

TALLINNA TEHNIKAÜLIKOOL
INFOTEHNOLOOGIA TEADUSKOND
ARVUTITEADUSE INSTITUUT

FACEBOOKIGA INTEGREERITUD AUTOJAGAMISE RAKENDUS
ANDROIDI PLATVORMIL

bakalaureusetöö

Üliõpilane: Janis Ardel
Üliõpilaskood: 123656IAPB
Juhendaja: Ago Luberg

Tallinn
2015

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.

(kuupäev)

(allkiri)

Annotatsioon

Töö eesmärgiks oli arendada Androidi platvormile autojagamise rakendus. Rakendus võtab andmeid Facebooki transporditeemalistest kommuunidest, kus jagatakse informatsiooni kahe Eesti vahelise asustuse autojagamiste kohta (näiteks on olemas kommuun nimega „Tartu – Tallinn – Tartu”, kus käib infovahetus sellel marsruudil soovivate reisijate ning pakkujate vahel). Rakendus töötleb Facebooki kommuuni postitusi selliselt, et neist on pärast võimalik välja lugeda sõidu toimumise aega, suunda, hinda ning muid parameetreid.

Facebooki postituste töötlemiseks kasutab rakendus peamiselt regulaaravaldisi, millega saadakse iga postituse kohta teada, millal postituse autor soovib sõita, kus suunal postituse autor soovib sõita ning mis tüüpi postitusega on tegu (kas tegu on sõidu pakkumisega või sõidu otsimisega).

Töö tulemusena valmis Androidi rakendus, kus kuvatakse Facebooki kommuuni kirjutatud postitused detailsemal kujul lõputöö autori valminud rakenduses. Lisaks arendati rakendusse postituste filtreerimiste ning sorteerimiste võimalused.

Lõputöö maht on 49 lehekülge, hõlmates endas 5 tabelit ning 12 joonist. Tööl on 1 lisa. Töö on kirjutatud eesti keeles ja lühikokkuvõte inglise keeles

Abstract

This thesis describes how is made transport related application for Android platform which is using data from Facebook communities. In Facebook, there are several transport related communities where people communicate with each other to manage shared rides between two Estonian settlements (for example there is a community named „Tallinn – Tartu - Tallinn”). The application is retrieving data from this type of communities and processeses these post messages to more readable posts.

To process Facebook community messages, the application is using mostly regular expressions. By regular expressions the application finds out additional information about each post. There are regular expressions for finding out the date – when user wants to ride and also for finding out in which direction user wants to travel (for example there is two different directions in „Tallinn – Tartu – Tallinn” community: „Tallinn – Tartu” direction and „Tartu – Tallinn” direction). Also application tries to find out how much the travel costs, what is the posts owner phone number and is the post for offering a ride or searching for a ride.

The result of the thesis was an application, which retrieves data from Facebook communities and processing users writed messages to get more information about this offering or searching post. The application shows these posts graphically and user can search for a posts more quickly than in Facebook. User can search for a post by date, by route or even by mode (whether the post is about searching for a ride or about offering a ride). By using this application the user can also post new messages to Facebook and comment other users writed posts.

Lühendite ja mõistete sõnastik

API	Rakendusliides; i.k. <i>Application Program Interface</i>
HTTP	Protokoll dokumentide vahetamiseks veebis; i.k. <i>HyperText Transfer Protocol</i>
JSON	Andmevahetusformaad; i.k. <i>JavaScript Object Notation</i>
URL	Internetiaadress; i.k. <i>Uniform Resource Locator</i>
IP aadress	Internetiaadress arvutivõrgus toimivate seadmete omavaheliseks suhtluseks; i.k. <i>Internet Protocol address</i>
GET päring	Pöördumine veebiserveri poole andmete küsimise eesmärgil; i.k. <i>GET request</i>
Access-token	Turvaelement kasutaja tuvastamiseks ning sisselogimiseks; i.k. <i>Access-token</i>
IDE	Programmeerimiskeskond; i.k. <i>Integrated Development Environment</i>
Sõidu pakkuja	Autori rakenduses isik, kes pakub oma transpordivahendiga kahe linna vahelist teenust teistele kasutajatele.
Sõidu otsija	Autori rakenduses isik, kes otsib transporti kahe linna vahel.

Jooniste nimekiri

Joonis 1: http://car.piu.ee veebilehe kasutajaliides.....	17
Joonis 2: Autori rakenduse Facebooki suhtluse arhitektuur.....	25
Joonis 3: Rakenduse kasutusjuhtude diagramm.....	26
Joonis 4: Rakenduse avamisel kuvatavad marsruudid.....	27
Joonis 5: Peale valitud marsruudi lehekülje avamist kuvatakse kasutajale süsteemi poolt defineeritud vaikimisi suund.....	28
Joonis 6: Lehe lõpus asub pildiga nupp, mille vajutamisel laetakse uued postitused rakendusse.....	29
Joonis 7: Suuna vahetamine.....	30
Joonis 8: Kuupäeva alusel postituste filtreerimine.....	30
Joonis 9: Kasutaja saab kirjutada oma kommentaari „Kirjuta kommentaar” lahtrisse ning selle seejärel Facebooki postitada.....	31
Joonis 10: Uue postituse lisamise vorm.....	33
Joonis 11: Postituse töötlemise meetodi 'getEntry()' jadadiagramm.....	37
Joonis 12: Rakenduse postituste graafilise kasutajaliidese elemendid.....	43

Tabelite nimekiri

Tabel 1: Autori rakenduse ideega analoogsete rakenduste võrdlus.....	19
Tabel 2: Rakenduse mittefunktsionaalsed nõuded.....	21
Tabel 3: Rakendusse võimalikud liidestavad marsruudid Eesti linnade vahel.....	23
Tabel 4: Facebooki kommuuni "Kuressaare-Tallinn-Kuressaare" postituste statistika.....	42
Tabel 5: Autori rakenduse poolne postituste töötlemiste tulemused.....	42

Sisukord

Annotatsioon	3
Abstract	4
Lühendite ja mõistete sõnastik	5
Jooniste nimekiri	6
Tabelite nimekiri	7
Sisukord	8
1. Sissejuhatus	10
1.1 Taust ja probleem.....	10
1.2 Ülesande püstitus.....	10
1.3 Metoodika.....	11
1.4 Ülevaade tööst.....	11
2. Rakenduse võrdlemine juba olemasolevate rakendustega	13
2.1 Uber.....	13
2.1.1 Rakenduse ülevaade.....	14
2.1.2 Uberi alternatiivid.....	14
2.2 Sidecar.....	14
2.2.1 Rakenduse ülevaade.....	15
2.3 http://car.piu.ee	16
2.3.1 Rakenduse ülevaade.....	17
2.4 Võrdluste kokkuvõte.....	18
3. Rakenduse mittefunktsionaalsed nõuded	20
4. Rakenduse lahendus	22
4.1 Rakenduse võimalikud marsruudid.....	22
4.2 Rakenduse probleemid Facebooki kommuunidega.....	23
4.3 Tehnilised piirangud.....	24
4.4 Valmisoleva rakenduse teostuse vahendid.....	24
4.4.1 Facebook'i rakendus.....	25
4.5 Valmisoleva rakenduse funktsionaalsuse analüüs.....	26
4.5.1 Kasutaja sisselogimine.....	26
4.5.2 Facebooki õiguste küsimine.....	26

4.5.3 Kasutaja poolt soovitud marsruudi valimine.....	27
4.5.4 Rakenduse seaded.....	28
4.5.5 Kahe linna vahelise marsruudi kuvamine.....	28
4.5.6 Marsruudil vaikumisi postituste kuvamine.....	29
4.5.7 Kahe linna vahelise suuna muutmine.....	29
4.5.8 Postituse tüübi vahetamine (sõidu pakkumine, sõidu otsimine).....	30
4.5.9 Postituste kuvamine kuupäeva alusel.....	30
4.5.10 Ainult sõprade postituste kuvamine.....	31
4.5.11 Leitud sõidu pakkumisele või sõidu otsimisele registreerimine.....	31
4.5.12 Uue postituse lisamine.....	32
4.6 Facebooki postituste töötlus.....	33
4.6.1 Facebook Graph API.....	34
4.6.2 Autori rakenduse päritavad Facebook Graph API andmed.....	34
4.6.3 Autori rakenduse poolne Facebooki postituse tekstitöötlus.....	37
4.6.3.1 Teenus EntryService.....	37
4.6.3.2 Teenus ModeService.....	38
4.6.3.3 Teenus UserService.....	38
4.6.3.4 Teenus DateService.....	39
4.6.3.5 Teenus PhoneNumberService.....	40
4.6.3.6 Teenus PriceService.....	40
4.6.3.7 Teenus CommentService.....	41
4.6.4 Facebooki postituste töötlemise statistika.....	41
4.6.5 Postituse graafiline kuvamine rakenduses.....	42
5. Kokkuvõte	45
Summary	47
Kasutatud kirjandus	48
Lisa 1	49

1. Sissejuhatus

Bakalaureusetöö eesmärk on programmeerida transporditeemaline rakendus, mis võimaldab selle kasutajatel jagada oma autot ühel kindal marsruudil toimuval reisirühmal teiste samal marsruudil reisida soovivate inimestega. Tulenevalt Facebooki [1] erinevate Eesti linnade vaheliste marsruutidega kommuunide suurest populaarsusest ja suurest kasutajate hulgast otsustas autor arendada rakenduse, mis suudab Facebooki kommuunide postitused töödelda rakenduse kasutajatele mugavamaks lugemiseks. Töö käigus arendati rakendus Androidi [2] platvormile.

