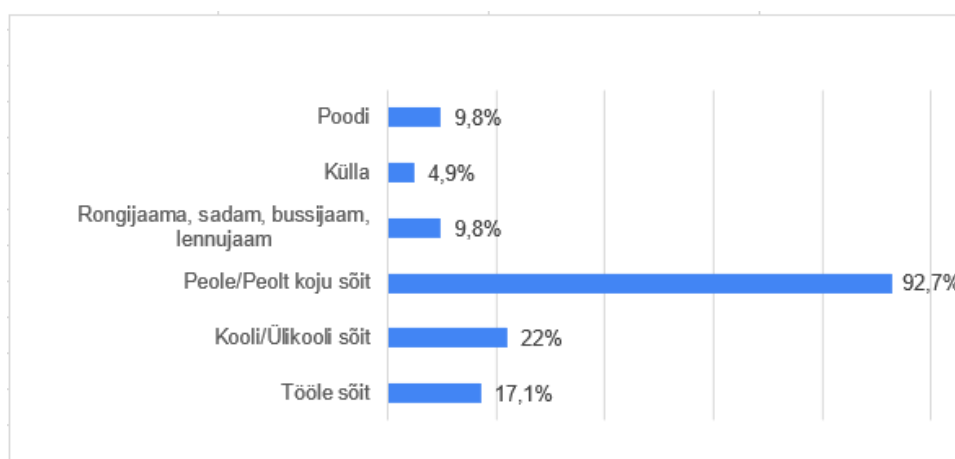
Joonis 1. Küsimus "Kus on teie peamine elukoht?"

Vastanutest vaid 6,6% on pärit suuremast linnast, 63,2% pärinevad väiksemast linnast, 11,3% alevikust ning 18,9% märkisid vastuseks küla (Joonis 2). Võib öelda, et antud küsimustik sisaldab peamiselt väiksemate piirkondade elanike vastuseid.



Joonis 2. Küsimus “Kust olete pärit?”

Edasi jätkus küsimustik vaid neile, kes kasutavad taksot, mis aitab töö autoritel välja selgitada, milline on potentsiaalne klient ja milline oleks kliendile meelepärane rakendus. Potentsiaalsetelt klientidelt uuriti, mis eesmärgil nad tavaliselt taksot kasutavad, selle küsimuse puhul sai vastaja valida mitu vastuse varianti. 92,7% vastanutest märkis, et kasutavad taksot peole või peolt koju minekuks. 17,1% kasutavad taksot, et sõita tööle, 22% kasutavad taksot kooli sõiduks ning ülejäänud vastusevariandid nagu näiteks rongijaama/bussijaama minekuks kasutavad taksot 9,8% vastanutest ja sõpradele külla minekuks valis 4,9% vastanutest (Joonis 3).

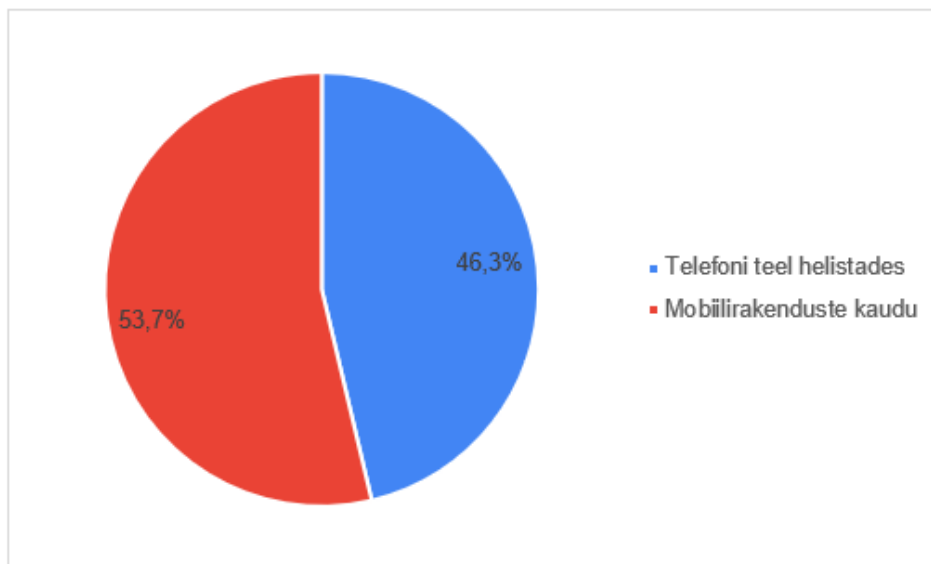


Joonis 3. Küsimus “Mis eesmärgil kasutate taksot?”

Antud küsimus aitab fikseerida populaarsemaid ajad takso kasutuseks. Peamised kasutusajad oleksid hommikud, kui kliendid lähevad tööle ja kooli ning õhtu kui tööpäev

on lõppenud. Samuti nädalavahetuse õhtud ja varahommikud, kui kliendid kasutaksid taksoteenust peole või peolt koju minekuks.

Kuna Rakveres ei ole mobiilirakenduse kasutamine nii populaarne, siis uuriti potentsiaalsetelt klientidelt, kuidas nad eelistavad taksot tellida endale – 46,3% märkisid, et nad helistavad taksofirmadele ning ülejäänud kasutavad mobiilirakendust (Joonis 4).



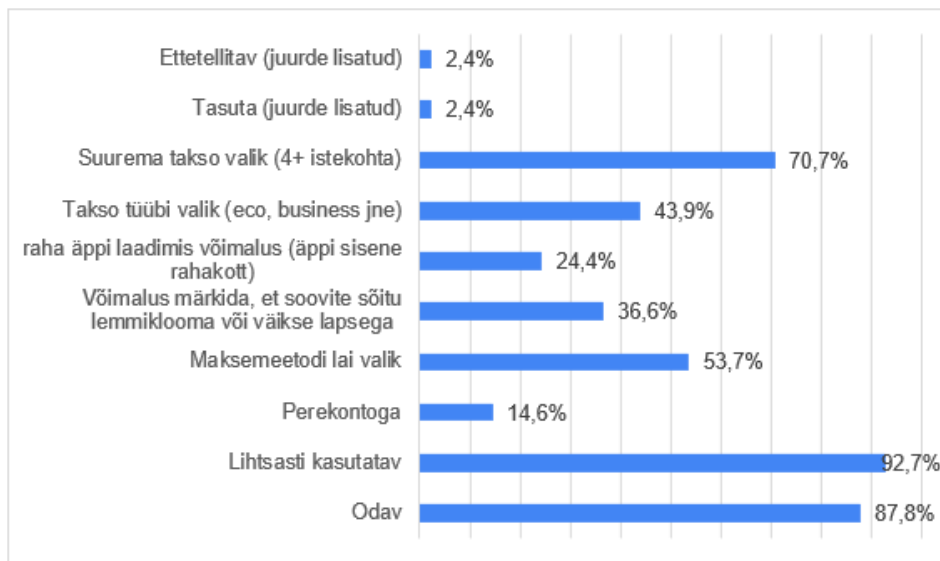
Joonis 4. Küsimus “Kuidas tellite endale taksot?”

Ligi 90% vastanutest märkis, et nad oleksid huvitatud heast takso tellimise mobiilirakendusest.

Vastanud tõid välja, et nad eelistavad rakendust helistamisele, sest see on lihtsam, mugavam ja kiirem, samuti võimalus sõidu eest maksta läbi rakenduse, mis tähendab, et sularaha puudumine ei ole probleemiks, mitmel korral toodi välja ka hirmu helistamise ees. Siiski oli ka neid, kes eelistavad helistamist takso firmadesse, sest on sellega juba harjunud.

Töö autorid uurisid ka vastajatelt, milline võiks olla hea taksosõidu mobiilirakendus, antud küsimuse juures sai vastaja valida mitu vastusevarianti. Ootuspäraselt olid kõige populaarsemad vastused odav ja lihtsasti kasutatav, vastavalt siis 87,8% ja 92,7%. Samuti oli populaarne valik ka see, et oleks võimalik tellida suuremat taksot, kus oleks üle 4 istekoha, selle märkis valituks 70,7% vastanutest. Vähem populaarseks osutusid maksemeetodi lai valik 53,7%, takso tüübi valik (*eco, business* jne) 43,9%, võimalus sõita lemmikloomaga valisid 36,6% vastanutest. Vaid 14,6% vastanutest märkisid, et rakenduses võiks olla ka perekonto võimalus ning 24,4% valisid raha äppi laadimise võimaluse, ehk siis virtuaalnerahakott. Küsimuse juures oli võimalik ka ise vastuseid

juurde lisada ning juurde märgiti, et rakendus võiks olla tasuta allalaetav ja võimalusega takso ette tellida (Joonis 5).



Joonis 5. Küsimus “Milline võiks olla hea taksosõidu mobiilirakendus?”

Oluline oli leida ka see, mis on kliendi jaoks kõige olulisem taksode juures. 75,6% märkis, et nende jaoks on kõige olulisem sõidu hind. 17,1% pidas oluliseks juhti ja juhi hinnangut, 4,9% pidasid kõige tähtsamaks automarki ja 2,4% valis puhtuse.

4 Ülevaade kasutatud tööriistadest

Käesolevas peatükis on detailselt kirjeldatud autorite poolt loodud mobiilirakenduse prototüübi realiseerimiseks kasutatud tööriistad ja nende valimise põhjused.

4.1 Android Studio

Taksoteenust osutava mobiilirakenduse loomiseks kasutati Android Studiot – programm, kus on võimalik ehitada mobiilirakendus Android tüüpi telefonidele. Android Studio on ehitatud JetBrains IntelliJ IDEA baasil ning antud programm toetab kirjutamist kahes programmeerimiskeeles – Java ja Kotlin [13].

Android Studio võimaldab luua rakendust, mis toetab erinevaid Androidi versioone. Programmis saab käivitada mobiilirakendust emulaatoris, mis käitub päris nutiseadmena. Emulaatoris võib luua erinevaid nutiseadmeid nagu näiteks telefon, tahvel, televiisor ja nutikell, millele saab määrata ekraanisuurusi ja Androidi versioone. Emulaatoriga saab testida, kas rakendus jääb nii nagu peaks, et olla kindel, et loodud rakendus sobib kasutamiseks erinevate Androidi versioonidega nutiseadmetes.

4.2 Java

Mobiilirakenduse realiseerimiseks valitud programmiks oli Android Studio, kus programmeerimiskeeleks valiti Java, kuna see on sarnane C# keelega, mida on töö autorid ka varasemalt läbitud ainetes käsitlenud.

Java on klassipõhine, objektorienteeritud programmeerimiskeel [14, p. 1]. See on disainitud nii, et sama koodi on võimalik käivitada igalpool, näiteks Windows masinal kirjutatud kood töötab ka Mac masina Java programmis samamoodi [15, p. 34]. Antud programmeerimiskeel on laialdaselt kasutatud, et luua töölauarakendusi, veebiservereid ning veebirakendusi. Samuti on see Android operatsioonisüsteemi keeleks [15, p. 36].

4.3 XML

Vaadete loomiseks kasutatakse Android Studios XML keelt. XML-i kasutatakse tavaliselt standardformaadina andmete vahetamiseks erinevate rakenduste vahel [16]. Antud juhul XML võimaldab luua visuaalselt nähtavaid vaateid Android rakenduse jaoks.

XML on disainitud korruga loetavaks inimesele ja masinale. Kood on lihtne ning kasutatav üle interneti. Peamiselt on XML fokuseeritud dokumentidele kuid siiski leiab laialdaselt kasutust ka andmestruktuuride esitamisel. Koodi kirjutamisel pannakse rõhk andmete tüübile [17].

4.4 Maps SDK

Kaardi vaate kasutamiseks rakenduses lisasid autorid Google poolt pakutud API - Maps SDK. Maps SDK võimaldab lisada loodavale rakendusele kaardi vaate. Kasutaja saab lisada kaardile asukoha markereid ning näha nende asukohtade informatsiooni [18].

4.5 Espresso UI Testing

Kasutajaliidese testimiseks kasutati Espresso UI Testing-ut. See on UI testimise raamistik, mis on Google poolt avalikustatud 2013 aastal. Raamistiku valimise põhjuseks on Espresso testimise arusaadavus ja lihtsus ning testide käivitamise optimaalne kiirus [19].

4.6 SQLite

Kasutajakontode ja nendega seotud informatsiooni hoidmiseks kasutasime lokaalset andmebaasi, milleks osutus SQLite.

SQLite on serverita andmebaas mida ei pea konfigureerima ning mille kood on avaliku juurdepääsuga ehk vaba igasuguseks kasutamiseks [20]. Kõikides telefonides ja enamustes arvutites on SQLite juba sisseehitatud [21].

