

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Ott Kossar 192885IADB

Sõidujagamise platvorm Eestis

Bakalaureusetöö

Juhendaja: Meelis Antoi
Magistrikraad

Tallinn 2022

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Ott Kossar

24.04.2022

Annotatsioon

Käesoleva lõputöö eesmärgiks on luua kasutatav veebirakendus, mis aitaks kokku viia autojuhid samal teekonnal liikuda soovivate inimestega. Lõputöö lisaeesmärkideks on disainida veebirakendus nii, et seda oleks võimalik kasutada nii mobiililt kui ka arvutist ning, et valmis arendatud veebirakendus oleks ka juurdepääsetav.

Enne arendust põhjendatakse lõputöö aktuaalsus läbi uurimustöö ning leitakse tähtsamad nõuded, mis kataks loodava lahenduse vajadused nii, et seda oleks võimalik eesmärgipäraselt kasutama hakata. Töö käigus pannakse suurt rõhku kasutatavusele ja juurdepääsetavusele läbi disaini, mille raames luuakse ka prototüüp.

Käesolevas lõputöös käsitletakse lisaks veel vajalikke tööriistu ja tehnoloogiaid, ning lõpuks luuakse ka töötav lahendus.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 46 leheküljel, 7 peatükki, 12 joonist, 2 tabelit.

Abstract

A Ridesharing Platform in Estonia

The aim of this final thesis is to create a usable web application that would help to connect drivers with people who want to travel the same route. Additional goals of this final thesis are to design the web application so that it can be used both, on a mobile phone and a computer, and so that the completed web application is also accessible.

Before the development, the topicality of the final thesis is substantiated through research and the most important requirements are found, that would cover the needs of the created solution so that it could be used purposefully. In the course of the work, great emphasis is placed on usability and accessibility through design, which also includes the creation of a prototype.

This final thesis also deals with the necessary tools and technologies, and finally a working solution is created.

The thesis is in Estonian and contains 46 pages of text, 7 chapters, 12 figures, 2 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> - rakendusliides
API endpoint	Rakendusliidese osa, mis võtab vastu ja vastab päringutele.
backend	Tagarakendus, teenusepoolne rakendus
code-first	Lähenedamine, milles luuakse andmebaas programmi koodi järgi.
DTO	<i>Data Transfer Object</i> – andmeedastus objekt. Objekt, mis kannab andmeid erinevate protsesside vahel.
front-end	Eesrakendus, rakenduse kliendipoolne osa
JSON	<i>JavaScript Object Notation</i> – andmete formaat
JWT	<i>JSON Web Token</i> – krüpteeritud sõne, mis sisaldab endas turvalisusega seotud informatsiooni.
objekt-orienteeritud	Programmeerimise paradigma, mis keskendub andmetele läbi struktureeritud objektide. Objektid ise koosnevad andmeväljadest ja meetoditest.
REST	<i>Representational State Transfer</i> - hajussüsteemidele mõeldud arhitektuur.
SOAP	<i>Simple Object Access Protocol</i> – andmete edastamise protokoll.
SQL	<i>Structured Query Language</i> – päringukeel, mida kasutatakse relatsioonilistes andmebaasides.

Sisukord

1 Sissejuhatus	10
2 Metoodika.....	11
3 Ülevaade probleemist	12
3.1 Eksisteerivad lahendused.....	12
3.1.1 Facebooki grupid.....	12
3.1.2 Transportaal.....	14
3.1.3 Bolt, Uber, Yandex.....	15
3.2 Uue lahenduse skoop.....	15
3.3 Lahenduse finantsvõimalused.....	16
4 Loodava rakenduse analüüs.....	17
4.1 Nõuded.....	17
4.1.1 Funktsionaalsed nõuded	18
4.1.2 Mittefunktsionaalsed nõuded.....	21
4.1.3 Leitud nõuete kokkuvõte	22
4.2 Tehnoloogiate valik	23
4.2.1 Tagarakenduse tehnoloogiate valik	25
4.2.2 Eesrakenduse tehnoloogiate valik	28
4.3 Andmebaasi valik	31
4.4 Arenduskeskkondade valik.....	32
4.5 Disain.....	33
4.5.1 Kasutajatee voodiagrammid	34
4.5.2 Prototüübi loomine	37
4.5.3 Andmebaasi disain.....	43
4.6 Ahitektuur.....	45
4.7 Haldus.....	47
4.8 Analüüsi kokkuvõte.....	48
5 Arendus.....	50
6 Tulemused	53
6.1 Loodud lahendus.....	53

6.2 Võimalused edasiarendusteks.....	54
7 Kokkuvõte	56

Jooniste loetelu

Joonis 1. Klient server mudel	24
Joonis 2. Sisselogimise ja registreerumise voodiagramm	35
Joonis 3. Transpordi soovide lehe kasutajatee voodiagramm	36
Joonis 4. Transpordi andmete täitmise kasutajatee voodiagramm	36
Joonis 5. Erinevad sisestuselemendid ja nupud.....	39
Joonis 6. Sisselogimata kasutaja avalehe disain.....	40
Joonis 7. Transpordi soovi detailvaade.....	41
Joonis 8. Transpordi soovi lisamise sammud	42
Joonis 9. Lehe "Minu plaan" põhikomponent arvuti vaates	42
Joonis 10. Mobiili vaated	43
Joonis 11. Lihtsustatud olemi-suhte diagramm	45
Joonis 12. Arhitektuur	47

Tabelite loetelu

Tabel 1. Suurimad Facebooki grupid koos keskmiste postituste arvuga päevas seisuga 12.03.2022	13
Tabel 2. Värvide valik	38

1 Sissejuhatus

Eesti ühistranspordi süsteem ja erasektori poolt pakutavad teenused ei ole võimelised katma kõikide inimeste transpordi vajadusi. Praegusel ebastabiilsel ajal on aina kallimaks muutumas ka igapäevased sõidud oma autoga, mis viib inimeste mõtted kokkuhoiu peale. Väiksem autode arv teedel aitab ka säästa loodust ja suurendada liiklusohutust. Erinevatel põhjustel on Eestis elavate inimeste poolt Facebooki loodud kümneid sõidujagamise grupe tuhandete liikmetega, kus otsitakse ja pakutakse transporti nii inimestele kui ka kaupadele.

Käesolevas lõputöös uuritakse ning analüüsitakse probleemi Eesti kontekstis. Olemasolevatest lahendustest uuritakse lahendusi, mis on Eestis saadaval. Probleemi lahenduseks on veebirakendus, mis võimaldaks inimestel mugavalt ja kiiresti sisestada vastavalt vajadusele oma transpordi soov või transpordi pakkumine. Lahendus annab lisaks võimaluse kasutajatel lisada oma transpordi soovid või pakkumised korduvate kuulutuste näol, mis säästab kasutajate aega samasuguste kuulutuste lisamise arvelt. Loodav veebirakendus aitab inimestel raha kokku hoida, säästa loodust, suurendada liiklusohutust läbi väiksema mootorsõidukite arvu teedel või teenida isegi lisatulu.

Kirjeldatud lahenduse puudumist täheldas käesoleva lõputöö autor mõned aastad tagasi, kui käis veel Pääskülast Lasnamäele tööle ja ühel hetkel oli vaja auto remonti viia üheks nädalaks. Ühistranspordiga kulus tööle minekuks sisuliselt 1 tund ja 30 minutit, kuna oli vaja vahepeal lihtsalt bussist maha minna, ning teist bussi oodata, seevastu autoga kulus hommikul selleks umbes 20 minutit. Taksoga oli ühe sõidu hind aga kallid, kuid sõiduaeg piirdus samuti 20 minutiga.

2 Metoodika

Käesoleva lõputöö raames kirjeldatakse kõigepealt probleemi, vaadeldakse olemasolevaid lahendusi ja tuuakse välja nende tugevused ning nõrkused. Olemasolevate lahenduste põhjal tehakse järeldused vajadusest lõputöö raames loodava lahenduse kohta. Probleemi kirjelduse ning eksisteerivate lahenduste vaatluse põhjal pakutakse välja probleemile sobiv lahendus, mis jääks käesoleva bakalaureusetöö skooopi ja võimaldaks edasiarendusi tulevikus. Lisaks on vaatluse all loodava lahenduse väljavaated oma ülalpidamiskulude katmiseks.

Lahenduse analüüsi alguses tuuakse välja peamised kasutajagrupid ja selgitatakse välja funktsionaalsed ning mitte funktsionaalsed nõuded kasutades kasutajalugusid. Kasutajalugude korrektsust ning arusaadavust hinnatakse INVEST printsiibi abil. Tehnoloogiliste vahendite analüüsis tuuakse välja peamiselt tasuta saadaolevaid tehnoloogiaid ja tehakse nende seast põhjendatud valik.

Kasutajaliidese disainis luuakse vastavalt püstitatud nõuetele kasutajateekonna diagrammid, valitakse rakendusele värvid ja luuakse prototüüp. Lisaks arvestatakse kasutajaliidese disainis kasutajakogemuse põhieesmärkidega, disaini üldprintsiipidega ja WCAG 2.1 nõuetega vähemalt tasemel A ning kasutatakse kohanduvat disaini.

Analüüsi faasis loodud disaini põhjal arendatakse lõpuks tagarakendus koos integratsiooni testidega ja eesrakendus, mida testitakse manuaalselt. Lõpus kirjeldatakse edasiarenduse võimalusi ning hinnatakse tehtud tööd.

3 Ülevaade probleemist

Sõidujagamise peamine idee on viia kokku autojuhid inimestega, kes soovivad liikuda samal teekonnal. Sõidujagamine sai alguses Teise Maailmasõja ajal Ameerika Ühendriikides, kui võimude poolt tulid soovitusel julgestamaks autoga tööle sõitjaid jagama sõitu naabruskonnas elavate inimestega säästmaks naftatooteid sõjatööstusele. Sõidujagamise populaarsus tõusis uuesti 1970ndatel, kui Ameerika Ühendriikides oli nafta kriis. Nende kriiside ajal loodi mitmeid klubisid sõitude jagamiseks [1].

Tänapäeva sõidujagamine läbi mobiili- ja veebirakenduste sai alguse 2010. aasta paiku kui tekkisid sellised ettevõtted nagu Uber, Lyft ja Sidecar [2]. Peamine probleem seoses sõidujagamisega Eestis on kasutatava rakenduse, mis oleks suunatud just sõidujagamisele, puudumine. Käesolevas peatükis uuritakse võimalusi sõidujagamiseks Eestis ja selgitatakse välja, millist võimalust inimesed kõige rohkem kasutavad sõidujagamiseks.

3.1 Eksisteerivad lahendused

Olemasolevatest lahendustest vaadeldi käesolevas lõputöös kavandatava lahendusega kõige sarnasemaid lahendusi, mida lõputöö autor suutis leida. Otsingu tulemustena leiti lisaks mitmeid viiteid sarnastele lahendustele, mida enam ei eksisteeri ja need jäeti kõrvale, sest neid lahendusi ei olnud võimalik järele proovida.

3.1.1 Facebooki grupid

Eestlased on loonud kümneid Facebooki grupe, mis viivad transpordivajadustega inimesed kokku transpordi pakkujatega. Sellest johtuvalt tegi käesoleva lõputöö autor uurimustöö Facebooki gruppides, et teada saada, kui suur on inimeste huvi sõidujagamise vastu.

Uurimustööd tehti vaatlusena, milles selgitati välja grupi liikmete arv ja keskmine postituste arv päevas. Uurimustööst jäeti kõrvale väiksemad Facebooki grupid ja lisaks jäeti kõrvale vene keelsed grupid. Vene keelsed grupid jäeti kõrvale põhjusel, et

käesoleva lõputöö autori vene keele oskus pole piisavalt hea hindamaks venekeelsete postituste sisu. Vaatluse all oli ühe nädalane ajalõik. Grupi liikmete ja postituste arvu näitab Facebook ka ise grupi infos ümardatuna. Facebooki poolt näidatud liikmete arv on kantud otse uurimistöö tulemustesse (Tabel 1), kuid postituste arvus oli käesoleva lõputöö autoril vaja veenduda. Veenduda oli vaja postituste sisus ja asjakohases, selleks oli vajalik postituste sisuga tutvuda. Suuremates gruppides olid postitused grupeeritud päevade või nädala kaupa, mille alla siis kasutajad lisasid kommentaaridena oma sõidu soove või pakkumisi. Sellistel puhkudel vaadeldi kommentaare ja arvestati need tulemuste hulka. Tulemused reastati gruppide liikmete arvu järgi.

Tehtud uurimustööst nähtub (Tabel 1), et nädala keskmine sõidu soovide ja pakkumiste arv päevas ületab 90 piiri. Vaatluse all olevaid gruppe analüüsid nähtub, et enamus suuremaid gruppe on seotud suuremate Eesti linnade vaheliste suundadega.

Tabel 1. Suurimad Facebooki grupid koos keskmiste postituste arvuga päevas seisuga 12.03.2022

Facebooki grupp	Liikmete arv	Postitusi päevas
Tartu - Tallinn - Tartu	41000	14,0
Kained juhid Tallinnas "CARLES" (rideshare)	26000	10,0
Pärnu-Tallinn-Pärnu	21000	7,1
Tallinn-Tartu-Tallinn	14000	3,0
Sõidud sõbralt sõbrale	14000	3,0
Kained autojuhid Tartus/Tartumaal	14000	10,0
Kuressaare-Tallinn-Kuressaare	12000	5,0
Transport Viljandi <---> Tallinn/ Tartu	11000	0,7
Kained juhid tallinnas ja harjumaal. Kiired ja Soodsad.	11000	8,0
По пути (Эстония), üle Eesti liiklejad	6100	4,4
Transport/väikeveod/Eestis	4600	6,0
Tallinn-Pärnu-Tallinn autokohtade jagamine	4400	1,4
Tartu-Viljandi-Tartu	4300	0,6
TRANSPORT -pakun/ otsin/ transporditeenus/ veoteenus/ kolimisteenus/taksoteenus	3600	5,0
Metsatehnika veod, rasketehnika veod, treileri veod.	2800	3,0
Sõidukite transport	2500	5,0
Paide-Tallinn-Paide	2400	1,4

Facebooki grupp	Liikmete arv	Postitusi päevas
Transport Eestis Ja Euroopas	1800	5,0
Transport Tallinn-Tartu-Võru jne	1200	0,7
Transport üle Eesti	940	2,0
Kokku:		96,3

Facebooki gruppide suurimaks plussiks on kohene suhtlemisvõimalus läbi Facebook Messenger rakenduse, suur kasutajaskond ning lihtne reklaamimine.

Facebooki gruppide halvad küljed sõidujagamise probleemi lahendamiseks on kindla struktuuri puudumine postitustes. Samuti ei ole gruppides postitustel ka kindlat järjekorda ning postitusi ei saa filtreerida, mistõttu võib tekkida probleeme sama postituse uuesti üles leidmisega. Kuna Facebook ise ei eemalda postitusi gruppidest, siis võivad ajakohased postitused ka segamini olla vanade postitustega, mille sõidukuupäev on möödas. Suuremates Facebooki gruppides on kasutatud postituste grupeerimist nii, et grupi administraator teeb ühe postituse kas ühe päeva või nädala jaoks, kuhu alla siis kasutajad panevad hoopis kommentaare oma soovide ja pakkumistega. Ühest küljest on selline lähenemine mõistlik, et hoida mingisugust korrapärasust, teisalt see limiteerib kasutajate võimalusi kui näiteks tahetakse postitada sellise kuupäevaga sõitu, mida valikus ei ole. Suur gruppide arv Facebookis raskendab kasutajal sõidu otsinguid, kui kasutajal on vaja liikuda erinevatesse kohtadesse.

3.1.2 Transportaal

Transportaal on keskkond, mis peamiselt pakub transpordi teenuste pakkujatele võimalusi lisada oma kuulutus nende keskkonda. Lisaks on antud keskkonnas ka foorum, milles on transpordi otsimise ja pakkumise teemad. Antud foorumis on ainult mõned postitused millest viimased on tehtud 2017. aastal [3].

Transportaal keskkonna foorumi osa haakubki kõige rohkem käesolevas bakalaureusetöös lahendatava probleemiga. Selline foorum küll tagab info jagamise transpordi pakkumisteks ja transpordi otsimiseks, kuid foorumi postitus ise on struktureerimata. Lisaks puudub foorumis otsingu võimalus. Struktureerimata andmete puhul oleks väga keerukas teha korralikku otsingut. Antud veebirakendusega tutvudes

hakkab silma rakenduse aegunud kasutajaliidese disain ning lähemal uurimisel selgub, et rakendus ei ole aktiivses kasutuses. Seda võib järeltada nii viimaste foorumi postituste kuupäevade järgi, kui ka mitte laadiva kaardirakenduse järgi.

3.1.3 Bolt, Uber, Yandex

Nii Bolt, Uber kui ka Yandex on Eestis ühed suurimad sõidujagamise stiilis taksoteenuseid pakkuvad ettevõtted. Neil on olemas korralikud ja funktsionaalsed rakendused ning suur klientuur [4], [5], [6].

Nende rakenduste puhul on võimalus leida kiirelt transporti ning ka oma sõitude pealt teenida, kuid Eestis on need rakendused kasutatavad vaid suuremates linnades ja on pigem mõeldud lühemate sõitude jaoks linnasiseselt. Linnadevahelised pikemad sõidud võivad maksma minna sadu euroseid. Lisaks puudub nendes rakendustes võimalus luua korduvaid sõite, mistõttu näiteks iga päev rakendust kasutav isik peab alati uuesti hakkama omale sõitu otsima. Samuti sõidetakse üldiselt sõiduautoga, ning puudub võimalus täpsustada kui tahetakse transportida saadetist. Pakiveo võimalus on Yandexil küll olemas, kuid see on saadaval vaid Tallinnas [6].

3.2 Uue lahenduse skoop

Olemasolevatest lahendustest nähtub, et Eestis ei ole hetkel mitte ühtegi rakendust, mis täielikult kataks lõputöös välja toodud probleemi lahenduse mugavalt ja kasutatavalt. Puudub keskkond, kus inimesed saaksid hea ülevaate alternatiivsetest transporti soovidest ning pakkumistest sõidujagamise näol. Lisaks olemasolevatel lahendustel puudub võimalus koostada korduvaid sõite.

Kasutatavate ning ülevaatlike lahenduste puudumise tõttu pakub käesoleva lõputöö autor välja uue veebirakenduse, mis viiks kokku sõidu soovijad ning inimesed, kes sooviksid oma sõite jagada. Esialgne lahendus keskendub Eestis elavatele sõidujagamise huvilistele, kuid ei piira ka väljaspool Eestit elavatel inimestel rakenduse kasutamist. Rakendus peaks pakkuma kasutajatele võimalust luua omale konto ning kirjeldada ära oma kontaktandmed. Rakenduse peamine eesmärk on sõidupakkumiste ning sõidusoovide kuulutused mida kasutajad saavad lisada. Kuna kirjeldatud kuulutused on sisult erineva eesmärgiga, siis peaksid need olema erinevatel lehtedel ja selgelt eristatavad. Korduvate sõitude või sõidusoovide probleemi lahendamiseks peaks

keskkond olema võimeline pakkuma kasutajatele korduvate sõitude loomise võimalust, et kasutajad ei peaks korduvate sõitude puhul hakkama uuesti sisestama andmeid uue kuulutuse jaoks.

Lõputöös valmib veebirakendus, mis peab olema kasutatav nii mobiililt, arvutist ja tahvelarvutist. Kuna käesolevas lõputöös valmivat rakendust on plaanis peale lõputöö valmimist edasi arendada, siis peab arvestama tulevaste võimalike edasiarendustega, sealhulgas ka mobiilirakenduse loomisega, mistõttu tuleb eesrakendus ja tagarakendus eraldada kasutades klient-server mudelit, milles on eesrakendus tagarakendusest eraldatud.

3.3 Lahenduse finantsvõimalused

Käesolevas lõputöös loodava veebirakenduse eesmärk ei ole raha teenimine. Kasutajatelt ei kavatseta hakata küsima raha käesoleva lõputöö skoobis arendatava rakenduse kasutamise eest.