1.1 Taust ja probleem

Antud töö põhineb Facebookis asuvatel kommuunidel, kus vahendatakse kasutajate transporditeemalisi soovide Eesti suuremate asukohtade vahel (näiteks on olemas kommuun nimega “Tallinn-Tartu-Tallinn”), kus Facebooki kasutajad sõidavad sellel marsruudil ning otsivad omale reisikaaslast. Antud töö on vajalik mitmetel põhjustel. Esiteks selleks, et kasutajad saaksid busside ja muude ühistranspordi vahendite kõrval võimaluse reisida teiste rakenduse kasutajate autodega. Samuti aitab rakendus vähendada nende Facebooki kasutajate ajakulu, kes soovivad leida omale planeeritud reisi kaaslast või kes otsivad autot koos juhiga, kes neid soovitud marsruudil teenindavad.

Töö tulemusena valminud rakendus on suunatud kõigile, kes soovivad sõita kahe Eesti asustuse vahel ning soovivad oma sõiduvahendit teiste inimestega jagada. Samuti on rakendus suunatud neile, kes soovivad otsida kahe Eesti asustuse vahelisi sõidupakkumisi. Üldiselt on sõidu pakujate eesmärk saada oma autoga sooritatava reisi jagamise eest ka sõidukompensatsiooni.

1.2 Ülesande püstitus

Transporditeemalistes Facebooki kommuunides toimub väga palju infovahetust ning tihti kulub palju aega, et omale sobiv sõit välja valida. Omale vajaliku sõidu välja otsimiseks ning vajaliku info kättesaamiseks peab läbi lugema terve postituse, et aru saada, mis pakkumisega

on tegu. Postitusi peab läbi lugema senikaua, kuniks on omale meelepärane sõit üles leitud ning välja valitud.

Autori rakendus lahendab selle ajakulu probleemi, pakkudes erinevaid filtreid sorteerimaks kirjutatud postitusi, millega omale sobivat pakkumist leida.

- Kuupäeva filter – kasutaja saab valida, mis kuupäevaga seotud postitusi soovib vaadata.
- Suuna filter – kasutaja saab valida, kummal suunal liiklejate postitusi soovib vaadata (näiteks „Tallinn – Tartu – Tallinn” marsruudil on eraldi suundadeks „Tallinn – Tartu” ning „Tartu - Tallinn”)
- Postituse tüübi filter – kasutaja saab valida, kas kuvada sõidu otsimise postitusi või sõidu pakkumise postitusi.
- Sõidupakkumise hinna filter – kasutaja saab hinna alusel järjestada sõidu pakkumiste postitusi.

1.3 Metoodika

Töö käigus arendatud rakendus valmis Androidi mobiilplatvormil. Androidi arendustöid teostati Android Studio [3] arenduskeskkonnas.

1.4 Ülevaade tööst

Töö on jaotatud kolmeks suuremaks etapiks:

- Hetkel olemasolevate transporditeemaliste rakenduste analüüs. Tuuakse välja, millised on suuremad autori ideega analoogsed transpordirakendused, mida kasutajal on võimalik täna kasutada. Etapis võrreldakse väljatoodud rakendusi omavahel, mis on aluseks autori loodavale rakendusele.
- Autori loodud rakenduse analüüs. Etapis kirjeldatakse, millised on rakenduse kasutuslood, kuidas on võimalik rakendust kasutada ning milline rakendus välja näeb.

- Autori loodud rakenduse tehniline teostus. Etapis kirjeldatakse, kuidas on teostatud kasutajate kirjutatud postitustele erinevate filtrite rakendus ning kuidas jõuab üks postitus Facebookist autori rakendusse.

2. Rakenduse võrdlemine juba olemasolevate rakendustega

Valik olemasolevate transporditeemaliste rakenduste vahel on kasutajatel juba üsna mitmekesine. Seega valis autor võrdlusesse selliseid rakendusi, mis vastaksid järgmistele küsimustele: Kas rakenduses saavad füüsilised isikud pakkuda oma autoga teistele reisijatele teenust ning kas rakenduses saavad kasutajad avaldada soovi sõita teatud suunal.

Rakendusest parema ülevaate saamiseks vastab autor iga rakenduse kohta järgmistele küsimustele:

1. Kas rakendust saab kasutada info jagamiseks kahe linna vahelises transpordiühenduses?
2. Kas rakenduses saavad füüsilised isikud pakkuda transporditeenust?
3. Kas rakenduses saab postitada nii sõidu otsimise kui ka pakkumise soove?
4. Kui palju on rakendusel kasutajaid?
5. Kas rakendust saab kasutada Eestis?

2.1 Uber

Uber [4] on autori arvates kõige populaarsem transpordirakendus, kus teenust saavad pakkuda taksojuhtide asemel tavalised inimesed. Uber töötab maailma suuremates linnades (samuti ka Tallinnas). Teenust osutavad nii taksojuhid kui ka tavalised autojuhid, kes pakuvad linnasiseseid transporditeenuseid oma rakenduse klientidele. Autojuht peab kliendi saamiseks olema iga hetk valmis kliendi soovidele vastama ning talle järgi sõitma. Klient vaatab rakendusest, kus on temale kõige lähedamal asuv või muul põhjusel kõige rohkem huvi pakkuv auto ning saadab autojuhile läbi rakenduse päringu oma sõidusooviga, millele autojuht peab teatud aja jooksul vastama. Kui autojuht selle aja jooksul sõidusoovijale vastust ei anna, siis jõuab sõidu soovijale info, et valitud autojuht ei vastanud soovile. Peamiselt on tegu taksoteenusega, kus taksojuhti saavad mängida ka tavalised autojuhid. Maksmine toimub samuti läbi Uber rakenduse.

Uber võtab partners.uber.com veebilehe kaudu vastu igäühe taotlusi, kes tahab hakata Uberi kaudu oma autoga raha teenima. Vastu võetakse need juhid, kes on vähemalt 21-aastased, juhtimisõigusega ning kes läbivad taustakontrolli. Uberi juhi sõidukil peab olema liikluskindlustus juhi nimel. Sõiduk ei tohi olla vanem kui 10 aastane ja peab olema vähemalt nelja uksega. [5]

2.1.1 Rakenduse ülevaade

1. *Kas rakendust saab kasutada info jagamiseks kahe linna vahelises transpordiühenduses?*

Rakendus on mõeldud ühe linna siseseks kasutamiseks taksoteenustele alternatiiviks. Kahe linna vaheline transport on võimalik, kuid tülikas ja mõttetult kulukas.

2. *Kas rakenduses saavad füüsilised isikud pakkuda transporditeenust?*
Jah, samamoodi saavad ka taksojuhid.

3. *Kas rakenduses saab postitada nii sõidu otsimise kui ka pakkumise soove?*
Ei, rakenduses on aktiivsed autojuhid, kes reageerivad kasutajate sõidusoovidele.

4. *Kui palju on rakendusel kasutajaid?*

Üle 8 miljoni.

5. *Kas rakendust saab kasutada Eestis?*

Jah, Tallinnas alates 15. mai 2015

2.1.2 Uberi alternatiivid

Kuna Uber on üle maailma populaarne rakendus, siis on sellele rakendusele tehtud ka üsna mitmeid samadel põhimõtetel töötavaid rakendusi. Üks näidetest on näiteks Lyft - Uberi peamine konkurent Ameerikas. Funktsionaalsus on samasugune nagu Uberil ainult kasutajaliidese disainis on erinevused. Rakendus töötab 50 suuremas Ameerika linnas.

2.2 Sidecar

Sidecar [6] töötab 10 suuremas Ameerika linnas. Sidecar rakendus on suunatud tavalistele liiklejatele, kes sõidavad oma isiklike autodega.

Sidecar rakendus pakub oma kasutajatele kolme suuremat teenust:

- Uberi stiilis sõiduotsijate ja -pakkujate ühendamine. Sõidu pakkujateks on tavalised inimesed oma isiklike autodega. [7]
- Autojagamine – teenus, kus autojuhid määravad teekonna, kuupäeva ja kellaaja ning kasutajad saavad autojuhiga ühendust võttes ennast reisikaaslaseks pakkuda. [7]
- Pakiveo teenus – kasutajad saavad läbi Sidecar rakenduse saata oma pakke samal päeval sihtkohta toimetada, kasutades selleks Sidecar sõitjaid. [7]

Lisavõimalused:

- „Lemmikute” sektsioon – kasutaja saab märkida sõitjaid oma lemmikute listi, et neid edaspidi mugavamalt üles leida.
- Reisi lisavõimaluste valik – sõitja saab ära kirjeldada milliseid lisamugavusi saab sõitja tema reisi ajal kasutada (näiteks snäkid, joogid või lisa istekoht).
- Reisi hinnanguline kestvus – reisija saab omale sõitu otsides vaadata, kui kaua sõit kestab.

2.2.1 Rakenduse ülevaade

Sidecar sarnaneb autori loodava rakenduse ideega selle poolest, et kasutajad saavad lisada oma igapäevaseid sõite pakkumiseks teistele rakenduse kasutajatele. Sõitjad määravad alguspunkti ja lõpp-punkti ning reisijad saavad valida erinevate sõitjate vahel omale meeldivamat sõidupakkujat.

