TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Risto Reiljan 179521IAIB

MOBIILIRAKENDUS ÜHISTRANSPORDIANDMETE HALDAMISEKS JA VAATAMISEKS

Bakalaureusetöö

Juhendaja: Elli Valla
MSc

Tallinn 2023

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: Risto Reiljan

22.05.2023

Annotatsioon

Selle bakalaureusetöö eesmärk oli luua mobiilirakendus Tallinna ja Harjumaa ühistranspordi andmete kuvamiseks koos mikroteenusega, mis töötleb andmeid ja võimaldab neid mobiilirakenduses kasutada.

Töö tulemusel loodi Android operatsioonisüsteemile mõeldud Kotlin keeles kirjutatud *native* mobiilirakendus, mis kuvab Tallinna ja Harjumaa ühistranspordiliinide sõiduplaane jaotatuna transporditüübi ja liini järgi. Rakenduses on võimalik kuvada liini kõiki peatuseid ja väljumisaegu. Lisaks saab valida kindla reisi ja näha selle peatumisi erinevates peatustes. Loodi ka Micronaut raamistikul põhinev mikroteenus, mille eesmärk on koguda ühistranspordi andmestikust kokku vajalik informatsiooni ja see teisendada mobiilirakenduse jaoks vajalikuks andmehulgaks. Mikroteenusega käib koos PostgreSQL andmebaas töödeldud andmete hoiustamiseks ja andmete kiiremaks pärimiseks.

Loodud rakenduse valideerimiseks anti see kasutajatele testimiseks ning paluti neil vastata tagasisideküsitlusele. Selgus, et kasutajad olid rakendusega valdavalt rahul ning tuli ka häid soovitusi, kuidas rakenduse arendamisega edasi liikuda.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 6 peatükki, 16 joonist.

Abstract

Mobile Application for Managing and Viewing Public Transport Data

The main goal of this thesis was to develop a mobile application to display Tallinn and Harjumaa's public transport information and a microservice that parses raw data to a usable format for the mobile application.

As a result, a native mobile application for Android in Kotlin was created. This application displays Tallinn and Harjumaa's public transport timetables grouped by transport type and route line. It is possible to display all stops of the selected route and departure times from the selected stop. In addition, every departure time from all trip stops can be shown. A microservice-based on the Micronaut framework was also created, which purpose was to parse raw public transport data and make it usable for the mobile application. The microservice is accompanied by a PostgreSQL database for storing parsed data and for faster data retrieval.

The application was validated by giving it to a group of users to use and having them fill out a feedback survey. Users' feedback was mostly positive with the application in general, and they also gave many exciting ideas for the next development cycle.

The thesis is written in Estonian and is 28 pages long, including 6 chapters and 16 figures.

Lühendite ja mõistete sõnastik

API	Application Programming Interface
CC BY-SA 3.0	Creative Commons - Autorile viitamine + Jagamine samadel tingimustel 3.0 Jurisdiktsiooniga sidumata
GTFS	General Transit Feed Specification
JSON	JavaScript Object Notation
NaPTAN	National Public Transport Access Nodes
OSM	OpenStreetMaps
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
TEHIK	Tervise ja Heaolu Infosüsteemide Keskus

Sisukord

Jooniste loetelu

1. Sissejuhatus

Selle bakalaureuse töö käigus valminud ühistranspordi liikumisi kuvav rakendus sündis nii autori isiklikust kui ka aina digimeelsemaks muutuva maailma vajadusest uute lahenduste järgi. Töö peamised eesmärgid olid rajada kompaktne, loogiline, kasutajasõbralik, kiire ning andmeid reaalajas kuvav rakendus. Paljud juba olemasolevad sarnased rakendused on pisivigadega ning ei vasta neile standarditele.

Töö taustaosa kirjutamiseks ja parema ülevaate saamiseks kasutati nii ajakirjanduslikke kui ka akadeemilisi allikaid. Taipamaks teiste riikide ühistranspordi korraldust, võeti abiks tuntumate teenusepakkujate kodulehed. Ka Eesti lahenduste puhul olid teenusepakkujate endi koostatud lehed suureks abiks.