Kuigi loodava rakenduse eesmärk ei ole raha teenimine, peab lõputöö autor vajalikuks kirjeldada võimalusi rakenduse elushoidmiseks tulenevalt loodava rakenduse domeeni ning majutuse tasudest. Majutuskulude suurusjärg sõltub rakenduse populaarsusest, kuid minimaalselt võiks arvestada loodava rakenduse kuludeks aastas umbkaudu kahesaja eurose väljaminekuga.

Peamised võimalused loodava lahenduse ülalpidamiseks:

- Reklaami pakkumine – rakenduse pinna pakkumine reklaamidele. Võimalus müüa pinda otse ettevõtetele või kasutada Google Adsense platvormi [7].
- Rakendusesisesed ostud – kasutajale anda võimalus kuvada lehte reklaamivabalt või kuvada kasutaja kuulutusi teiste kuulutustest eespool tasu eest.
- Transpordialased reklaamid – luua rakendusele lisa leht, millele saavad reklaami osta näiteks transpordi valdkonnas tegutsevad ettevõtted. Sellise variandi puhul ei jää tüütud reklaamid kasutajatele ette, kui nad ei lähe sellele lehele. Lisaks annab kasutajatele võimaluse leida omaile transporti läbi teiste teenusepakkujate.

4 Loodava rakenduse analüüs

Käesolevas peatükis leitakse loodavale lahendusele nõuded, millele peab loodav lahendus ka valmimisel vastama. Nõuete leidmine aitab arendust kiirendada, sest on täpselt teada mida teha on vaja. Arenduse lihtsustamiseks ja parima lahenduse leidmiseks analüüsitakse käesolevas peatükis veel lahenduse disaini erinevatest aspektidest. Lisaks leitakse arendamiseks vajalikud vahendid ja tehnoloogiad.

4.1 Nõuded

Uue lahenduse nõuded on koostatud kasutajalugude abiga ja on jagatud funktsionaalseteks ning mittefunktsionaalseteks nõueteks. Kasutajalugu on väiksem mõõdetav tööühik agiilses raamistikus, mis kirjeldab rakenduse kasutaja mingisugust soovitud eesmärki. Kasutajaloo fookuses on kasutaja ning tema vajadus. [8]

Nõuete koostamisel hindas käesoleva lõputöö autor kasutajalugusid INVEST meetodiga. INVEST on akronüüm inglise keelsetest sõnadest [9]:

- Independent – iseseisev. Kasutajalugu peaks olema võimalikult eraldiseisev, sõltumatu.
- Negotiable – läbiräägitav või edasiantav. Kasutajalugu ei tohiks tekitada küsimusi vaid diskussiooni.
- Valuable – väärtuslik, väärtust andev. Kasutajalugu peab andma rakendusele ärilist väärtust.
- Estimable – hinnanguline. Kasutajaloole peab olema võimalik anda hinnangut arenduseks kuluva aja suhtes.
- Small – väike. Piisavalt väike, et jõuaks ettemääratud arendustsükli jooksul valmis teha.

- Testable – testitav. Kasutajalugu peab andma selge ettekujutuse, et mida ning kuidas testida.

Käesoleva lõputöö skoobi põhjal on koostatud järgnevates alapeatükkides välja toodud funktsionaalsed ja mittefunktsionaalsed nõuded. Nõuded on esitatud nimekirjana koos INVEST hinnangutega, milles kasutajalood on esile toodud rasvases kirjas. Lisaks on kokkuvõtlikult lisatud nõuded tabelisse (Lisa 2).

Kuna loodav lahendus on suunatud just transpordi otsijatele ning ka transpordi pakkujatele, siis on rakenduse põhirolliks kasutaja, kelle all peetakse silmas nii transpordi soovijat kui ka transpordi pakkujat. Teiseks rolliks on rakenduse haldaja, kes tegeleb rakenduse koodihalduse, majutuse ja kasutajate küsimustega.

4.1.1 Funktsionaalsed nõuded

Funktsionaalsete nõuete all peetakse silmas rakenduse kasutaja tegevusi sisaldavaid nõudeid. Funktsionaalsete nõuete abil saab hea ülevaate loodavasse rakendusse planeeritavatest funktsioonidest.

Kasutaja funktsionaalsed nõuded:

- **Kasutajana soovin registreerimisvormi täites registreeruda rakenduse kasutajaks.** – Täidab kõik INVEST printsiibis välja toodud nõuded. Kirjeldatud nõue sõltumatu ja ka läbiräägitav. Arutleda saab näiteks selle üle, millised väljad on registreerimise vormil kohustuslikud ja kuidas toimub valideerimine. Kasutaja saab väärtust isikliku konto näol millega saab rakendust kasutada. Lisaks on võimalik anda hinnangut kasutajaloo realiseerimiseks ja on ka piisavalt väike. Testimisel on võimalik veenduda kas konto on loodud.
- **Kasutajana soovin läbi sisselogimise vormi rakendusse sisse logida.** – Ei täida iseseisvuse nõuet, sest eelnevalt on kasutajal vaja luua, ehk registreerida konto. Väärtus seisneb selles, et kasutajad saavad sooritada rakenduse peamisi tegevusi sisse logituna. Arendusele minevat aega on võimalik hinnata tänu nõudes oleva täpsustusele. Sisselogimise vormi kasutuse täpsustus teeb käesoleva nõude ka piisavalt väikeseks. Testimisel tuleks veenduda, et kasutaja on sisse logitud.

- **Kasutajana soovin rakendusest välja logida.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreeruda ja ka sisse logida. Väärtus seisneb turvalisuses. On pigem väike ülesanne, mille tõttu on ka kerge hinnata valmimiseks kuluvat aega ja lihtne testida.
- **Kasutajana soovin enda kasutajakonto andmeid muuta.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreeruda ja sisse logida. Väärtus on kasutaja jaoks suur, kui kasutaja kontaktandmed peaksid muutuma. Käesolev nõue on piisavalt väike, et aru saada valmimise ajakulust ja testimise võimalustest.
- **Kasutajana soovin kustutada enda kasutajakonto.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida konto ja sisse logida. Väärtus seisneb kasutaja privaatsuse soovides, kui ei soovita enam rakendust kasutada. Väike ja konkreetne kasutajalugu, mille tõttu on kerge hinnata funktsionaalsuse valmimise ajakulu ja testida.
- **Kasutajana soovin sisestada transpordi otsimise kuulutusi.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida ja sisse logida. Arutluse võimalused on olemas ja väärtus on suur, sest see on rakenduse üks põhitegevustest. Lisaks on nõue konkreetne, mis määrab nõudele piisava suuruse ja aitab ajakulu hinnata. Testimisel on vaja veenduda, et kuulutus ka salvestuks ja oleks liigendatud transpordi otsimiseks. Testimine ei eelda vaatamise võimalust, sest sisestamise edukust saab kontrollida ka andmebaasist.
- **Kasutajana soovin sisestada transpordi pakkumise kuulutusi.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida ja sisse logida. Arutluse võimalused on olemas ja väärtus on suur, sest see on rakenduse üks põhitegevustest. Nõudes olev täpsustus aitab aru saada täpsemast tegevusest ja aitab hinnata ajakulu. Testimisel on vaja veenduda kuulutuse salvestumisel.
- **Kasutajana soovin kuulutuse sisestamisel luua korduvat kuulutust.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida ja sisse logida. On eelnevalt mainitud kuulutuste sisestamise täpsustus nii soovi kui ka pakkumiste kuulutustele. On piisavalt väike, et aru saada ajakulus ja testimise võimalustest.

- **Kasutajana soovin muuta enda kuulutusi.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida, sisse logida ning kuulutus sisestada. Väärtust annab kasutajale valede andmete parandamise võimaluse ja kuulutuste vaatajatele õige informatsiooni esitamisenä.
- **Kasutajana soovin kustutada enda kuulutusi.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida, sisse logida ning kuulutus sisestada. Väärtus seisneb kasutajate jaoks asjakohaste kuulutuste näol rakenduses.
- **Kasutajana soovin vaadata kuulutusi.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida, sisse logida ning kuulutusi peab olema võimalik ka eelnevalt sisestada. Väärtus seisneb selles, et see on rakenduse üks põhitegevustest.
- **Kasutajana soovin otsida kuulutusi alguskoha ja sihtkoha järgi.** – Ei täida iseseisvuse nõuet, sest eelnevalt peab olema võimalik kuulutusi sisestada. Väärtus seisneb kasutaja mugavusel ja kuulutuste otsimise ajakulus.
- **Kasutajana soovin otsida kuulutusi sõidu kuupäeva järgi.** – Ei täida iseseisvuse nõuet, sest eelnevalt peab olema võimalik kuulutusi sisestada. Väärtus seisneb kasutaja mugavusel ja kuulutuste otsimise ajakulus.
- **Kasutajana soovin anda oma huvist teise kuulutuse vastu märku avaldades soovi või pakkudes transporti.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida, sisse logida ning kuulutusi peab olema võimalik ka eelnevalt sisestada ja ka vaadata. Väärtus seisneb kasutajamugavusel läbi rakenduse broneerida koht või leida sõidukaaslane.
- **Kasutajana soovin tühistada oma registreeringuid teiste kuulutustele.** – Ei täida iseseisvuse nõuet, sest eelnevalt on vaja registreerida, sisse logida ning kuulutusi peab olema võimalik ka eelnevalt sisestada, vaadata ja ka registreeruda.
- **Kasutajana soovin klaviatuuriga rakenduses navigeerida.** – Ei täida iseseisvuse nõuet, sest teiste funktsionaalsete nõuete puhul peab selle nõudega arvestama. Teine võimalus on arendada klaviatuuriga liikumise funktsionaalsus peale kõiki teisi funktsionaalseid nõudeid. Käesoleva nõude suurust ja ajakulu on

võimalik hinnata läbi teiste funktsionaalsete nõuete. Testides peaks veenduma, et on saab ilma hiireta rakendust kasutada.

- **Kasutajana soovin lihtsasti leida abi informatsiooni rakenduse kohta.** – Täidab kõik nõuded, isegi kui haldaja ei saa hallata veel korduvate küsimuste lehte, siis on võimalik esialgu abiinformatsioon sisestada otse koodi. Väärtus seisneb nii kasutajale kui ka haldajale – kasutaja saab peamistele küsimustele otse vastuseid ja haldaja saab sellevõrra vähem küsimusi kasutajatelt.

Haldaja funktsionaalsed nõuded:

- **Haldajana soovin, et kuulutuste detailinfole pääsevad ligi ainult sisselogitud kasutajad.** – Iseseisev ei ole, sest kuulutuste detailinfo lehe loomisel peab arvestama selle nõudega. Väärtus seisneb kasutajaandmete privaatsuses.
- **Haldajana soovin hallata korduvate küsimuste lehte.** – Iseseisev ei ole, sest eelnevalt on vaja sisse logida.

4.1.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded ei sisalda endas rakenduse kasutaja poolt tehtavaid tegevusi. Mittefunktsionaalsed nõuded annavad hea ülevaate rakenduse üldisemast olemusest ning erinevatest omadustest.

Kasutaja mittefunktsionaalsed nõuded:

- **Kasutajana soovin lihtsasti eristada erinevat tüüpi kuulutusi.** – Ei ole iseseisev, sest kuulub kokku kasutajale kuulutuste kuvamisega. Väärtus seisneb loodava lahenduse paremas arusaadavuses kasutajatele.
- **Kasutajana soovin, et minu kontaktandmed ei ole kuulutuses mittekasutajatele nähtavad.** – Ei ole iseseisev, sest eelnevalt ei saa testida, kui kuulutusi on võimalik kuvada. Annab väärtust kasutaja privaatsusele.
- **Kasutajana soovin kasutada veebirakendust arvutist.** – Täidab kõik INVEST nõuded. Annab väärtust, kuna ei limiteeri kasutaja võimalusi rakendust kasutada.

- **Kasutajana soovin kasutada veebirakendust tahvelarvutist.** – Täidab kõik INVEST nõuded. Annab väärtust, kuna ei limiteeri kasutaja võimalusi rakendust kasutada.
- **Kasutajana soovin kasutada veebirakendust nutitelefonist.** – Täidab kõik INVEST nõuded. Annab väärtust, kuna ei limiteeri kasutaja võimalusi rakendust kasutada.
- **Kasutajana soovin, et veebirakendus oleks eestikeelne.** – Ei täida iseseisvuse nõuet, sest iga arendus peab sellega arvestama. Annab väärtust eesti keelest aru saavatele kasutajatele.

Haldaja mittefunktsionaalsed nõuded:

- **Haldajana soovin, et rakenduse kood oleks dokumenteeritud.** – Ei ole iseseisev, sest iga arenduse juures peab arvestama dokumentatsiooni loomise või ajakohasena hoidmisega. Lisaks on raske ennustada rakenduse koodibaasi suurust, mistõttu ei ole sellele lihtne hinnangut anda.

4.1.3 Leitud nõuete kokkuvõte

Nõudeid analüüsid nähtub, et enamik nõuded ei täida iseseisvuse printsiipi INVEST printsiipide kogumist. Enamasti on iseseisvuse printsiibi mitte täitmine seotud kasutaja registreerimise ja sisse logimisega, sest rakenduse põhitegevusi saab sooritada ainult sisse loginud kasutaja. Sisse logimata kasutajal piiratakse tegevusi, sest tema tegevusi ei ole võimalik andmebaasi salvestada selliselt, et ta hiljem oma tegevusi ka jätkata saaks. Lisaks ei leiaks kasutajad sisse logimata kasutaja kontaktandmeid kuulutustelt, kui neil peaks olema täiendavaid küsimusi.

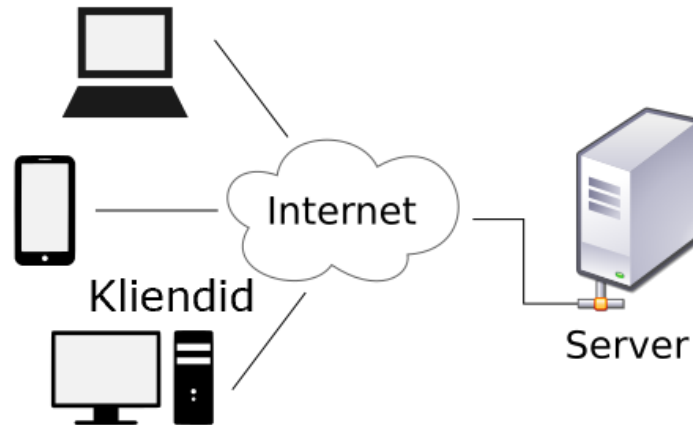
Teiseks peamiseks iseseisvuse printsiibi mittetäitmise põhjuseks on tükeldatud kasutajalood, mida saaks tegelikult ka üldistada kokku üheks kasutajalooks. Üldistamisega on aga oht, et kasutajalugu tekitab liiga palju küsimusi, ja ei ole piisavalt konkreetne. Kasutajaloo väiksemateks tükideks tegemine annab kindlamad testi stsenaariumid ja lisaks on arendamise ajal ka väga konkreetne ülesanne, mida on vaja teha.

Kasutajalugudest lähtuvalt on vaja arendada kõige esimesena registreerimise ja sisselogimise funktsionaalsused. Lisaks on vaja arendada enne kuulutuste lisamise funktsionaalsused, kui kuulutuste muutmise funktsionaalsused.

4.2 Tehnoloogiate valik

Käesoleva lõputöö üheks eesmärgiks oli arvestada tuleviku arendustega. Põhjusel, et üks võimalik tulevikuarendus on mobiilirakenduse loomine, siis on vajalik sellele planeerimise faasis, sealhulgas ka tehnoloogiate valikul, ka mõelda. Lahenduseks on eraldada eesrakendus tagarakendusest. Eesrakenduse eraldamine tagarakendusest eraldab vaated, mida kasutaja näeb, täielikult ärioloogikast ning võimaldab neid eraldiseisvalt arendada. Lisaks annab selline lähenemine suuremat paindlikkust nii programmeerimiskeelte kui ka raamistike valikul. Eesrakenduse tagarakendusest eraldamise miinuspoolena võib välja tuua lisatöö tegemise andmete liigutamisel. See väljendub selles, et nii kliendi kui ka teenuse pool on vaja teha andmetele lisatöötusi. Näiteks kliendi poolses rakenduses on vajadus uuesti defineerida kõik andmeobjektid, mis tegelikkuses juba teenuse pool on defineeritud.

Eesrakenduse eraldamise tagarakendusest, ehk klient-server mudelis (Joonis 1) suhtleb eesrakendus tagarakendusega läbi API ehk rakendusliidese [10]. Eesrakendus ja tagarakenduse suhtlust läbi API tehakse tänapäeval peamiselt läbi REST API. REST ehk *representational state transfer* on arhitektuuriliste piirangute kogum, mida arendajad saavad implementeerida erinevatel viisidel [11], [12]. Peamiselt liigutatakse andmeid kliendi ja teenuse vahel JSON ehk *JavaScript Object Notation* formaadis, kuid on võimalik ka teised formaadid, kaasa arvatud lihttekst. JSON on andmeformaadi standard [11].



Joonis 1. Klient server mudel [11]

Kuna käesoleva lõputöö autor soovib kasutada laialt levinud turvalisi tehnoloogiaid siis on kindlasti REST API kasutamine JSON formaadis andmete liigutamisel mõistlik valik. Lisaks on see väga hästi toetud laialt levinud programmeerimiskeelte ning raamistike hulgas ning seda on kerge kasutada.

Alternatiiviks oleks kasutada SOAP ehk *Simple Object Access Protocol* protokoll, kuid SOAP protokoll on keerukam, ning seda oleks mõistlikum kasutada keerukamates andmesüsteemides, nagu näiteks finantsteenustes, kus tulevad esile SOAP protokoll plussid [13], [14].

Tehnoloogiate valikul lähtub käesoleva lõputöö autor oma kogemusest, tehnoloogia õppimiskurvist, tehnoloogia kohta saadaolevast dokumentatsioonist ning ka lisakuludest, mida vaatluse all olevad tehnoloogiaid võivad tekitada. Lisaks hinnatakse tehnoloogia turvalisust ning tehnoloogia populaarsust.

Isiklikku kogemust, õppimiskurvi ning dokumenteeritust on vajalik arvestada tehnoloogia valikul järgnevatel põhjustel:

- Arendusvigade minimeerimiseks. Arendaja poolt uus või vähe kasutatud tehnoloogia võib tekitada segadust või valesti mõistmist ja sellest tulenevalt tekitada vigu.

- Valitud tehnoloogia poolt pakutavate võimaluste paremaks ära kasutamiseks. Kui arendaja jaoks on tehnoloogia võõras, võib arendaja teadmatusest või kehva dokumentatsiooni tulemusena teha lisatööd.
- Käesoleva lõputööga ajagraafikus püsimiseks. Võõra või vähem kasutatud tehnoloogia puhul tuleb kindlasti arvestada lisa ajakuluga. Lisaks kehva dokumentatsioon võib raisata arendaja aega ja põhjustada tehnoloogia vahetust arenduse ajal.

Kuna käesoleva lõputöö peamine eesmärk on luua inimestele alternatiivne viis transpordi otsimiseks või pakkumiseks, ning raha selle eest küsida ei ole plaanis, siis on tähtsal kohal tehnoloogia maksumus.

Tehnoloogia turvalisus on tähtis alati, kuid kuna loodavas lahenduses on kasutajatel vajalik kasutajakontode olemasolu, siis on kasutajate andmete kaitse aspektist tehnoloogia turvalisus ülimalt tähtis.

Populaarsuse aspekt on tähtis, sest sellest kumab läbi valitud tehnoloogia tuleviku perspektiiv. Vähem populaarsemate tehnoloogiate arendamised võidakse seisma panna ning vähem populaarsemate tehnoloogiate puhul on raskem saada abi või leida näiteid internetist ehk siis populaarsematel tehnoloogiatel on suuremad kogukonnad, kust võib probleemide korral abi saata.

4.2.1 Tagarakenduse tehnoloogiate valik

Tagarakendus ehk *backend* on teenus, mis hoolitseb andmetele ligipääsu ja äri loogika eest [15]. Selleks peab tagarakendus olema suuteline vastu võtma päringuid, päringutest saadud andmeid töötleva ning olema võimeline ka andmebaasiga suhtlema. Käesolevas alapeatükis on vaatluse all tehnoloogiad, mis aitavad realiseerida tagarakenduse eesmärke.