Android Studios ei ole lokaalsete andmebaaside kuvamist sisseehitatud, seepärast valiti andmete kuvamiseks lokaalsest andmebaasist programm DB Browser for SQLite (Lisa 3). See on avatud lähtekoodiga tööriist, mis võimaldab luua, disainida ja muuta andmebaase [22].

4.7 Trello

Rakenduse loomiseks tuli teha mitmeid erinevaid *task*-e, mida oli vaja täita. Selleks, et need meeles püsiks ning saada pidevat ülevaadet, millega iga meeskonnaliige hetkel tegeleb oli kasutusse võetud Trello (Lisa 4).

Trello on visuaalne tööriist, kus igas suuruses meeskonnal on võimalik töötada ühe projekti kallal. Rakenduses saab hallata erinevaid “tahvleid”, mis kujutavad projekte ja meeskondi. Ühel tahvlil on võimalik luua mitmeid nimekirju ning soovi korral ühte *task*-i ühes nimekirjast teise liigutada kuni *task* ei saa täidetud [23].

Task-idele on võimalik lisada märke (näiteks punane märk – raske, kollane märk – keskmise raskusega, roheline märk – kerge) ja meeskonna liiget, kes sellega tegelema peab. Püstitatud ülesannetele on võimalik lisada tähtaeg, millal ülesanne peab valmis saama, samuti võib lisada erinevaid liste, et saaks ülesannet teha väiksemateks osadeks. Trellos toimuvad kõik uuendused reaalajas ehk kui keegi muudab ülesannet Trellos siis muudatust näevad kohe ka teised meeskonnaliikmed – nad näevad aktuaalset informatsiooni [23].

5 Funktsionaalse prototüübi loomine

DuckeyTaxi on rakendus, kus klient saab tellida endale taksot ning juht saab pakkuda sõidu teenust (Lisa 5). Rakendust avades seisab kasutaja ees rolli valik (Joonis 6) – kas kasutaja soovib tellida taksot ja olla takso klient või pakkuda oma takso teenuseid ja olla taksojuht.



Joonis 6. DuckeyTaxi rakenduse avanemise vaade

Rakenduses on peidetud tavakasutajate silmade eest kolmas roll – administraator, kelle ülesanne on vastu võtta taksojuhtide taotlusi. Administraatori rollile saame ligi kui vajutada DuckeyTaxi logole.

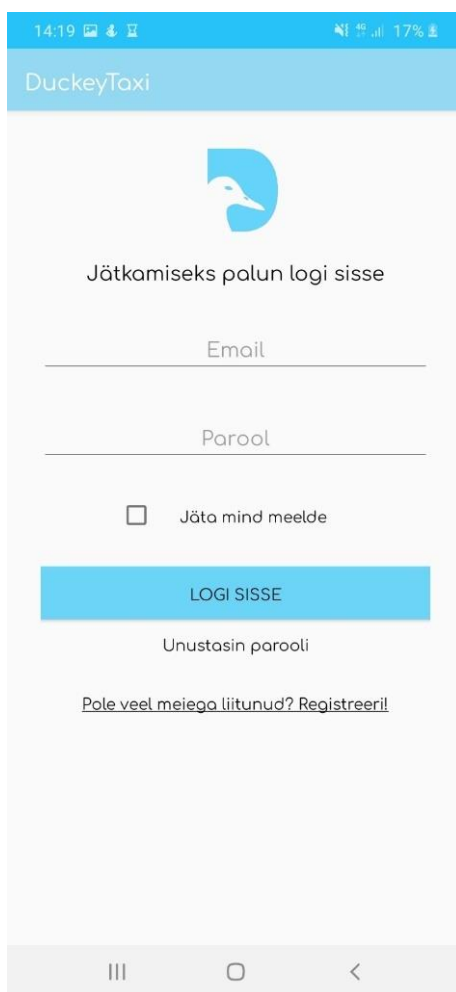
Rakenduse peamiste klasside diagrammi saab näha lisa 6.

5.1 Klient

Kliendiks on kasutaja, kes soovib tellida takso. Kliendil on rakenduses olemas sellised vaated nagu sisselogimine, registreerimine, navigatsiooniriba ning kliendi peavaade, kus on võimalik tellida endale takso.

5.1.1 Sisselogimine ja kliendiks registreerimine

Valides rakenduse avavaatel kliendi rolli, suunatakse kasutaja edasi sisselogimise lehele (Joonis 7), kus tal on võimalus sisse logida või kasutaja puudumisel end registreerida.



Joonis 7. Kliendi sisse logimise vaade

Valides “registreeri” avaneb kasutajale registreerimise vaade, kus on kohustuslik sisestada ees- ja perekonnanimi, email (mida edaspidi kasutatakse sisse logimisel), parool, telefoni number, sünnikuupäev (DuckyTaxi klient-kasutaja peab olema vähemalt 16 aastat vana) ja pangakaardi andmed. Soovi korral saab klient-kasutaja lisada oma profiilile pildi.

Peale registreerimist suunatakse klient-kasutaja koheselt edasi vaatele, kus on võimalik tellida endale taksot.

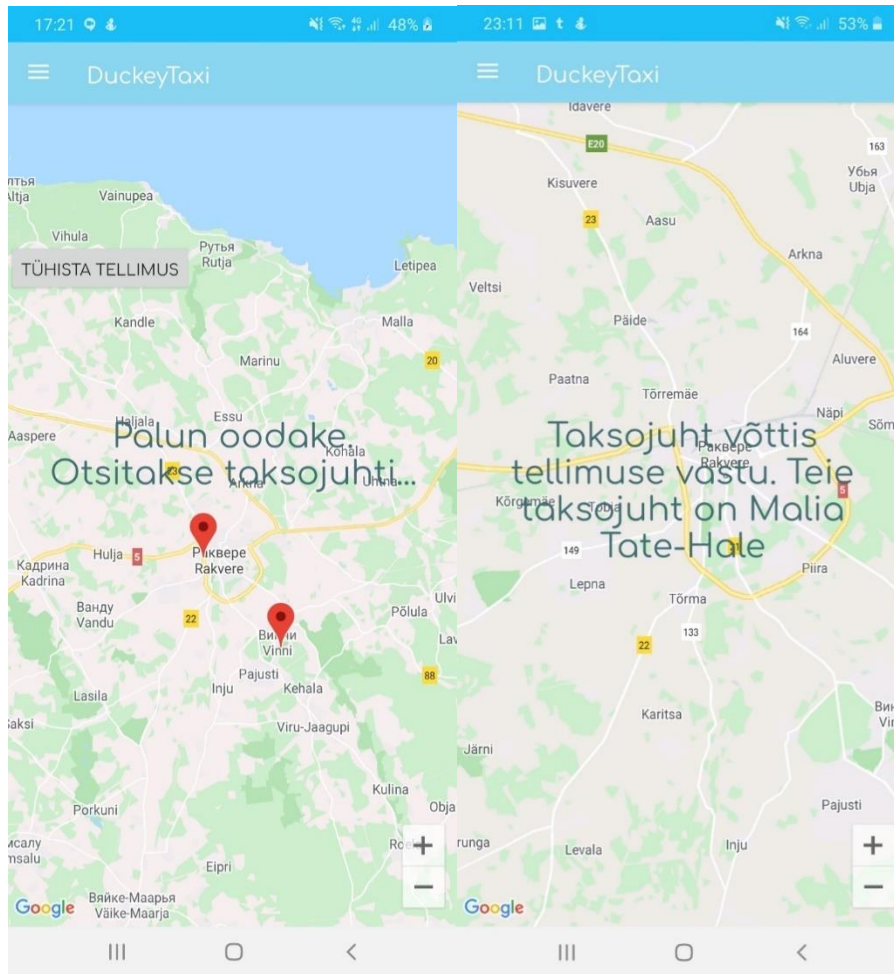
Registreerimine ja sisselogimine on realiseeritud läbi SQLite andmebaasi. Registreerimisel määratakse kliendile unikaalne ID kood, mille abil saab andmebaasist kätte ülejäänud informatsiooni konkreetse klient-kasutaja kohta. Kliendi objektile on parameetrina lisatud tulevase tellimuse ID kood, mis enne tellimuse tegemist omab väärtust *NULL*.

5.1.2 Kliendi peavaade

Peavaatel (Lisa 7) on kliendil võimalik tellida endale taksot. Takso tellimiseks on vaja vajutada "Telli takso" nuppu, peale mida on võimalus sisestada alg- ja lõpp-punkt. Algpunktiks saab valida enda hetkekohta ehk kus kasutaja sel momendil asub, selleks on vaja vajutada ikoonile, mis asub vasakul pool algkoha väljast.

Tellimuse esitamiseks tuleb vajutada "Edasi" nuppu, peale mida saab täiendada oma tellimust valides takso tüüp ja lisades lisainformatsiooni nagu näiteks, et klient soovib sõitu oma lapsega või lemmikloomaga. Tellimust saab esitada vajutades "Kinnita tellimus" nuppu.

Peale seda kui tellimus saab esitatud, avaneb kliendile sõnum, et otsitakse taksojuhti. Kui sobiv taksojuht, kes vastab kliendi tellimuse tingimustele võtab tellimuse vastu, siis kuvatakse kliendile sõnum "Taksojuht võttis tellimuse vastu. Teie taksojuht on (taksojuhi ees- ja perekonnanimi)" (Joonis 8).



Joonis 8. Taksojuhti otsimine (vasakul) ja taksojuht on võtnud tellimuse vastu (paremal)

Kaart on realiseeritud Google poolt pakutud Maps SDK abil, mis sisaldab funktsioone nagu asukoha määramine markerina. Aadressi saamine koordinaatidest ehk *reverse geocoding* toimub Android ressursides oleva Geocoder klassi kaudu.

5.2 Taksojuht

Taksojuht on kasutaja, kes soovib osutada taksoteenuseid ja saada oma töö eest vastavat raha. Taksojuhil on vastavad vaated nagu sisselogimine ja registreerimine, navigatsiooniriba, taksojuhi peavaade ja kliendi tellimuse vaade.

5.2.1 Sisse logimine ja taksojuhi taotluse saatmine

Valides taksojuhi rolli, suunatakse kasutaja edasi sisselogimis vaatesse (Joonis 9). Taksojuht-kasutaja saab kohe sisse logida kui tal on kasutaja olemas, kui pole kasutajat siis saab juht esitada taotluse.



Joonis 9. Taksojuhi sisse logimise vaade

Taksojuht ei saa koheselt registreerida ja sisse logida. Esmalt peab kasutaja esitama taotluse (Lisa 8). Taotluse enamus väljasid, kaasa arvatud pilt juhust, on kohustuslikud täitmiseks – selleks, et reisijad saaksid tuvastada juhti. Taksojuht saab valida, kas ta on nõus ja võimeline sõidutama kliente koos lastega (vajab laste turvaistet), sõidutama kliente nende lemmikloomadega (kassid, koerad ja teised väiksemad lemmikloomad), kas juhi auto mahutab rohkem kui neli klienti ning kas juht on valmis abistama kliente, kui neil abi on vaja, seda näiteks vanemaealiste klientide puhul.

Taotluse esitamisel antakse kasutajale teada, et tema taotlus vaadatakse läbi nii kiirelt kui võimalik. Administraator saab taotluse kätte (Lisa 9) ning kontrollib andmete sobivust ja kui taotluse esitaja vastab kõikidele parameetritele siis avaldus võetakse vastu (Lisa 10), määratakse autole takso tüüp ning antakse esialgne parool, millega kasutaja saab nüüd sisse logida kui taksojuht ja võtta vastu klientide tellimusi.

Taksojuhi registreerimine ning sisselogimine realiseeriti sarnaselt kliendi sisselogimise ja registreerimise SQLite andmebaasi kaudu. Erinevuseks on see, et taksojuht peab

saatma taotluse ja mobiilirakenduse administraatorid on võimelised taotlust tagasi lükkama või vastu võtta ning täiendada andmeid, nagu näiteks auto tüüp. Peale taotluse vastu võtmist on alles taksojuht rakenduses registreeritud.

5.2.2 Taksojuhi kaardi vaade

Taksoteenuste osutamiseks peab taksojuht vajutama nuppu “Tööle”, peale nupu vajutamist ilmnevad tellimused, mis vastavad konkreetsele taksojuhi-kasutajale (Joonis 10). Näiteks kui kasutaja registreerimisel valis, et ta on valmis sõidutama kliente koos nende lastega ja tema auto klassiks määrati *normal*, siis näeb ta nende klientide tellimusi, kes soovisid *normal* tüüpi taksot ja/või kes vajavad lapse turvaistet.