Erinevused on selles, et reisijad saavad sõitjatele taotluse, et antud kasutaja soovib reisist osa võtta ning peale seda võib sõitja selle pakkumise veel tühistada

1. *Kas rakendust saab kasutada info jagamiseks kahe linna vahelises transpordiühenduses?*

Jah

2. *Kas rakenduses saavad füüsilised isikud pakkuda transporditeenust?
Jah.*
3. *Kas rakenduses saab postitada nii sõidu otsimise kui ka pakkumise soove?
Ei.*
4. *Kui palju on rakendusel kasutajaid? Rakendus on kasutusel kümnes Ameerika linnas*
5. *Kas rakendust saab kasutada Eestis?
Ei*

2.3 <http://car.piu.ee>

Tegemist on Eestis töötava rakendusega. Põhiline idee ning arhitektuur on sarnane esialgse autori rakenduse ideega, kuid antud veebilehel saab liikuda vaid ühel, Tallinn – Tartu marsruudil. Uusi postitusi saab lisada läbi rakenduse veebilehe ning hiljem saab seda Facebookis antud kommuuni jagada.

Lehe arendamisel on rakendatud autori ideega sarnast tehnoloogiat. Üks erinevustest on see, et antud rakenduses on kasutusel enda server, kus salvestatakse kasutajate lisatud postitused. Facebooki postitusi rakendus ei töötle ning Facebooki kirjutatud postitus ei jõua antud rakendusse. Küll aga saab rakenduse serverisse lisatud postitust jagada hiljem Facebooki kommuuni alla. See teeb rakenduses lihtsamaks postituse objekti loomise - selle hinna, suuna ja muude parameetrite väärtustamise. Autori arvamuse kohaselt paistab rakendust jälgides välja, et selline süsteem kasutajate arvule hästi ei mõju. Kasutajad on harjunud kõike ühest kohast leidma ning sellist uut eraldiseisvat autojagamise rakendust pole hästi omaks võtnud.

Antud rakenduses autokohtade jagamisel on marsruut Tallinn – Tartu – Tallinn jagatud kaheks suunaks: Tallinn – Tartu ning Tartu – Tallinn. Ühest postitusest võib välja lugeda selle kuupäeva, kellaaja ning hinna.

The screenshot shows the website 'Autokohtade jagamine' with a navigation bar containing 'AVALAHT', 'TALLINN → TARTU', 'TARTU → TALLINN', 'LEMMIKUD', and 'LISA SÕIT'. A main heading reads 'Autod Tartust Tallinnasse'. A car listing for 'Andrus Luhtla' is shown with details: 'Väljasõit: 18 mai 06:00' and 'Hind: 5 €'. To the right, a sidebar contains three sections: 'JÄRGISED SUUNAL TALLINN → TARTU' (Autosid pole), 'JÄRGISED SUUNAL TARTU → TALLINN' (E (18.05) 06:00 sõidab Andrus Luhtla), and 'SÕIDUD KUUPÄEVA JÄRGI' (a calendar for May 2015). The calendar shows the 18th as the current date. At the bottom, there is contact information for 'Autor: Piu' and a 'Jaga' button with a count of 217.

Joonis 1: <http://car.piu.ee> veebilehe kasutajaliides

Lisavõimalused:

- Hindamisfunktsioon – kasutajad saavad hinnata teisi kasutajaid selle põhjal, kas tegu on hea juhiga või mitte.
- „Lemmikud” sekstioon – kasutaja saab märkida sõitjaid oma lemmikute listi, et neid edaspidi mugavamalt üles leida.

2.3.1 Rakenduse ülevaade

1. Kas rakendust saab kasutada info jagamiseks kahe linna vahelises transpordiühenduses?
Jah, kuid marsruute on üks: Tallinn – Tartu
2. Kas rakenduses saavad füüsilised isikud pakkuda transporditeenust?
Jah.
3. Kas rakenduses saab postitada nii sõidu otsimise kui ka pakkumise soove?
Ei.
4. Kui palju on rakendusel kasutajaid? Teadmata. Esmasel hinnangul on võimalike kasutajate arv alla 100.

5. *Kas rakendust saab kasutada Eestis?*

Jah

2.4 Võrdluste kokkuvõte

Järgnevalt on tabelis 1 ära kirjeldatud kolme rakenduse omavaheline võrdlus. Võrdlustest selgub, et Eestis on kasutatavad kaks rakendust: Uber ning veebilehel <http://car.piu.ee> asuv rakendus. Kõige sarnasemad autori ideega on Sidecar ning car.piu.ee autojagamise rakendus, mis mõlemad on suunatud autojuhtidele, kes teevad kindlat marsruuti mööda oma tavapäraseid sõite ning soovivad kaasata teisi reisijaid, et jagada oma reisiga kaasnevaid kulusi.

Erinevaid transporditeemalisi rakendusi analüüses leidis autor, et Eestis ei ole töötavat rakendust, mis pakuks kasutajatele võimalust pikematel marsruutidel (näiteks kahe Eesti asustuse vahel) jagada oma autot teiste kasutajatega, mis on ajendiks uue rakenduse loomisele.

Kõikides olemasolevates rakendustes on olemas hindamisfunktsioon sõitjatele ning samuti ka reisijatele, mida autor plaanib kasutada ka oma rakenduses. Samuti avaldas autorile muljet Sidecari ning Uberi rakenduste lahendus, mis kuvab autosid interaktiivsel kaardil, mille kasutamist autor plaanib rakenduse tegemisel või edasiarendustes.

Kuigi <http://car.piu.ee> aadressil asuv autojagamise rakendus on autori ideega kõige sarnasem, ei kasuta autor seal kasutatud tehnoloogiat ega kasutajaliidese kujundust (joonis 1), kus uue postituse lisamine käib läbi rakenduse vaheserveri, vaid hoopis suhtleb otse Facebooki andmebaasiga.

	Uber	Sidecar	car.piu.ee
Kas saab kasutada transpordi leidmiseks kahe linna vahel?	Jah, kui autojuht/ taksojuht on nõus, üldjuhul linnasisesteks sõitudeks	Jah	Jah, hetkel töötab marsruudil Tallinn - Tartu
Kas on võimalik kasutada Eestis? (25.05.2015 seisuga)	Jah	Ei	Jah
Kas on võimalik kasutada mobiilil?	Jah, eraldi mobiilirakendus Androidi, IOS ning Windowsi platvormidel	Jah, eraldi mobiilirakendus Androidi ning IOS platvormidel	Eraldi mobiilirakendust ei ole, kuid veebileht on mobiilisõbralik
Kas saab sisestada uusi sõidu pakkumise postitusi? (kindla ajaga ja marsruudiga)	Ei	Jah	Jah
Kas saab vaadata kõiki lisatud postitusi valitud marsruudil?	Ei	Jah, saab näha jooksvalt kaardil	Jah
Kas saab sisestada uusi sõidu otsimise postitusi?	Ei, saab teada anda oma sõidusoovist	Ei, saab teada anda enda valitud juhile oma sõidusoovist	Ei, saab teada anda enda valitud juhile oma sõidusoovist
Kas saab maksta sõidu eest läbi rakenduse?	Jah	Jah	Ei
Kas sõitjaid ja reisijaid saab hinnata?	Jah	Jah	Jah

Tabel 1: Autori rakenduse ideega analoogsete rakenduste võrdlus

3. Rakenduse mittefunktsionaalsed nõuded

Tüüp	Nõude kirjeldus
Keel	Rakenduse kasutajaliides peab olema eesti keeles, kuid arendustööd peavad olema realiseeritud nii, et uute keelte lisamine ei nõua suuri ümbermuudatusi rakenduse arhitektuuris.
Kasutajaliides	Kasutajaliides on mobiili/tahvelarvuti põhine.
Töökiirus	Töökiirus on suuresti sõltuv kasutaja internetiühendusest. Kuid keskmise mobiilse internetiühenduse kiiruse juures peaksid kehtima järgmised nõuded töökiiruse osas: <ul style="list-style-type: none"> • Esimene laadimine (peale rakenduse käivitamist ja marsruudi valimist) ei tohi aega võtta üle 5 sekundi. • Marsruudi suuna vahetamine (näiteks “Tallinn – Tartu” suuna muutmine “Tartu – Tallinn” suunaks) ei tohi aega võtta üle 5 sekundi. • Otsingufiltri realiseerimine ning sorteeritud andmete kuvamine ei tohi aega võtta üle 5 sekundi.
Töökindlus	Kõikidest postitustest, mis asuvad Facebooki seinal, peab autori rakendusse jõudma 99% Facebooki kommuunis asuvatest postitustest. Autori rakenduse poolne postituste töötlemine peab andma järgmised tulemused: <ul style="list-style-type: none"> • 95% postituste suundadest peavad vastama tegelikkusele • 99% hindadest peavad vastama tegelikkusele • 99% telefoninumbritest peavad vastama tegelikkusele • 90% postituse tüüpidest peavad vastama tegelikkusele
Ligipääsetavus	Rakenduses peab olema andmete lugemisõigus kõikidele Facebooki kasutajatele, kui tegu on avaliku kommuuniga. Rakenduses peab olema andmete lugemis – ning kirjutamisõigus kõikidele Facebooki kommuuni liikmetele, kui tegu on avaliku või privaatsel kommuuniga.
Turvalisus	Rakenduse enda serveri puudumise tõttu on kõik andmed Facebookist päritavad ning turvalisus sõltub suuresti Facebooki turvalisusest.