Suures pildis võib tagarakenduse tehnoloogiate valiku jagada tagarakenduses kasutatava programmeerimise keele ja raamistiku valikuks. Tehniliselt on võimalik ka raamistik kõrvale jätta, ja ainult programmeerimiskeelega tagarakendus valmis programmeerida, kuid selline lähenemine ei ole mõistlik, sest raamistikud annavad arendajale võimaluse kasutada juba kellegi poolt valmis tehtud tükke. Nendeks tükkideks võivad olla näiteks kasutaja autoriseerimine, andmebaasipäringud, andmete validatsioon või kasutajate

haldus. Lisaks aitavad raamistikud seada arendatavale rakendusele kindla struktuuri. Raamistike kasutamine tõstab turvalisust ning aitab eesmärki saavutada kiiremini [16]. Sellest tulenevalt on vajalik välja selgitada nii kasutatav programmeerimise keel, kui ka raamistik, mis sobiks valitud programmeerimise keelega.

Tagarakenduse tehnoloogiad:

- C# programmeerimiskeel – on *objekt-orienteeritud* programmeerimiskeel, mida saab kasutada .NET raamistikus. C# ja .NET raamistik on turvaline valik ja ka kogukond on suur. .NET raamistikul on ka hea valik teeke, mis aitavad arendust veelgi kiirendada. .NET raamistikule lisaks peaks veebiarenduseks kasutama ASP.NET raamistikku. ASP.NET on .NET raamistiku osa ja on veebiarenduse raamistik. Lisaks on dokumentatsioon väga heal tasemel - peale dokumentatsiooni leidub dokumentatsiooni juurde ka erinevaid veebikursuseid õppimiseks [17]. Käesoleva lõputöö autoril on kõige rohkem kogemust ning oskusi kirjeldatud tehnoloogiatega.
- Java – on üks populaarsemaid programmeerimise keeli maailmas ning seda kasutatakse palju erinevates valdkondades, kaasa arvatud veebiarenduses. Erinevalt C# -ist töötab Java erinevatel platvormidel [18]. Java kõige populaarsemaks raamistikuks veebiarenduses on Spring [19]. Kuna Java on üks populaarsemaid programmeerimise keeli, siis on ka Spring üks populaarsemaid raamistike veebiarenduses. Kirjeldatud tehnoloogiate dokumentatsioon on samuti väga heal tasemel [20]. Käesoleva lõputöö autoril on kogemusi nende tehnoloogiatega, kuid valides need tehnoloogiad, siis tuleks arvestada aega, mis kulub meelde tuletamiseks ning juurde õppimiseks.
- Kotlin – tugineb Java programmeerimise keelele [21]. Lisaks on võimalik kasutada Kotlinit koos Javaga, käivitades Java koodi Kotlinist või vastupidi [22]. Kotlin ise tuli välja 2016. aastal ja on suhteliselt uus kuid kõvasti populaarsust kogunud. Kotlini puhul on võimalus kasutada mitmeid raamistikke, näiteks Ktor, Vert.x või Javalin. Kuid Kotliniga saab kasutada ka Spring raamistikku, mis oleks mõistlik valik, kuna Spring on turul olnud kauem [23]. Käesoleva lõputöö autoril on kogemusi Kotliniga Androidi platvormile lihtsamate rakenduste arendamisel ja Spring raamistikuga koos Javaga, kuid koos neid tehnoloogiaid kasutanud ei

ole. Õppimiskõver ei ole kuigi suur, kuid arvestades lõputöö tähtaegasid, siis eelistab lõputöö autor omale paremini tuntud tehnoloogiaid, et tähtaegadest kinni pidada.

- Python – on samuti üks populaarsemaid programmeerimise keeli maailmas, millest tulenevalt on Pythoni arendajatel suur kogukond. Python on oma olemuselt lihtsam ja süntaksit on ka algajatel lihtsam lugeda, aru saada ja õppida. Populaarseim raamistik Pythonile on Django [24]. Django on paindlik ja lihtsasti skaleeritav raamistik [25]. API loomiseks oleks siiski mõistlik Django raamistikku kasutada koos Django Rest raamistikuga, sest Django Rest pakub API-de loomiseks sisseehitatud võimalusi, nagu näiteks andmete serialiseerimine [26]. Käesoleva lõputöö autoril on Pythoniga kogemusi, kuid need kogemused on olnud pigem algoritmilised. Django raamistikuga kogemused puuduvad.
- PHP – suure kasutajaskonnaga ja populaarne programmeerimise keel. PHP plussiks on tema lihtsus . PHP on siiski küllaltki vana tänapäevane programmeerimise keel – PHP -d tutvustati 1994. aastal. Tänapäeval enam PHP -d niivõrd ei eelistata ja tema populaarsus on langustrendis [27]. PHP kõige populaarsemaks raamistikuks on Laravel [28]. Kirjeldatud tehnoloogiatel on olemas korralik dokumentatsioon. Käesoleva lõputöö autori kogemus PHP -ga on ebapiisav ja kogemus Laraveliga puudub.
- Ruby – objekt orienteeritud programmeerimise keel. Süntaksilt on Ruby väga sarnane Pythoniga. Rubyga kasutatakse peamiselt Ruby on Rails raamistikku [29]. Käesoleva lõputöö autoril ei ole kogemust selles punktis kirjeldatud tehnoloogiatega.
- JavaScript – dünaamiline programmeerimise keel, kuid JavaScriptis on ka toetatud objekt-orienteeritud programmeerimine. JavaScripti suureks plussiks on, et teda saab kasutada nii eesrakenduse arendamiseks kui ka tagarakenduse arendamiseks. JavaScript on koos Pythoni ja Javaga samuti üks kõige populaarsemaid programmeerimise keeli maailmas. Nii JavaScripti kui ka Java kaubamärgid kuuluvad ettevõttele nimega Oracle kuid JavaScript ise aga erineb Javast ning neid ei tasuks kuidagi samastada [30]. JavaScript oli esialgu kasutuses eesrakendustes, kuid Node.js tulekuga 2009. aastal algas ka JavaScripti edulugu

tagarakendustes [31]. Node.js on keskkond serveris, mis suudab käitada JavaScripti. Üheks populaarsemaks raamistikuks tagarakenduste arendamisel JavaScriptiga on Express.js, millel on minimalistlik lähenemine [32]. Express.js on küll kiire ja paindlik, kuid erinevalt ASP.NETi või Springi raamistikust, ei ole sellel näiteks valmis ehitatud tükke autentimiseks [33]. Käesoleva lõputöö autoril on kogemusi JavaScriptiga, kuid Node.js ja Express.js on võõrad.

Turvalisuse poole pealt on Ruby on Rails ja Express.js rohkem ohustatud, sest neile arendatakse palju funktsionaalsust juurde paljude erinevate arendajate poolt pakettide näol, mida saab kasutada nimetatud raamistiketes. Nende arendajate hulka võib sattuda keegi pahatahtlik inimene halbade kavatsustega. Kuigi ka näiteks nii Spring kui ka ASP.NET raamistikele saab sarnaselt pakette arendada, siis nimetatud raamistike turvalisus on paremal tasemel läbi sisse ehitatud autoriseerimise ja autentimise süsteemide [34]. Turvalisuse osas ei tohiks jääda lootma ainult raamistikule vaid arendaja peab siiski lõpuks ise ka veenduma arendatava rakenduse turvalisuses kirjutades läbimõeldut ja turvalist koodi ning seda ka testimata.

Kõik eelpool kirjeldatud tehnoloogiad on tasuta ja nende kasutamine nii käesolevas lõputöös kui ka edasistes arendustes, ei nõuaks rahalist väljaminekut. Lisaks on neil kõigil olemas korralik dokumentatsioon ning küllaltki suured kogukonnad ja nad on laialdaselt ka kasutusel üle maailma.

Käesoleva lõputöö autor hindab enda jaoks parimaks variandiks kasutada C# ja .NET raamistikku koos ASP.NET-ga kavandatava lahenduse tagarakenduse arendamiseks. Valikus sai otsustavaks heal tasemel dokumentatsioon, kõrge turvalisus ja olemasolevad turvalahendused. Lisaks on ka käesoleva lõputöö autoril kõige rohkem kogemusi C# ja ASP.NET raamistikuga. Kogemused aitavad arendusel kulgeda sujuvamalt ja vähendab veaohlike lahenduste implementeerimist.

4.2.2 Eesrakenduse tehnoloogiate valik

Eesrakendus, ehk *front-end* on rakenduse kliendipoolne osa, mis hoolitseb kasutaja vaate eest. Eesrakenduse ülesanne on võtta vastu sisend kasutajalt ja sellele vastavalt reageerida. Põhiülesandeks on päringute tegemine tagarakendusse ja vaadete muutmine.

Eesrakenduses kasutatavad kolm peamist alustehnoloogiat on [35]:

- HTML – *HyperText Markup Language* – veebilehe aluskomponent mis defineerib veebilehe struktuuri.
- CSS – *Cascading Style Sheets* – kirjeldab veebilehe välimuse.
- JavaScript – kasutatakse veebilehele funktsionaalsuse andmiseks.

Neist on kolmest tehnoloogiast on ainult JavaScript programmeerimise keel. JavaScriptile lisaks on veel valikus TypeScript. TypeScript on tugevalt tüübitud programmeerimise keel, samas JavaScript on nõrgalt tüübitud keel. JavaScriptis ei saa defineerida andmetüüpe kuid TypeScriptis on see võimalik. Tugevalt tüübitud keeles tulevad andmetüübi vead juba koodi kirjutades välja. Lihtsustatult kirjeldades teisendatakse TypeScript käivitades lõpuks siiski JavaScriptiks. Lisaks on võimalik kutsuda välja JavaScripti koodi TypeScriptist ja vastupidi. Lisaks on võimalik JavaScripti raamistikke või teke kasutada TypeScriptiga [36], [37].

Eesrakenduse JavaScripti ja TypeScripti raamistikud ja teegid:

- Angular – keerukas ja suure õppimiskurviga TypeScripti põhine raamistik. Samas on Angular üks populaarsemaid [38] eesrakenduse raamistikke suure kogukonnaga ja hea dokumentatsiooniga [39]. Käesoleva lõputöö autoril puuduvad kogemused Angulariga.
- Vue – oma olemuselt lihtsam raamistik võrreldes Angulariga. Lisaks on Vue samuti üks populaarsemaid valikuid eesrakendustes [38]. Vuel on hea dokumentatsioon ning tänu oma populaarsusele suur kogukond [40]. Käesoleva lõputöö autoril on põgusalt kogemusi Vue raamistikuga ja hindab oma oskusi selles rahuldavaks.
- Aurelia – siinses nimekirjas kõige vähem populaarsem raamistik, kuid sellele vaatamata on Aurelia võimekas. Aurelia on oma olemuselt lihtsam kui Angular ja sellest tulenevalt ka väiksema õppimiskurviga. Aurelial on olemas küll korralik dokumentatsioon, kuid kogukonna väiksuse tõttu võib tekkida probleeme keerukamate probleemide lahendamisel [41]. Käesoleva lõputöö autoril on põgusalt kogemusi Aureliaga ning hindab oma oskusi Aureliaga madalaks.

- React – ei ole raamistik vaid on teek [42]. React on koos Vue ja Angulariga üks populaarsemaid valikuid eesrakenduste arendamise tööriistade valikul [38]. Reactil on väga hea dokumentatsioon ja suur kogukond tulenevalt tema populaarsusest. Reacti suureks eeliseks on lisaks veel see, et mobiilirakenduste arendamiseks Androidile ja IOS'ile on olemas raamistik nimega React Native, milles siis teisendatakse Reactis kirjutatud kood vastavale platvormile vajaminevaks koodiks [43]. Arvestades, et tulevikus on plaanis arendada mobiilirakendus käesolevas lõputöös loodavale veebirakendusele, siis oleks React hea valik, sest enamuse loodava eesrakenduse koodist saaks taaskasutada mobiilirakenduse arendamisel. Käesoleva lõputöö autoril on Reacti teegiga head kogemused ja autor hindab oma oskusi Reactis küllaltki heaks. Lisaks on käesoleva lõputöö autoril kogemusi ka React Native raamistikuga.

Lisaks JavaScriptile ja TypeScriptile leidub veel võimalusi eesrakenduste arendamiseks. Nimelt on võimalus kasutada Dart programmeerimise keelt koos Flutter raamistikuga. Dart koos Flutteriga on väga populaarne valik mobiilirakenduste arendamises [44], kuid neid tehnoloogiaid on võimalik kasutada ka veebiarenduses. Flutteri suureks eeliseks on tema võimalused rakenduste arendamiseks korraka veebi, Androidile ja IOS'ile. Lihtsustatult kirjeldades teisendatakse veebibrauseri jaoks Dart programmeerimise keeles kirjutatud kood lõpuks JavaScriptiks. Valides Flutteri, oleks positiivseks pooleks veel see, et kui käesolevas lõputöös planeeritava lahendusele hakata mobiilirakendust arendama, siis saaks mobiilirakenduse jaoks enamuse eesrakenduse koodi taaskasutada [45]. Veebiarenduses pole Flutter nii kiiresti populaarsust kogunud, nagu mobiilirakenduste arendustes, ning Flutter on pigem vähem populaarne veebiarenduse raamistik. Käesoleva lõputöö autori kogemused Darti ja Flutteriga piirduvad mõne lihtsa mobiilirakenduse arendamisega.

Käesolevas lõputöös valib autor omale eesrakenduse arendamiseks React teegi. Valikus said otsustavaks autori enda kogemused ja ka Reacti koodi taaskasutuse võimalus tulevikus arendatavas mobiilirakenduses. Kuigi ka Flutterit valides saab tulevikus mobiilirakenduse arendamiseks koodi ära kasutada, siis suhtub käesoleva lõputöö autor Flutterisse veebirakenduse poole pealt ettevaatlikkusega ja ei vali seda.

Eesrakenduse arendamisel on mõistlik valida ka CSS raamistik, mis aitab eesrakenduses luua kohanduva disainiga vaateid, elemente mugavalt paigutada ja hoida ühtlast stiili üle

rakenduse. Lisaks aitab CSS raamistiku kasutusele võtt aitab kiirendada arendusprotsessi [46].

Mõned populaarsemad CSS raamistikud on [47]:

- Tailwind CSS;
- Bootstrap;
- Material UI;
- Bulma;

Käesoleva lõputöö autoril on kogemusi nii Bootstrapi kui ka Material UI-ga ning valib loodavale lahendusele Material UI raamistiku, hea dokumentatsiooni tõttu. Kuigi Material UI kodulehelt nähtub ka hinnakiri, siis Material UI tuumik osa on täiesti tasuta. Lisaks on Material UI dokumentatsioonis välja toodud informatsiooni juurdepääsetavuse kohta ning lisaks on võimalus kasutada valmiskomponente, mis kiirendab arendust [48].

4.3 Andmebaasi valik

Andmebaasi valikul lähtub käesoleva lõputöö autor andmebaasi maksumusest ja andmebaasi sobivusest käesolevas lõputöös loodavasse lahendusse, lisaks peab arvestama ka turvalisust. Sellest tulenevalt on käesolevas lõputöös vaatluse all vabavaralised andmebaasid.

Andmebaasid jagunevad ise mitmetesse erinevatesse kategooriatesse ning veebirakenduse puhul tehakse valik üldiselt relatsiooniliste ja NoSQL andmebaasi tüüpide vahel. NoSQL andmebaasid on skaleeritavamad, kui relatsioonilised andmebaasid, kuid võrreldes relatsiooniliste andmebaasidega, on NoSQL andmebaasidel väiksem kasutajaskond [49] ning ka käesoleva lõputöö autoril puuduvad kogemused NoSQL andmebaasidega. Sellest tulenevalt tehakse andmebaasi valik relatsiooniliste andmebaaside seast.

Relatsioonilistes andmebaasides hoitakse andmeid tabelites, kus andmed on omavahel seotud läbi tabeli ridade. Ühe tabeli ridasid seotakse teise tabeli ridadega läbi

välisvõtmete, mille abil on võimalik leida seotud ridu. SQL – *Structured Query Language* on päringukeel, mis on kasutusel relatsioonilistes andmebaasides [50].

Järgnevalt tuuakse välja peamised vabavaralised relatsioonilised andmebaasid:

- MySQL – maailmas üks kõige populaarsematest valikutest vabavaraliste relatsiooniliste andmebaaside hulgast. Lisaks on MySQL turvaline ja ka skaleeritav ning seda saab kasutada erinevate operatsiooni süsteemidega [51].
- MariaDB – relatsiooniline andmebaas, mis on edasi arendatud MySQL andmebaasi pealt. Võrreldes MySQL andmebaasiga on MariaDB kiirem. Saab samuti kasutada mitmete erinevate operatsioonisüsteemidega [52].
- PostgreSQL - skaleeritav, turvaline ja rikkalike funktsioonidega relatsiooniline andmebaas [53]. Kuid lihtsate päringute puhul ei ole nii optimaalne võrreldes MySQL'iga [54].
- SQLite – lihtne, kergelt õpitav ning kompaktne failipõhine andmebaas. SQLite piiranguteks on vähene mahutavus ja andmetüüpide vähesus [55].

Käesoleva lõputöö autoril on siin väljatoodud andmebaasidest kõige rohkem kogemusi SQLite andmebaasiga, kuid skaleeritavuse ja SQLite andmebaasi andmetüüpide vähesuse tõttu ei valita SQLite. Käesoleva lõputöö autor valib loodavas lahenduses kasutatavaks andmebaasiks PostgreSQL-i. MySQL andmebaasi ei valita selletõttu, et see ei ole lõpuni vabavaraline, ja kuulub Oracle-le [52]. Samuti on PostgreSQL populaarsem ja suurema kogukonnaga, kui MariaDB [56].

4.4 Arenduskeskkondade valik

Arenduskeskkondade valik jaguneb eesrakenduse, tagarakenduse ja koodihalduse keskkondade valikuks. Arenduskeskkondade valikul arvestab käesoleva lõputöö autor arenduskeskkonna maksumuse, isikliku kogemusega ning arenduskeskkonna sobilikkusega eesmärgi saavutamiseks.

Kuigi arenduskeskkondade valik eesrakenduse arendamiseks on lai [57], siis valib lõputöö autor eesrakenduse arendamiseks Visual Studio Code nimelise arenduskeskkonna. Visual Studio Code on täiesti tasuta ning sellega on võimalik

arendada React raamistikul põhinevaid rakendusi. Lisaks on Visual Studio Code-le võimalik lisada erinevaid laiendeid, et muuta arendamine veelgi lihtsamaks ja sujuvamaks [58]. Käesoleva lõputöö autoril on väga head kogemused kirjeldatud arenduskeskkonnaga.

Tagarakenduse arenduskeskkondade valikul tuleb esmalt arvestada, et tagarakendust hakatakse arendama C# programmeerimise keeles ja .NET raamistikus. Nende jaoks on arenduskeskkondade valikus kaks kõige enim kasutatavat integreeritud arenduskeskkonda – Visual Studio ja Rider [59]. Mõlemad on tasulised, kuid Visual Studio -l on tasuta versioon, millega saab enamuse arendamiseks vajalikest ülesannetest täita. Visual Studio tasulised versioonid on tunduvalt kallimad, kui Rideri litsentsitasud [60], [61]. Käesoleva lõputöö autoril on kogemusi mõlemaga neist ja valib kasutatavaks arenduskeskkonnaks Visual Studio, sest sellel on olemas korralik tasuta versioon. Lisaks kui mingil põhjusel peaks vaja minema tasuta versiooni, siis saab alati arenduskeskkonna vahetada odavamaks Rideri vastu.

Koodihalduskeskkondade valik tehakse kolme populaarseima keskkonna vahel: Github, GitLab ning BitBucket [62]. Kõik kolm mainitud keskkonda on kasutajatele tasuta kättesaadavad ning oma võimalustelt üpris sarnased. Käesoleva lõputöö autoril on enim kogemusi GitHubiga ning sellest tulenevalt ka valitakse GitHub koodihalduskeskkonnaks.

4.5 Disain

Käesoleva lõputöö disaini faasis luuakse kasutajakogemuse disain koos prototüübiga ning luuakse ka andmebaasi disain. Disaini loomisel lähtutakse kasutajalugudest, mis on kirjeldatud nõuete peatükis. Kasutajakogemuse disainis luuakse kasutaja liikumise illustreerimiseks voodiagrammid, valitakse loodavale kasutajaliidesele värvid ning nende põhjal luuakse ka prototüüp. Andmebaasi disainis luuakse üldistatud andmebaasi diagramm ning ka detailsem olemi-suhte diagramm.