Joonis 10. Taksojuht on valmis klientide tellimusi vastu võtma

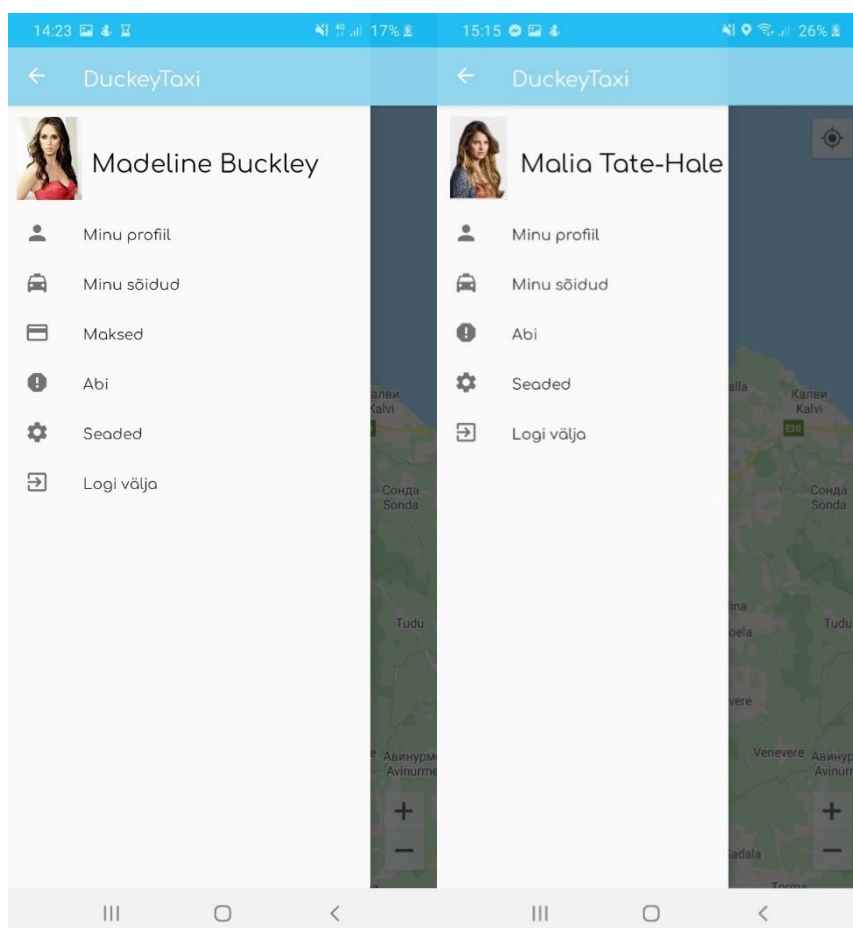
Kui kaardile ilmneb marker siis tuleb sellele vajutada, et näha kliendi nime ja lõppkohta (Lisa 11). Vajutades ilmunud kliendi info *messagebox*-le avaneb *pop-up* kus on näha kliendi ees-ja perekonnanimi, telefoninumber, asukoht ja lõppkoht ning lisa informatsioon, mida ta on valinud (soovib sõitu loomaga, lapsega, on vaja abi, on vaja

suuremat autot, kuhu mahuks rohkem kui neli klienti). *Pop-up* lõpus on võimalik võtta tellimus vastu vajutades “Võta vastu” nuppu, kui aga soovi või võimalust kliendi tellimust vastu võtta ei ole siis saab akna sulgeda vajutades ristile, mis asub paremal üleval.

Kui taksojuht võtab kliendi taksotellimuse vastu, saab klient teada, et tema tellimus on vastuvõetud ja kes teda sõidutab. Nii klient kui ka taksojuht saavad tühistada tellimuse vajutades vastavalt “Tühista tellimus” nupule.

5.3 Navigatsiooniriba

Taksojuhi ja kliendi kaardi vaadetes asub navigatsiooniriba (Joonis 11), mille avades saab kasutaja liikuda teistele vaadetele nagu “Minu profiil”, “Minu sõidud”, “Abi”, “Seaded”, “Logi välja” ning klient-kasutaja navigatsiooniriba puhul ka “Minu maksed”. Navigatsiooniribal samuti saab näha sisse logitud kasutaja profiilipilti ning nime.



Joonis 11. Kliendi (vasakul) ja taksojuhi (paremal) navigatsiooniribad

“Minu profiil” alt on võimalik näha kasutaja profiili ning soovi korral teha muudatusi. Klient-kasutaja vaade kuvab kasutaja pildi, nime, sünnipäeva, email, telefoninumbri ja

pangakaardi andmeid. Taksojuhi vaatel on olemas sama, mis kliendil ja lisaks ka juhiloa number, teenindajakaardi andmed, informatsioon autost (auto bränd, mudel ning auto number). Samuti taksojuhi profiilis kuvatakse ka lisa informatsioon juhi otsustes, kas juht on valmis sõidutama kliente lastega ja/või valmis sõidutada kliente nende lemmikloomadega ja/või abistada kliente ja/või auto on piisavat mahukas, et mahutada rohkem kui neli klienti.

Klient-kasutaja saab muuta kõiki andmeid välja arvatud sünnipäev ja email. Taksojuht-kasutaja saab muuta ise vaid enda parooli, telefoninumbrit ja pangakontot. Teiste andmete muutmiseks peab juht pöörduma administraatori poole, kuna ülejäänud andmed puudutavad tema taksojuhi staatust rakenduses peab administraator kontrollima uute andmete sobivust (näiteks kas uuel autol on vajalik sõiduluba olemas, kas juhiloa on kehtivad jne). Profiili muutmise all on võimalus ka kustutada oma kasutajakonto.

“Minu sõidud” alt avaneb vaade (Lisa 12), kus kuvatakse kõik kliendi varasemad sõidud. Iga sõit kuvatakse eraldi väikse kaardina, kus on marsruudi pilt, millal sõit toimus, kust kuhu sõideti ja palju sõit maksma läks.

“Abi” vaate (Lisa 13) all kuvatakse küsimusi, mis võiksid korduma kippuda. “Seaded” alt saab kasutaja valida, kas ta tahab, et talle helistatakse või mitte, näiteks kui klient-kasutaja paneb, et ta tahab siis juht võib talle helistada kui jõudis kohale. “Logi välja” nupule vajutades on võimalik kasutajal välja logida ning suunduda tagasi logi sisse vaatesse.

5.4 Testid

Enamasti rakendusest oli käsitsi testitud ehk mobiilirakendus lülitati sisse ning iteratiivselt vaadati üle olemasolevad vaated ja käivitati stsenaariume. Samuti oli kasutusel võetud kasutajaliidese testimisraamistik Espresso UI Testing ja Android Studio sisse ehitatud *unit* testid (Joonis 12).

```
public class AddressParserTest {
```

```
    AddressParser ad;
    String originalAddress;
    String expectedAddress;
    String newAddress;
```

```
    @Before
    public void setup(){
        ad = new AddressParser();
    }
}
```

```
    @Test
    public void shorten_NormalAddress(){
        originalAddress = "Tööstuse 82-31, 10487, Tallinn, Estonia";
        expectedAddress = "Tööstuse 82-31, 10487";
        newAddress = ad.removeTownAndCountryFromAddress(originalAddress);
        Assert.assertEquals(expectedAddress,newAddress);
    }
}
```

```
    @Test
    public void shorten_addressWithNoPostAddress(){
        originalAddress = "Tööstuse, Tallinn, Estonia";
        expectedAddress = originalAddress;
        newAddress = ad.removeTownAndCountryFromAddress(originalAddress);
        Assert.assertEquals(expectedAddress,newAddress);
    }
}
```

```
public class ClientMapViewActivityUITest {
```

```
    @Rule
    public ActivityTestRule<ClientMapViewActivity> clientMapViewActivityRule =
        new ActivityTestRule<>(ClientMapViewActivity.class);
```

```
    @Test
    public void aSetup(){
        //setup method, clicks on button "Tellima"
        onView(withId(R.id.clientMap)).check(matches(isDisplayed()));
        onView(withId(R.id.orderButton)).check(matches(isDisplayed()));
        onView(withId(R.id.orderButton)).perform(click());
    }
}
```

```
    @Test
    public void onMapLongClickTest(){
        aSetup();
        onView(withId(R.id.clientMap)).perform(LongClick());
        onView(withId(R.id.clientCurrentLocation)).check(matches(not(withText(""))));
        onView(withId(R.id.clientDestination)).check(matches(withText("")));
        //checks that text appears in Algkoht
        onView(withId(R.id.clientMap)).perform(LongClick());
        onView(withId(R.id.clientDestination)).check(matches(not(withText(""))));
        //checks that text appears in Lõppkoht
        onView(withId(R.id.clientMap)).perform(LongClick());
        onView(withId(R.id.clientDestination)).check(matches(withText("")));
        //checks that text appears in Algkoht, Lõppkoht is empty
    }
}
```

Joonis 12. Ühiktestid aadressi lühendamise meetodi jaoks (vasakul) ning pika klikki Espresso UI testid kliendi kaardil (paremal).

6 Analüüs ja järeldused

Käesolevas peatükis tehakse kokkuvõtte küsimustiku analüüsist ning räägitakse millest andmebaas koosneb, kirjeldatakse projekti raames tehtud peamisi otsuseid nagu valitud tehnilised lahendused ja nõuded rakendusele ning esitatakse tulevasi võimalikke täiendusi ning samuti tehakse teostatute tööde hinnangut meeskonnas.

6.1 Küsimustiku analüüsi kokkuvõte

Potentsiaalsete klientide seas läbiviidud küsimustik andis aimu, mis suunas oleks vaja rakendusega liikuda, mida kliendid tegelikult ootavad ja tahavad. Sihtgruppi kuulusid peamiselt väiksemast linnast ning selle ümbruses olevates alevikes ja küldes elavad inimesed.

Küsimustik aitas välja selgitada populaarsemad ajad takso kasutamiseks, milleks olid tööle ja kooli mineku ajad, õhtu poolikud, kui tööpäevad on lõppenud ja nädalavahetuse õhtud ja varahommikud, kui inimesed liiguvad peole ja peolt koju.

Enamus vastanutest eelistaksid mobiilirakendust helistamisele, sest see on kiirem, mugavam ning sõidu eest on võimalik tasuda läbi rakenduse.

Rakenduse loomiseks oli oluline leida, mida kliendid ootavad. Populaarsemateks vastusteks olid, et rakendus oleks odav ning lihtsasti kasutatav. Oluliseks peeti ka seda, et oleks erineva takso tüübi tellimise võimalus ning vajadusel saaks tellida ka taksot, kuhu mahub rohkem kui 4 klienti või, et saaks hoopis taksot kasutada koos lemmiklooma või lapsega, kes vajaks turvaistet.

Klientide arvates on kõige olulisem takso juures hind, sellele järgnes juhi hinnang ning automark, kõige vähem valiti auto puhtust.

6.2 Tehnilised otsused

Otsus teha just mobiilirakendus tuli sellest, et taksoteenuse platvormi on mugavam kasutada nii klientidel kui ka taksojuhtidel enda nutitelefonis või tahvelarvutis. Kuna takso kasutamiseks on vaja liikumist ning interneti ühendust siis on kasutajal mugavam

teha seda väiksemas seadmes mitte arvutis. Eraldi mobiilirakenduse olemasolu teeb kasutaja elu lihtsamaks, kuna kõik vajalik on ühes kohas kokku koondatud ja uusi rakendusi ei avane (näiteks kui veebirakenduses klikkad sisse ehitatud Google kaardile siis ikkagi avatakse kaart Google Maps mobiilirakenduses).

Otsus teha mobiilirakenduse prototüüpi Android operatsioonisüsteemiga nutitelefonidele tulenes potentsiaalsete DuckeyTaxi kasutajate arvust – Eestis eelistatakse rohkem Android operatsioonisüsteemil töötavaid nutitelefone kui iOS operatsioonisüsteemil töötavaid iPhone. Kasutajaid, kellel on nutitelefoni Android baasil, on kõikide nutitelefonide seast kokku 60% ning kasutajaid, kes omavad telefone iOS platvormil on 20% [24].

Lokaalset andmebaasi otsustati kasutada kuna pilvepõhise andmebaasiga, mis koosneks kõigist vajalikest osadest nagu *server*, URL ja viis, kuidas pöörduda (*client*), ei saadud ühendust. Internetis oleva andmebaasiga ühendamine vajab suurt uurimistööd ja lõpuni seda viia ei õnnestunud.

Võrreldes selliste andmebaasidega nagu näiteks SQL Anywhere, Oracle TimesTen ja Oracle-i Database Lite, ainult SQLite on avatud lähtekoodiga, koormamata litsentsitasudega ja mõeldud just manustatud andmebaasina kasutamiseks [25, p. 1].