1.1 Probleem

Rakenduse ideele pani alguse Leedu ettevõtte Trafi otsus lõpetada Tallinna ühistranspordi sõiduplaanide kuvamise toe. Trafi oli Eestis üks parimaid ja populaarsemaid ühistranspordi liikumisi kuvavaid rakendusi. Rakendus võimaldas ühistranspordi kasutajal olla kursis busside, trammide, trollide ja rongide liikumisega nii Tallinnas kui ka mujal Harjumaal. Rakendus oli asendamatu tänu oma reaalajas uuenevale ning usaldusväärsele infole. [1] Ettevõtte otsus Eesti turult lahkuda jättis endast maha mitmeid olematu kasutusmugavusega ning aegunud rakendusi. Lõplik idee luua uuendusmeelne ühistranspordi infot kuvav rakendus tekkis autoril Austrias ÖBB Scotty rakendust kasutades.

ÖBB Scotty on väga kompaktne, kasutajasõbralik ja paljude võimalustega rakendus. See on loogiliselt üles ehitatud ning konkreetne. Info reaalajas uuenemine teeb kasutamise mugavaks ning kiireks. Oma uue rakenduse loomisel kasutab autor neid samu põhimõtteid ning loob lõpuks ka Eestisse ühe hea ühistranspordi infot edasiandva rakenduse. Töö esimene eesmärk on saada rakendus toimima nii, et see kuvaks Tallinna ja Harjumaa ühistranspordi informatsiooni. Võttes arvesse, et ka mujal Eestis on sellise teenuse järgi vajadus, on järgmine eesmärk laiendada sedasama süsteemi üle Eesti ning teha sellest üle kogu riigi töötav rakendus.

1.2 Töökäik

Lõputöö eesmärk on arendada Android platvormil mobiilirakendus ning seda toetav mikroteenus. Mobiilirakendus peab suutma kasutajale kuvada Tallinna ja Harjumaa ühistranspordi infot. Mikroteenuse eesmärk on toetada mobiilirakenduse tööd, tagades sellele tarvilikke andmed.

Mobiilirakendus peab võimaldama kasutajal arusaadaval kujul kuvada Tallinna ja Harjumaa ühistranspordi liinide graafikuid. Rakenduses peab olema võimalik valida erinevate transporditüüpide vahel ning kuvada tüübi liine. Liini valimisel peab rakendus kuvama selle peatused ning peatuse all valitud liini väljumisajad. Mobiilirakenduses peaks vältima andmetöötlust, et hoida mälukasutus võimalikult madal ja tagada rakenduse kiire ja tõrgeteta töö. Lisaks sellele on oluline, et lahendus töötaks ka võrguühenduseta ehk mikroteenusest laetud andmeid on vaja lokaalselt hoiustada. Veel on tähtis, et mobiilirakendus teeks mikroteenuse pihta võimalikult vähe päringuid, et vähendada mobiilse andmeside kasutamist.

Rakendust toetav mikroteenus peab suutma algandmestikust leida vajalikud andmeosad ning need salvestama andmebaasi. Mikroteenuse ülesanne on tegeleda andmetöötlusega ja andmete ettevalmistamisega mobiilirakenduses kasutamiseks. Algandmete värskendamine peab toimuma iseseisvalt, koos võimalusega neid manuaalselt uuendada.

1.3 Ülesehitus

Selle uurimistöö teooriaosa algab sissejuhatusega, kus antakse lühike ülevaade töö idee tekkimisest ning sõnastatakse lahendatav probleem, töö eesmärk ja nõuded loodavale süsteemile. Seejärel avatakse töö taust. Teises peatükis antakse lühiülevaade erinevatest ühistranspordiga seotud digilahendustest Eestis ja välismaal. Lisaks kirjutatakse olemasolevatest lahendustest ning nende puudujääkidest. Kolmandas peatükis tutvustatakse lahenduse loomiseks kasutatud metoodikat koos põhjendustega, miks sellised valikud on tehtud. Seejärel kirjeldatakse rakenduse erinevate osade arenduskäiku. Viimaseks kirjeldatakse valminud rakenduse tähtsamaid komponente, analüüsitakse saadud tagasisidet ning pakutakse välja ideid rakenduse edasiarendamiseks. Valminud rakendusega saab tutvuda skannides Lisa 2. asuvat QR-koodi.

2. Taust

Ühistranspordi kohta käivat informatsiooni on võimalik edasi anda mitut eri moodi alates traditsioonilisest paberile printimisest ning lõpetades spetsiaalsete digiplatvormide loomisega. Maailm liigub järjest enam digilahenduste poole, mis taotlevad kasutajamugavust ja andmetele kiiret ligipääsu. Seevastu on Eestis olevad mobiilsed lahendused ajale üsna soiku jäänud ning vajavad uuendamist. Vajadusest uute lahenduste järgi tuligi idee luua rakendus Transport Info.