Kasutajakogemuse disainis on arvestatakse, et loodav lahendus täidaks kasutatavuse peamisi eesmärke, milleks on [63, pp. 14-18]:

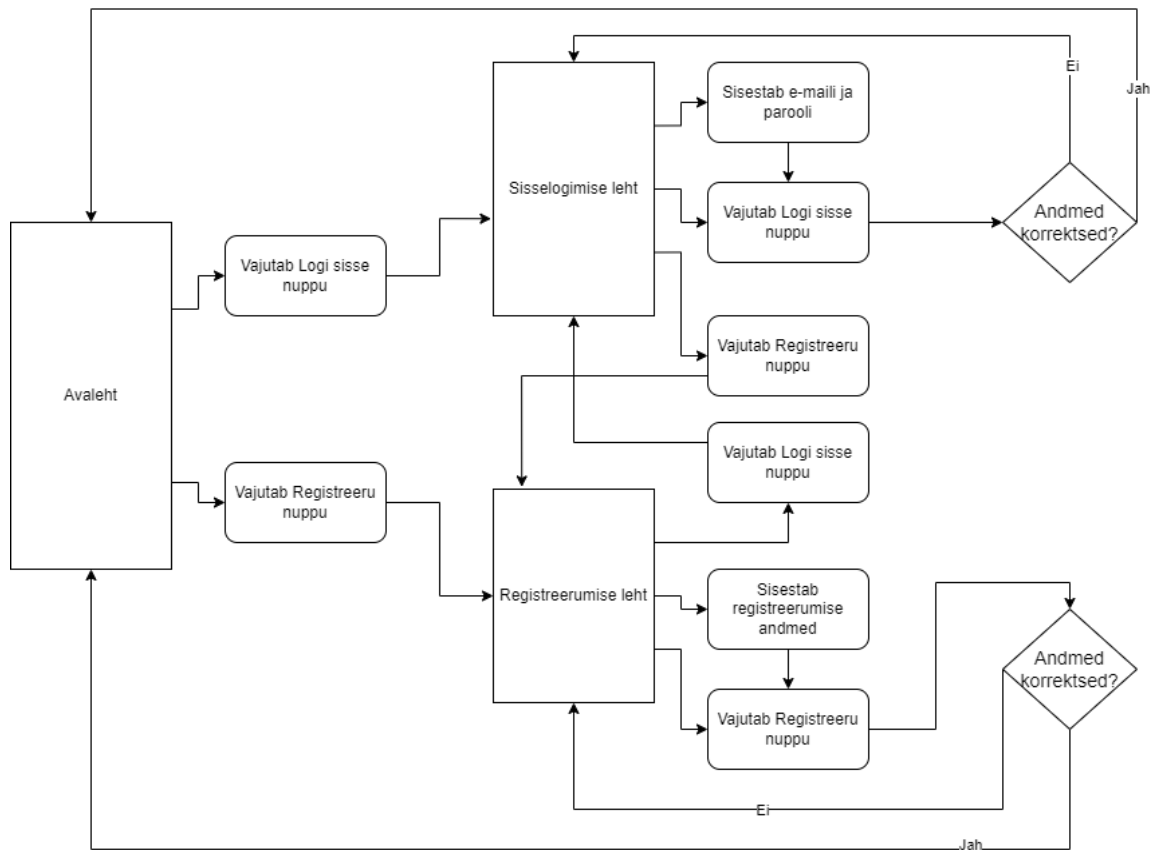
- Efektiivsus – üldine eesmärk, mis näitab kui hea on süsteem tegemaks seda, milleks see mõeldud on.

- Tõhusus – näitab, kui hea on süsteem toetades kasutajat tema tegevustes.
- Turvalisus – näitab, kui hästi süsteem suudab kaitsta kasutajat mittesoovitud ja ohtlike olukordade eest.
- Õpitavus – näitab, kui lihtsasti on võimalik süsteemi kasutama õppida.
- Meeldejäätvus – näitab, kui lihtne on süsteemi kasutamist meelde jätta, kui juba ollakse süsteemiga tutvunud.

4.5.1 Kasutajate voodiagrammid

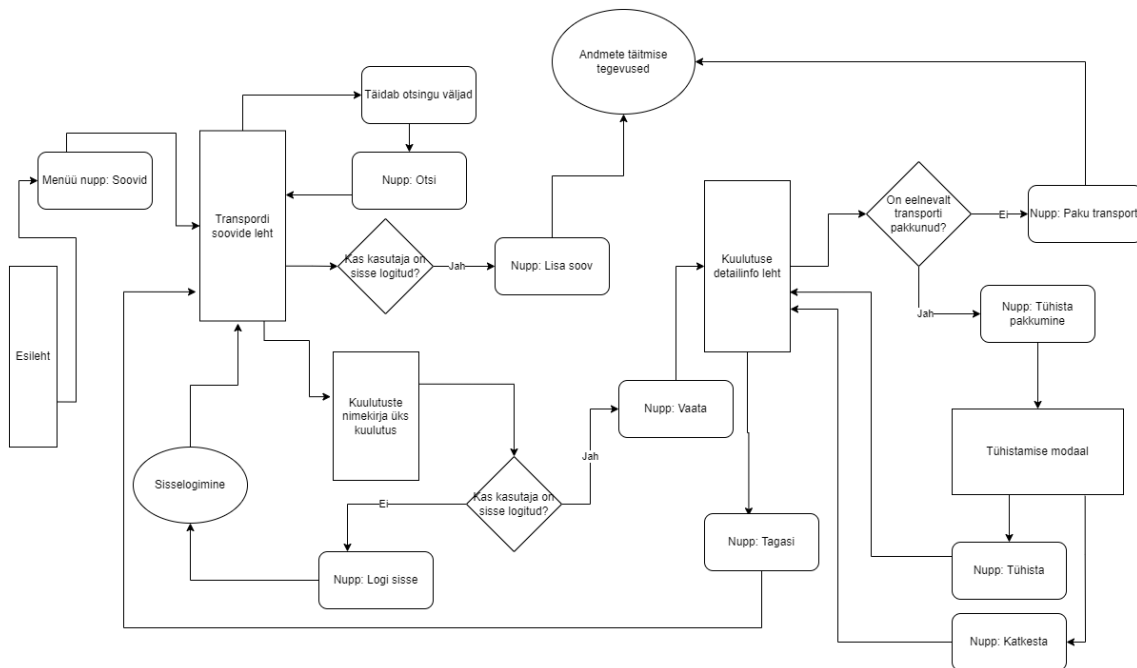
Kasutajate voodiagrammid on loodud loodava lahenduse põhitegevustele illustreerimaks kasutajate liikumist loodavas rakenduse kasutajaliideses. Peamised põhitegevused on sisselogimine ja registreerimine, kuulutuste sirvimine, enda kuulutustega seotud tegevused ning teiste kuulutustele pakkumiste tegemine. Lisaks saab kasutaja vaadata enda ajaplaani. Diagrammide loomisel on võetud arvesse eelnevalt kirjeldatud kasutatavuse eesmärgid.

Rakenduse sihipäraseks kasutamiseks on kasutajal vajalik registreeruda või sisse logida. Joonisel (Joonis 2) on ära toodud kasutaja sisselogimise ja registreerumise kasutajate voodiagramm. Sisselogimise ja registreerumise vormid viitavad üksteisele, mis aitab tõsta rakenduse kasutatavust nii tõhususe aspektist kui ka turvalisuse aspektist. Kui kasutaja on näiteks kogemata läinud registreerumise lehele, kuigi ta soovis sisse logida, siis tal on võimalus kiirelt liikuda sisse logimise lehele. Lisaks vigaste andmete sisestamise puhul ei suunata kasutajat mujale, vaid ta jääb samale vormile.



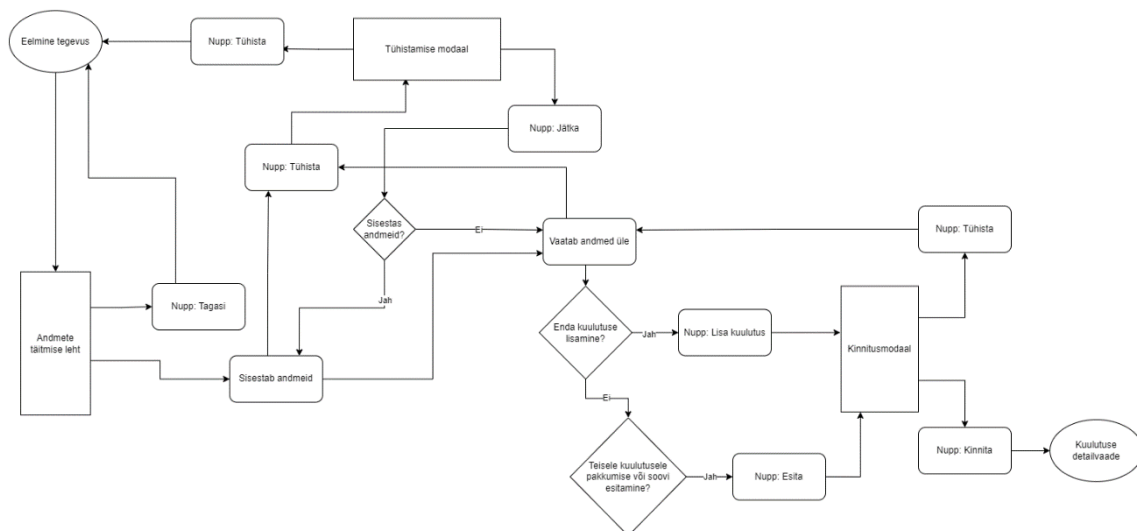
Joonis 2. Sisselogimise ja registreerumise voodiagramm

Kasutajate põhitegevus loodavas lahenduses on seotud põhiliselt kuulutustega. Kasutaja saab sisestada nii transpordi pakkumiste, kui ka transpordi soovide kuulutusi. Lisaks saavad kasutajad teha oma pakkumisi ja sooviavaldusi teiste kuulutustele, mille vastuvõtmisel nad registreeritakse sõidule. Joonisel (Joonis 3) on välja toodud loodava lahenduse transpordi soovide lehe kasutajatee voodiagramm. Joonisel (Joonis 3) olevas voodiagrammis on „Andmete täitmise tegevus“ toodud välja joonisel (Joonis 4). Kasutatavuse turvalisuse aspektist lähtudes saab esile tuua täiendavad kinnitamisest põhitegevuste eesmärkide saavutamiseks või tehtu tühistamiseks.



Joonis 3. Transpordi soovide lehe kasutajatee voodiagramm

Joonisel (Joonis 4) on näha kasutajate tegevus, kui kasutaja sisestab uut transpordi soovi või transpordi pakkumise kuulutust. Lisaks on sisuliselt samad tegevused ka kellegi teise kuulutusele pakkumise tegemisel, kus mõned väljad ja nupud on teistsuguse tekstiga.



Joonis 4. Transpordi andmete täitmise kasutajatee voodiagramm

Loodavasse lahendusse on kavandatud ka transpordi pakkumiste leht, ning ka selle jaoks on loodud eraldi voodiagramm. Tegevused nimetatud lehel kattuvad joonisel 3 välja

toodud tegevustega kuid erinevuseks on tegevuste nimetused. Transpordi pakkumiste lehe kasutajatee voodiagramm on välja toodud käesoleva lõputöö lisades (Lisa 2).

4.5.2 Prototüübi loomine

Kõige parema ettekujutuse loodavast lahendusest annab kasutajaliidese disain. Kasutajaliidese disainis tehakse loodava rakenduse värvide valik ning luuakse prototüüp. Prototüüp on disainilahenduse piiratud esitus, mis võimaldab selle sobivust hinnata ning annab esialgse ülevaate loodavast lahendusest [63, pp. 240-241].

Värvide valik on tähtis, sest värvid annavad kasutajatele esmamulje rakendusest, ning halbade värvikombinatsioonide tõttu võivad kasutajad veebirakenduse kasutamisest kohe loobuda. Värvide valiku puhul on tähtis saavutada tasakaal kasutajaliidese kasutatavuse ja kasutajaliidese esteetilisuse vahel [63, p. 27]. Käesoleva lõputöö autor kasutas värvide valikus veebirakendust nimega Material Design Pallette. Nimetatud veebirakenduse valiku tingis värvide valiku keerukus, ning käesoleva lõputöö autor soovis võimalikult lihtsat ja ka kindlat värvide kombinatsiooni, mis läheks omavahel kokku. Tabel (Tabel 2) kirjeldab valitud värve ja nende peamisi kasutuskohtasid kasutajaliidese.

Värvide valikul oli tähelepanu all lisaks ka värvide kontrastsus. Värvide kontrastsus on oluline, et nii tekst kui ka kasutajaliidese teised põhielemendid paistaks hästi välja ka nägemisprobleemidega inimestele. Värvide kontrastsust käsitleb WCAG (*Web Content Accessibility Guidelines*) 2.1 tase AA [64], kuid käesolev lõputöö miinimumeesmärk oli luua lahendus minimaalselt tasemel A. Kuna käesoleva lõputöö autor leiab, et loodava lahenduse sisu ja tegevused peaksid ka halvema silmanägemisega inimestele paremini silma paistma, siis testiti peamisi teksti ja teksti taustavärvide kombinatsioone veebirakendusega WebAIM. Nimetatud rakendus annab kahe värvikoodi sisestamisel tagasisidet nii värvide kontrastsuse suhte osas, kui ka WCAG värvide kontrastsuse nõuete täitmise osas [65].

Tabel 2. Värvide valik

Värv	Peamised kasutuskohad
	Rakenduse menüüriba ning jalus. Lisaks transpordi pakkumiste kuulutuste päised.
	Tervituse teksti taust kasutajale, kes ei ole sisse loginud.
	Nupud, mis tähendavad tegevust, mis ei ole kasutajale ohtlikud. Transpordi soovide kuulutuste päis. Aktiivsed sisestuselemendid.
	Pealkirjad ja rõhutatud tekst.
	Tekst, mis ei ole rõhutatud. Mitte aktiivsed sisestuselemendid.
	Sektsioonide jagajad.
	Tekst, mis tumedal taustal. Sisestuselementide ja teksti taustavärv.
	Nupud, mis tähistavad millegi katkestamist või kustutamist. Sisestuselementide värv vea korral ning veatekst.

Peale värvide valikut hakati looma kasutajateede voodiagrammide põhjal prototüüpi. Käesolevas lõputöös oli eesmärk luua kohanduva disainiga veebirakendus. Kohanduv disain on disaini meetod, kus kasutajaliides suudab automaatselt kohaneda erinevates mõõtmetes ekraanidega [66]. Kohanduva disaini tõttu luuakse prototüüp nii arvutile kui ka nutitelefonile. Käesolevas lõputöös kasutatakse prototüübi loomisel Figma rakendust.

Figmat on lihtne kasutada ja lisaks on Figma üks populaarsemaid prototüüpimise tööriistu maailmas [67].

Lisaks arvestatakse kasutajaliidese disainis järgnevate disaini üldiste printsiipidega [63, pp. 21-24]:

- Nähtavus – mida nähtavamal peamised funktsioonid on, seda lihtsam on kasutajal kasutajaliidest kasutada.
- Tagasiside – süsteemi poolt info tagasi andmine kasutajale tema tegevuste käigus või peale tegevust, et kasutaja saaks aru mida ta on saavutanud.
- Piirangud – teatud kasutaja tegevuste piiramine, mida antud ajahetkel ei ole võimalik teha.
- Järjepidevus – kõik sama tegevuse või mõttega kasutajaliidese disaini elemendid peaksid ka üle kasutajaliidese samasugused välja nägema.

Joonisel (Joonis 5) nähtuvad erinevad sisestuselemendid millest osadel on erinev olek. Lähtudes nimetatud disaini üldistest printsiipidest annab erinev olek kasutajale tagasisidet tema tegevustes. Lisaks kasutatakse sama disainiga sisestuselemente üle terve kasutajaliidese, mis annab loodavale lahendusele järjepidevust.

The diagram illustrates three states of an email input field:

- Empty state:** The input field contains the placeholder text "E-mail".
- Filled state:** The input field contains the text "E-mail" and "Abi tekst" (Help text).
- Error state:** The input field contains the text "E-mail", "Sisestatud viga" (Entered error), and "Abi tekst".

Below the input fields are two buttons: "TÜHISTA" (Clear) and "LOGI SISSE" (Log In).

Joonis 5. Erinevad sisestuselemendid ja nupud

Jooniselt (Joonis 6) nähtub sisselogimata kasutaja esialgne vaade arvuti ekraanilt, kus kasutajat kutsutakse rakenduse kasutajaks registreeruma või sisse logima. Lisaks on näha ka esialgne navigatsiooni menüü. Eelpool nimetatud disaini üldistest põhimõtetest järgib joonisel (Joonis 6) olev sisselogimata kasutaja vaade nähtavuse printsiipi – kasutajale tuuakse üle terve vaate igal pool esile, et tal on vaja teatud tegevuste jaoks sisse logida või registreeruda. Lisaks on näha kuidas värvide ja tekstiga on eristatud transpordi pakkumiste ja transpordi soovide kuulutustega.

The screenshot shows the 'Tahan Küüti' website interface. At the top, there is a blue navigation bar with the text 'Tahan Küüti' on the left and 'Avaleht', 'Soovid', and 'Pakkumised' in the center. On the right side of the bar are 'KKK', 'LOGI SISSE', 'või', and 'REGISTREERU' buttons. Below the navigation bar, the main heading 'Tahan Küüti' is displayed in large white text on a blue background, followed by the subtitle 'Teretulemast sõidujagamise platvormile!'. Below this, there are two more 'LOGI SISSE' and 'REGISTREERU' buttons. The main content area is titled 'Viimased kuulutused' and contains three listing cards. Each card has a header bar (green for 'Soov' and blue for 'Pakkumine'), followed by details: 'Alguskoht: Tallinn', 'Sihtkoht: Kilingi - Nõmme', 'Aeg: 27.04.22 15:00', and 'Piirhind: 10 €'. A 'Vaataamiseks LOGI SISSE' button is located at the bottom right of each card. The footer contains the text 'Kõik õigused kaitstud © 2022' and 'rideshare@info.ee'.

Joonis 6. Sisselogimata kasutaja avalehe disain

Joonisel (Joonis 7) on näha, et kasutaja on parajasti sisse logitud ja vaatab kindlat transpordi soovi. Kuna käesoleva lõputöö lahenduse skoobis ei ole rakendusesisese suhtluse arendamist, siis peab lõputöö autor vajalikuks lisada kuulutuste detailvaadetesse kuulutaja kontaktandmeid selleks, et kasutajad saaks omavahel suhelda. Nii kasutajalugude kui ka suhtluse vajaduse põhjustel on ka kuulutuste detailvaade nähtav

ainult sisselogitud kasutajatele, et tagada privaatsust. Lisaks nähtub antud jooniselt, et navigatsiooni menüüle on tekkinud täiendav link „Minu plaan“. Kuna ainult sisselogitud kasutajad saavad kuulutustele ligi, siis sisselogimata kasutajate eest on see menüü link ära peidetud.

Tahan Küüti Avaleht Soovid Pakkumised Minu plaan KKK Ott OK Logi välja

Transpordi soov

TALLINN → RAKVERE

Alguskoht: **Kadaka 88, Tallinn, Harjumaa, Eesti**
Kollase maja juures parklas.

Sihtkoht: **Kuuse 32, Rakvere, Eesti**

Aeg: **27.04.2022 15:00**

Inimeste arv **1**

Piirhind **5 €**

Kommentaar -

Soovija: **Taaniel Tamm**

E-mail: **vvv@hot.ee**

Telefon: **55777999**

Järgmised sõidud

25.04.2022 15:00

TALLINN → RAKVERE

Kõik õigused kaitstud © 2022 rideshare@info.ee

Joonis 7. Transpordi soovi detailvaade

Joonis (Joonis 8) on näide transpordi soovi lisamisest. Transpordi soovi või transpordi kuulutuste lisamine toimub mitmes erinevas grupeeritud sammus, kus kasutajal on alati võimalus oma tegevus tühistada või minna tagasi eelneva sammu juurde. Lisaks saab kasutaja tagasisidet progressi faasi osas.