6.3 Andmebaas

Andmed hoitakse lokaalselt SQLite andmebaasis. Hetkel on projektis kokku kaks andmebaasi – DuckeyTaxiUsers ja DuckeyTaxiOrders, kus esimene vastutab kasutaja andmete ning teine tellimuste hoiustamise eest. Andmebaas DuckeyTaxiUsers sisaldab kolme tabelit – ClientUsers, TESTDrivers ja TaxiDriverUsers.

ClientUsers tabel hoiab ClientUser objekte parameetritega ID, ees- ja perekonnanimi, email, parool, sünnikuupäev, telefoninumber, pangakaardi number ja profiilipilt ning hetke tellimuse ID, mis kasutaja loomisel saab automaatväärtuse *NULL*. Kliendi registreerimisel, kui klient ise ei lisanud endale sobivat profiili pilti siis lisatakse andmebaasi vaikimisi kasutaja pildi.

TESTDrivers on tabel, mis hoiab neid DriverUser objekte, mis ootavad kinnitust administraatorite poolt. See tabel sisaldab potentsiaalse juhi andmeid, mis on taotlusega esitatud. Objektid on selliste parameetritega nagu ID, ees- ja perekonnanimi, juhi isikukood, telefoninumber, sünnikuupäev, email, pangakonto number, profiilipilt, juhiloa

number, teenindajakaardi number, auto bränd ja number ning *boolean* väärtused, kas juht on nõus sõidutama kliente lastega, lemmikloomadega või kas nõus sõidutama suuremaid kompaniisi (rohkem kui neli inimest) ja vajadusel abistama klienti. Samuti omab objekt automaatväärtust *NULL* hetke tellimuse ID parameetris kui taksojuht alles saatis taotluse. Objektid liiguvad TESTDrivers tabelist TaxiDriverUsers tabelisse kui rakenduse administraator kiidab heaks juhi kandidatuuri ja kinnitab taksojuhi. Taksojuhi kinnitamisel määrab administraator auto tüübi (kas auto on *business*, *normal* või elektriauto) ja soovi korral määrab talle ise parooli (muidu pannakse juhile *random* parool). TaxiDriverUsers tabel erineb TESTDrivers tabelist kahe parameetri võrra – parool ja auto tüüp, kõik ülejäänud parameetrid on samad.

DuckyTaxiOrders andmebaas omab ühte tabelit, Orders, mis sisaldab objekte, mille parameetriteks on ID, kliendi ees- ja perekonnanimi, algpeatuse koordinaadid, lõpppeatuse koordinaadid, tellimuse staatus (*awaiting*, *in_process*, *done*), telefoninumber, taksojuhi ID, kliendi ID, taksojuhi auto tüüp ning *boolean* väärtused, kas klient märkis lisa informatsioonina, et sõidab lapse ja/või lemmikloomaga, soovib taksojuhi abi või et on vaja suuremat taksot, mis mahutaks rohkem kui neli klienti.

Kõik ID koodid on unikaalsed ning mitte ükski mõlema andmebaasi vältel ei kordu.

6.4 Nõuded

Takso tellimisel on vaja vähemalt kahte osapoolt – taksojuhti ja klienti, ilma nende kaheta ei toimu taksosõitu. Taksosõit toimub mugavamalt ja kiiremini kui on olemas ka kolmas pool, kes pakuks klientidele taksosi ja taksojuhtidele kliente.

Taksoteenust osutav mobiilirakendus peab omama kasutajate süsteemi ehk kasutaja sisse ja välja logimist ning registreerimist, et nii taksojuhil kui ka kliendil oleks võimalik oma andmeid edastada üksteisele läbi tellimuse esitamise.

Klient peaks saama esitada takso tellimuse määrates alg- ja lõpp-punkt ja vajadusel lisama lisa informatsiooni, näiteks et klient soovib sõita lapsega, ning samuti valima takso tüübi, mis jääks kliendile sobivasse hinna vahemikku.

Taksojuhil on taksoteenuse osutamiseks teada kliendi andmeid – kust kuhu on kliendil soov sõita. Samuti peab taksojuht olema teadlik kliendi lisainformatsioonist, sest kui näiteks klient soovib sõita lapsega siis peab juhil olema lapseturvatool.

Vajaduse korral peab saama nii klient kui ka taksojuht tühistada sõitu. Kui sõidu tühistab klient siis on tal kohustus maksta tühistustasu, kui sõidu tühistab taksojuht siis ta ei maksa tühistustasu ja kliendile otsitakse edasi uut sobivat taksojuhti.

Eriolukordade tekkimisel ja ka kasutaja teadmiseks peab olema kättesaadav sõitude ajalugu, näiteks kui klient unustas midagi taksosse siis on tal võimalik näha kindla sõidu kohta informatsiooni.

6.5 Hinnang projekti teostamise kohta

Projekti juhtimine oli korraldatud *Scrum*-i järgi. *Scrum* on projekti läbiviimise raamistik, mis aitab tiimi produktiivselt juhtida keeruliste toodete realiseerimises [26].

Üks kord nädalas viidi läbi koosolek, kus igaüks rääkis tehtust ja takistustest, millega on kokku puutunud. Juhendaja rollis olid kõik meeskonnaliikmed, ja ülesannete jagamine toimus arutamise kaudu. Omavahel jagati ülesanded järgmiseks nädalaks ning nädala jooksul hoiti sidet ja teatati kõikidest tehtud osadest või takistustest. Kui püstitatud ülesandega oli palju segadusi, siis see kas jäeti vahele, anti teisele meeskonna liikmele käsitlemiseks või tehti koos.

Peale eriolukorra jõustumist Eestis ei olnud võimalik kohtuda meeskonnaliikmetega füüsiliselt. Koosolekud hakkasid toimuma Google Hangouts keskkonnas. Kõned toimusid tihedamalt kui kord nädalas – meeskonna liikmed pidevalt arutasid üksteisega, kuidas midagi teha või läbi ekraani jagamise töötasid ülesannetega koos.

Katse luua ühendust interneti andmebaasiga võttis oodatust rohkem aega ja lõpuni ei jõudnud. Serveriga ei õnnestunud ühendust saada ja pilvepõhilise andmebaasiga ühendamisest tuli loobuda. Sellepärast oli otsustatud kasutada andmete hoidmiseks lokaalset andmebaasi SQLite, et enam-vähem saada valmis toote ettekujutuse.

Takistused tekkisid ka navigatsiooniriba rakendamisel. Läbi prooviti mitmeid erinevaid versioone, millest paljud ei sobinud. Peale navigatsiooniriba loomist ilmnes uus probleem vaatega, vaadete avamised toimisid kuid nupud uuel vaatel ei olnud funktsioneerivad. Probleem sai lõpuks lahenduse, kuid kindlasti kulus kogu navigatsiooniriba loomiseks oodatust rohkem aega.

Vähem aega võttis ühendamine Google Maps API-ga ning mõningate väiksemate lisadega nagu näiteks kui kasutaja vajutab telefonil olevat tagasi nuppu ei tohiks rakendus

minna sisselogimis vaatele vaid avavaatele ja ka tagasi noole lisamine vaadetele, kuhu jõuti läbi navigatsiooni riba ning oleks vaja tagasi liikuda peavaatele.

Seoses eriolukorraga suhtlus juhendajaga toimus meili teel, mis raskendas õigeaegset tagasiside saamist seega tuli tööd jätkata ise edasi proovides või jääda ootama ideid, kuidas edasi liikuda.

6.6 Teostatud tööde logi

Taksoteenust osutava mobiilirakenduse protüübi projekti hoidsime üleval Gitlab-is, kus igal meeskonnaliikmel oli ligipääs – igaüks omas staatust *maintainer* ehk omanik.

Projektis on kokku üks haru ja 288 *commit*-i (Joonis 13).



Joonis 13. Projekti päis GitLab-is

6.7 Täiendused, mida võiks tulevikus teha

Kui projekti finantseeritakse, avaneb võimalus rakendada tasulisi funktsioone nagu näiteks marsruudi loomine mitme koha vahel (Directions API) ja *Place Autocomplete* ehk aadresside automaatne pakkumine tähtede sisestamisel (Places API). Marsruudi funktsionaalsuse abil on edasi võimalik rakendada sõidu hinna arvutamist kilomeetrilisele kaugusele tuginedes ning leida ka sõiduks kuluv aeg.

Kuna Google Maps ei võimalda kasutada reaalaja navigatsiooni, mis on autojuhtide jaoks oluline, siis tulevikus tuleb kasutada teist kaardi funktsionaalsuse pakkumist *turn-by-turn* ehk samm-sammulise navigatsiooni rakendamiseks.

Tegelik takso mobiilirakendus peab olema iga kasutaja jaoks sünkroniseeritud ja sisaldama aktuaalseid andmeid kasutajate ja tellimuste kohta. Selleks on oluline omada *cloud-based* ehk pilvepõhelist andmebaasi, mis töötaks ja uueneks reaajas.

Tulevikus oleks võimalik ka tuua rakenduse iOS platvormile, mis võimaldaks ka iPhone ja iPad kasutajatel tellida taksot väljaspool suuri linnu.

6.8 Meeskondliku töö analüüs

Meeskond koosnes kolmest liikmest – Iris Nael, Darja Molodtsova ja Keidi Kaasik. Projekti loomiseks jagati ülesandeid iganädalaselt, kui toimusid meeskonna koosolekud. Koosolekutel arutati, kes milliste ülesannetega tegeles ja kaugele jõudis ning räägiti, mis probleemid ülesannete täitmisel tekkisid. Samuti jagati ära uue nädala ülesanded ehk, kes millega edasi tegelema hakkab. Kui ühel liikmetest tekkisid raskused või takistused toimusid lisa koosolekud, kus üritati ühiselt leida lahendus probleemile.

Eriolukorra kehtestamise tõttu tuli koosolekud üle viia *online* keskkonda. Esiialgu ei olnud need nii produktiivsed nagu füüsiliselt kokku saades, kuid olukorraga harjudes muutusid ka koosolekud paremaks.

Töö jagati nii, et lõputöö kõik olulised osad saaksid õigeaegselt tehtud, keegi panustas peamiselt koodile ja keegi tegeles ka lõputöö käigus tehtud küsimustiku ning rakenduste võrdlusega, et tekiks ülevaade, mida on veel vaja koodi lisada.

Iris Naela peamiseks ülesanneteks oli projekti sidumine andmebaasiga, kasutajate registreerimise, muutmise ning kustutamise võimaluste loomine, enamiku vaadete loomine, email ja parooliga sisse logimine ning sessiooni olemasolu ehk et kasutaja ei peaks uuesti sisse logima kui ta pole varasemalt välja loginud (Lisa 15).

Darja Molodtsova peamiseks ülesanneteks oli testide kirjutamine ning igasuguste kaardi vaadetega seotud funktsionaalsuste lisamine projekti, sealhulgas kaardi integreerimine, aadressi saamine kaardilt ja klassidevaheline andmete edastus rakenduse sünkroonimiseks (Lisa 16).

Keidi Kaasiku peamiseks ülesanneteks oli sarnaste olemasolevate rakenduste analüüsi koostamine ning potentsiaalsetele klientidele mõeldud küsimustiku koostamine ja analüüs, et leida, mida peaks lisama taksoteenust osutavasse mobiilirakendusse. Lisaks sellele navigatsiooni menüü loomine igale osapoolle, erinevate lisa lahenduste testimine nagu näiteks Facebooki ja Google sisselogimine ja kaardile marsruudi loomine, mis hetkel projekti prototüüpi ei jäänud (Lisa 17).

Iga meeskonnaliige oli ühtemoodi oluline ning liikmete panus projekti loomisel väga tähtis. Irise panus andmebaaside ning igasuguste vaadete loomise ja nende haldamise osas oli oluliselt suurem kui teistel meeskonnaliikmetel. Darja panustas oluliselt rohkem kaardi vaadete funktsionaalsuste loomisele ning testide kirjutamisele. Keidi tegeles oluliselt rohkem konkurentide ning küsimustiku analüüsimisega, navigatsiooni menüü loomisega ning hilisema rakenduse korrektse eesti keele kontrolliga.

7 Kokkuvõte

Enamus rakendused toimivad suurtes linnades, kus on kliente palju, kuid piirkonnad, kus on inimesi vähem või vahemaad suuremad ei ole juba olemasolevad sõidujagamisteenust osutavad takso mobiilirakendused juhisobralikud.

Probleemiks on võimetus taksot tellida või osutada takso teenust Eesti väikelinnades mobiilirakenduse kaudu, kuna enamik sealseid taksojuhte jäävad kahjumisse või selles kindlas linnas ei saa taksot tellida mobiilirakenduse kaudu.

Käesoleva töö eesmärgiks on valmis teha takso tellimiseks mõeldud mobiilirakenduse prototüüp, mis oleks kasutatav klientidele ja autojuhtidele Eesti väikelinnades. Prototüübiks on DuckeyTaxi, mis on Android platvormil töötav takso tellimise mobiilirakendus.