Tüüp	Nõude kirjeldus
Testitavus	Rakenduses peavad olema testidega kaetud järgmised komponendid <ul style="list-style-type: none"> • Postituse tüübi otsimise teenus • Postitusest kuupäeva otsimise teenus • Postitusest telefoninumbri otsimise teenus • Postitusest hinna otsimise teenus

Tabel 2: Rakenduse mittefunktsionaalsed nõuded

4. Rakenduse lahendus

4.1 Rakenduse võimalikud marsruudid

Järgnevalt kirjeldatakse ära hetkel Facebookis registreeritud kommuunid, kus käib infovahetus sõidu otsimistest ja sõidu pakkumistest Eesti asustuste vahel ning on seega võimalik autori rakendusega liidestada.

Kriteeriumid, millele peavad kommuunid vastama:

- Vähemalt 1000 kasutajat grupis.
- Hinnanguliselt 90% kommuuni seina postitustest on seotud reise pakkumise/otsimisega.

Võimalikud liidestatavad marsruudid:

Kommuuni nimi	Staatuse ¹	Liikmeid (04.05.2015 seisuga)
Tallinn – Tartu – Tallinn	privaatne	16 400
Tallinn – Pärnu – Tallinn	privaatne	9 500
Tallinn – Kuressaare – Tallinn	avalik	6 500
Tallinn – Tartu - Tallinn	avalik	5 500
Viljandi – Tartu/Tallinn	privaatne	5 100
Tallinn – Rakvere – Tallinn	avalik	3 000
Tallinn – Haapsalu – Tallinn	avalik	3 000
Tartu – Pärnu – Tartu	avalik	2 000

¹ Staatuse näitab seda, kas kommuuni postitused on avalikud, mida saavad näha kõik Facebooki kasutajad, kuid uusi postitusi saavad lisada vaid kommuuni liikmed. Privaatne tähendab, et kommuuni tegevust saavad jälgida ainult need, kes on selle kommuuniga liitunud.

Kommuuni nimi	Staatus	Liikmeid (04.05.2015 seisuga)
Tallinn – Ida-Virumaa – Tallinn	privaatne	1 800
Tallinn – Rapla – Tallinn	avalik	1 500
Tartu saarlased	avalik	1 400
Tartu – Jõhvi - Tartu	privaatne	1 200
Tartu – Haapsalu - Tartu	avalik	1 100
Paide – Tallinn - Paide	avalik	1 000
Valga-Tartu/Tallinn-Valga	avalik	1 000
		59 000

Tabel 3: Rakendusse võimalikud liidestavad marsruudid Eesti linnade vahel.

4.2 Rakenduse probleemid Facebooki kommuunidega

Esilekerkinud probleemid Facebooki kommuunidega

- Osad kommuunid on privaatsed. See tähendab, et kommuunis toimuvat infot saavad jälgida ainult need Facebooki kasutajad, kes on end sinna kommuuni registreerinud.
- Avalikes kommuunides ei saa kasutaja lisada uut postitust ning samuti ei saa olemasolevaid postitusi kommenteerida ilma, et ta oleks selle kommuuniga liitunud.
- Sõltuvus kommuunisisestest reeglitest. Üldine poliitika kommuunis on selline, et igaüks, kes soovib või pakub reisi, postitab uue kirje kommuuni seinale. On aga näiteks kommuun „Tallinn – Tartu – Tallinn”, kus kommuuni administraator/haldaja postitab iga päeva kohta uue kirje seinale ning kel on soovi otsida/pakkuda reisi, see kommenteerib antud postitust. Sellega aga kaasnevad järgmised probleemid:
 - Selliselt rakendatud kommuune tuleb käsitleda ja töödelda eraldi ülejäänutest kommuunidest.

- Kui postitamise poliitikat otsustatakse kommuuni siseselt muuta, siis peab muutma ka loogikat, kuidas rakendus küsib Facebooki Graph API [8] käest

4.3 Tehnilised piirangud

- Facebook Graph API piirangud:
 - 600 sekundi jooksul saab teha ühelt access-tokenilt ja IP aadressilt 600 päringut (igale kasutajale antakse oma access-token, seega üks kasutaja saab 10 minuti jooksul teha 600 päringut). [9]
 - 100 miljonit päringut saab sooritada 24h jooksul (piirang on Facebooki rakenduse tasemel, seega kõik kasutajad kokku saavad sooritada päevas 100 miljonit päringut). [9]
 - Maksimaalselt 5 miljonit kasutajat võivad ühes kuus rakendust kasutada. [9]

Eelnevalt välja toodud Facebooki piirangud ei ole takistuseks rakenduse edukaks toimimiseks.

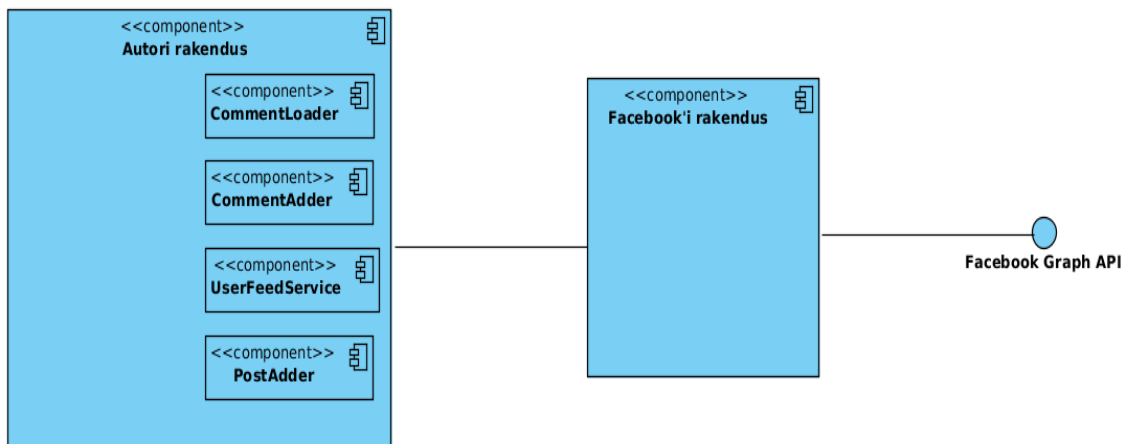
4.4 Valmisoleva rakenduse teostuse vahendid

Rakenduse on kirjutatud Android Studio [3] platvormi kasutades. Testimiseks kasutati peamiselt „Android Virtual Device” Nexus 5 emulaatorit ning reaalselt seadet Samsung Galaxy S3 mini.

4.4.1 Facebook'i rakendus

Rakenduse toimimiseks on vajalik registreerida Facebooki arendajatele mõeldud keskkonnas uus rakendus. Facebookis registreeritud rakendus on vahekiht autori rakenduse ning Facebook Graph API vaheliseks suhtluseks (Joonis 2).

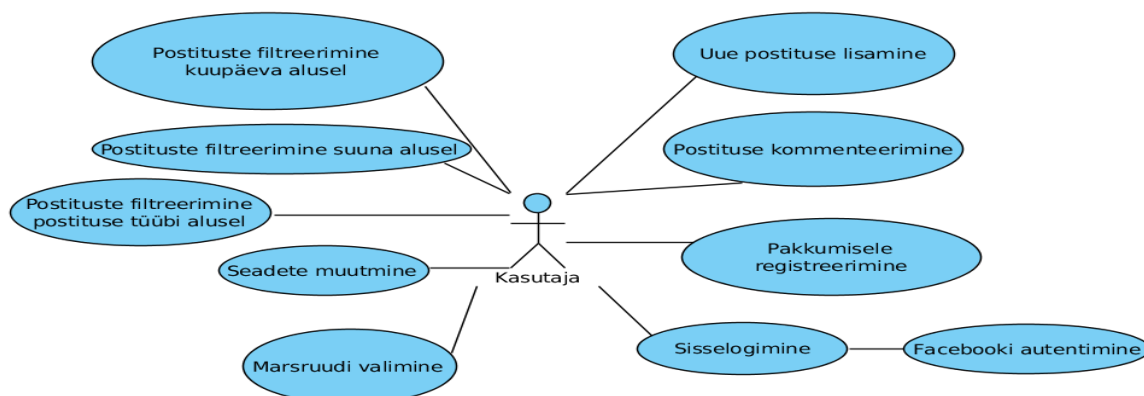
Uue Facebook'i rakenduse loomisel genereeritakse rakendusele unikaalne identifikaator, mis on vajalik Androidi rakenduses Facebooki Graph API poolsete päringute tegemisel. Lisaks tuleb Facebook'i rakenduses ära määratleda, milliseid õigusi kasutaja käest küsitakse, kui kasutaja Facebooki sisse logib.



Joonis 2: Autori rakenduse Facebooki suhtluse arhitektuur

Rakenduse kasutajaliides on mobiilipõhine. Rakendus töötab Androidi platvormi telefonidel ning tahvelarvutitel.

4.5 Valmisoleva rakenduse funktsionaalsuse analüüs



Joonis 3: Rakenduse kasutusjuhtude diagramm

4.5.1 Kasutaja sisselogimine

Rakendusse esimest korda sisenedes kuvatakse kasutajale nupp, mille abil on võimalik end Facebooki kontoga autentida. Rakenduse kasutamiseks on kohustuslik kasutajal end Facebooki kontoga siduda.