Joonis 8. Transpordi soovi lisamise sammud

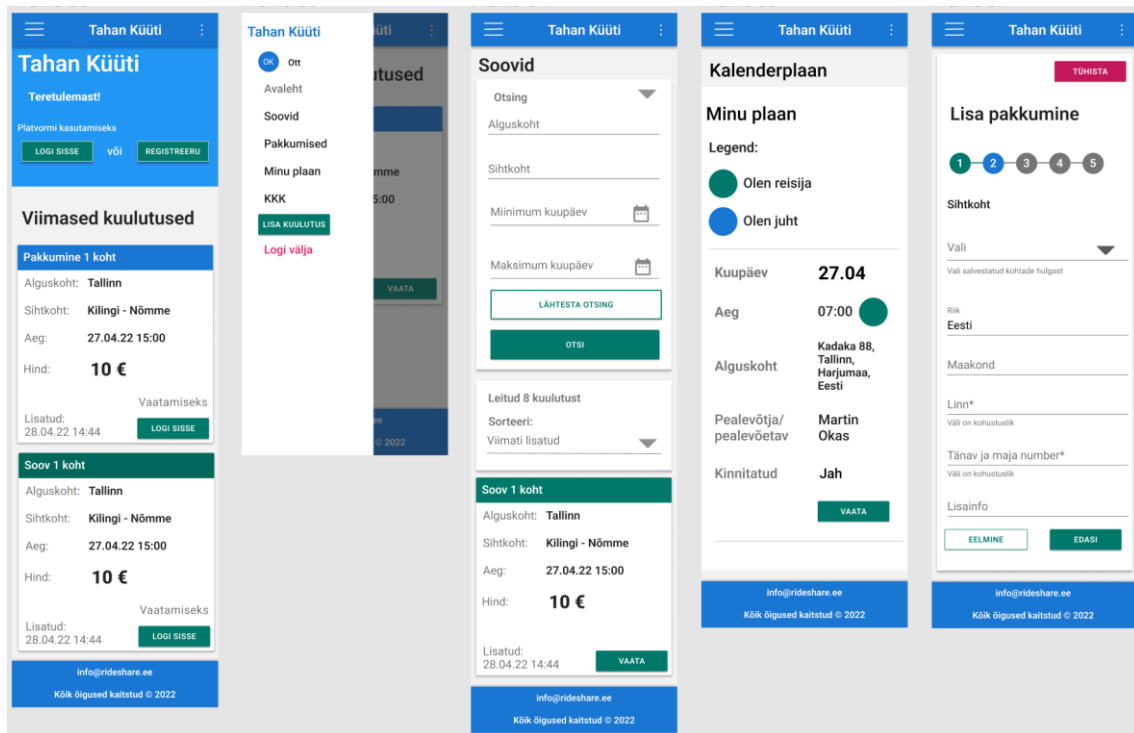
Joonisel (Joonis 9) on näha lehe „Minu plaan“ põhikomponent, milleks on kalenderplaan, mille abil kasutaja saab ülevaate nii oma kuulutustest kui ka nende kuulutuste aegadest ning saab ka nupu kaudu otse liikuda detailifoni. Ka siin lehel on hoitud värvide stiilis sama joont. Kuna plaani vaade on üsna lai, siis erineb see teistest lehtedest selle poolest, et katab peaaegu terve lehe ära laiuse poolelt.

Minu plaan Legend: ● Olen reisija ● Olen juht

Kuupäev	Aeg	Pealevõtmine	Kinnitatud	Pealevõtja/pealevõetav	
27.04	● 07:00	Kadaka 88, Tallinn, Harjumaa, Eesti	Jah	Martin Okas	VAATA
	● 16:15	Kause 33, Tartu, Eesti	Jah	Nimi Uus	VAATA
28.04	● 07:00	Kadaka 88, Tallinn, Harjumaa, Eesti	Jah	Martin Okas	VAATA
29.04	● 07:00	Kadaka 88, Tallinn, Harjumaa, Eesti	Ei	-	VAATA
30.04	● 07:00	Kadaka 88, Tallinn, Harjumaa, Eesti	Ei	-	VAATA
02.05	● 07:00	Kadaka 88, Tallinn, Harjumaa, Eesti	Jah	Mina	VAATA

Joonis 9. Lehe "Minu plaan" põhikomponent arvuti vaates

Kohanduva disaini huvides valis käesoleva lõputöö autor Figma rakenduses mobiili versiooni prototüübi loomiseks võimalikult kitsa lahenduse, et oleks võimalik paremini aru saada disaini elementide paigutusest väga kitsal ekraanil ja et arenduse faasis kasutajaliidest arendades ei tekiks üllatusi. Joonisel (Joonis 10) on näha, et navigatsiooni menüü on täielikult ära peidetud kasutaja eest ja kasutaja peab seda avama nupust. Lisaks on navigatsiooni elemendid paigutatud üksteise alla. Samalt jooniselt nähtub veel, et „Kalenderplaani“ komponenti on muudetud võrreldes arvuti vaate näitega (joonis 9) nii, et veergude pealkirjad on toodud iga infovälja ette. Peamisteks muudatusteks prototüübi mobiili vaadete tegemiseks võrreldes arvuti vaatega oligi elementide paigutamine rohkem üksteise alla.



Joonis 10. Mobiili vaated

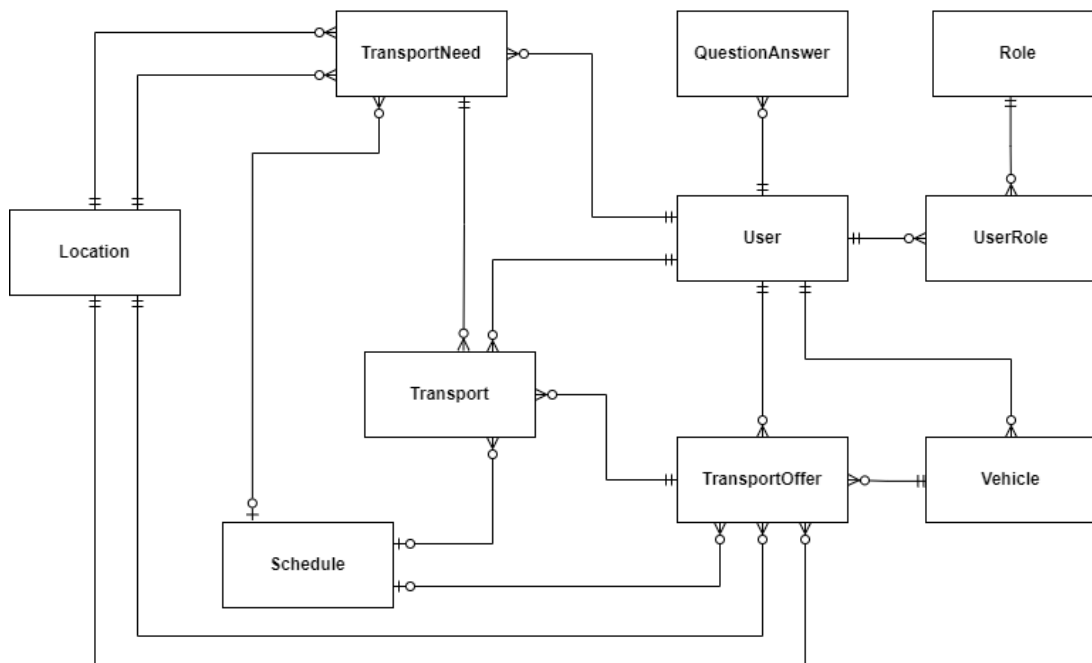
Käesoleva lõputöö autor lõi prototüübi tervele rakendusele nii mobiili kui ka arvuti vaadetes, ning ülejäänud vaated ja komponendid on leitavad lisas (Lisa 3).

4.5.3 Andmebaasi disain

Käesolevas analüüsi osas on kogutud informatsiooni põhjal võimalik luua andmebaasi disain. Kasutajalugude põhjal on praeguseks loodud kasutajatee diagrammid ja kasutajaliidese vaated mis annavad piisavalt informatsiooni olemi-suhte diagrammide

loomiseks. Käesoleva lõputöö autor kasutas olemi-suhte diagrammide loomiseks Windowsi rakendust nimega draw.io Diagrams. Esmalt disainis käesoleva lõputöö autor lihtsustatud versiooni olemi-suhte diagrammist, kus kirjeldati ära olemite nimed ning nendevahelised seosed (joonis 11). Inglise keele kasutamine aitab hoida loodava lahenduse stiili ühtlasena, sest ka valitud programmeerimiskeeled ja raamistikud on inglise keelsed. Lisaks annab inglise keele kasutamine tulevikus võimaluse ka võõrkeelsetel huvilistel loodavast lahendusest aru saada.

Joonisel () oleval olemi-suhte diagrammis kasutatakse peamiselt üks-mitmele seoseid, mis tähendab, et ühel objektiga võib olla seotud mitu teisest tabelist pärit objekti. Kuna andmebaasis otse ei looda mitu-mitmele suhet, siis on kasutaja (*User*) ja rolli (*Role*) vahele lisatud vahetabel, mis aitab luua mitu-mitmele seoseid. Kasutajal saab olla mitu rolli, ning ka ühel rollil saab olla mitu kasutajat. *QuestionAnswer* olemis hoitakse loodava lahenduse abiküsimusi ja vastuseid ning *Vehicle* tabelis hoitakse andmeid transpordi pakkumistes kasutatava sõiduki üle. Sõiduk ise saab kuuluda mitmesse erinevasse transpordi pakkumisse (*TransportOffer*), sest kasutaja saab oma transpordi pakkumisse sõiduki ka valida juhul, kui tal see salvestatud on. Transpordi pakkumisel (*TransportOffer*) on kaks erinevat seost asukohaga (*Location*), millest üks seos on alguskoht ning teine seos on sihtkoht. Samasugused seosed on asukoha olemil (*Location*) veel ka transpordi sooviga (*TransportNeed*). *Transport* olem ise viib kokku transpordi pakkumise (*TransportOffer*) ja transpordi soovi (*TransportNeed*) kui kasutaja avaldab kellegi kuulutusele transpordi soovi või teeb kellegi transpordi soovile pakkumise. Olemis *Schedule* on loodud eesmärgiga siduda mitu andmebaasis olevat sama kuulutust, sest kasutajad saavad luua korduvaid kuulutusi.



Joonis 11. Lihtsustatud olemi-suhte diagramm

Koos andmeväljade ja andmetüüpide olemi-suhte diagramm asub ruumi kokkuhoiu mõttes käesoleva lõputöö lisades (Lisa 5). Viidatud olemi-suhte diagrammi andmetüübid on C# programmeerimise keelest põhjusel, et kasutatakse *code-first* lähenemist, mis tähendab, et andmebaas luuakse koodi põhjal [68].

4.6 Ahitektuur

Loodaval rakendusel kasutatakse kihelist arhitektuuri, mis tähendab, et igal programmi osal, ehk kihil on omad ülesanded.

Tagarakenduse peamisteks kihtideks on:

- Andme kiht – suhtleb on teenusekihi ja andmebaasi vahendaja. Peamine ülesanne on andmebaasiga suhtlemine (andmete küsimine, salvestamine, muutmine ja kustutamine).
- Teenuse kiht – vastutab rakenduse äriloogika eest. Teenusekiht manipuleerib andmebaasist tulevate andmetega ja muudab need ärioloogilistest vajadustest lähtuvalt andmeedastus objektideks API kihile.

- API kiht – võtab vastu eesrakendusest tulevaid päringuid ja edastab sissetulnud andmed edasi teenuse kihile.

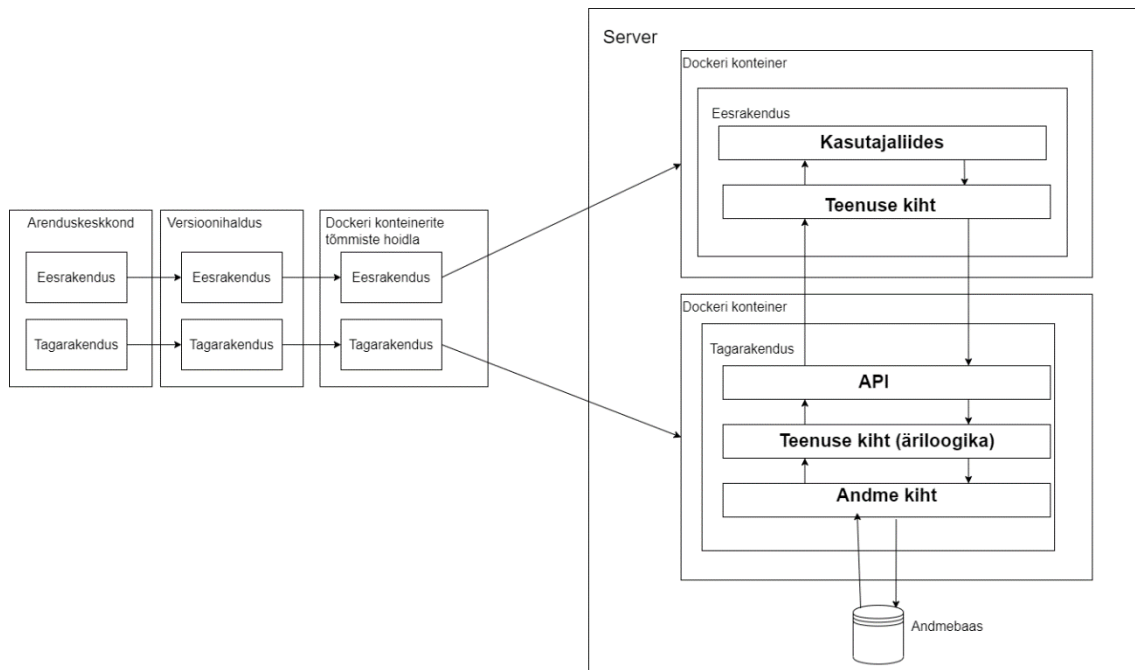
Eesrakenduse põhilised kihid on:

- Teenuse kiht – vahendab informatsiooni kasutajaliidese ja tagarakenduse vahel.
- Kasutajaliides – peamiseks ülesandeks on kasutajaga suhtlemine ja kasutaja sisendi saatmine eesrakenduse teenusekihile. Kasutajaliides koosneb komponentidest, kus erinevad kasutajaliidese osad on eraldi komponendid.

Lisaks eelpool mainitule sõltuvad mõlemad rakendused välistest teekidest ning on ühenduses versioonihalduse platvormiga (GitHub).

Nii eesrakendust kui ka tagarakendust serveeritakse arenduse faasis testimiseks ja publitseeritud faasis kasutajatele läbi Dockeri konteinerite. Dockeri konteinerid on väikesed ja kergekaalulised keskkonnad, mis töötavad üksteisest isoleerituna [69]. Dockeri konteinerid lihtsustavad ja kiirendavad rakenduse publitseerimist ning vajadusel on lihtsasti võimalik pöörduda tagasi eelneva versiooni peale. Lisaks annab Dockeri konteinerite kasutamine rakenduse käituse keskkonna ühtluse nii arenduskeskkondades rakendust käivitades kui ka valmistoodangu keskkondades. Dockeri konteiner lisab veel paindlikkust teisaldatavuse näol erinevate keskkondade vahel [70]. Konteineri tõmmiste hoidmiseks kasutatakse populaarset konteineri tõmmiste hoidlat Docker Hub, milles olevate andmete põhjal luuakse serveris konteinerid rakenduse käitamiseks [71].

Jooniselt (Joonis 12) nähtub rakenduse üldine arhitektuur lihtsustatud kujul. Eesrakendus ja tagarakendus arendatakse eraldi ning nad paiknevad ka versioonihalduskeskkonnas kui ka konteinerite tõmmiste hoidlates eraldi. Joonisel olev rakenduse teekond serverisse hõlmab esmalt seadistamist, kuid hiljem toimib rakenduse teekond versioonihaldusest serverisse automaatselt. Esialgu luuakse serverisse üks eksemplar rakendusest, mida saab avalikkus kasutada, hilisemalt luuakse lisaks ka avalikkusele suletud testkeskkond mis kasutab eraldi konteinerihoidlat ja versioonihalduses eraldi haru.



Joonis 12. Arhitektuur

4.7 Haldus

Loodava lahenduse haldus koosneb peamiselt kahest suuremast osast: rakenduse tehnilisest haldusest ja loodava rakenduse kasutajatega seonduvate probleemide haldusest.

Rakenduse tehnilise haldusega peetakse silmas nii koodi haldust kui ka rakenduse publitseerimisega seotud tegevusi. Arenduskeskkondade valikul otsustati ka kasutada koodihaldus keskkond, milleks valiti Github. Githubis on suurepärased võimalused loodud rakenduse edasiseks publitseerimiseks. Teatud tegevused on võimalik automatiseerida, kaasa arvatud testide jooksumine ja Dockeri konteineri tõmmise loomine ja edasi saatmine tõmmiste hoidlasse.

Loodavas lahenduses kujunevad kõige kulukamaks serveriteenused. Arendaja töö lihtsustamiseks ja haldamise osakaalu minimeerimiseks valitakse teenusepakkuja, kes pakub nii PostgreSQL andmebaasi, kui ka vajalikku teenust Dockeri konteinerite jaoks. Lisaks peaks teenusepakkuja poolt pakutavad teenused sisaldama endas infrastruktuuri haldust ning teenused peaksid olema skaleeritavad. Valiku puhul tuleb väga tähelepanelik olla hinnakujundeses ja äärmise kriitilisusega vaadata tasuta pakutavate teenuste

tingimusi. Üldiselt sobivad tasuta pakutavad teenused arenduseks või testimiseks [72]. Käesoleva lõputöö autor valib serveriteenuse pakkujaks DigitalOcean platvormi, mille hinnad on võrreldes näiteks Microsofti Azure platvormiga paremad ning ka hinnakujundese poliitika on lihtsamini arusaadavam [73], [74]. Lisaks on DigitalOceani platvorm üpris populaarne ning soodne ja ei ole liigselt keerukas [75].

Rakenduse kasutajatega seonduvate probleemide all peetakse silmas peamiselt kasutajate abistamisega või modereerimisega seonduvaid ülesandeid. Kasutajate abistamiseks luuakse esialgu korduma kippuvate küsimuste jaoks eraldi leht, kuid kõiki probleeme see ei lahenda. Probleemiks võivad olla näiteks pahatahtlikud kasutajad, kes ei ole ausad ja on teiste vastu ebaviisakad. Selliste kasutajate eemaldamine rakendusest peaks olema võimalikult kiire ja efektiivne otse kasutajaliidesest. Käesoleva lõputöö lahenduse skoobis ei ole administraatori jaoks arendatavaid võimalusi kasutajate blokeerimiseks või kasutajate abistamiseks teistes küsimustes. Arendused administraatori ja kasutajate suhtluse hõlbustamiseks on rakenduse kasutatavuse seisukohalt väga tähtsad ja ka käesoleva lõputöö autorile prioriteetsed ja arendatakse esimesel võimalusel. Käesolevas lõputöös plaanis olevate arenduste juures saavad kasutajad pöörduda rakenduse haldaja poole e-maili teel ning haldajal on võimalik otse andmebaasist kustutada või blokeerida pahatahtlikke kasutajaid.

4.8 Analüüsi kokkuvõte

Käesoleva lõputöö analüüs toetub peatükis 3 tehtud Facebooki gruppide uuringule, kus selgitati välja vajadus luua veebirakendus, kus inimesed saaksid oma autos olevaid vabu kohti teistele pakkuda ja teisalt saaksid inimesed ka oma sõidusoove kuulutuste näol postitada.

Kõigepealt koostati funktsionaalsed ja mittefunktsionaalsed nõuded kasutajalugude formaadis ja hinnati kasutajalugude sobivust INVEST printsiibi alusel. Edasi valiti tehnoloogiad tagarakenduse ja eesrakenduse arendamiseks. Tagarakendust hakatakse arendama C# programmeerimise keeles ning .NET raamistikus ja eesrakendust arendatakse TypeScript programmeerimise keeles React raamistikus. Lisaks valiti tagarakenduse arenduskeskkonnaks Visual Studio Community, mis on Visual Studio tasuta variant. Eesrakendust hakatakse arendama Visual Studio Code arenduskeskkonnas ja loodava lahenduse andmebaasiks valiti PostgreSQL.