Eesmärgi saavutamiseks oli vajalik võrrelda olemasolevaid mobiilirakendusi ja läbi viia potentsiaalsete klientide arvamuse küsimustik. Mobiilirakenduste võrdlemise tulemusel tuletati prototüübi tariifid arvestades ka suuremaid vahemaid, et autojuht ei jääks kahjumisse. Potentsiaalsete klientide seas läbi viidud küsimustikku tulemusi ning ettepanekuid arvestades üritati luua mobiilirakendus, mis leiaks laialdast kasutust.

Prototüüp otsustati teha Android mobiilirakenduse programmis Android Studio kasutades Java programmeerimiskeelt tavakoodi kirjutamiseks ja XML keelt kasutaja vaadete ehitamiseks. Andmete hoiustamiseks kasutati lokaalset andmebaasi SQLite.

Otsus teha just mobiilirakendus tuli sellest, et taksoteenuse platvormi on mugavam kasutada nii klientidel kui ka taksojuhtidel enda nutitelefoni või tahvelarvutis. Lokaalset andmebaasi otsustati kasutada, et simuleerida andmebaasile pöördumist tegevuslikus mobiilirakenduses ning kuna pilvepõhise andmebaasiga ühendust ei saanud. Projekti juhtimine oli korraldatud *Scrum*-i järgi ehk üks kord nädalas viidi läbi koosolek, kus igaüks rääkis tehtust ja takistusest, millega on kokku puutunud ning toimus arutamise kaudu uute ülesannete jagamine. Tulevikus peaks mobiilirakendus omama *cloud-based* ehk pilvepõhilist andmebaasi ning samuti oleks võimalik tuua rakenduse iOS platvormile.

Kasutatud kirjandus

- [1] ERR, „Ööl vastu 1. maid hakkab Tallinnas sõitma Yandex Taxi,“ 30. 04. 2018. [Võrgumaterjal]. Available: <https://www.err.ee/827569/ool-vastu-1-maid-hakkab-tallinnas-soitma-yandex-taxi>. [Kasutatud 24. 04. 2020].
- [2] Yandex Taxi mobiilirakendus, 2020.
- [3] Yandex Taxi, „Sõidutasu: Premium teenseklass, Tallinn,“ [Võrgumaterjal]. Available: <https://taxi.yandex.ee/tallinn/tariff/ultimate>. [Kasutatud 01. 05. 2020].
- [4] Yandex Taxi, „Sõidutasu: Säästuklass teenuseklass, Tartu,“ [Võrgumaterjal]. Available: <https://taxi.yandex.ee/tartu/tariff>. [Kasutatud 03. 05. 2020].
- [5] Yandex Taxi, „Sõidutasu: Säästuklass teenuseklass, Narva,“ [Võrgumaterjal]. Available: <https://taxi.yandex.ee/narva/tariff>. [Kasutatud 02. 05. 2020].
- [6] M. Mägi, „Uber laieneb Tallinnasse,“ 09. 02. 2015. [Võrgumaterjal]. Available: <https://www.ituudised.ee/article/2015/2/9/uber-laieneb-tallinnasse>. [Kasutatud 24. 04. 2020].
- [7] Uber mobiilirakendus, 2020.
- [8] P. Brottier, „Oluline teave reisi maksumuse arvutamise kohta Tallinnas,“ 14. 06. 2018. [Võrgumaterjal]. Available: <https://www.uber.com/et-EE/blog/oluline-teave-reisi-maksumuse-arvutamise-kohta-tallinnas/>. [Kasutatud 29. 04. 2020].
- [9] Ira, „Kuidas Sulle meeldiks veel mugavam sõidukogemus? UberSELECT on kohal!,“ 30. 08. 2018. [Võrgumaterjal]. Available: <https://www.uber.com/et-EE/blog/uber-select-3>. [Kasutatud 29. 04. 2020].
- [10] Bolt Press. [Võrgumaterjal]. Available: <https://bolt.eu/et/press/>. [Kasutatud 24. 04. 2020].
- [11] Bolt mobiilirakendus, 2020.
- [12] Bolt, „Bolt Linnad,“ [Võrgumaterjal]. Available: <https://bolt.eu/et/cities>. [Kasutatud 02. 05. 2020].

- [13] Android Developers, „Documentation,“ [Võrgumaterjal]. Available: <https://developer.android.com/docs>. [Kasutatud 22. 04. 2020].
- [14] J. Gosling, B. Joy ja G. Steele, The Java Language Specification, 1996.
- [15] M. Lassofo, Java Programming for Beginners, 2017.
- [16] Z. Brahmia, H. Hamrouni ja R. Bouaziz, „XML data manipulation in conventional and temporal XML databases: A survey. Computer Science Review.,“ 2020.
- [17] GeeksforGeeks. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/xml-basics/>. [Kasutatud 05. 05. 2020].
- [18] Google Developers, „Google Maps Platform Documentation,“ [Võrgumaterjal]. Available: <https://developers.google.com/maps/documentation>. [Kasutatud 02. 05. 2020].
- [19] D. Zelenchuk, Android Espresso Revealed: Writing Automated UI Tests, 2019.
- [20] SQLite, „About SQLite,“ [Võrgumaterjal]. Available: <https://www.sqlite.org/about.html>. [Kasutatud 07. 05. 2020].
- [21] SQLite kodulehekül. [Võrgumaterjal]. Available: <https://sqlite.org>. [Kasutatud 05. 05. 2020].
- [22] DB Browser for SQLite kodulehekül. [Võrgumaterjal]. Available: <https://sqlitebrowser.org>. [Kasutatud 16. 04. 2020].
- [23] Atlassian, „Trello,“ [Võrgumaterjal]. Available: <https://www.atlassian.com/software/trello>. [Kasutatud 05. 05. 2020].
- [24] Elisa, „Kas Android või iOS – milliseid telefone kasutavad eestlased kõige rohkem?,“ 26. 02. 2020. [Võrgumaterjal]. Available: <https://digi.geenius.ee/blogi/elisa-blogi/kas-android-voi-ios-milliseid-telefone-kasutavad-eestlased-koige-rohkem/>. [Kasutatud 05. 05. 2020].
- [25] G. Allen ja M. Owens, The Definitive Guide to SQLite, 2010.
- [26] Scrum, „WHAT IS SCRUM?,“ [Võrgumaterjal]. Available: <https://www.scrum.org/resources/what-is-scrum>. [Kasutatud 06. 05. 2020].

Lisa 1 – Taksode võrdlus tabel

Funktsionaalsus	Uber	Yandex.Taxi	Bolt	DuckeyTaxi
Facebook-i sisselogimine	+		+	
Google-i sisselogimine	+			
E-posti sisselogimine/registreerimine				+
Telefoni numbriga sisselogimine/registreerimine	+	+	+	
Takso ette tellimine	+			
Mitme peatuse lisamine	+	+	+	
Lapsega reisija		+	+	+
Loomaga reisija				+
Pere/töö konto	+	+	+	
Nähtav sõitude ajalugu	+	+	+	
Fikseeritud sõidu hind	+	+	+	
Salvestatud aadressid	+	+	+	
Kellelegi teisele takso tellimine		+		
Valik erinevat tüüpi autode vahel		+	+	+
XL auto		+	+	+
Assist tüüpi auto(juhi abi)			+	+
Elektriauto			+	+

Lisa 2 - Küsimustik

Jaotis 1/5

Tere! Oleme Tallinna Tehnikaülikooli Äriinfotehnoloogia tudengid Keidi, Iris ja Darja. Teeme oma lõpuprojektiks taksosõidu mobiilirakenduse prototüüpi, ideede saamiseks, vajaduste välja selgitamiseks ja edasi tegutsemiseks oleks meil tarvis Teie abi. Küsimustik ei võta aega kauem kui paar minutit, vastused on anonüümsed ja kasutatakse vaid lõputöö kirjutamiseks.

Teie vanus

- Noorem kui 16
- 16-20
- 21-25
- 26-30
- 31-40
- 41+

Teie sugu

- Naine
- Mees

Kus on Teie peamine elukoht?

- Suur linn(>50 000 inimest)
- Väike linn (<50 000 inimest)
- Alevik
- Küla

Kust olete pärit?

- Suur linn(>50 000 inimest)
- Väike linn (<50 000 inimest)
- Alevik
- Küla

Kas kasutate taksot?

- Jah
- Ei

Jaotis 2/5 Takso kasutajad

Mis eesmärgil kasutate taksot?

- Tööle sõit
- Kooli/Ülikooli sõit
- Peole/Peolt koju sõit
- Poodi sõit
- Muu

Kui tihti kasutate takso teenuseid?

- Vähemalt kord nädalas
- Vähemalt kord kuus
- Vähemalt kord aastas
- Kuidas tellite endale taksot?
- Telefoni teel helistades
- Mobiilirakenduse kaudu

Jaotis 3/5 Telefoni teel takso tellija

Kas oleksite huvitatud takso äppi kasutamisest?

- Jah
- Ei

Põhjenda oma valikut lühidalt

Jaotis 4/5 Mobiilirakenduse kasutaja

Mis takso õppi olete kasutanud?

- Bolt
- Yandex.Taxi
- Uber

Jaotis 5/5 Hea taksosõidu mobiilirakendus

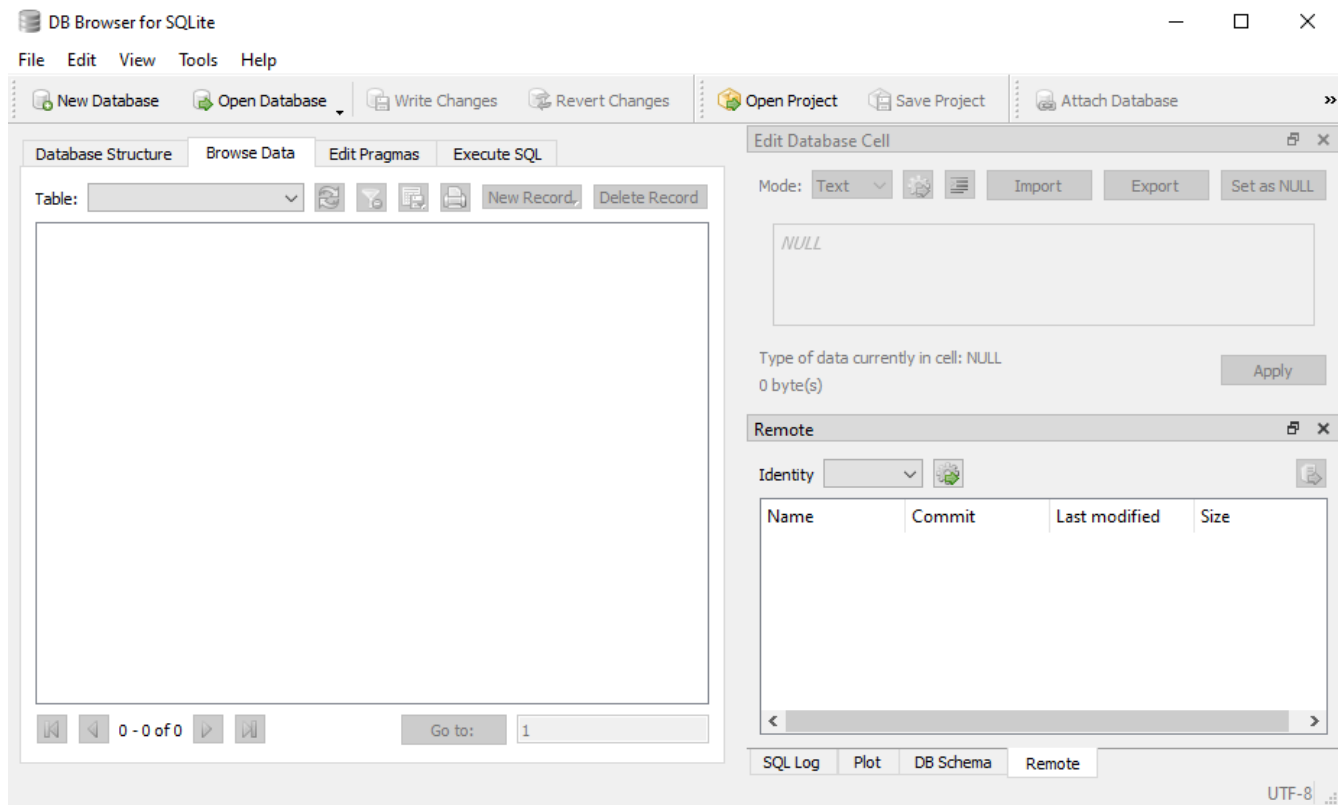
Milline võiks olla hea taksosõidu mobiiliakendus?