4.5.2 Facebooki õiguste küsimine

Rakendus vajab Facebooki Graph API'ga suhtlemisel kasutajapoolset nõusolekut selleks, et rakendus saaks informatsiooni kasutaja kohta. Rakendus küsib järgmisi nõusolekuid

- *public_profile* – kasutaja informatsioon, mida autori rakendus kasutab: eesnimi, perekonnanimi, kasutaja id, kasutaja profiilipildi url. [10] Nõusolekut on vaja selleks, et rakendus saaks teada kasutaja nime ning profiilipildi.
- *user_friends* – kasutaja sõprade nimekiri, mida kasutatakse postituste sorteerimisel kui kasutaja on valinud seadetes, et soovib näha vaid oma sõprade postitusi.

- *user_groups* – lubab rakendusel lugeda kommuunide andmeid, millega kasutaja on liitunud. [10]
- *user_managed_groups* – lubab rakendusel lugeda kommuunide andmeid, kus kasutaja on administraator või kommuuni omanik. [10]
- *publish_actions* – lubab rakenduse kasutajal oma Facebooki konto alt postitada Facebooki seinale. [10] Rakendus vajab nõusolekut selleks, et kasutaja saaks lisada uue postituse Facebooki andmebaasi.

4.5.3 Kasutaja poolt soovitud marsruudi valimine

Rakendust avades kuvatakse kasutajale nappudena erinevate linnade vahelisi marsruute ning lisaks valikut minna rakenduse seadetes (Joonis 4). Vaikimisi kuvatakse kõik marsruudid. Kasutaja teeb oma valiku kuvatavate marsruutide seast ning seejärel suunatakse kasutaja vastava marsruudiga leheküljele.



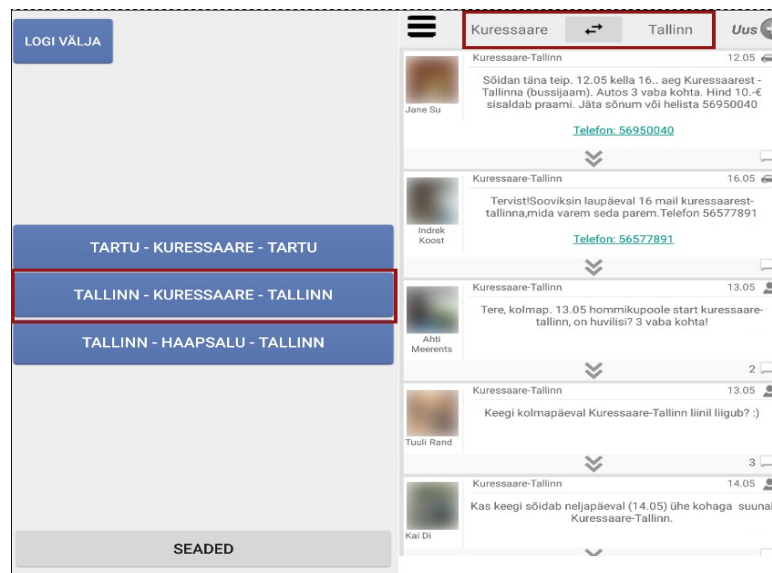
Joonis 4: Rakenduse avamisel kuvatavad marsruudid

4.5.4 Rakenduse seaded

Rakenduse seadetes on kasutajal võimalik muuta rakenduse avamisel kuvatavaid marsruutide nimekirja. On võimalik seadistada, milliseid marsruute kuvatakse rakenduse avamisel ning milliseid mitte. Vaikimisi kuvatakse kõik marsruudid, mis võimalikud.

4.5.5 Kahe linna vahelise marsruudi kuvamine

Kasutaja on valinud peamisest menüüst ühe marsruudi, mille postitustest kasutaja soovib leida endale vajalikku informatsiooni. Peale valiku tegemist ning soovitud marsruudi nupul klikkamist suunatakse kasutaja edasi valitud marsruudi põhilehele (Joonis 5). Valitud marsruut on jagatud suundade alusel kaheks. Vaikimisi kuvatakse ühte süsteemi poolt vaikimisi suunaks määratud suunda (näiteks valitud marsruut „Tallinn – Kuressaare -Tallinn” ning sellel marsruudil on kaks erinevat suunda: Tallinn – Kuressaare ja Kuressaare – Tallinn. Rakenduses kuvatakse marsruudi lehe esimesel laadimisel vaikimisi Kuressaare – Tallinn suunda).



Joonis 5: Peale valitud marsruudi lehekülje avamist kuvatakse kasutajale süsteemi poolt defineeritud vaikimisi suund.

4.5.6 Marsruudil vaikumisi postituste kuvamine

Rakenduses valitud marsruudi lehekülje avamisel kuvatakse süsteemi poolt määratud vaikumisi suunda valitud marsruudil ning selle suunaga seotud kõiki postitusi. Esimesel laadimisel tehakse päring, mis kuvab minimaalselt 5 postitust juhul, kui see on võimalik (juhul kui niipalju postitusi on Facebooki laetud) ning maksimaalselt kuvatakse 10 postitust esimesel laadimisel. Rohkemate postituste kuvamiseks tuleb kasutajal kerida leht lõpuni, kus asub nupp, mille vajutamise peale käivitatakse funktsioon, mis laeb uuesti maksimaalselt 10 postitust rakendusse (Joonis 6).



Joonis 6: Lehe lõpus asub pildiga nupp, mille vajutamisel laetakse uued postitused rakendusse.

Postitused tulevad vaikumisi selles järjekorras, kuidas neid kuvatakse Facebooki seinal ehk kõige enim tulevad viimati postitatud või viimati kommenteeritud postitused.

4.5.7 Kahe linna vahelise suuna muutmine

Kasutaja soovib vaadata sõidu otsimisi või sõidupakkumisi vastupidisel suunal, kui rakendus hetkel kasutajale pakub. Selleks kasutaja vajutab menüüribal olevat nuppu (joonisel 7. märgitud nupp) ning rakendus seejärel laeb ette postitused vastupidisel suunal.



Joonis 7: Suuna vahetamine

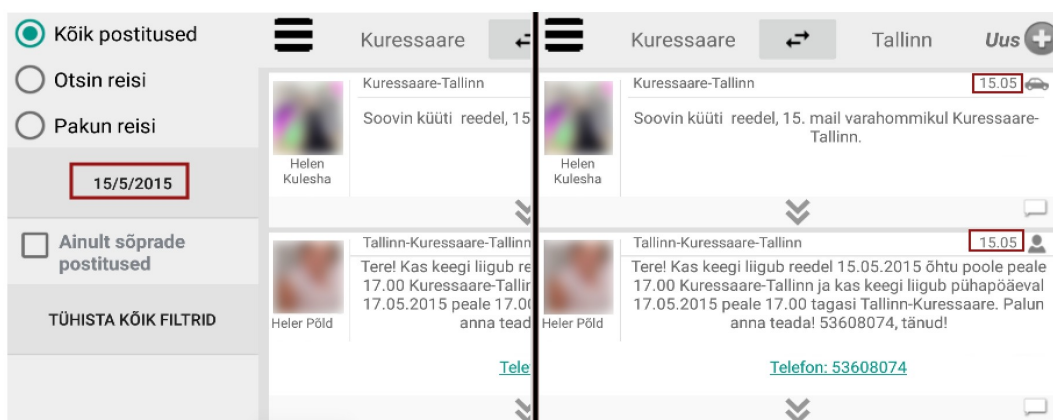
4.5.8 Postituse tüübi vahetamine (sõidu pakkumine, sõidu otsimine)

Kasutaja soovib vaadata endale rohkem huvi pakkuvat sõidu tüüpi ning avab selleks vasaku menüü, milles avaneb võimalus muuta, milliseid sõidu tüüpe rakendus parasjagu kuvab. Valikuteks on sõidu pakkumise tüüp, sõidu otsimise tüüp ning kõik tüübid. Viimase puhul kuvatakse kõiki postitusi ning antud filtrit postituste töötlemisel ei rakendata.

4.5.9 Postituste kuvamine kuupäeva alusel

Kasutaja soovib vaadata postitusi, mille tegevus toimub kasutaja poolt määratud kuupäeval. Selleks kasutaja avab vasaku menüü, kus kuvatakse nupp „Vali kuupäev” mille muutmisel kuvatakse kasutajale postitusi antud kuupäeval (Joonis 8). Edasiarendustes on plaanis teha kuupäeva vahemik, et kasutaja saaks valida vahemikku, mille ulatuses postitusi kuvada.

Kuupäev ei näita seda, millal postitus on Facebooki lisatud, vaid seda, millal kasutaja soovib sõita või pakub sõitu.



Joonis 8: Kuupäeva alusel postituste filtreerimine.

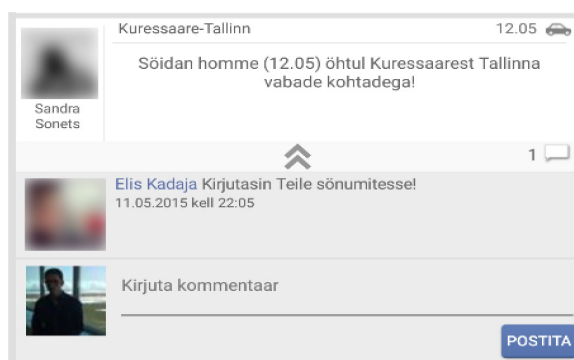
4.5.10 Ainult sõprade postituste kuvamine

Kasutaja soovib otsida postitusi, mis on seotud tema sõpradega. Selleks avab kasutaja vasakpoolse menüü, kus kuvatakse kasutajale checkbox'i „Ainult sõprade postitused”, mille aktiveerimisel rakendus kuvab kasutajale ainult neid postitusi, mille on lisanud Facebooki kommuuni seinale kasutaja Facebooki sõber.