Analüüsi disainifaasis arvestati kasutajakogemuse loomisel peamiste kasutajakogemuse eesmärkidega ning ka peamiste disaini printsiipidega. Nende alusel loodi voodiagrammid kasutajate põhitegevustega rakenduses, milleks oli autoriseerimine, kuulutuste vaatamine ja lisamine ning teiste kuulutustele vastamine. Lisaks valiti ka rakenduse värvid arvestades nägemisprobleemidega inimestega. Edasi loodi rakenduse kasutajaliidesele prototüüp enamuse vaadetega. Disaini lõpuosas loodi andmebaasi disain kasutades olemissuhte diagramme.

Analüüsi lõpuosas kirjeldati rakenduse arhitektuuri ja haldust. Arhitektuuri osas toodi välja, et rakendus serveeritakse kasutajatele läbi Dockeri konteinerite ja tehti üldine joonis rakenduse arhitektuurist. Halduse osas leiti serveri teenuse pakkuja ja selgitati kasutajate ja rakenduse haldaja probleeme, mida käesolevas lõputöös ei lahendata.

5 Arendus

Arendust alustatakse GitHubi versioonihalduskeskkonnas koodihoidla loomisega nii tagarakendusele kui ka eesrakendusele. Koodi paremaks haldamiseks kasutatakse ka rakendust nimega Sourcetree, mis aitab visuaalse poole pealt näha erinevate koodiharude kasutusi ja lisaks on sellega ka mugav luua uus haru [76]. Seejärel laetakse koodihoidlast Sourcetree abiga vajalik kaust, millesse luuakse nii eesrakenduse kui ka tagarakenduse projektid.

Kuigi eesrakenduse arendus toimub eraldi tagarakenduse arendusest, siis arendus toimub tegelikult vaheldumisi. Kõigepealt arendatakse tagarakenduses vajalik funktsionaalsus valmis, ning peale seda tehakse tagarakenduse arenduse peale ka eesrakenduse arendus. Seda tehakse nii põhjusel, et arenduse hetkeseisust oleks parem ülevaade läbi kasutajaliidese, ning lisaks, et avastada tagarakenduse integratsioonitestidest välja jäänud stsenaariumeid vigade leidmisel.

Viimane samm enne tagarakenduse arendust oli luua arenduse jaoks andmebaas. Käesoleva lõputöö autor kasutas teenusepakkuja poolt pakutavat andmebaasi, kui ka lokaalset PostgreSQL andmebaasi.

Tagarakenduse arendust alustati uue projekti loomisega. Visual Studioga uue projekti alustamisel lisab mugavust projekti täpse raamistiku ja põhja valiku võimalus. Käesoleval juhul valiti raamistikuks ASP.NET Core ja põhjaks Web API. Edasi valiti .NET versiooniks 6.0, millele Microsoft pakub pikaajalist tuge. Pikaajaline tugi tähendab 3 aastast tuge Microsofti poolt [77]. Lisaks valiti projekti loomisel ka Dockeri ja Swaggeri tugi loodavale rakendusele. Swagger on lahendus, mis API *endpointide* loomisel rakenduses kuvab need rakendust käima pannes teatud veebiaadressil, ja aitab sellega paremini visualiseerida ja manuaalselt testida [78]. API *endpoint* on üks rakendusliidese osa mis võtab vastu sissetulevaid päringuid ja vastab nendele [79].

Loodud projekti alustati andmekihi loomisega ja esimesena defineeriti ära andmebaasi minevad objektid. Peale seda lisati projekti teek nimega Entity Framework Core, mis annab võimaluse suhelda andmebaasiga otse koodist ilma SQL-i kirjutamata [80]. Tänu

sellele saab eelnevalt loodud objektidest luua andmebaasi tabelid ja suhelda nende tabelitega otse koodist. Lisaks pakub Entity Framework Core ka kasutajate haldamisega seotud abivahendeid, mida kasutati kasutajate registreermise ja sisselogimisega seotud tegevustest. Andmebaasiga ühenduse saamiseks oli vajalik lisada veel `Npgsql.EntityFrameworkCore.PostgreSQL` nimeline laiendus, mis aitab luua Entity Framework -il ühenduse PostgreSQL andmebaasiga. Erinevate andmebaaside jaoks on vaja ühenduse loomiseks erinevaid laiendusi [81].

Edasi loodi tagarakendusele teenuse kiht, mis vastutab äri loogika eest ja on vahendajaks andme ja API kihi vahel. Nii andme-, teenuse-, kui ka API kihil on omad DTO-d, ehk andmeedastus objektid. DTO on andmeedastus objekt, mis kannab andmeid erinevate protsesside vahel [82]. Selle jaoks, et andmeedastuse objekte mõistlikult luua, kasutatakse käesolevas lahenduses teeki nimega Automapper, mis aitab sarnaseid objekte teisendada teist tüüpi objektideks [83]. Objekte on vaja teisendada sest *C#* on objekt-orienteeritud programmeerimise keel, ja iga meetodi sisendiks või väljundiks saab olla ainult teatud tüüpi objekt. Peale teenuse kihti arendati API kiht.

Kasutajate sisselogimise ja sisselogimist nõudvate tegevuste jaoks oli vajalik leida moodus, kuidas aru saada, kes parajasti rakendust kasutab. Selle jaoks kasutatakse JWT ehk *JSON Web Token*-it. JWT on krüpteeritud sõne, mis sisaldab endas turvalisusega seotud informatsiooni. JWT genereeritakse tagarakenduse poolt ning saadetakse kasutaja sisselogimisel tagasi eesrakendusele. Eesrakenduse ülesanne on see iga päringu puhul tagasi saata tagarakendusele, mille alusel tagarakendus valideerib päringut tegeva kasutaja [84].

Peale API kihi valmimist hakati looma integratsiooni teste. Integratsiooni testide toimimiseks oli vaja kasutada mälusisest andmebaasi, et vältida suhtlust päris andmebaasiga. Selle jaoks lisati rakendusele teek nimega In-Memory. Edasi defineeriti iga API *endpointi* jaoks testid, mis testisid erinevate stsenaariumite abil API kihti. Vigade esinemise korral oli vajalik üle kontrollida test ise ja ka vastav API *endpoint*, mida testiti.

Eesrakenduse arendust alustati käsurealt, et luua uus React projekt kasutades TypeScripti. Kui projekt oli loodud, siis oli vajalik lisada loodud projektile teegid, et oleks võimalik mugavalt alustada arendamisega. Peamised teegid, mis lisati eesrakendusele olid:

- React Router – teek, mis aitab hallata eesrakenduse kasutajaliideses navigeerimist [85].
- Material UI – kasutajaliidese raamistik, mis valiti tehnoloogiate valikus.
- Axios – teek, mis aitab teha päringuid tagarakendusele, lisaks aitab teisendada objektid JSON-iks ja vastupidi [86].

Edasi oli vajalik teenusekihis defineerida sarnased andmeobjektid, mis tagarakendusest JSON kujul saadetakse, et lugeda välja kõik andmed ja saata ka tagarakendusele andmed loetaval kujul. Lisaks loodi teenusekihti meetodid tagarakendusele päringute tegemiseks kasutades Axiose teeki. Teenuse kihil oli lisaks vajalik panna igale sisselogitud kasutaja päringule kaasa ka *JWT token*.

WCAG juhendite jälgimiseks oli vaja kõige rohkem eesrakenduse kasutajaliidese arendamisel tähelepanu pöörata, sest WCAG juhendites on suur osa juhistest seotud just eesrakenduse arendamise koodi poolega seotud. Näiteks oli vaja HTML märgenditesse lisada andmeid, mille järgi nii ekraanilugejad kui ka arvuti klaviatuur saaks aru, mis elemendiga on tegu ja kuidas seda elementi peaks tõlgendama ekraanilugeja poolt või klaviatuuriga liikumisel.

6 Tulemused

Käesolev lõputöö oli suuremas osas autorile väga väljakutsuv. Lõputöö jooksul leiti lahendused esialgu väga rasketena tunduvatele küsimustele. Kõige raskemaks osutus analüüsi osa, kus oli vaja leida loodavale lahendusele kasutatav disain. Käesolev peatükk kirjeldab loodud lahendust ja põhilisi edasiarendusi, mis on käesoleval hetkel plaanis.

6.1 Loodud lahendus

Lõputöö jooksul valmis kohanduva disainiga veebirakendus, kuid kogu nõuetes kajastuvat funktsionaalsust ei suudetud teostada käesoleva lõputöö jooksul. Kasutajad saavad registreeruda ja sisse logida, ning ka kuulutusi vaadata, lisada ja muuta. Samuti saavad kasutajad lisada korduvaid kuulutusi. Lisaks on rakendust võimalik ka klaviatuuriga kasutada ja värvide valik tekstide ümber on piisavalt kontrastne ja vastab enamuses WCAG nõuetele.

Valmis ei jõutud ühte tähtsaimat rakenduse osa – läbi rakenduse otse kuulutusele pakkumise tegemist või soovi avaldamist. Käesoleva lõputöö autorile valmistas raskusi esialgu prototüübi võimaliku disaini leidmine ja värvide valik. Kuigi käesoleva lõputöö skoobis ei olnud kontrastsus, siis autor soovis enamuse rakenduse luua piisavalt kontrastsena, et nägemisraskustega inimestele oleks samuti rakendus kasutatav. Lisaks oli käesoleva lõputöö autoril esialgu raske kasutada eesrakenduse arendamiseks valitud CSS raamistikku, mille kasutamine esialgu aeglustas arendusprotsessi. Probleem seisnes selles, et viimati kasutati nimetatud raamistikku ammu ja oli raske meelde tuletada. Lisaks vajasid WCAG nõuded erilist tähelepanu.

Käesolevas lõputöös loodud prototüüp kujutab endas peaaegu täielikku rakenduse kasutajaliidest. Prototüübi valmimine on loodud lahenduse juures üks suurimaid kordaminekuid, sest kujutab endas sisuliselt kogu rakendust. Prototüüp aitab ka valmimata rakenduse funktsionaalsust kiiremini saavutada, sest on täpne ettekujutus vajaminevast tööst. Loodud prototüüp ja tegelikkuses arendatud eesrakendus on väga

sarnased, selles võib veendud käesoleva lõputöö lisas (Lisa 6), kus nähtub sisseloginud kasutaja avalehe vaated nii mobiiltelefoni ekraanilt kui ka arvuti ekraanilt.

Käesoleva lõputöö raames arendatud rakenduse koodihalduskeskkonna lingid on saadaval käesoleva lõputöö lisades (Lisa 7).

6.2 Võimalused edasiarendusteks

Enne edasiarendusi lõpetatakse käesoleva lõputöö skoobis olevad arendused ja testitakse rakendust põhjalikult. Lisaks parandatakse osade lehtede puudujäägid WCAG nõuete kohapealt. Lisaks antakse ka rakendus kasutajatele testida ja positiivse tagasiside puhul publitseeritakse rakendus avalikult kasutamiseks.

Enne esimesi edasiarendusi plaanitakse rakendus viia vastavusse WCAG 2.1 tase AAA nõuetega, et rakendust saaksid mugavamalt kasutada puuetega inimesed.

Edasiarendusi peaks alustama eesrakendusele testide loomisega, et tulevikus, kui juba edasiarendusi on loodud, ei peaks korruga suurt hulka teste looma hakkama. Edasiarendustes on prioriteetsel kohal haldaja ja kasutajate vahelise suhtluse lihtsustamine. Kasutajad võiksid saada võimalusi anda haldajale märku pahatahtlikest ning rakendust mittesihipäraselt kasutavate kasutajate osas analoogselt Facebookiga, kus saab teavitada postitusest, mis ei sobi keskkonda. Lisaks oleks vajalik ka haldajale luua võimalused kasutajate eemaldamiseks või blokeerimiseks otse kasutajaliidest, mis on võrreldes otse andmebaasis muudatuste tegemisega ohutum.

Ilmselt kõige raskem edasiarendus on kuulutuste otsing algus- ja sihtpunkti järgi arvestades teekonnale jäävaid kohtasid. Nimelt praegune rahvastiku jaotus suurematesse linnadesse soosib neid kasutajaid, kes ise soovivad sõita suuremate linnade vahel. Selletõttu luuakse ka kuulutusi, mille algus- ja sihtkohad on suuremad linnad. Selline lahendus jätab väiksemates kohtades elavad või väiksematesse kohtadesse sõita soovivad inimesed tõenäoliselt ilma otsingutulemustest. Näiteks, kui keegi soovib sõita Tallinnast Tartusse ja sisestab ka sellised parameetrid transpordi soovijate lehel olevasse otsingusse, siis ta võiks sellise otsingu peale leida ka antud teekonnale jäävate sõiduotsijate kuulutused.

Tuleviku edasiarenduste hulgas on prioriteetsel kohal ka Facebooki, Google või Apple konto abil registreerumine ja sisselogimise implementeerimine. Põhjused, et Facebooki gruppide uuringus (Tabel 1) selgus, et leidub mitmeid gruppe, kus otsitakse transporti ka kaupadele, tuleks lisada ka käesoleva rakenduse kasutajatele võimalused täpsustada oma sõidukit ning mida nad soovivad transportida.

Lisaks võiks rakenduses olla kasutajatel võimalus lisada mõni pilt enda kuulutusele kas endast või oma autost.

7 Kokkuvõte

Käesoleva töö eesmärk oli luua veebirakendus, millega Eestis elavad inimesed saaksid nutikalt raha säästa või teenida tulu läbi teiste inimeste abistamise ja keskkonna säästmise.

Lõputöös leitud lahenduse skoobis oli luua klient-server veebirakendus, kus kasutajal on võimalik ennast sisse logida, luua nii transpordi pakkumise, kui ka transpordi soovi kuulutusi. Lisaks lihtsalt kuulutuste tegemisele oli suur rõhk pandud kalenderplaanile ehk korduvatele kuulutustele. Lõputöös planeeritud lahenduse eesmärk oli aidata inimestel leida sõidukaaslane, hoida kokku kuludelt ja luua pidevalt samal marsruudil liikuvate inimestega püsikontakte, mis annab inimestele suure kokkuhoiu nii ajas kui ka rahas.

Käesoleva lõputöö analüüsist sai sisulise ülevaate rakenduse peamisest funktsionaalsusest ning kasutatavatest tehnoloogiatest. Analüüsi faasis tehtud andmebaasi disain andis hea põhja, mille peale rakendust hakata üles ehitama, ning kasutajakogemuse disain ja prototüüp andsid hea aluse nii tagarakenduse kui ka klientrakenduse arendamisele.

Antud lõputöö raames loodud rakendusele on plaanis teha mitmeid edasiarendusi, et luua inimestele mugav ning interaktiivne keskkond, mis aitaks inimestel leida sõidukaaslasti ning seeläbi hoida kokku raha ja säästa loodust.

Kasutatud kirjandus

- [1] rafeeqapp.com, „HISTORY OF CARPOOLING,“ 28 07 2020. [Võrgumaterjal]. Available: <https://rafeeqapp.com/en/news/history-of-carpooling>. [Kasutatud 10 03 2022].
- [2] B. Bernstein, „A Brief History of Ride Sharing,“ 1 04 2015. [Võrgumaterjal]. Available: <https://medium.com/@bhbern/a-brief-history-of-ride-sharing-7d1eca9e4654>. [Kasutatud 28 03 2022].
- [3] „Transportaal,“ Transportaal, [Võrgumaterjal]. Available: <http://www.transportaal.ee/>. [Kasutatud 13 03 2022].
- [4] Bolt, „Bolt,“ [Võrgumaterjal]. Available: <https://bolt.eu/et/>. [Kasutatud 13 03 2022].
- [5] Uber, „Uber,“ [Võrgumaterjal]. Available: <https://www.uber.com/ee/et/>. [Kasutatud 13 03 2022].
- [6] Y. Taxi, „Yandex Taxi,“ [Võrgumaterjal]. Available: <http://yandex-taxi.ee/>. [Kasutatud 14 03 2022].
- [7] Google, „Google AdSense,“ [Võrgumaterjal]. Available: <https://www.google.com/adsense/start/>. [Kasutatud 22 03 2022].
- [8] S. Blumstengel, „The Business Analyst’s Guide to Writing User Stories,“ Freshworks, 03 04 2019. [Võrgumaterjal]. Available: <https://freshworks.io/business-analyst-guide-to-user-stories/>. [Kasutatud 22 03 2022].
- [9] B. Hartman, „New to agile? INVEST in good user stories,“ 14 05 2009. [Võrgumaterjal]. Available: <https://agileforall.com/new-to-agile-invest-in-good-user-stories/>. [Kasutatud 23 03 2022].
- [10] S. Dixit, „Beginners Guide to Client Server Communication,“ 30 07 2019. [Võrgumaterjal]. Available: <https://medium.com/@subhangdxt/beginners-guide-to-client-server-communication-8099cf0ac3af>. [Kasutatud 24 03 2022].
- [11] R. D. John Au-Yeung, „Best practices for REST API design,“ 02 03 2020. [Võrgumaterjal]. Available: <https://stackoverflow.blog/2020/03/02/best-practices-for-rest-api-design/>. [Kasutatud 24 03 2022].
- [12] „What is a REST API?,“ 08 05 2020. [Võrgumaterjal]. Available: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. [Kasutatud 25 03 2022].
- [13] „REST vs. SOAP,“ 08 04 2019. [Võrgumaterjal]. Available: <https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest>. [Kasutatud 25 03 2022].
- [14] B. Pedro, „Common Cases When Using SOAP Makes Sense,“ 02 03 2015. [Võrgumaterjal]. Available: <https://nordicapis.com/common-cases-when-using-soap-makes-sense/>. [Kasutatud 25 03 2022].

- [15] „Backend,“ 11 04 2020. [Võrgumaterjal]. Available: <https://techterms.com/definition/backend>. [Kasutatud 28 03 2022].
- [16] „Web Development Using Frameworks: Why Does It Matter?,“ 06 04 2021. [Võrgumaterjal]. Available: <https://erpsolutions.oodles.io/blog/web-development-frameworks-benefits/>. [Kasutatud 28 03 2022].
- [17] Microsoft, „.NET documentation,“ Microsoft, [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/>. [Kasutatud 28 03 2022].
- [18] „Java Introduction,“ [Võrgumaterjal]. Available: https://www.w3schools.com/java/java_intro.asp. [Kasutatud 29 03 2022].
- [19] M. WARREN, „The 10 Most Popular Java Frameworks,“ 04 10 2021. [Võrgumaterjal]. Available: <https://www.bairesdev.com/blog/the-10-most-popular-java-frameworks/>. [Kasutatud 28 03 2022].
- [20] „Java Documentation,“ [Võrgumaterjal]. Available: <https://docs.oracle.com/en/java/>. [Kasutatud 28 03 2022].
- [21] P. Böllhoff, „Kotlin vs Java: strengths, weaknesses and when to use which,“ 20 01 2022. [Võrgumaterjal]. Available: https://kruschecompany.com/kotlin-vs-java/#What_is_Kotlin. [Kasutatud 29 03 2022].
- [22] „Calling Java from Kotlin,“ 13 04 2022. [Võrgumaterjal]. Available: <https://kotlinlang.org/docs/java-interop.html>. [Kasutatud 20 04 2022].
- [23] E. Moecklin, „An opinionated Kotlin backend service - Framework,“ 28 04 2021. [Võrgumaterjal]. Available: <https://medium.com/nerd-for-tech/an-opinionated-kotlin-backend-service-part-1-87f814e3dffd>. [Kasutatud 20 04 2022].
- [24] „3 top picks for backend development in 2022,“ 24 01 2022. [Võrgumaterjal]. Available: <https://www.merixstudio.com/blog/backend-development/>. [Kasutatud 29 03 2022].
- [25] „Django - The web framework for perfectionists with deadlines,“ [Võrgumaterjal]. Available: <https://www.djangoproject.com/>. [Kasutatud 29 03 2022].
- [26] „Django Rest framework,“ [Võrgumaterjal]. Available: <https://www.django-rest-framework.org/>. [Kasutatud 29 03 2022].
- [27] „Top 10 Backend Technologies,“ [Võrgumaterjal]. Available: https://blog.back4app.com/backend-technologies/#PHP_Features. [Kasutatud 31 03 2022].
- [28] „Top 10 PHP Frameworks,“ [Võrgumaterjal]. Available: <https://blog.back4app.com/php-framework/>. [Kasutatud 31 03 2022].
- [29] „Ruby,“ [Võrgumaterjal]. Available: <https://www.ruby-lang.org/en/>. [Kasutatud 02 04 2022].
- [30] „JavaScript,“ [Võrgumaterjal]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Kasutatud 04 04 2022].
- [31] „Introduction to Node.js,“ [Võrgumaterjal]. Available: <https://nodejs.dev/learn/introduction-to-nodejs>. [Kasutatud 04 04 2022].
- [32] T. Kaneriya, „12 Best Nodejs Frameworks for Web Apps in 2022,“ 05 01 2021. [Võrgumaterjal]. Available: <https://www.simform.com/blog/best-nodejs-frameworks/#Express>. [Kasutatud 04 04 2022].
- [33] „Express,“ [Võrgumaterjal]. Available: <https://expressjs.com/>. [Kasutatud 04 04 2022].