- Odav
- Lihtsasti kasutatav
- Perekontoga
- Maksemeetodi lai valik
- Raha äppi laadimis võimalus (äppi sisene rahakott)
- Võimalus märkida, et soovite sõitu lemmiklooma või väikse lapsega
- Takso tüübi valik (eco, business jne)
- Suurema takso valik (4+ istekohta)
- Muu

Mis on Teie jaoks takso juures kõige olulisem?

- Sõidu hind
- Automark
- Juht ja tema hinnang

Lisa 3 - DB Browser For SQLite

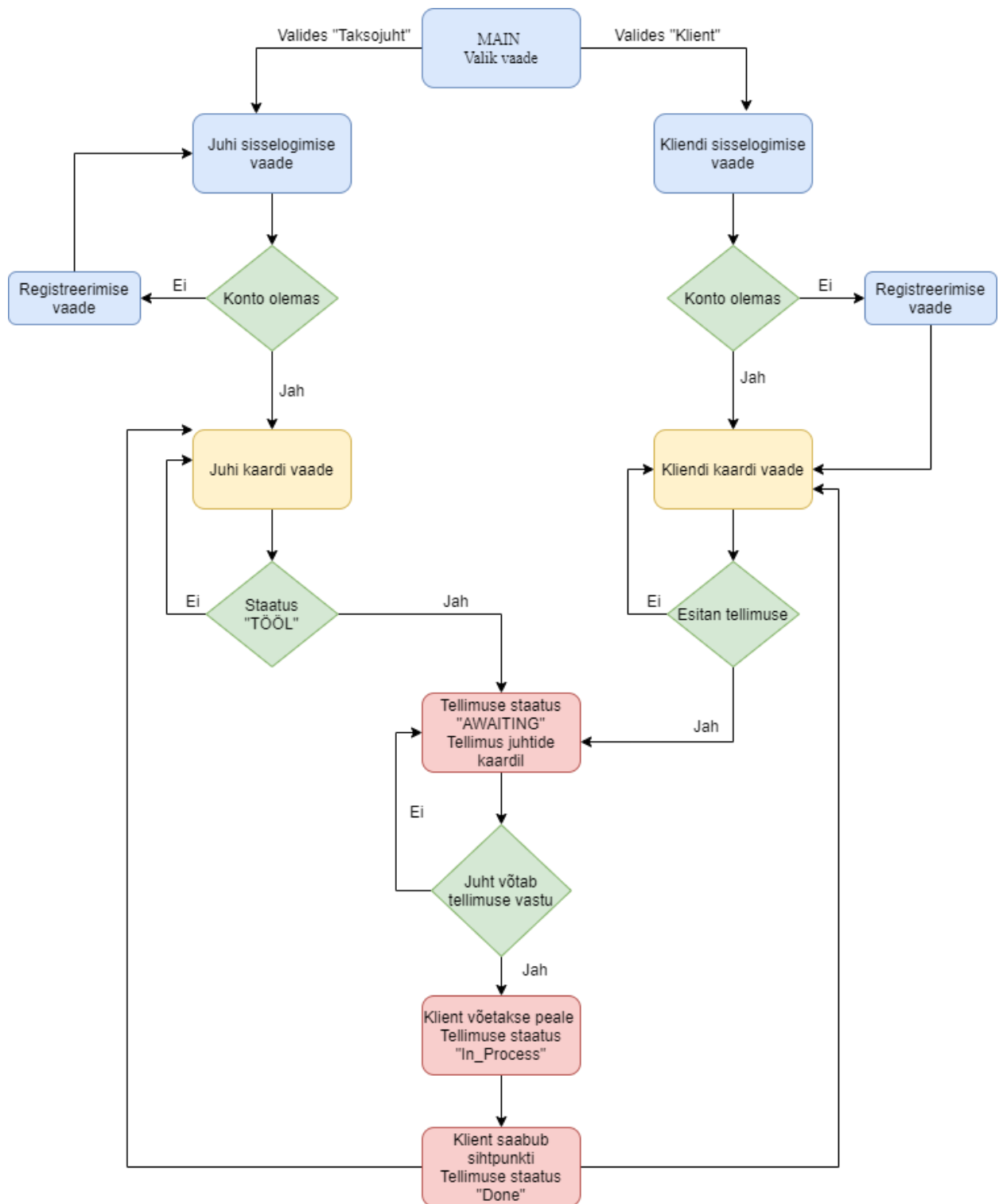


Lisa 4 - Trello

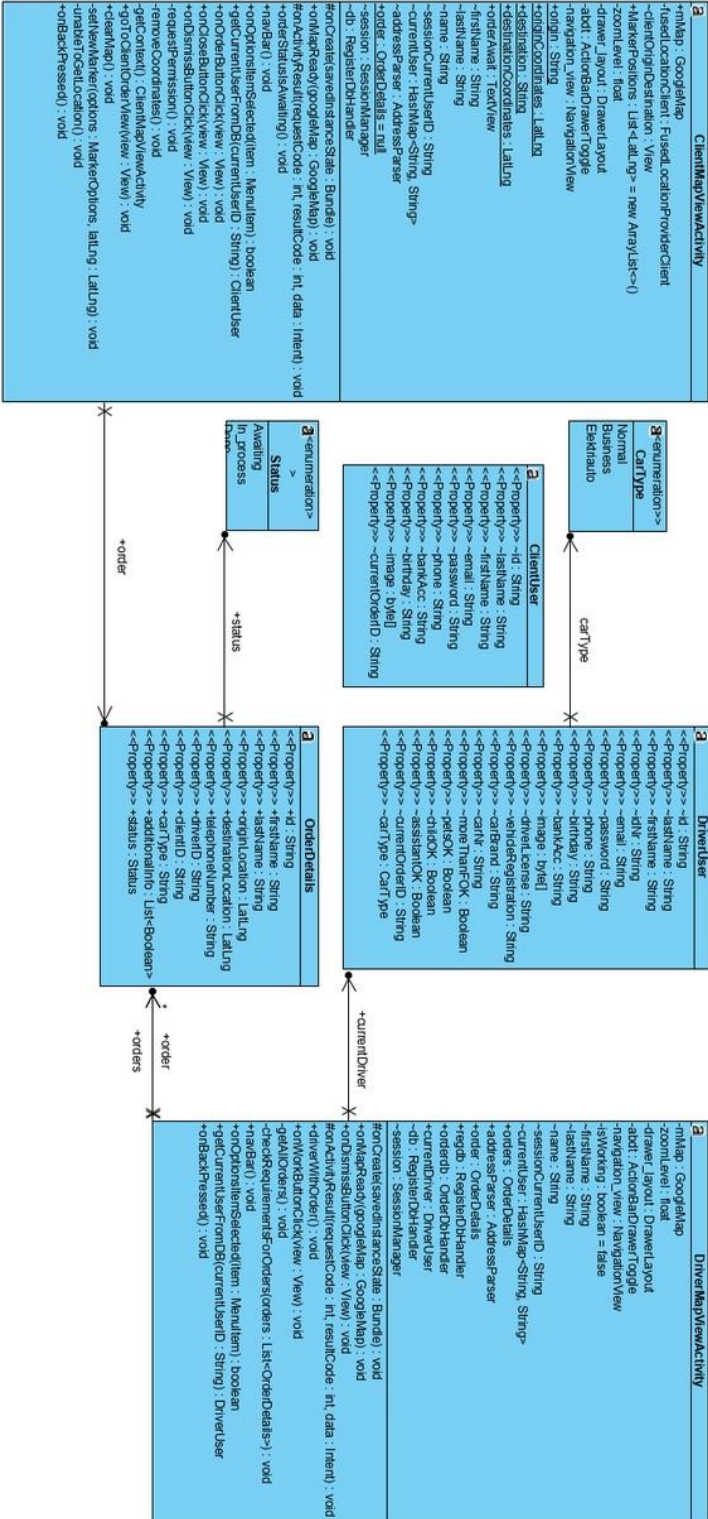
The screenshot shows a Trello board for 'DuckyTaxi' with the following columns and cards:

- What to do:**
 - Teha korda järgnevad asjad (mitte kood) (0/3)
 - Regamisel peab kontrollima: (0/3)
 - Taksojuhi regamisel peab samuti autol olema ka sõidukaart! (0/3)
 - ÜLE VAADATA DISAIN!!!! JA EESTI KEEL!!! (2/4)
- In process:**
 - Pildid! (7/19)
 - Tellimuse hind! (3/4)
 - Session (8/12)
 - Unustasin parooli: (3/6)
 - Luuva vaated (sõidud, abi, seaded) (7/29)
 - Navbar - testing
- Done:**
 - Küsimustiku analüüs
 - Juht saab muuta parooli
 - Kliendi registreerimisel tuleb muuta välja hint.
 - Eemaldada sessiooni staatuse (true, false)
 - Sessiooni salvestamine - kui oleme sisse logitud siis ilma registreerimise/sisselogimise vaadeteta minnakse otse kaardi vaatele
- Dokumentatsioon:**
 - Tegevusplaan (2/4)
 - Kirjutada olemasolevatest taksoarandustest
 - 1 - sissejuhatus (3/4)
 - 2 - rakenduste võrdlus (4/4)
 - 3 - mida kasutasime+küsimustiku analüüs (0/2)
 - 4 - mida tegime (0/9)
 - 5 - mis teha tahame/täiendada
- Set aside for now:**
 - Connecting to MySQL Online Database (2/7)
 - Autocomplete when typing in an address (TASULINE)
 - Vaadata kas on võimalik luua marsruute uuemas AS versioonis (0/3)
 - implement Directions API - route creating (1/2)
 - Admin view (3/6)
 - Admin registreerimine, sessioon,

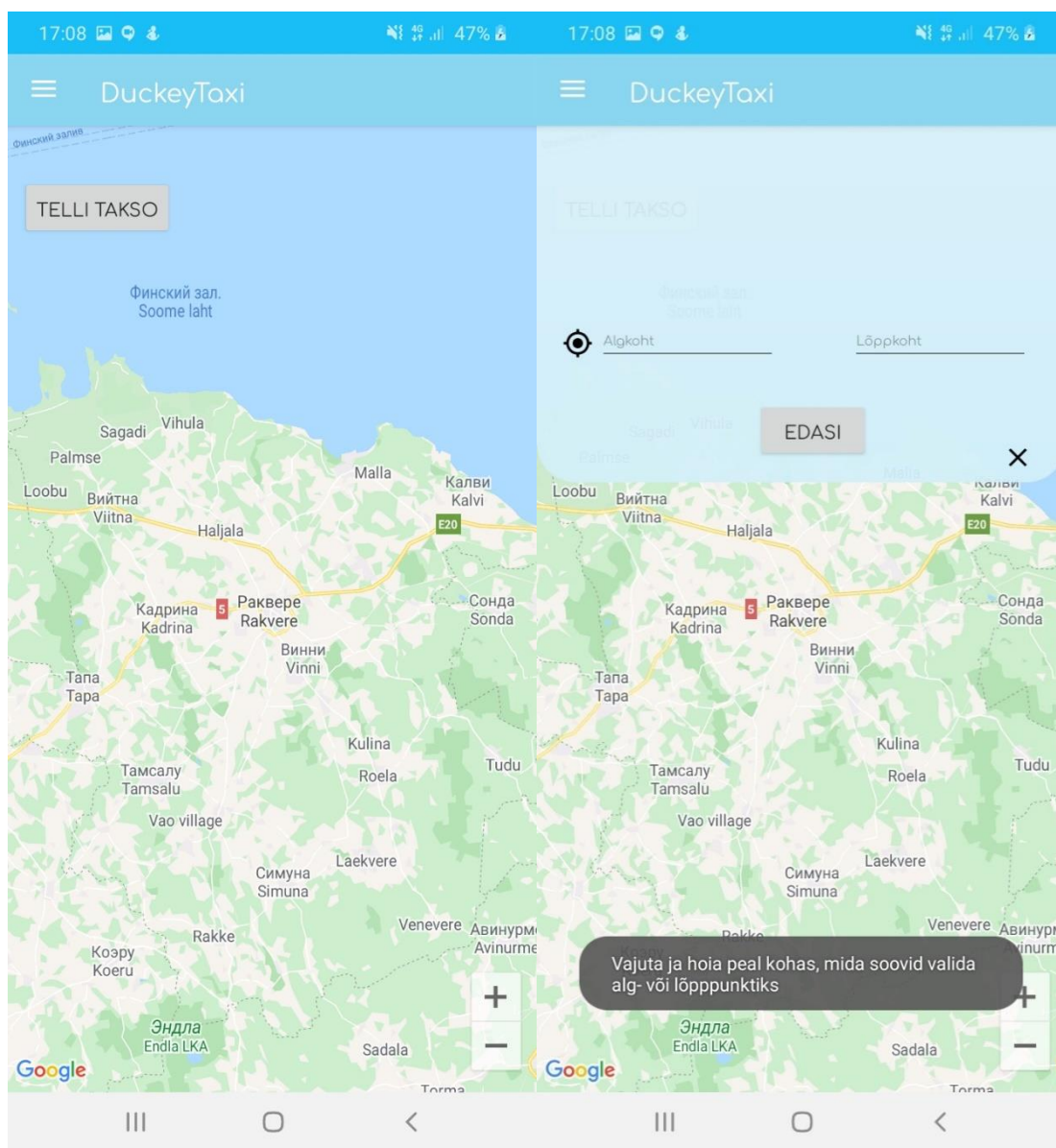
Lisa 5 – Taksojuhi ja kliendi tegevusskeem



Lisa 6 - Klassidiagramm



Lisa 7 – Kliendi vaade “Telli takso”



Lisa 8 – Taksojuhiks taotlemine

The image displays two side-by-side screenshots of the DuckyTaxi mobile application interface, showing the 'Taksojuhi taotluse ankeet' (Taxi Driver Application Form). The app's header is blue with the 'DuckyTaxi' logo. The status bar at the top shows the time as 14:27 and battery level at 18%.