4.5.11 Leitud sõidu pakkumisele või sõidu otsimisele registreerimine

Kasutaja on leidnud endale rakendusest sobiva sõidu pakkumise või sõidu otsija ning soovib leitud isikuga ühendust võtta. Selleks on rakenduse kasutajal kaks võimalikku varianti.

- Rakenduse kasutaja kommenteerib postitust: Rakenduse kasutaja avab postituse kommentaaride sektsiooni ning seal avaneb tal võimalus antud postitusele kommentaar lisada (Joonis 9). Lisatud kommentaar jõuab samuti Facebooki kommuuni.



Joonis 9: Kasutaja saab kirjutada oma kommentaari „Kirjuta kommentaar” lahtrisse ning selle seejärel Facebooki postitada

- Rakenduse kasutaja soovib saata kasutajale privaatset sõnumit: Rakenduse kasutaja klikib postituse omaniku profiilipildil ning seejärel suunab rakendus

kasutaja Facebooki rakendusse, kust avaneb võimalus privaatsete sõnumite edastamiseks.

4.5.12 Uue postituse lisamine

Kasutaja soovib rakendust kasutades lisada uut postitust valitud marsruudi Facebooki lehele. Selleks klikkab kasutaja üleval paremal olevat „Uus” nuppu, mille peale rakendus kuvab kasutajale uut vormi (Joonis 10). Vorm pakub automaatselt teksti, mida Facebooki seinale kuvada. Tekst on seotud vormiväljade väärtustega. Iga vormivälja muutus toob kaasa muudatuse tekstis, mida kasutajale Facebooki postitamiseks pakutakse. Vormiväljad ise on järgmised:

1. „Pakun reisi / Otsin reisi” valik

Kasutaja valib, kumba tüüpi postitust ta soovib lisada.

2. Suuna valik

Kasutaja valib, millisel suunal ta soovib liikuda. Valikus on lisaks põhimarsruudile erinevaid valikuid kahe linna vaheliste marsruutide kombinatsioone (näiteks marsruudil „Kuressaare – Tallinn – Kuressaare” marsruudil on lisaks „Kuressaare - Tallinn” ning „Tallinn – Kuressaare” suundadele veel „Tallinn – Muhu”, „Tallinn – Orissaare” jne.). Selle valiku muutmisel uuendatakse postituse teksti – lisatakse valitud suund.

3. Kuupäeva valik

Kasutaja soovib valida kuupäeva, millal kasutaja otsib/pakub sõitu. Valiku muutmisel muudetakse postituse teksti – lisatakse kuupäev, millal sõidetakse.

4. Sõidu hinna valik

Kasutaja soovib määrata hinna, kui palju üks koht tema sõidukis maksab. Väli on nähtaval ainult siis, kui kasutaja on valinud reisi tüübiks „Pakun reisi”.

5. Vabade kohtade arvu valik

Kasutaja soovib öelda, kui palju on tema sõidukis valitud reisi ajal vabu kohti, mida ta on nõus teistele reisijatele pakkuma.

6. Telefoninumbri valik

Kasutaja, kes soovib sõitu pakkuda, soovib lisada postitusele oma telefoninumbri, mille kaudu temaga kontakti saab.

7. Lõplik tekst

Kasutaja soovib muuta teksti, mida Facebooki postitada. Teksti muudetakse iga teise välja muudatuste peale, kuid kasutajal on ka ise võimalus seda muuta.

Lisa uus postitus ✕

Pakun reisi Otsin reisi

Vali suund
Kuressaare - Tallinn

Millal liigud?
29.5.2015

Ühe koha hind 7 €

Tere, liigun reede, 29 mai
Kuressaare - Tallinn suunal. Ühe
koha hind on 7 eurot

POSTITA

Eelvaade

Janis Ardel Kuressaare - Tallinn 29.05 🚗

Tere, liigun reede, 29 mai Kuressaare - Tallinn suunal. Ühe koha hind on 7 eurot

Joonis 10: Uue postituse lisamise vorm.

4.6 Facebooki postituste töötlus

Järgnevalt on ära kirjeldatud, kuidas jõuab üks postitus Facebooki kommuuni seinalt autori rakendusse. Eeltingimusteks on see, et kasutajal on Facebooki konto ning on liitunud

Facebookis asuva soovitud kahe linna vahelise marsruudi nimelise kommuuniga, millele on autori rakenduse poolne tugi.

Esmalt on vaja lisada postitus Facebooki. Selleks kasutaja lisab uue postituse soovitud kommuuni, mille tulemusena salvestatakse postitus Facebooki andmebaasi.

4.6.1 Facebook Graph API

Facebook Graph API [8] on peamine viis, kuidas saada ning kuidas postitada infot Facebooki sotsiaalvõrgustikku. Tegu on HTTP protokollil põhineva API'ga, millega saab andmeid Facebookist küsida, uusi postitusi lisada, pilte üles laadida ning mitmeid teisi tegevusi sooritada. Kuna see põhineb HTTP'l siis on seda võimalik liidestada kõikide programmeerimiskeeltega, mis toetavad HTTP teeki. [11]

Facebooki Graph API struktuur koosneb kolmest peamisest osast, mida Facebook ise nimetab järgmiselt:

- **Tipud** – põhiliselt objektid, nagu näiteks kasutaja, kommuun, kommentaar, seinapostitus. [11]
- **Kaared** – ühendused objektide vahel, nagu näiteks kommuuni postitused, postituse kommentaar, kommuuni pildid. [11]
- **Väljad** – Informatsiooniväljad objektide kohta, nagu näiteks kasutaja sünnipäev, kommuuni nimi. [11]

Hetkeseisuga (03.05.2015) on kõige uuem Facebook Graph API versioon v2.2.

4.6.2 Autori rakenduse päritavad Facebook Graph API andmed

Facebook Graph API tagastab päritavaid andmeid üle HTTP protokollile ning kõik vastused on JSON formaadis.

Autori rakendus suhtleb Facebook Graph API'ga autori enda kirjutatud klassis, *FeedService.java*, mis vajab parameetrina kaasa kommuuni id. Klassi käivitamisel tehakse rakenduse taustal GET päring *graph.facebook.com/v2.2/id* aadressile, kus id on valitud

kommuuni unikaalne identifikaator. GET päringuga antakse kaasa vajalikud parameetrid, mille peale Facebook genereerib oma vastuse.

Graph API peamise GET päringu parameetrid ja nende väärtused koos selgitustega:

Näidisurl: `graph.facebook.com/v2.2/12345678901/fields=feed.limit(10){message,actions,comments.limit(15){from{name,picture},message,created_time},from{name,picture},created_time}`

- `fields`
 - `feed.limit(10)` – feed, ehk voog on põhiparameeter ning tähendab, et võetakse kommuuni andmetest välja kõik postitused, ning `limit` näitab, mitu postitust võetakse. Antud näitel võetakse 10 postitust
 - `message` – kõikides postitustes kuvatakse sõnumit. `Message` on ühe postituse kõige tähtsam komponent, see on kasutaja poolt sisestatud tekst kommuuni seinale.
 - `comments.limit(15)` – kõikides postitustes kuvatakse kommentaare selle postituse kohta. Vaikimisi on lubatud 15 kommentaari ühe postituse kohta, kui on rohkem kommentaare, siis rakenduses vanemaid kommentaare ei näidata.
 - `from` – kõikide postituste kommentaaride kohta kuvatakse, kes on selle kommentaari autor.
 - `name` – kommentaari autori nimi.
 - `picture` – kommentaari autori profiilipildi URL.
 - `message` – kommentaari tekst, mille autor on postitusele lisanud.
 - `created_time` – kommentaari lisamise kuupäev ning kellaaeg.
- `from` - kõikide postituste kohta kuvatakse, kes on selle postituse autor.
 - `name` – postituse autori nimi.

- `picture` – postituse autori profiilipildi URL.
- `created_time` - postituse lisamise kuupäev ja kellaaeg.

Andmed, mis Facebooki Graph API poolt päringuga saadakse, tulevad ajaliselt järjestatud. Kõige hilisema tegevusega (hiljuti kommenteeritud postitused, hiljuti lisatud postitused) postitused tulevad Graph API'lt kõige esimesena.