- [34] J. Boyer, „How Secure Are Popular Web Frameworks? Here Is a Comparison,“ 17 07 2018. [Võrgumaterjal]. Available: <https://www.veracode.com/blog/secure-development/how-secure-are-popular-web-frameworks-here-comparison>. [Kasutatud 04 04 2022].
- [35] „HTML: HyperText Markup Language,“ [Võrgumaterjal]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML>. [Kasutatud 05 04 2022].
- [36] „Difference between TypeScript and JavaScript,“ 18 02 2022. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/difference-between-typescript-and-javascript/>. [Kasutatud 06 04 2022].
- [37] „TypeScript,“ [Võrgumaterjal]. Available: <https://www.typescriptlang.org/>. [Kasutatud 05 04 2022].
- [38] A. IVANOV, „The Most Popular Front-end Frameworks in 2022,“ 19 02 2022. [Võrgumaterjal]. Available: <https://stackdiary.com/front-end-frameworks/>. [Kasutatud 08 04 2022].
- [39] „Angular,“ [Võrgumaterjal]. Available: <https://angular.io/>. [Kasutatud 07 04 2022].
- [40] „Vue.js,“ [Võrgumaterjal]. Available: <https://vuejs.org/>. [Kasutatud 07 04 2022].
- [41] „Aurelia,“ [Võrgumaterjal]. Available: <https://aurelia.io/>. [Kasutatud 08 04 2022].
- [42] „React,“ [Võrgumaterjal]. Available: <https://reactjs.org/>. [Kasutatud 08 04 2022].
- [43] „React Native,“ [Võrgumaterjal]. Available: <https://reactnative.dev/>. [Kasutatud 09 04 2022].
- [44] „Why Flutter is the most popular cross-platform mobile SDK,“ [Võrgumaterjal]. Available: <https://stackoverflow.blog/2022/02/21/why-flutter-is-the-most-popular-cross-platform-mobile-sdk/>. [Kasutatud 10 04 2022].
- [45] „Flutter,“ [Võrgumaterjal]. Available: <https://flutter.dev/>. [Kasutatud 10 04 2022].
- [46] M. Lawrence, „What is a CSS Framework?,“ 03 05 2019. [Võrgumaterjal]. Available: <https://medium.com/html-all-the-things/what-is-a-css-framework-f758ef0b1a11>. [Kasutatud 11 04 2022].
- [47] „Top 5 CSS Frameworks to check out in 2022.,“ 12 12 2021. [Võrgumaterjal]. Available: <https://nextjsdev.com/top-5-css-frameworks-to-check-out-in-2022/>. [Kasutatud 15 04 2022].
- [48] „MUI,“ [Võrgumaterjal]. Available: <https://mui.com/>. [Kasutatud 14 04 2022].
- [49] „SQL versus NoSQL: Pros and Cons,“ 29 09 2021. [Võrgumaterjal]. Available: <https://www.datastax.com/blog/sql-vs-nosql-pros-cons>. [Kasutatud 13 04 2022].
- [50] R. Powell, „SQL vs NoSQL databases,“ 10 09 2021. [Võrgumaterjal]. Available: <https://circleci.com/blog/SQL-vs-NoSQL-databases/>. [Kasutatud 13 04 2022].
- [51] J. A., „MariaDB vs MySQL – Key Differences, Pros and Cons, and More,“ 07 04 2022. [Võrgumaterjal]. Available: <https://www.hostinger.com/tutorials/mariadb-vs-mysql>. [Kasutatud 14 04 2022].
- [52] R. Peterson, „MariaDB vs MySQL: What is the Difference Between MariaDB and MySQL,“ 26 02 2022. [Võrgumaterjal]. Available: <https://www.guru99.com/mariadb-vs-mysql.html>. [Kasutatud 14 04 2022].
- [53] „The benefits of PostgreSQL,“ [Võrgumaterjal]. Available: <https://www.prisma.io/dataguide/postgresql/benefits-of-postgresql>. [Kasutatud 14 04 2022].

- [54] K. Hristozov, „MySQL vs PostgreSQL -- Choose the Right Database for Your Project,“ 19 07 2019. [Võrgumaterjal]. Available: <https://developer.okta.com/blog/2019/07/19/mysql-vs-postgres>. [Kasutatud 16 04 2022].
- [55] „SQLite,“ [Võrgumaterjal]. Available: <https://sqlite.org/index.html>. [Kasutatud 16 04 2022].
- [56] „TOPDB Top Database index,“ [Võrgumaterjal]. Available: <https://pypl.github.io/DB.html>. [Kasutatud 16 04 2022].
- [57] M. Timotic, „14 Best Web Development IDE in 2021 [CSS, HTML, JavaScript],“ 05 11 2018. [Võrgumaterjal]. Available: <https://tms-opensource.com/blog/posts/web-development-ide/>. [Kasutatud 13 04 2022].
- [58] „Visual Studio Code,“ [Võrgumaterjal]. Available: <https://code.visualstudio.com/>. [Kasutatud 13 04 2022].
- [59] „Best IDEs for .NET Development,“ 26 09 2020. [Võrgumaterjal]. Available: <https://hackernoon.com/best-ides-for-net-development-7x193tp9>. [Kasutatud 13 04 2022].
- [60] „Buy Visual Studio,“ [Võrgumaterjal]. Available: <https://visualstudio.microsoft.com/vs/pricing/?tab=business>. [Kasutatud 13 04 2022].
- [61] „Subscription options & Pricing,“ [Võrgumaterjal]. Available: <https://www.jetbrains.com/rider/buy/#personal?billing=monthly>. [Kasutatud 13 04 2022].
- [62] T. Lisowski, „Top Git hosting services for 2022,“ 08 12 2021. [Võrgumaterjal]. Available: <https://gitprotect.io/blog/top-git-hosting-services-for-2022/>. [Kasutatud 13 04 2022].
- [63] J. Preece, Y. Rogers ja H. Sharp, Interaction Design, John Wiley & Sons, Inc., 2002, p. 519.
- [64] W3C, „Web Content Accessibility Guidelines (WCAG) 2.1,“ 05 06 2018. [Võrgumaterjal]. Available: <https://www.w3.org/TR/WCAG21/>. [Kasutatud 10 04 2022].
- [65] „Contrast Checker,“ [Võrgumaterjal]. Available: <https://webaim.org/resources/contrastchecker/>. [Kasutatud 07 04 2022].
- [66] S. Berry, „What Is Responsive Web Design? (The Non-Developer’s Cheat Sheet),“ 26 02 2021. [Võrgumaterjal]. Available: <https://www.webfx.com/blog/web-design/what-is-responsive-web-design/>. [Kasutatud 07 04 2022].
- [67] J. Pereira, „10 Prototyping Tools for Designers in 2021,“ 12 01 2021. [Võrgumaterjal]. Available: <https://www.uxbooth.com/articles/10-best-prototyping-tools-for-ux-designers-in-2021/>. [Kasutatud 07 04 2022].
- [68] P. Vadapalli, „Code First Approach in MVC: Everything You Need to Know,“ 25 02 2021. [Võrgumaterjal]. Available: <https://www.upgrad.com/blog/code-first-approach-in-mvc/>. [Kasutatud 12 04 2022].
- [69] S. Carey, „What is Docker? The spark for the container revolution,“ 02 08 2021. [Võrgumaterjal]. Available: <https://www.infoworld.com/article/3204171/what-is-docker-the-spark-for-the-container-revolution.html>. [Kasutatud 08 04 2022].

- [70] S. Yegulalp, „Why you should use Docker and containers,“ 10 10 2018. [Võrgumaterjal]. Available: <https://www.infoworld.com/article/3310941/why-you-should-use-docker-and-containers.html>. [Kasutatud 08 04 2022].
- [71] „Docker Hub,“ [Võrgumaterjal]. Available: <https://www.docker.com/products/docker-hub/>. [Kasutatud 08 04 2022].
- [72] Microsoft, "App Service pricing," [Online]. Available: <https://azure.microsoft.com/en-us/pricing/details/app-service/windows/>. [Accessed 08 04 2022].
- [73] „Our products,“ [Võrgumaterjal]. Available: <https://www.digitalocean.com/pricing>. [Kasutatud 14 04 2022].
- [74] Microsoft, „Azure pricing,“ [Võrgumaterjal]. Available: <https://azure.microsoft.com/en-us/pricing/#product-pricing>. [Kasutatud 14 04 2022].
- [75] Matt, „DigitalOcean Hosting: Dev-Friendly But Do They Get The Basics Right?,“ 23 02 2022. [Võrgumaterjal]. Available: <https://digital.com/best-web-hosting/digitalocean/>. [Kasutatud 14 04 2022].
- [76] Sourcetree, „Sourcetree,“ [Võrgumaterjal]. Available: <https://www.sourcetreeapp.com/>. [Kasutatud 10 04 2022].
- [77] Microsoft, „.NET and .NET Core Support Policy,“ [Võrgumaterjal]. Available: <https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core>. [Kasutatud 17 04 2022].
- [78] „API Design,“ [Võrgumaterjal]. Available: <https://swagger.io/solutions/api-design/>. [Kasutatud 18 04 2022].
- [79] „API Endpoints - What Are They? Why Do They Matter?,“ [Võrgumaterjal]. Available: <https://smarter.com/learn/performance-monitoring/api-endpoints/>. [Kasutatud 17 04 2022].
- [80] „What is Entity Framework?,“ [Võrgumaterjal]. Available: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>. [Kasutatud 17 04 2022].
- [81] Microsoft, „Database Providers,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/ef/core/providers/?tabs=dotnet-core-cli>. [Kasutatud 18 04 2022].
- [82] „The DTO Pattern (Data Transfer Object),“ 31 03 2022. [Võrgumaterjal]. Available: <https://www.baeldung.com/java-dto-pattern>. [Kasutatud 16 04 2022].
- [83] „What is AutoMapper and using it in ASP.NET Core,“ [Võrgumaterjal]. Available: <https://www.pragimtech.com/blog/blazor/using-automapper-in-asp.net-core/>. [Kasutatud 16 04 2022].
- [84] „What is JWT?,“ 8 12 2020. [Võrgumaterjal]. Available: <https://www.akana.com/blog/what-is-jwt>. [Kasutatud 17 04 2022].
- [85] „Beginner’s Guide to React Router,“ 05 04 2016. [Võrgumaterjal]. Available: <https://www.freecodecamp.org/news/beginner-s-guide-to-react-router-53094349669/>. [Kasutatud 18 04 2022].
- [86] „What is Axios?,“ [Võrgumaterjal]. Available: <https://axios-http.com/docs/intro>. [Kasutatud 20 04 2022].
- [87] „Facebook,“ Facebook, [Võrgumaterjal]. Available: <https://www.facebook.com/>. [Kasutatud 12 03 2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Ott Kossar

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

24.04.2022

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

Lisa 2 – Kasutajalood INVEST hinnanguga

Funktsionaalsed nõuded:

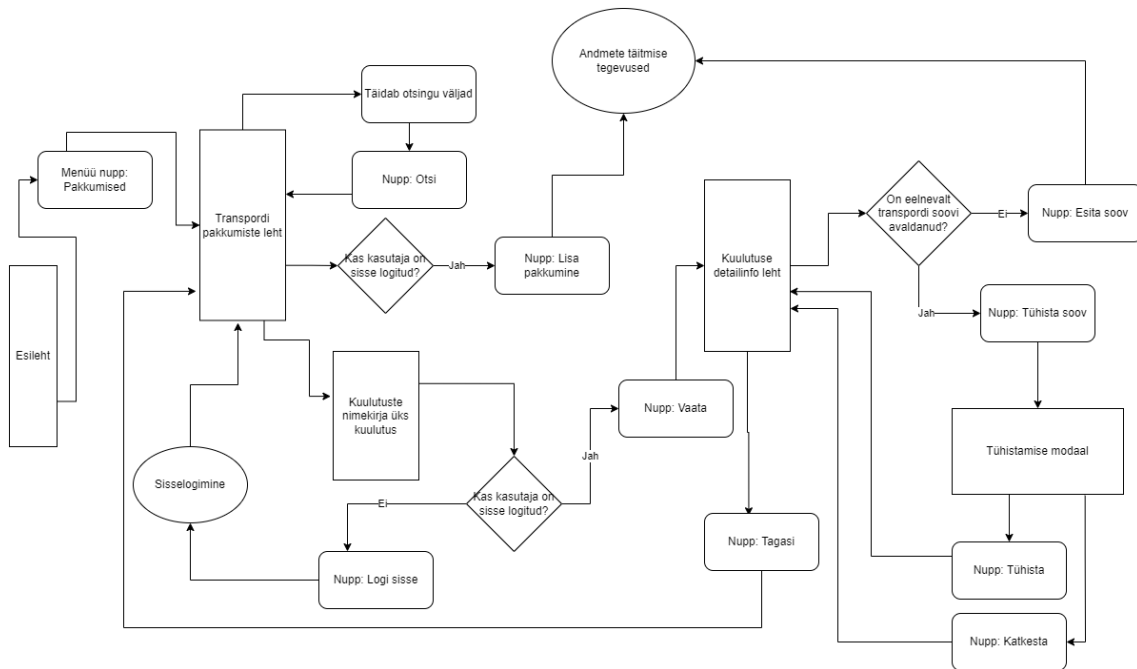
Kasutajalugu	I	N	V	E	S	T
Kasutajana soovin registreerimisvormi täites registreeruda rakenduse kasutajaks.	x	x	x	x	x	x
Kasutajana soovin läbi sisselogimise vormi rakendusse sisse logida.	-	x	x	x	x	x
Kasutajana soovin rakendusest välja logida.	-	x	x	x	x	x
Kasutajana soovin enda kasutajakonto andmeid muuta.	-	x	x	x	x	x
Kasutajana soovin kustutada enda kasutajakonto.	-	x	x	x	x	x
Kasutajana soovin sisestada transpordi otsimise kuulutusi.	-	x	x	x	x	x
Kasutajana soovin sisestada transpordi pakkumise kuulutusi.	-	x	x	x	x	x
Kasutajana soovin kuulutuse sisestamisel luua korraga korduvat kuulutust.	-	x	x	x	x	x
Kasutajana soovin muuta enda kuulutusi.	-	x	x	x	x	x
Kasutajana soovin kustutada enda kuulutusi.	-	x	x	x	x	x
Kasutajana soovin vaadata kuulutusi.	-	x	x	x	x	x
Kasutajana soovin otsida kuulutusi alguskoha ja sihtkoha järgi.	-	x	x	x	x	x
Kasutajana soovin otsida kuulutusi sõidu kuupäeva järgi.	-	x	x	x	x	x
Kasutajana soovin anda oma huvist teise kuulutuse vastu märku avaldades soovi või pakkudes transporti.	-	x	x	x	x	x
Kasutajana soovin tühistada oma registreeringuid teiste kuulutustele.	-	x	x	x	x	x
Kasutajana soovin klaviatuuriga rakenduses navigeerida.	-	x	x	x	x	x
Kasutajana soovin lihtsasti leida abi informatsiooni rakenduse kohta.	x	x	x	x	x	x
Haldajana soovin, et kuulutuste detailinfole pääsevad ligi ainult sisselogitud kasutajad.	-	x	x	x	x	x

Kasutajalugu	I	N	V	E	S	T
Haldajana soovin hallata korduvate küsimuste lehte.	-	x	x	x	x	x

Mittefunktsionaalsed nõuded:

Kasutajalugu	I	N	V	E	S	T
Kasutajana soovin lihtsasti eristada erinevat tüüpi kuulutusi.	-	x	x	x	x	x
Kasutajana soovin, et minu kontaktandmed ei ole kuulutuses mittekasutajatele nähtavad.	-	x	x	x	x	x
Kasutajana soovin kasutada veebirakendust arvutist.	x	x	x	x	x	x
Kasutajana soovin kasutada veebirakendust nutitelefonist.	x	x	x	x	x	x
Kasutajana soovin kasutada veebirakendust tahvelarvutist.	x	x	x	x	x	x
Kasutajana soovin, et veebirakendus oleks eesti keelne.	x	x	x	x	x	x
Haldajana soovin, et rakenduse kood oleks dokumenteeritud.	-	x	x	-	-	x

Lisa 3 – Transpordi pakkumiste kasutajatee voodiagramm



Lisa 4 – Prototüübi vaated

Tahan Küüti

Avaleht Soovid Pakkumised

KKK LOGI SISSE või REGISTREERU

Logi sisse

E-mail _____

Parool _____

Jäta mind meelde.

LOGI SISSE

Kontot veel ei ole?

REGISTREERU

Kõik õigused kaitsstud © 2022 rideshare@info.ee

Tahan Küüti

Avaleht Soovid Pakkumised

KKK LOGI SISSE või REGISTREERU

Registreeru

Eesnimi* _____

Perekonna nimi* _____

Telefoni number* _____
Palun sisestada koos suunakoodiga

Parool* _____
Peab sisaldama vähemalt 8 tähemärki.

Parooli kinnitus* _____

REGISTREERU

Konto juba olemas?

LOGI SISSE

Kõik õigused kaitsstud © 2022 rideshare@info.ee

Tahan Küüti Avaleht Soovid Pakkumised Minu plaan Tegevused KKK Ott OK Logi välja

Viimased kuulutused

Soov: 1 koht Lisatud: 28.04.22 14:44

Alguskoht: **Tallinn** Piirhind: **10 €**

Sihtkoht: **Kilingi - Nõmme**

Aeg: **27.04.22 15:00** VAATA

Soov: 1 koht Lisatud: 28.04.22 14:44

Alguskoht: **Tallinn** Piirhind: **10 €**

Sihtkoht: **Kilingi - Nõmme**

Aeg: **27.04.22 15:00** VAATA

Pakkumine: 1 koht Lisatud: 28.04.22 14:44

Alguskoht: **Tallinn** Hind: **10 €**

Sihtkoht: **Kilingi - Nõmme**

Aeg: **27.04.22 15:00** VAATA

Järgmised sõidud

25.04.2022 15:00 VAATA

TALLINN → **RAKVERE**

Kõik õigused kaitstud © 2022 rideshare@info.ee

Tahan Küüti Avaleht Soovid Pakkumised Minu kuulutused Tegevused KKK Ott OK Logi välja

Transpordi soovid

LISA SOOV

Otsing

Alguskoht Sihtkoht

Miinimum kuupäev Maksimum kuupäev

LÄHTESTA OTSI

Leitud 8 kuulutust Sorteeri: Viimati lisatud

Soov: 1 koht Lisatud: 28.04.22 14:44

Alguskoht: **Tallinn** Piirhind: **10 €**

Sihtkoht: **Kilingi - Nõmme**

Aeg: **27.04.22 15:00** VAATA

Soov: 1 koht Lisatud: 28.04.22 14:44

Alguskoht: **Tallinn** Piirhind: **10 €**

Sihtkoht: **Kilingi - Nõmme**

Aeg: **27.04.22 15:00** VAATA

Järgmised sõidud

25.04.2022 15:00 VAATA

TALLINN → **RAKVERE**

Kõik õigused kaitstud © 2022 rideshare@info.ee

Transpordi pakkumised

LISA PAKKUMINE

Otsing

Alguskoht _____ Sihtkoht _____

Minimimum kuupäev _____ Maksimum kuupäev _____

LÄHTESTA OTSI

Leitud 8 kuulutust

Sorteeri: Viimati lisatud

Pakkumine: 1 koht		Lisatud: 28.04.22 14:44
Alguskoht: Tallinn	Hind: 10 €	
Sihtkoht: Kilingi - Nõmme		
Aeg: 27.04.22 15:00		VAATA

Pakkumine: 1 koht		Lisatud: 28.04.22 14:44
Alguskoht: Tallinn	Hind: 10 €	
Sihtkoht: Kilingi - Nõmme		
Aeg: 27.04.22 15:00		VAATA

Järgmised sõidud

25.04.2022 15:00

VAATA

TALLINN → RAKVERE

Minu soov

KUSTUTA

Transpordi soov TALLINN → RAKVERE

Alguskoht: **Kadaka 88, Tallinn, Harjumaa, Eesti**
Kollase maja juures parklas.