Left Screenshot: Üldine informatsioon

Taksojuhi taotluse ankeet

Üldine informatsioon

Eesnimi

Perekonnanimi

Isikukood

Email

Telefoni number

Sünnipäev

Informatsioon juhiloa, auto ning teenindaja kaardi kohta

Juhiloa number

Teenindajakaardi number

Auto mark (nt Toyota Yaris)

Auto number

Right Screenshot: Pangakaardi informatsioon

Taksojuhi taotluse ankeet

Informatsioon juhiloa, auto ning teenindaja kaardi kohta

Juhiloa number

Teenindajakaardi number

Auto mark (nt Toyota Yaris)

Auto number

Pangakaardi informatsioon

Pangakonto

Pange linnukesed nende kirjete juurde, millega olete nõus

Sõidutama lapsi

Sõidutama kliente lemmikloomadega

Sõidutama ning abistama vanemaid inimesi või invaliide

Võimeline sõidutama rohkem kui neli inimest

Lisad

Lisa 8 - jätkub

14:27 18%

DuckeyTaxi

Taksojuhi taotluse ankeet


Pangakonto

Pange linnukesed nende kirjete juurde, millega olete nõus

- Sõidutama lapsi
- Sõidutama kliente lemmikloomadega
- Sõidutama ning obistama vanemaid inimesi või invaliide
- Võimeline sõidutama rohkem kui neli inimest

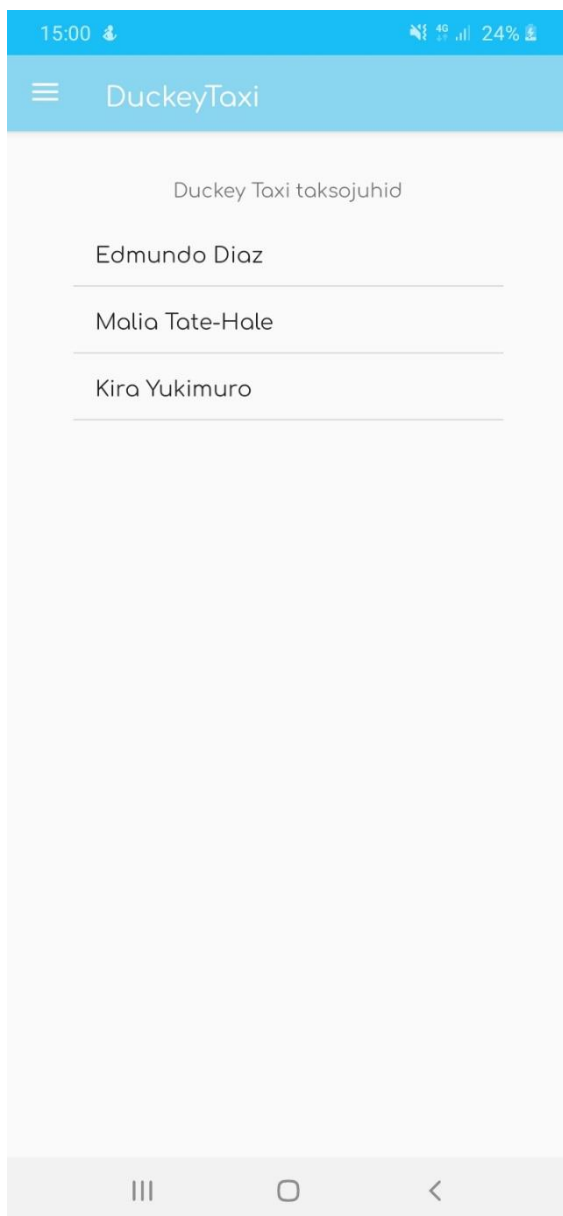
Lisad

LISA PILT



SAADA TAOTLUS

Lisa 9 – Administraator vaade - kõik taotlused



Lisa 10 – Administraator vaade - üksiku taksojuhi taotlus

15:01 4G 24%

← DuckeyTaxi

 ID: 69d754b6-48f9-4c19-af8e-570328b459ab
Nimi: Edmundo Diaz

Isikukood: 38709211121
Email: ed.diaz@gmail.com
Telefoni number: 56210096
Sünnipäev: 21/9/1987
Pangakaart: EE12123456567800001
Juhiluba: ET123456
Teenindajakaardi number: AV12345
Auto mark: Volvo S60
Auto number: ABC123

Valmis sõidutada lastega
Valmis anda abi sõitjatele

Normal Business Elektriauto

Sisestage parooli uue taksojuhi jaoks

KINNITA TAKSOJUHT

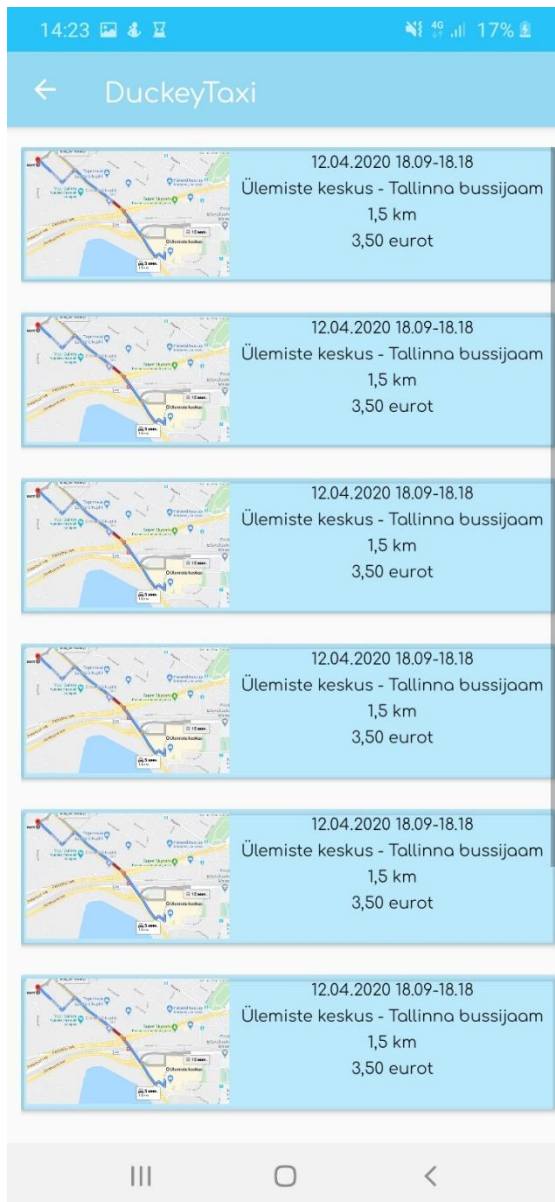
KUSTUTA

III ○ <

Lisa 11 – Taksojuhi kaardi vaade – marker kaardil koos kliendi andmetega



Lisa 12 - “Minu sõidud” vaade



Lisa 13 - “Abi” vaade



Lisa 14 – Scrum

<https://docs.google.com/spreadsheets/d/1MvZCyfUY1OaESbKNysRfzepxbwfbNYsLPaNvmkDYMh4/edit?usp=sharing>

Lisa 15 – Iris isiklik panus

Minu tegevused projektis

Lõputöö käigus võtsin osa kõigist kolmest kasutaja osapooltest (klient, taksojuht, administraator). Eelkõige lõin suurema osa vaadetest DuckyTaxi rakenduse jaoks ning ühendasin projekti lokaalse andmebaasiga.

Eialgu proovisin ühendada projekti pilvepõhise MySQL andmebaasiga, kuid ei saanud ühendust. Peale mitut katset otsustasime, et on vaja minna üle lokaalse andmebaasi loomisele.

Registreerimisel tuli saada kõiki andmeid registreerimise väljadelt nagu näiteks ees- ja perekonnanimi, email, parool, sünnikuupäev, telefoninumber, panga informatsioon (pangakonto juhil ja kaardinumber kliendil), pilt ja taksojuhil lisaks sellele veel teisi spetsiifilisi andmeid (juhiluba, teenindajakaart jt). Pildi korral tuli tüüpi *imageView*-lt konverteerida *byte array*-sse, et oleks võimalik seda hoida andmebaasis ja pildi kuvamisel mõnel vaatel tuli konverteerida seda tagasi.

Peale registreerimise sain tehtud kasutaja andmete muutmise ning muudetud andmete salvestamise andmebaasi. Samuti sain tehtud kasutaja kustutamise.

Sisse logimisel kontrollitakse, kas sisestatud email ja parool vastavad konkreetse kasutaja objekti parameetritega andmebaasis. Juhul kui vastavad siis saab kasutaja sisse logida, kui ei siis kuvatakse kasutajale sõnum, et sisestatud parool või email olid valed.

Potentsiaalsete kasutajate mugavuseks tegin sessiooni võimaluse ehk kui kasutaja (klient või taksojuht) on sisse logitud ja näiteks sulgeb rakenduse välja logimata siis rakenduse taasavamisel ei pea ta uuesti sisse logima ja saab kohe peavaatesse kaardiga. Juhul kui rakendust kasutab inimene, kes on samal ajal nii klient kui ka taksojuht, siis saab ta olla sisse logitud ainult ühes kasutajas. Näiteks kui kasutaja on loginud sisse kui taksojuht ja pole sellest välja loginud, kuid soovib tellida endale takso ja läheb kliendi sisse logimise vaatesse siis tuleb ette *messagebox*, mis annab teada, et enne peab taksojuhi kasutajast välja logida ning annab talle võimaluse teha seda kohe seal samas *messagebox*-is vajutades “Logi välja taksojuhi kasutajast”.

Taksoteenuste osutamisel üheks tähtsaks asjaks on tariifid. Vaatasin läbi ning analüüsisin konkurentide tariife erinevates linnades. Samuti uurisin, millised taksotüübid on teistel taksoteenust pakkuvatel mobiilirakendustel. Löin ligikaudse sobiva tariifi DuckeyTaxi taksosõitude jaoks vastavalt takso tüübile.

Eneseanalüüs

Projekti loomise käigus esines mitmeid väljakutseid ning suurim neist oli taksoteenuste rakenduse prototüübi loomine, kuna varasemalt pole mobiilirakendusi loonud ja Android Studio-ga tööd teinud.

Taksorakenduse loomisel tuli mõelda korraga nii potentsiaalsete klientide kui ka taksojuhtide peale ja mõningates küsimustes leidma kuldse keskmise, mis sobiks nii kliendile kui ka taksojuhile.

Seoses Eesti Vabariigis välja kuulutatud eriolukorraga suleti ülikoolid ning kehtestati piirangud mistõttu ei olnud võimalik enam teha meeskonnatööd nii nagu enne harjunud olin, kogu õppimine ja töö projektiga toimus teisiti ning varasemad meeskonna kokku tulekud tuli läbi viia interneti ühendusel, mis alguses ei olnud nii resultatiivne.

Vaadete loomisel tuli arvesse võtta potentsiaalsete kasutajate nutitelefonide erinevusi ehk kasutajatel on erinevad Android platvormil töötavad telefonid, mis on erineva suuruse ja resolutsiooniga. XML *design*-is nägi vaade välja üht moodi, emulaatoris teist ning enda telefonis kolmandat – teha oli vaja selline, mis oleks korrektne kõikides Android nutitelefonides.

Töötamine SQLite andmebaasiga oli meeldiv võrreldes MySQL-ga, millega proovisin projekti ühendada mitu korda ning erinevate võimalustega, kuid ühendust ei suutnud ikka saavutada. Tulevikus peaks pilvepõhisele andmebaasile rohkem tähelepanu pöörama.