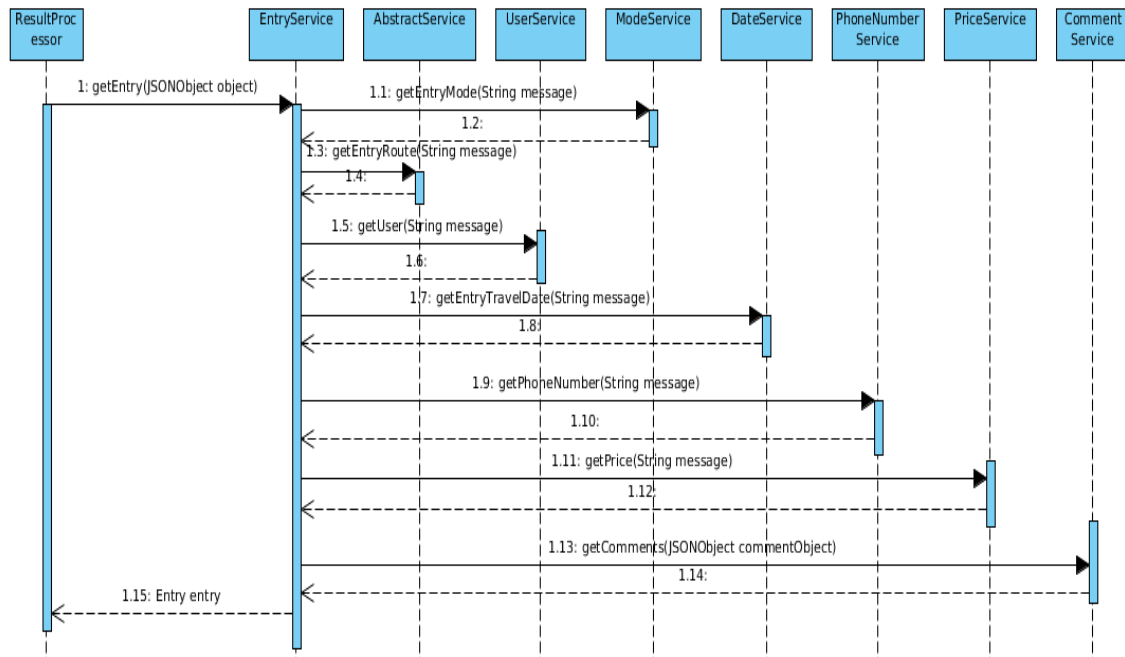
Graph API JSON vastuse näide:

```
{
  „data“: [{
    „message“: „Kas keegi sõidab homme Tallinna?“,
    „from“: {
      ...
    },
    „comments“: [{
      „message“: „Mina lähen“,
      „from“: {
        ...
      }
    }],
    „created_time“: „...“,
    ...
  }],
  „paging“: {
    „after“: „WTI5dGJXVnVkRjlqZFh..“,
    „before“: „WTI5dGJXVnVkRjlqZ.“
  }
}
```

API vastusesse tuleb JSON kujul kogu info, mis parameetrina küsitakse ning lisaks sellele veel üks JSON objekt `paging`, mille elemendid `after` ning `before` viitavad Graph API URL'ile, kust saab vastavalt järgmise ning eelmise lehekülje jagu andmeid.

4.6.3 Autori rakenduse poolne Facebooki postituse tekstitöötlus

Järgnevalt tehakse ülevaade sellest, kuidas jõuab üks postitus Facebooki seinalt autori rakendusse ning milliseid filtreid see läbib ja kuidas töödeldakse erinevaid andmeid postituse sõnumist välja. Joonisel 11 on näidatud, milliseid komponente läbib rakendus Facebooki postituse töötlemisel.



Joonis 11: Postituse töötlemise meetodi 'getEntry()' jadadiagramm.

4.6.3.1 Teenus *EntryService*

Meetod: `getEntry(JSONObject object)` – `object` on Graph API poolt saadud tulemus, mis sisaldab endas Facebooki kommuuni seinapostitusi.

Eeltingimused: Graph API meetod on välja kutsutud, ning selle tulemus on teisendatud `JSONObject` (`org.json` poolt pakutav teek JSON töötlemiseks) tüübiks.

Kirjeldus: Meetod kutsub välja teisi teenuseid, et saada vajalikke väärtuseid `Entry` objekti loomiseks. Meetod ise võtab JSON objektist välja järgmised väärtused:

- `Message(String message, String messageId)`

Tagastatav objekt: Meetod tagastab ühe lõpliku `Entry` objekti, mis sisaldab kõiki vajalikke väljasid.

4.6.3.2 Teenus *ModeService*

Meetod: `getEntryMode(String message)`

Eeltingimused: `EntryService` on võtnud JSON objektist välja sõnumi (`message`).

Kirjeldus: `ModeService` teenus kasutab sõiduviisi leidmiseks kasutaja sisestatud sõnumist kahte regulaaravaldist:

```
FINDER_REGEX = ^(.*((kas)|k(e){2}gi)|(kes)|(s(o){2}vi((ks)|(n)|(me)))|
(otsi)).*)|(.*\?\?.*)$
```

```
OFFER_REGEX = .*((l[i]{1,2}gun)|s[õö]ida((n)|(me))|(s[o]{1,2}vib)).*
```

Kui kasutaja Facebooki seinal loodud sõnum sobitub `FINDER_REGEX`'iga, siis on tegu postitusega, mille tüüp on *otsin sõitu*, `OFFER_REGEX`'i puhul on tüübiks *pakun sõitu*. Kui teenus ei suuda tuvastada mõlema regulaaravaldise abil, kumma sõidu tüübiga on tegu, siis vaikimisi tüüp on „pakun sõitu”.

Tagastatav objekt: Meetod tagastab Java Enum tüüpi objekti (`Mode.java`), enum koosneb kõikidest võimalikest sõidu tüüpidest (*sõidu pakkumine*, *sõidu otsimine*).

4.6.3.3 Teenus *UserService*

Meetod: `getUser(JSONObject object)`

Eeltingimused: `EntryService` on võtnud JSON objektist välja alamobjekti `from`.

Kirjeldus: Teenus parsib JSON objektist välja vajalikud parameetrid (nimi, pildi URL ning id) ja teeb nendest uue Java objekti (`User`)

Tagastatav objekt: `User` – objekt postituse omaniku andmetega.

4.6.3.4 Teenus *DateService*

Eeltingimused: `EntryService` on võtnud JSON objektist välja postituse teksti `message`.

Kirjeldus: `DateService` kasutab kuupäeva otsimiseks kolme erinevat viisi:

- 1) Teenus proovib sõnumist leida kasutaja poolt kirjutatud kuupäeva, millal kasutaja soovib oma sõitu alustada. Selleks kasutab teenus järgmist regulaaravaldist:

```
([0-9]{1,2}[^A-Za-z0-9]*\s*[monthOfYearFirstNum]  
[monthOfYearLastNum-(monthOfYearLastNum+1)]
```

`monthOfYearFirstNum` ja `monthOfYearLastNum` on hetkel käimasoleva kuu numbrilised väärtused, vastavalt hetkel oleva kuu esimene ning teine number, näiteks 11.05 `monthOfYearFirstNum` = 0, `monthOfYearLastNum` = 5. Need muutujad on vajalikud selleks, et sõnumist loetakse kindlalt just kuupäev välja, mitte kellaeg või muu numbriline väärtus.

- 2) Teenus proovib sõnumist leida kasutaja poolt kirjutatud nädalapäeva, millal kasutaja soovib sõitu alustada. Selleks kasutab teenus järgmisi regulaaravaldisi:

```
private String[] weekdaysRegexStringArray = new String[]  
{ " (esmas)", " (teisi)", " (kolmap)", " (neljap)", " (reede)",  
" (laup)", " (p[üy]hap) "};
```

```
String todayRegexString = " (t[äa]na) ";
```

```
String tomorrowRegexString = " (hom(m)?e) ";
```

Esmalt kontrollib teenus sõnumi vastavust regulaaravaldistele, mis kontrollib, kas sõit toimub täna või homme, ning seejärel kontrollib vastavust nädalapäevade regulaaravaldistele, millisel nädalapäeval sõit toimub. Kui nädalapäevade regulaaravaldistega on leitud, millisele nädalapäevale sõnum vastab, siis seadistatakse sõidu alguskuupäevaks kuupäev, mis järgneb

postituse lisamise kuupäevale ning mille nädalapäev võrdub leitud nädalapäevaga.

- 3) Juhul, kui eelpool nimetatud regulaaravaldiset ei anna kasutaja sisestatud sõnumi puhul tulemust, seadistatakse sõidu kuupäevaks see kuupäev, millal kasutaja postituse Facebooki üles laadis.

Tagastatav objekt: Kuupäev (`Date` objekt), millal antud postitusele vastav sõit toimub.

4.6.3.5 Teenus *PhoneNumberService*

Meetod: `getPhoneNumber(String message)`

Eeltingimused: `EntryService` on võtnud `JSON` objektist välja postituse teksti `message`.

Kirjeldus: `PhoneNumberService` kasutab kasutaja poolt sisestatud postitusest telefoninumbri otsimiseks järgmist regulaaravaldist:

`(([+])?[35][\d\s]{6,14})`

Tagastatav objekt: Telefoninumber Java `String` objekti kujul.

4.6.3.6 Teenus *PriceService*

Meetod: `getPrice(String message)`

Eeltingimused: `EntryService` on võtnud `JSON` objektist välja postituse teksti `message`.

Kirjeldus: `PriceService` on vajalik postitusest hinna välja otsimiseks. Teenust kasutatakse postituse töötlemiseks ainult juhul, kui postitus on „pakun sõitu” tüüpi, kus kasutaja pakub teistele kasutajatele küüti. Teenus kasutab kasutaja poolt sisestatud postitusest hinna otsimiseks järgmist regulaaravaldist:

`(\d{1,2}(\.|\,)?\d*)\s?(eur|€|(\.|\,)-|raha)|hind\s(\d{1,2})`

Tagastatav objekt: Postituse autori poolt määratud sõidu hind Java `Double` objekti kujul.

4.6.3.7 Teenus *CommentService*

Meetod: `createComments (JSONObject commentObject)`

Eeltingimused: `EntryService` on võtnud JSON objektist välja alamobjekti `comments`.

Kirjeldus: Kui JSON objekt `comments` on olemas, mis tähendab, et postitust on läbi Facebooki kommenteeritud, siis salvestatakse kõik ühe postituse kommentaarid listi. Ühe kommentaari kohta küsitakse andmed kommentaari sisestaja kohta (kasutajanimi, id ja profiilipildi url) kommentaari lisamise kuupäeva ning kommentaari sisu kohta.