2.1 Ühistranspordi digilahendused

Viimastel aastakümnetel on üleminek paberilt digitaalmaailmale eriti intensiivne olnud. Töötatakse selle nimel, et kasutajale kõik võimalikult käepäraseks teha. Esikohal on kasutajamugavus ning kiire ligipääs vajaminevatele andmetele.

2.1.1 Digilahendused Eestis

Esimene laiahaardeline Eesti ühistranspordi digitaliseerumise laine saabus 2012. aasta detsembri keskpaigas. Kuues Tallinna bussipeatuses hakkasid tööle esimesed elektroonilised tablood. Tabloode asukohad valiti vastavalt liinide prioriteedisüsteemile ning busse ootavate inimeste hulga järgi. Esimesed tablood püstitati Vabaduse väljaku, Estonia, Taksopargi, Zoo ja Lepistiku peatustesse. Tabloode suurus tulenes sellest, kui palju erinevaid liine vastavas peatuses peatus. Iga liini jaoks oli mõeldud üks rida, mis kuvas mitme minuti pärast järgmine selle liini ühissõiduk peatusest väljub. Tablool sai ka vajadusel kuvada erakorralist infot kajastavat teksti. Näiteks ühissõiduki tulemata jätmise kohta või takistusest teel. [2]

Lisaks tabloode kasutusele võtmisele avati Tallinna-sisene ühistranspordi reaalajas jälgimiseks mõeldud koduleht transport.tallinn.ee. Lehel oli võimalk vaadata kõikide Tallinna ühistranspordiliinide sõiduplaane ning ühissõidukite paiknemist reaalajas. Alates sellest on Tallinnas paigaldatud peaaegu igasse suuremasse peatusesse elektrooniline tabloo. 2019. aastal andis Tallinna Transpordiameti juht Andres Harjo ERR-ile teada, et aastate 2019–2022 eelarvestrateegias on plaanis tabloodega varustatud peatuseid veelgi juurde luua. Plaanis oli varustada tabloodega veel umbes 50 peatust. [3]

2013. aasta alguses alustas Transpordiamet (sellal Maanteeamet) oma ühistranspordi-

registri avaandmete spetsifikatsiooni esimese täisversiooni koostamisega, mis lõpetati sama aasta suveks. Avaandmetes on kirjeldatud kõik Riiklikku Ühistranspordiregistrisse kantud andmed, mis kirjeldavad ühistranspordiliinide graafikuid, peatusi ja nende asukohti. Avaandmed on kõigile soovijatele vabalt kättesaadavad ja kasutatavad. Avalikest allikatest saadud info taaskasutamisel tuleb viidata algallikale ning veenduda, et andmed ei oleks vanemad kui seitse päeva. Spetsifikatsiooni koostamisel on järgitud Google'i arendatud GTFS spetsifikatsiooni koos mõningate lubatud regionaalsete lisadega. [4] Avaandmete kogu kasutatakse Transpordiameti reisiplaneerija veebilehel peatus.ee. Selle veebilehe eesmärk on leida parim viis sihtkohta jõudmiseks ühistransporti kasutades. Koondrakenduse asemel oli varem kasutusel kolm eraldiseisvat lehekülge: peatus.ee, tartu.peatus.ee ning mobiilivaade m.peatus.ee. Vana lahendus oli aga ajale jalgu jäänud, selles esines vigu ning seda polnud mugav kasutada. [5]

Tallinnas on ainukese linnana Eestis saadaval ka ühissõidukite asukohtade reaalajas jälgimine. Süsteemi loomise eeltööga alustati juba 2008. aastal, kui suurem osa bussidest varustati pardakompuutri ja GPS-seadmetega [6]. Juba neli aastat hiljem, kui avati esimesed elektroonilised tablood bussipeatustes, tehti ka Tallinna ühsissõidukite ajakohastatud positsiooniinfo avalikuks. Neid andmeid saavad kõik soovijad vabalt kohandada ja jagada. Kogu avaandmestik kuulub CC BY-SA 3.0 (Creative Commons – Autorile viitamine + Jagamine samadel tingimustel 3.0 Jurisdiktsiooniga sidumata) [7] litsentsi alla.