Oodatavust kergem oli sessiooni integreerimine projekti, et kasutajal ei tuleks korduvalt sisse logida.

Mobiilirakenduse loomine Android baasil töötavatele nutitelefonidele oli väga arendav.

Taksoteenuse rakenduse prototüübi loomine oli vägagi meeldiv kogemus ning selle tegemisel oli vaja teha paljusid ülesandeid, mõned neist olid olulised ning mõningad väheolulisemad. Projekti kirjutamisel tekkis ka palju mõtteid, mida võiks veel integreerida projekti, näiteks võimalus logida sisse VKontakte (vene sotsiaalvõrgustik)

abil. Üheks vähem olulisemaks ülesandeks projektis, mis sai tehtud, oli näiteks “Abi” ja “Seaded” vaadete loomine.

Projekti kirjutamist kergendas varasem C# ja Java programmeerimiskeelte õppimine. Android Studio kasutamise õppimist tegi kergemaks varem kasutatud programm IntelliJ IDEA, kuna Android Studio on loodud selle baasil.

DuckeyTaxi arendamisel samuti tuli kasuks *Scrum*-i kasutamine. Koosolekutel iga meeskonnaliige rääkis, mida ta on teinud, milliseid takistusi tal oli ning mida plaanib edasi teha.

Meeskonnatöö alates eriolukorra välja kuulutamisest toimus täienisti veebikeskkonnas ning selleks ei olnud mina ega meie meeskond ette valmistunud - vähemalt mitte sellises ulatuses, et suurem osa koostööst toimub internetis.

Lisa 16 - Darja isiklik panus

Minu tegevused projektis

Projekti käigus tegelesin kliendi ja autojuhi peavaadet puudutava funktsionaalsuse ja disainiga. Sealhulgas oli näiteks Google Maps API integreerimine projekti, kaardi funktsioonide mõistmine ning sobiva kasutajaliidese rakendamine. Samuti tegelesin administraatori peavaatega (potentsiaalsete autojuhtide list ja nende detailid) ning testide kirjutamisega kasutajaliidese ja funktsionaalsuse jaoks.

Sai tehtud kliendi vaatel oleva asukohta valimise vaheaken, kus algpunkti on võimalik määrata kasutades oma seadme tegelikku asukohta või kaardi peal pikka klikki tehes. *Geocoder* klass võimaldab koordinaate tõlgendada teatud aadressina. Minu poolt oli ka tehtud kliendi tellimuse *pop-up* aken, mis võimaldab määrata kliendi erisoove takso valimisel. Oli lisatud funktsionaalsus tellimuse kinnitamisel saata andmed otse andmebaasi ning muuta kliendil oleva parameetri "CurrentOrderID" selle tellimuse ID'ks.

Autojuhi vaatel on realiseeritud olemasolevate tellimuste kättesaamine ja kaardi peal markerite kujul kliendi asukohad. Teatud markeri peale klikkides avaneb *infowindow* millele tehes veel ühe klikki avaneb *pop-up* aken kliendi tellimuse andmetega. Tellimuse vastu võttes autojuhi objekt omandab tellimuse ID ning enam ei ole näha teisi tellimuste markereid peale võetud tellimuse oma.

Oli lisatud meetod tellimuse staatuse kontrollimiseks igal autojuhi või kliendi peavaate käivitamisel. Kui tellimuse ID pole kasutajal tühi, siis sõltuvalt sellest muutub tema peavaade. Kliendi peavaatel puudub "Tellima" nupp uueks takso tellimiseks kui tellimus on juba kinnitud ning uue tellimuse tegemiseks on vaja tühistada olemasolev või viia see lõpuni. Autojuhi peavaade muutub konstantseks staatuseks "on tellimuse vastu võtnud", kus autojuht ei saa võtta rohkem tellimusi enne kui ta keeldub olemasolevast või viib selle lõpuni.

Administraatori peavaatel oli realiseeritud kinnitust oodatavate autojuhtide list, kus teatud autojuhi nime peale klikkides avaneb uus vaade autojuhi andmetega. Seal on võimalik

autojuhile määrata parooli ja auto tüüpi ning salvestada aktiivse autojuhina tegelevate autojuhtide andmebaasi või kustutada tema taotlus.

Tegelesin ka autojuhi, kliendi ja tellimuse objektide käsitlemisega SQLite andmebaasis.

Eneseanalüüs

Suurimaks väljakutseks võib vaadelda mobiilirakenduse prototüübi loomist, kuna eelnev mobiilirakenduse loomise kogemus puudub. Võib samuti öelda, et väljakutseks olid raskused kohtumiste läbiviimisel meeskonnaliikmete vahel ning juhendajalt õigeaegne tagasiside saamine eriolukorra tingimustel.

Kuna enne pole puutunud kokku kaardi integreerimisega projekti, siis oli näiteks Maps API rakendamine projekti päris uus asi. Vaatlesin ka Directions API ja Places API marsruudi loomise uurimiseks ja *Autocomplete* ehk aadressi automaatse lõpetamise funktsiooni rakendamiseks vastavalt, kuid seal esinesid probleemid selles, et API-d vajasisid funktsioonide kasutamiseks tasulise konto.

Keeruliseks võib nimetada vajadust kohaneda meeskonnaliikmete telefonide erineva ekraanisuurustega. Vaadete loomisel pidime arvestama, et igal liikmel kuvatakse see erinevalt.

Samal ajal aga oli vaadete disainimine lihtsam osa koodi kirjutamisest. Android Studios on vaadete loomine mugav – saab koodi kirjutada ning paralleelselt kõrvalt vaadata, kuidas see välja näeb. Samuti on võimalik aken disainida koodi kirjutamata.

Lihtsamaks osaks oli ka projekti SQLite andmebaasiga ühendamine. SQLite on väga intuitiivne ja arusaadav, ning selle integreerimisel probleeme ei esinenud.

Kõige arendavam oli serveriga ühendamise uurimine, kuna see andis ettekujutust sellest, kuidas *server-client* suhtlemine toimub pilvepõhises andmebaasis. Kahjuks, sellist andmebaasi ei õnnestunud edukalt integreerida. Arendavaks oli samuti mobiilirakenduse loomisel omandatud arusaam telefonis olemasolevate funktsioonidest.

Oleks võinud keskenduda rohkem pilvepõhise andmebaasiga ühendamisele, et oleks võimalik ära näidata sünkroonimist ühe serveriga korraga mitme seadme vahel ning simuleerida päris taksomobiilirakendust. Samuti oleks arendav rohkem uurida Espresso UI Testing raamistikku funktsioonidest, kuna ei saanud lõpuni automaattestide abil testida erinevaid stsenaariume rakenduses.

Eelnevast õppimisest tuli kasuks Java ja C# keelte tundmine ning samuti kogemus JetBrains IntelliJ IDEA-ga, mis aitas Android Studio-ga töötada, kuna antud programm on selle baasil ehitatud. Uueks kogemuseks oli mobiilirakenduse arendamine ja selle käivitamine tegelikus seadmes.

Ei olnud piisavat ettevalmistust interneti andmebaasi seadistamise osas, kuna enne ei tegelenud pilvepõhise andmebaasiga ühendamisega.

Lisa 17 – Keidi isiklik panus

Minu tegevused projektis

Projekti käigus tegelesin peamiselt erinevate lisadega. Esialgse plaani põhjal pidi rakenduses olema ka Facebooki ja Google kontoga sisselogimine, mille uurimiseks ning tööle saamiseks kulus rohkelt aega. Projekti käigus aga otsustasime need sisselogimised siiski hetkel välja jätta ning keskenduda ainult sisselogimisele läbi emaili.

Minu ülesandeks oli luua ka navigatsiooniriba kliendile, juhile ja administraatorile. Antud ülesandele kulus oodatust rohkem aega kuna küljepeale avanev menüü ei ole enam nii populaarne ja pigem kasutatakse rakendustes ekraani all osasse jäävat menüüd. Sobiva lahenduse leidmine võttis aega ning ka see ei jäänud esialgu täiuslik seega tuli selle probleemi juurde hiljem jälle tagasi tulla.

Navigatsiooni menüü igal osapoolel on erinev seega tuli luua nad eraldi ning iga menüü valik teha klikitavaks, et sealt pääseks edasi soovitud vaatele. Navigatsiooni riba ei tule uuele vaatele kaasa, seega on üleval riba peal tagasi nool, mida vajutades saab klient tagasi liikuda eelmisele vaatele. Navigatsiooni menüü kohale on lisatud kasutaja profiilipilt ning tema nimi. Kui klient soovib oma kontost välja logida, saab ta seda teha kasutades navigatsiooni menüül olevat nuppu "logi välja".

Alustasin ka marsruudi loomise koodiga, kuid see jäi hetkel poolikuks, kuna info saamiseks vajalik API on tasuline ning vajaks finantseerimist.

Minu ülesandeks oli ka rakenduse teadete ja nuppude kontroll, et need oleksid korrektses eesti keeles ning kasutajale arusaadavad. Lisaks ka mõned väiksemad täiendused üle rakenduse nagu näiteks, kui kasutaja vajutab telefonil "tagasi" nuppu läks rakendus alguses tagasi sisselogimis vaatesse, kuid ei tohiks sinna minna kui klient on juba sisse logitud ning ka üldine kontroll, et kõik nupud viiks õigesse kohta ja probleemi leidmisel tuli see parandada. Registreerimisele tuli panna ka tingimused, näiteks, et parool peab olema vähemalt 8 tähemärki, telefoni number sisaldab vähemalt 7 numbrit jne.

Lisaks koodile oli ka muid ülesandeid nagu näiteks viia läbi küsitlus, mis oli mõeldud potentsiaalsetele klientidele ja teha sellest kokkuvõtte, mida kliendid rakenduselt tahta võiks ja millele rohkem keskenduda. Selleks, et oleks ka parem ettekujutus, milline peaks

sõidujagamis rakendus olema tuli uurida ka juba olemasolevad rakendusi nagu Uber, Bolt ja Yandex.Taxi.

Rakendusele tuli välja mõelda ning valmis teha ka logo. Minu ülesandeks oli luua erinevaid variante ning leida neist parim, mida võiksime antud rakenduses kasutada. Kuna rakenduse nimi oli DuckeyTaxi tundus igati loogiline logoks teha part. Logol on kaks versiooni (Joonis 14), ühel neist on logo koos kirjaga DuckeyTaxi, antud versiooni kasutasime vaid esimesel vaatel ning teine ilma kirjata oli kasutusel sisselogimis vaatel ning on rakenduse ikooniks telefoni menüüs.



Joonis 14. DuckeyTaxi logo

Eneseanalüüs

Suurim väljakutse oli kindlasti mobiilirakenduse loomine ning Android Studio kasutamine kuna eelnevalt ei olnud sellega kokku puutunud ning kõik tuli selgeks saada kasutades internetis olevat informatsiooni.

Keeruliseks osutus kindlasti see, et Eestis kuulutati välja eriolukord, mistõttu suleti koolid ning inimestel soovitati olla kodus, see aga tähendas, et meil ei olnud võimalik enam meeskonnana kokku saada. Koosolekud hakkasid toimuma interneti teel, mis tegi asja keerulisemaks. Oodatust kergemaks aga osutusid mõned funktsionaalsused, mille puhul olin valmistunud, et koodi tuleb rohkem või tööle saamiseks kulub rohkem aega kui tegelikult. Samuti arvan, et oodatust kergem oli Java keele kasutamine. Varasemalt pole sellega nii suures mahus kokku puutunud kuid kuna keel on üpris sarnane juba varasemalt õpitud C# keelega siis läks ka projekt lihtsamini.

Isiklikult arvan et kogu see protsess oli minu jaoks väga arendav ja huvitav, kuna varasemalt ei olnud loonud ühtegi mobiilirakendust ja ka Java ning XML-iga oli kokkupuude olnud varasemalt minimaalne. Ka teiste loodud koodiga tuli tutvuda ning mõista, mida mille jaoks vaja on.

Üks asi millele oleks võinud veel keskenduda on Facebooki ja Google sisse logimine, mis oli juba pooleldi valmis kuid siiski nõudis veel palju uurimist ning jäi lõpuks pigem tahaplaanile. Nende sisselogimiste kasutamine oleks eeldanud ka kontoandmetest informatsiooni saamist, mida klient muidu sisestab registreerimisel.

Eriolukorra välja kuulutamine oli miski milleks me keegi valmis ei olnud ning see muutis ka meie edasist töö tegemist. Pidime hakkama saama interneti teel üksteisega suhtlemisega, ning sellele aitas kaasa Scrumi ja Trello kasutamine, pidevalt oli olemas ülevaade, mida keegi teeb või mida on vaja teha.