Tagastatav objekt: `List Comment` objektidest.

4.6.4 Facebooki postituste töötlemise statistika

Järgnevalt kirjeldab autor kui palju rakendusse jõutavatest postitustest vastavad tegelikkusele, mida kasutajad on Facebooki kirjutanud. Valimiks valiti 75 postitust ning postitused võeti „Kuressaare-Tallinn-Kuressaare” Facebooki kommuunist.

Tabelist 4 selgub, et sõidu otsimisi ning pakkumisi on kommuunis peaaegu võrdne arv. Teemaväliseid postitusi on 75 postituse hulgast kaks, millest üks oli postitatud reklaam ning teine link internetiaadressiga.

Veel selgub, et sõidu pakkumiste postitustes mainitakse ära hind 60,5% ning telefoninumber 42,1% juhtudest. Sõidu otsimiste postitustes hinda ning telefoninumbrit ei ole mainitud.

Tabelist 5 selgub, et antud valimi korral suudab autori rakendus kasutajate sisestatud sõnumist telefoninumbri ning hinna alati välja lugeda. Postituse töötlemise tulemusena leitud kuupäev on õige 90,4% juhtudest ning õige suund („Kuressaare - Tallinn – Kuressaare” marsruudil) on 91,8% juhtudest. Juhul kui rakendus ei suuda õiget marsruuti tuvastada, kuvab rakendus seda postitust mõlemal suunal.

	Leitud	Osakaal [%]
Teemaväliseid postitusi	2	2,7
Transporditeemalisi postitusi	73	97,3
Sõidu pakkumisega seotud postitusi	38	52,1
Sõidu otsimisega seotud postitusi	35	47,9
Kasutaja poolt sisestatud hinnaga sõidu pakkumise postitusi	23	60,5
Kasutaja poolt sisestatud telefoninumbriga sõidu pakkumise postitusi	16	42,1
Kokku	75	100

Tabel 4: Facebooki kommuuni "Kuressaare-Tallinn-Kuressaare" postituste statistika.

	Leitud	Osakaal [%]
Õige suunaga postitusi	67	91,8
Õige kuupäevaga postitusi	66	90,4
Õige hinnaga postitusi	23	100
Õige telefoninumbriga postitusi	16	100
Kokku	75	100



Tabel 5: Autori rakenduse poolne postituste töötlemiste tulemused



4.6.5 Postituse graafiline kuvamine rakenduses

Järgnevalt on joonisel 12 ära kirjeldatud postituste graafiline esitus autori rakenduses – mida tähendavad erinevad ikoonid ning millest postituste sekstioonid koosnevad.



Joonis 12: Rakenduse postituste graafilise kasutajaliidese elemendid.

- (1) Postituse sõnum.
- (2) Postituse autori profiilipilt ning kasutajanimi. Profiilipildil vajutades avatakse uues internetibrauseri aknas postituse autori Facebooki kasutaja profiil.
- (3) Postituse sõnumist välja loetud marsruut.
- (4) Postituse sõnumist välja loetud kuupäev, millal sõit toimub. Kuupäev on formaadis dd.MM
- (5) Postituse sõnumist välja loetud postituse tüüp
 -  - postituse autor pakub oma postituses sõitu.
 -  - postituse autor otsib oma postitusega küüti.

- (6)  - Kommentaari ikoon. Number ikooni ees näitab, kui palju on kommentaare postitusel.
- (7)  - Kommentaaride sektsiooni avamine/sulgemine. Vaikimisi on kõik kommentaaride sektsioonid suletud.
- (8) Postituse kommentaar. Kommenteerija profiilipildil klikkides avatakse uues internetibrauseri aknas postituse kommentaari autori Facebooki kasutaja profiil.
- (9) Uue kommentaari lisamise sektsioon. „Kirjuta kommentaar” lahtrisse saab kasutaja kirjutada oma soovitud kommentaari.
- (10) Kommentaari lisamine Facebooki andmebaasi.
- (11) Postituse sisust välja loetud hind. Hind loetakse postitusest välja ainult juhul, kui postituse tüüp on „pakun sõitu”.

5. Kokkuvõte

Käesolevas töös on arendatud Androidi platvormile transpordirakendus. Rakendus on mõeldud inimestele, kes soovivad kindlal marsruudil liikudes oma autot jagada teiste rakenduse kasutajatega või inimestele, kes otsivad kindlal marsruudil omale sobivat sõitu. Rakendus integreeriti Facebooki rakendusliidesega. Rakenduse abil saavad selle kasutajad otsida omale valitud marsruudil sõitjaid, kes on valmis võtma oma autosse väikese lisatasu eest kaasreisijaid. Samuti on rakenduse abil võimalik kasutajal end ise valitud marsruudil kaasreisijaks pakkuda.

Töös realiseeriti nii graafilise kasutajaliidese pool Androidi seadmetele, kui ka suhtlus Facebooki rakendusliidesega ning selle seest küsitud postituste töötlus.

Rakendus on eelkõige vajalik inimestele, kes peavad olude sunnil käima kodulinnast teises Eesti linnas koolis või tööl, aga ka teistele inimestele, kes peavad pidevalt liikuma kahe Eesti linna vahel.

Rakenduse käigus realiseeriti täielikult kaks erinevat marsruuti kahe Eesti linna vahel, Tallinn – Kuressaare – Tallinn ning Tallinn – Haapsalu – Tallinn. Võimalus uute marsruutide lisamiseks on aga olemas ning uue marsruudi lisamine ei vaja suuri arendustöid.

Töös kasutati Facebooki postituste töötlemisel regulaaravaldiste kontrole, leidmaks postitusest välja kõik vajalikud andmed, mis teevad kasutaja otsustamised ning otsingud mugavamaks. Vajalike andmete hulka kuuluvad kuupäev, millal sõit toimub, postituse suund, postituse tüüp ning postituse autori telefoninumber.

Püstitatud eesmärgid filtrite arendamisel täideti. Küll aga ei saa väita, et kõik filtrid töötaksid igasuguste kasutajate sisendite puhul. Kasutajad on leidlikud ning kirjutavad postitusi väga erinevalt ja samuti ei saa ka kirjavigade teket välistada. Seepärast on ka raske tagada 100% töökindlust töös realiseeritud filtrite rakendamisel.

Edasiarendustes on esimene plaan lisada rakendusse oma serverikiht. Seoses sellega oleks võimalik rakendusele lisada kohe hulk huvitavaid ja kasutajate jaoks atraktiivseid funktsionaalsusi. Näiteks sõitjate hindamine ning sõitjate edetabel tehtud sõitude arvu poolest. Edasiarendustes on kindlasti plaanis ka uusi marsruute lisada rakendusse.

Lisaks on edasiarendustes plaanis arendada rakendusele admini vaade, kus kommuuni administraator saaks lihtsalt kustutada vanu postitusi (mille kuupäev on mööda läinud). Hetkel käib postituste kustutamine kõik käsitsi, aga oleks võimalik ka see funktsionaalsus rakendusse kirjutada, mis suudaks ühe nupuvajutusega kustutada kõik ajaliselt mööda läinud postitused.

Summary

The aim of this work was to develop a new car sharing application for Android platform. The application integrates with Facebook communities, where people are offering other community members to share their travel costs while driving along with them. Also in these communities, there are people who are searching a drivers to get a lift between two cities. Races are carried out between two Estonian cities, the community is also named according to these cities (for example “Talinn – Tartu – Tallinn”).

The main problem in these communities is that there are always too many information shown. Some members are searching drivers and the others are offering races. On the other hand, some members are travelling in one route (for example from Tallinn to Tartu) and others are travelling in the second route (for example from Tartu to Tallinn). This type of information sharing is causing a big mess in these communities. Authors application is solving this problem by letting people choose whether they want to offer a ride or do they want to search for a ride.

The application is requesting Facebook community feed posts from Facebook Graph API and showing these posts in application. Posts are all processed into more readable objects. Application tries to understand what the members have posted – is it an offering or a searching request, when member wants to ride, how much money driver wants and in which route race is going to be. These processes have been implemented by using regular expressions. Each post processing service is using its own regular expression to get the information from post message.

Facebook community member can also comment existing posts or even post new message to this community via authors application.

Kasutatud kirjandus

- 1: , Facebook, , www.facebook.com
- 2: Astrium, Android, 2014, <https://www.android.com/>
- 3: , Android Studio Overview, 2015, <http://developer.android.com/tools/studio/index.html>
- 4: , Uber, , <https://www.uber.com>
- 5: , Uber driving requirements, 2015, <https://www.uber.com/driver-jobs>
- 6: , Sidecar, 2015, <https://www.side.cr/>
- 7: Sidecar Technologies, Inc, Our Story | Sidecar, 2015, <https://www.side.cr/about/>
- 8: , Graph API, 2015, <https://developers.facebook.com/docs/graph-api>
- 9: Facebook inc, Platform Policy, 2015, <https://developers.facebook.com/policy/>
- 10: , Permissions with Facebook Login, 2015, <https://developers.facebook.com/docs/facebook-login/permissions/v2.3>
- 11: , Facebook Graph Api overview, 2015, <https://developers.facebook.com/docs/graph-api/overview/>

Lisa 1

Viide rakenduse avalikule koodi repositooriumile - <https://github.com/janisar/Transporter>