2.1.2 Digilahendused mujal maailmas

GPS-i ja mobiilsete seadmete võidukäik 2000ndate keskpaigus tõi kaasa ühistranspordi ajakohastatud andmete süsteemi loomise ja avalikuks kasutamiseks muutmise mitmel pool üle maailma. Suurbritannias loodi juba 2003. aastal NaPTAN (*National Public Transport Access Node*) süsteem. Selle eesmärk oli anda igale Suurbritannia ühistranspordi peatusele unikaalne identifikaator. [8] See võimaldas Transport for Londonil arendada välja oma avaandmekogu ühissõidukte graafikutest, teekondadest ning peatustest. Erinevalt Tallinnast, pole Londonis ühissõidukite asukohtade info ajakohastatud. [9]

Üks ainukesi riike maailmas, kus on terve riigi ühistranspordi informatsioon ja ühissõidukite asukohad reaalajas kättesaadavad, on Austria. 2009. aastal lõi Austria raudtee rakenduse ÖBB Scotty, mis sai oma nime ühe Star Treki tegelase järgi. Rakenduse eesmärk oli kuvada terve Austria ja teisi Austriaga piirnevate alade ühistranspordi graafikud. Lisaks loodi võimalus planeerida teekonda kahe punkti vahel. [10] Ajapikku lisandus rakendusse mitmeid uusi võimalusi. Nüüd saab leida lähedal paiknevaid peatusi, kuvada kõiki ühisõidukite ajakohastatud asukohti ja võrguühenduseta kaarte ning samuti osta pileteid. [11]

2023. aastal alustas Saksamaal esimestel rongiliinidel pilootprojekt kuvamaks rongide täituvust reaalajas. Andmeid kuvatakse nii peatustes, rongil kui ka selleks spetsiaalselt loodud rakenduses. Projekti eesmärk on aidata reisijatel kiiremini leida vabu kohti ja muuta selle abil rongiliiklus efektiivsemaks. Saksamaal on rongiliikluse üks suurimaid probleeme hilinemine, mis on tingitud reisijate ebaefektiivsest rongidele paigutamisest. Uue süsteemiga peaksid reisijad olema ooteplatvormil rohkem hajutatud ja tänu sellele peaks rongile minek muutuma kiiremaks. Ustel asuvad sensorid tuvastavad reisija ja tema pagasi liikumise rongi ja sellelt maha. [12]

2.2 Olemasolevad alternatiivid

Tallinna ühistranspordi infot kuvavaid suuremaid ja väiksemaid rakendusi on saadaval päris mitu. Suurim nendest on Tallinna Ühistransport, mis kuvab informatsiooni otse Tallinna Transpordiameti veebilehelt [13]. See on selle töö käigus loodava rakenduse suurim konkurent. Selle rakenduse suurim miinus on see, et tegu on lihtsalt Tallinna Transpordiameti veebilehe mobiilivaate *wrapperiga*. Piltlikult tähendab see seda, et tegu veebilehega, mida kuvatakse mobiilirakenduse kestas.

Lisaks sellel on Google Play Store'is saadaval mitmed väiksemad rakendused, mille funktsioonid on sarnased, aga seevastu on nende kasutajaliides ja -mugavus ajale jalgu jäämas. Üldiselt kipuvad alternatiivsed rakendused olema väga värvikirevad ning need ei järgi Google'i Material Design'i juhendeid. Need väiksemad rakendused ei ole töö käigus loodavale rakendusele pikemas perspektiivis konkurentideks. Rakenduse esmane versioon keskendub küll ainult Tallinnale ja Harjumaale, aga pikemas perspektiivis saab rakenduse suurimaks konkurendiks Eesti ühistranspordi infot koondav keskkond peatus.ee, millel puudub mobiilirakendus.

3. Metoodika

3.1 Tallinna ja Harjumaa avaandmed

Töös kasutatud andmestik pärineb Tallinna linnavalitsuse poolt Eesti avaandmete teabeväravasse [14] üles laetud andmekogudest. Arhiiviandmete järgi oli kuni 2022. aasta keskpaigani Tallinnal kasutusel oma avaandmete veebisait [15], mis ühendati hiljem Eesti avaandmete teabeväravaga. Töös on kasutusel Tallinna ühistranspordi peatuste ja marsruutide andmekogu [16] ning ühistranspordivahendite reaalajas asukohtade andmekogu [17]. Lisaks nendele on kasutatud Tallinna ja Harjumaa ühistranspordi *General Transit Feed Specification* (GTFS) andmekogu. Seda andmekogu poldud töö kirjutamise ajal Tallinna avaandmete kogust Eesti avaandmete teabeväravasse üle viidud, kuid Tallinna Strateegiakeskuse Tulevikulinna büroo juhi kohusetäitja Hanna-Greta Veersalu sõnul kantakse ka need andmed teabeväravasse.