Sihtkoht: **Kuuse 32, Rakvere, Eesti**

Aeg: **27.04.2022 15:00**

Inimeste arv **1**

Piirhind **5 €**

Kommentaar -

Järgmised sõidud

25.04.2022 15:00

TALLINN → RAKVERE

Tehtud pakkumised

Pakkumine

Alguskoht: **Kadaka 88, Tallinn, Harjumaa, Eesti**

Sihtkoht: **Kuuse 32, Rakvere, Eesti**

Aeg: **27.04.2022 14:30**

Inimeste arv **1**

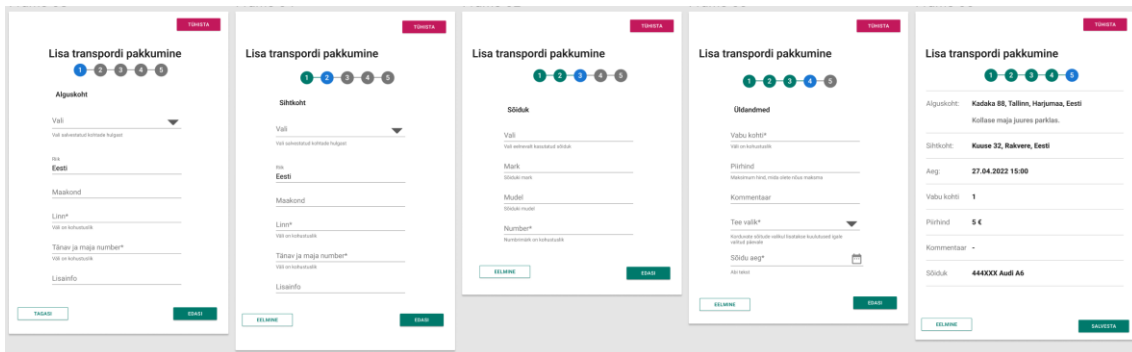
Hind **5 €**

Kommentaar -

Pakkuja **Taniel Tamm**

E-mail: **vvv@hot.ee**

Telefon: **55777999**



Tahan Küüti
Avalaht Soovid Pakkumised Minu plaan KKK Ott OK Logi välja

Korduma kippuvad küsimused

Otsing

Otsi

Otsimiseks sisestage otsitav sõna ja vajutage "Otsi"

LÄHTESTA
OTSI

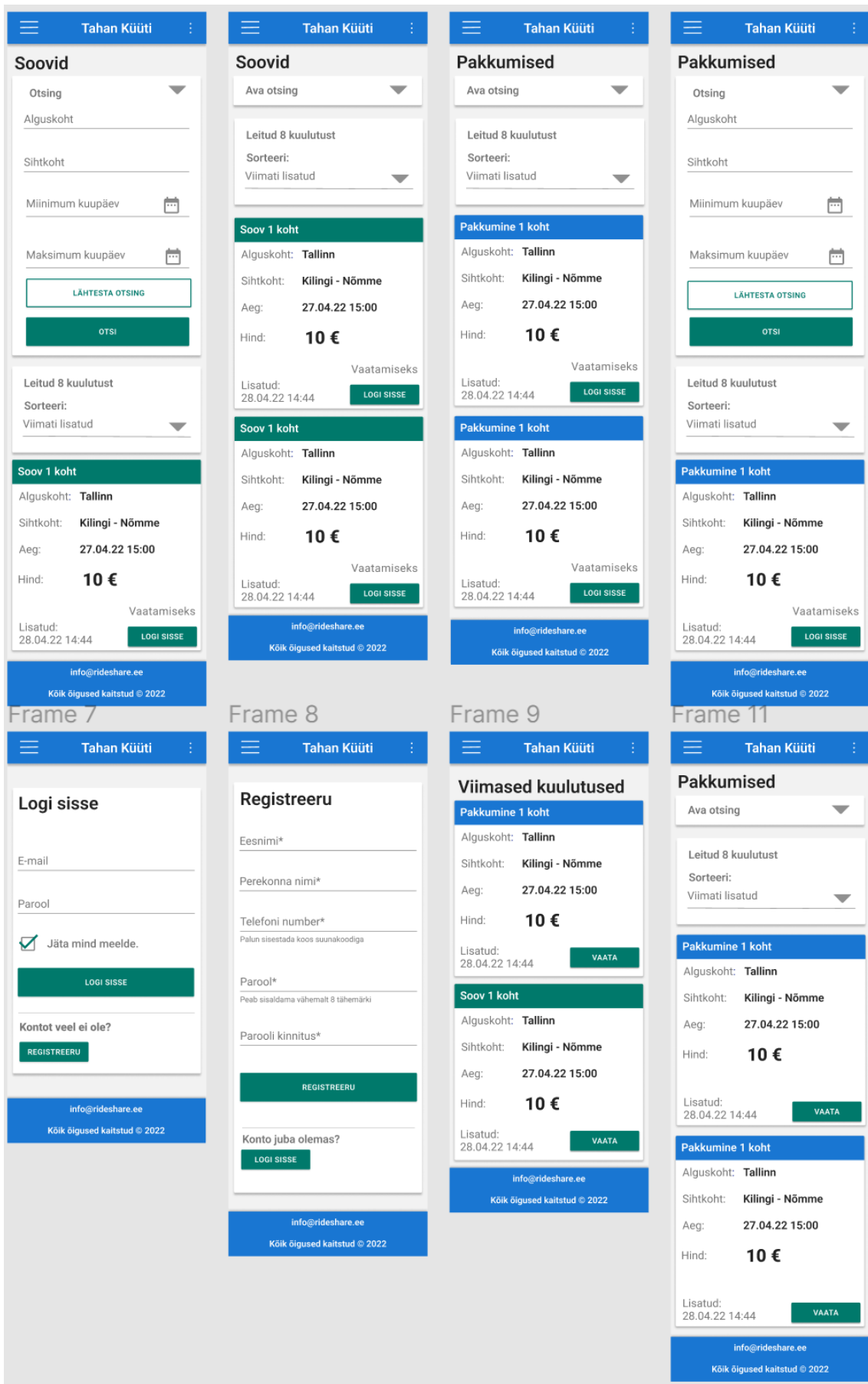
1. Kuidas ma sisse logida saan?

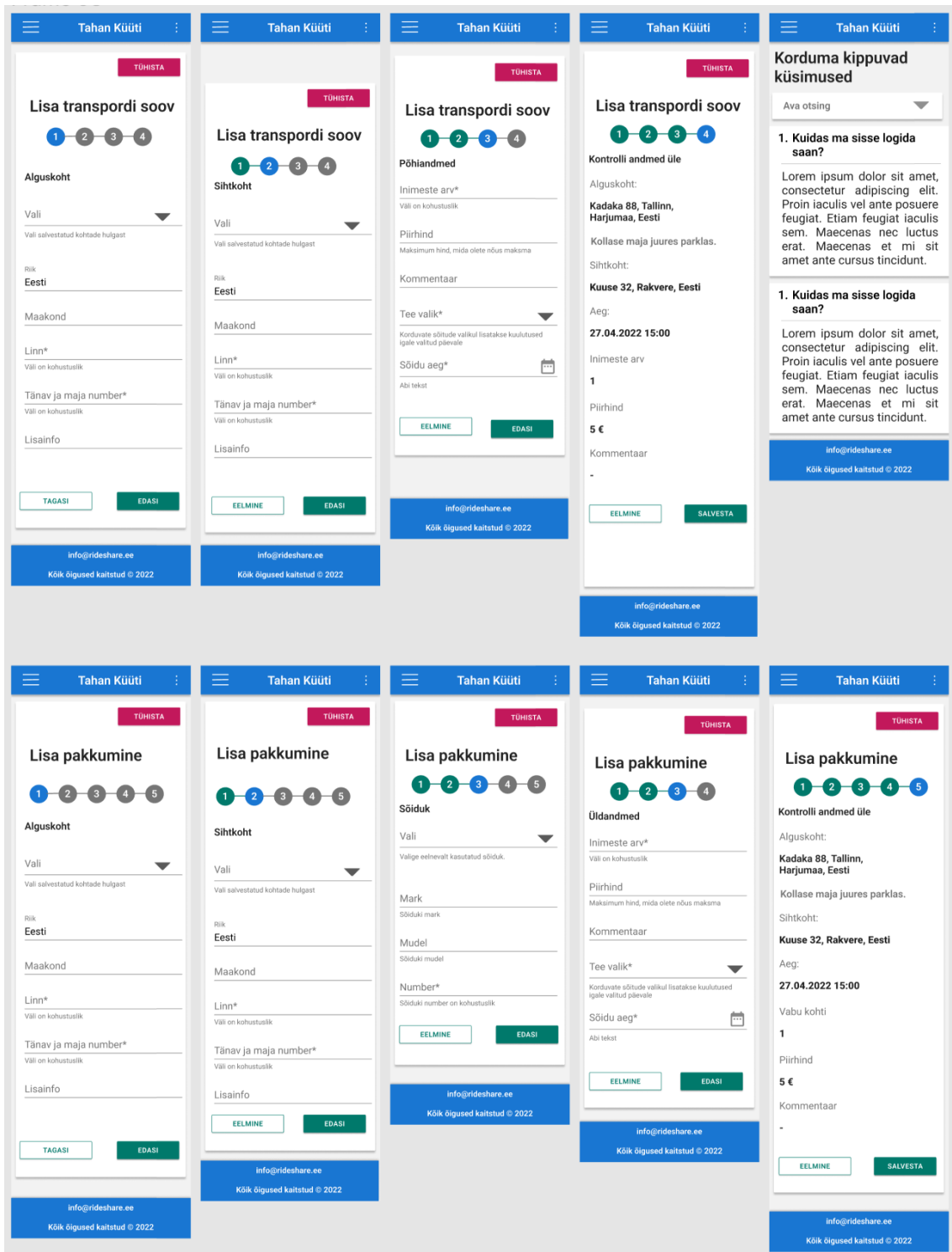
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin iaculis vel ante posuere feugiat. Etiam feugiat iaculis sem. Maecenas nec luctus erat. Maecenas et mi sit amet ante cursus tincidunt.

1. Kuidas ma sisse logida saan?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin iaculis vel ante posuere feugiat. Etiam feugiat iaculis sem. Maecenas nec luctus erat. Maecenas et mi sit amet ante cursus tincidunt.

Kõik õigused kaitstud © 2022
rideshare@info.ee





Tahan Küüti

Kalenderplaan

Legend:

- Olen reisija
- Olen juht

Kuupäev	27.04
Aeg	07:00
Alguskoht	Kadaka 88, Tallinn, Harjumaa, Eesti
Pealevõtja/pealevõetav	Martin Okas
Kinnitatud	Jah

[VAATA](#)

Kuupäev	27.04
Aeg	18:00
Alguskoht	Kadaka 88, Tallinn, Harjumaa, Eesti
Pealevõtja/pealevõetav	Teine Nimi
Kinnitatud	Jah

[VAATA](#)

Kuupäev	27.04
Aeg	18:00
Alguskoht	Kadaka 88, Tallinn, Harjumaa, Eesti
Pealevõtja/pealevõetav	Teine Nimi
Kinnitatud	Jah

[VAATA](#)

info@rideshare.ee
Kõik õigused kaitsitud © 2022

Tahan Küüti

Minu kuulutus

[KUSTUTA](#)

Transpordi soov

TALLINN → RAKVERE

Alguskoht:
Kadaka 88, Tallinn, Harjumaa, Eesti

Kollase maja juures parklas.

Sihtkoht:
Kuuse 32, Rakvere, Eesti

Aeg:
27.04.2022 15:00

Inimeste arv
1

Piirhind
5 €

Kommentaar
-

[MUUDA](#)

[TAGASI](#)

Pakkumised

Pakkumine

Alguskoht:
Kadaka 88, Tallinn, Harjumaa, Eesti

Sihtkoht:
Kuuse 32, Rakvere, Eesti

Aeg:
27.04.2022 15:00

Inimeste arv
1

Hind
5 €

Kommentaar
-

Pakkuja:
Taaniel Tamm

E-mail:
vvv@hot.ee

Telefon:
55777999

[KINNITA](#)

[KEELDU](#)

info@rideshare.ee
Kõik õigused kaitsitud © 2022

Tahan Küüti

Minu kuulutus

[KUSTUTA](#)

Transpordi pakkumine

TALLINN → RAKVERE

Alguskoht:
Kadaka 88, Tallinn, Harjumaa, Eesti

Kollase maja juures parklas.

Sihtkoht:
Kuuse 32, Rakvere, Eesti

Aeg:
27.04.2022 15:00

Inimeste arv
1

Hind
5 €

Kommentaar
-

Auto
Audi A6

Auto number
777XXX

[MUUDA](#)

[TAGASI](#)

Soovid

Soov

Alguskoht:
Ülemiste, Tallinn, Harjumaa, Eesti

Sihtkoht:
Vaksali 10, Rakvere, Eesti

Aeg:
27.04.2022 15:00

Inimeste arv
1

Piirhind
5 €

Kommentaar
-

Soovija
Taaniel Tamm

E-mail:
vvv@hot.ee

Telefon:
55777999

[KINNITA](#)

[KEELDU](#)

info@rideshare.ee
Kõik õigused kaitsitud © 2022

Tahan Küüti

Minu kuulutus

[KUSTUTA](#)

Transpordi pakkumine

TALLINN → RAKVERE

Alguskoht:
Kadaka 88, Tallinn, Harjumaa, Eesti

Kollase maja juures parklas.

Sihtkoht:
Kuuse 32, Rakvere, Eesti

Aeg:
27.04.2022 15:00

Inimeste arv
1

Hind
5 €

Kommentaar
-

Auto
Audi A6

Auto number
777XXX

[MUUDA](#)

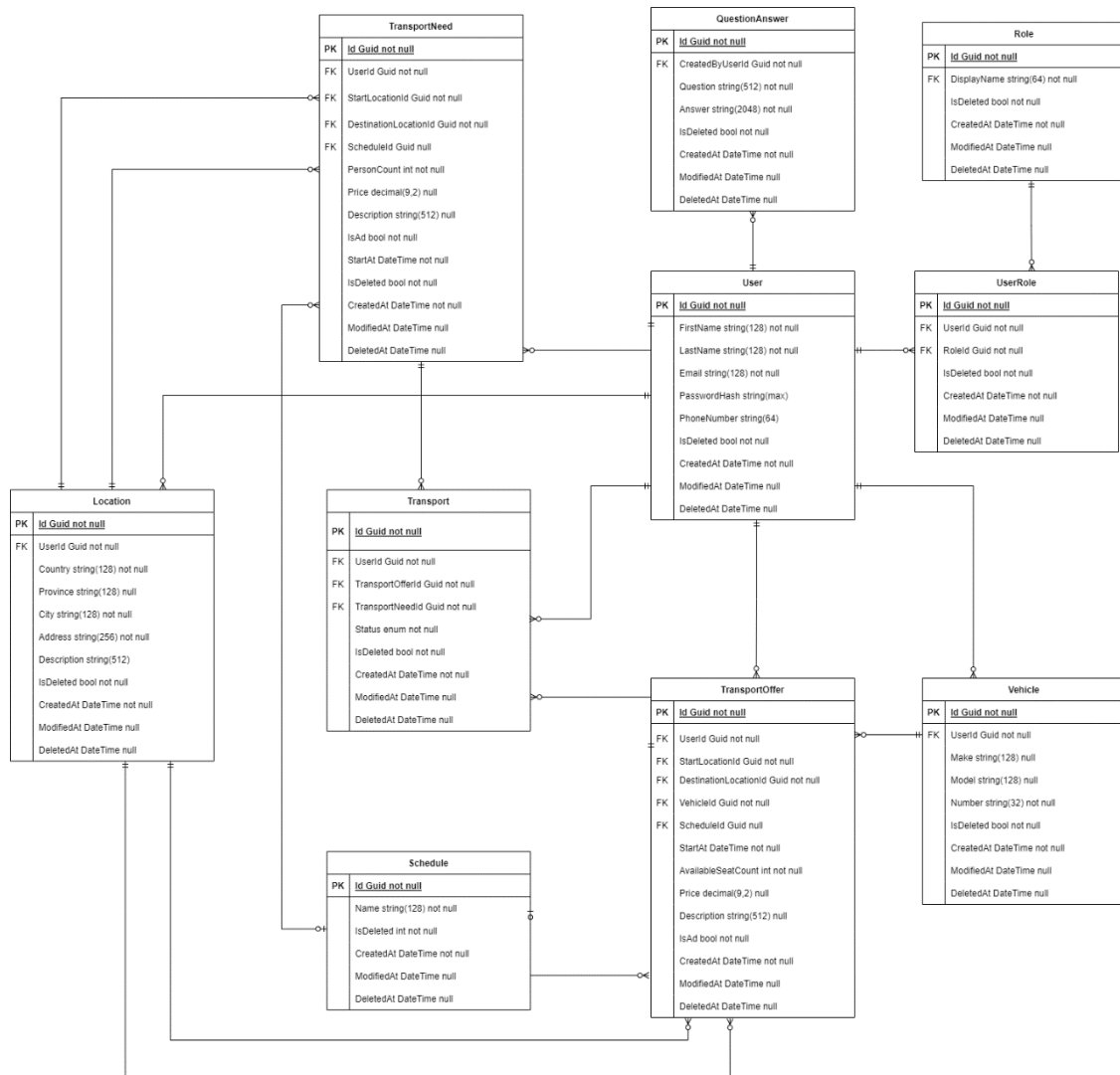
[TAGASI](#)

Soovid

Hetkel ei leidu ühtegi soovi...

info@rideshare.ee
Kõik õigused kaitsitud © 2022

Lisa 5 – Olemi-suhte diagramm



Lisa 6 – Valminud rakendus

The screenshot shows a desktop view of a web application. At the top, there is a blue header with the text "Tahan Küüti" on the left and navigation links "AVALEHT", "SOOVID", "PAKKUMISED", "MINU PLAAN", "KKK", "TESTNIMI", and a "TJ" logo on the right. Below the header, the main content area is titled "Viimased kuulutused". It contains three announcement cards. The first card has a green header "Soov: 1 koht" and "Lisatud: 05.14.2022 02:05". It lists "Alguskoht: kkkkkk ajdfsfb", "Sihtkoht: sdjgsdlgn", and "Algusaeg: 05.31.2022 02:05" with a price of "Hind: 0 €" and a "VAATA" button. The second card has a blue header "Pakkumine: 1 koht" and "Lisatud: 05.14.2022 12:05". It lists "Alguskoht: asd", "Sihtkoht: sd", and "Algusaeg: 05.18.2022 10:05" with a price of "Hind: 12 €" and a "VAATA" button. The third card has a green header "Soov: 1 koht" and "Lisatud: 05.14.2022 11:05". It lists "Alguskoht: Tartu", "Sihtkoht: juuuu", and "Algusaeg: 05.25.2022 11:05" with a price of "Hind: 0 €" and a "VAATA" button.

The screenshot shows a mobile view of the same web application. It features a blue header with a hamburger menu icon on the left and the text "Tahan Küüti" on the right. The main content area is titled "Viimased kuulutused". It displays two announcement cards. The first card has a green header "Soov: 1 koht" and lists "Alguskoht: kkkkkk ajdfsfb", "Sihtkoht: sdjgsdlgn", and "Algusaeg: 05.31.2022 02:05" with a price of "Hind: 0 €" and a "VAATA" button. The second card has a blue header "Pakkumine: 1 koht" and lists "Alguskoht: asd", "Sihtkoht: sd", and "Algusaeg: 05.18.2022 10:05" with a price of "Hind: 12 €" and a "VAATA" button.

Lisa 7 - Versioonihalduse kood

Tagarakendus: <https://github.com/kossar/RideShare>

Eesrakendus: <https://github.com/kossar/RideShare-FE>