Google'i välja töötatud GTFS-i eesmärgiks on ühtlustada ühistranspordi sõidugraafikute ja nendega seotud geoandmete formaati [18], et lihtsustada erineva ühistranspordiga seotud rakenduste loomist. Süsteem loodi esialgu Google'i rakenduste siseseks kasutamiseks, aga 2010. aastal sai akronüüm oma tänapäevase nime, kirjeldamaks täpsemalt spetsifikatsiooni Google'i-välist kasutamist. Paljude teiste ühistranspordi andmesüsteemidest tõusis GTFS esile oma kindlate reeglite ja lihtsusega nii inimeste kui ka masinate jaoks. [19]

3.2 Versioonihaldus

Tarkvaraarenduses on väga olulisel kohal versioonihaldus, mis võimaldab tarkvaras tehtud muudatusi säilitada ja vajadusel eri versioonide vahel ringi liikuda. Kõige tuntum ja laialdasemalt kasutusel olev versioonihaldus süsteem on Git. Selle süsteemi lõi 2005. aastal Soome-Ameerika tarkvaraarendaja Linus Torvalds [20], kes on tuntud ka kui Linuxi kerneli looja. Giti ainuke miinus on esialgne kõrge keerukuse aste, mistõttu on sellele olemas lihtsamaid, aga samas aeglasemaid ja vähem võimekaid alternatiive. Nende hulgas on näiteks Mercurial ja Subversion (SVN) [21]. Sellegipoolest on tarkvaraarenduse *de facto* versioonihaldussüsteem Git.

3.3 Taustarakenduse majutamine

Taustarakenduse avalikuks jooksutamiseks on peamiselt kaks varianti: kohalik majutus ja pilvemajutus. Mõlemal variandil on omad plussid ja miinused.

Serveri kohaliku majutamise suurim eelis on täielik kontroll kõige üle, mis puudutab serverit. Majutaja saab vabalt valida riist- ja tarkvara, millel serverit jooksutada. Ta saab valida veel, kas ja millal tarkvara värskendada ning kuidas andmeid hallata. See võimaldab seadistada serverit väga täpselt kasutaja finantsiliste võimaluste ja kasutusvajaduste järgi. Ühest küljest võib selline lähenemine olla algajale kasutajale pisut keeruline ja aeganõudev protsess. See eeldab, et kasutajal on piisavalt teadmisi, kuidas valida serveri jaoks riist- ja tarkvara ja oskust läbi mõelda, mille jaoks serverit vaja on. Teisalt saab serveritarkvara jooksutada peaaegu et ükskõik millisel riistvaral, olgu selleks spetsiaalsed serverarvutid või vanad laua- ja sülearvutid. Kui kasutajal on juba käepärast mõni vana arvuti, siis saab selle kerge vaevaga serverarvutiks ringi teha. Samas on kogu muu infrastruktuuriga palju toimetamist, kui plaanis on serverit efektiivselt ja katkestusteta töös hoida.

Pilvemajutuse üks suurimaid eeliseid on kiire skaleeritavus. Ressursse, nagu RAM ja mälu, saab vastavalt vajadusele läbi teenusepakkuja konsooli lisada vaid mõne nupuvajutusega. See võimaldab rakendusel säästlikumalt kasvada, sest makstakse vaid selle eest, mida päriselt kasutatakse, ning siis ei pea mõtlema infrastruktuuri laiendamise peale. Lisaks pakub pilvemajutus kaitset riistvararikete vastu. Suure tõenäosusega kasutab teenusepakkuja liiasuse printsiipi (*redundancy principle*). See tähendab, et võrgustik koosneb mitmest erinevast ja mitte samas asukohas paiknevast seadmest. See tagab andmetele ligipääsu ka juhul, kui mõnd seadet tabab voolukatkestus, riist- või tarkvararike. Selline süsteem duplikeerib alati andmeid kõikide võrgus olevate masinate vahel. [22] Teisalt on sellisel juhul ka andmete lokaalselt varundamine oluline, et vältida pilvemajutuse suurimat miinust.

Kuna pilvemajutust pakub üldjuhul keegi kolmas osapool, siis puudub täielik ülevaade, mida nende juures hoiustatavate andmetega päriselt tehakse ja kui turvaliselt neid hoiustatakse. Isegi suurimatest pilveteenuste pakkujate serveritest lekivad aeg-ajalt andmed. Selle taga võivad olla näiteks häkkerid. Üks selline juhtum oli 2022. aastal Shanghai politseil, kui nad hoidsid oma andmeid Hiina suurimas pilvemajutusteenuses Alibaba Cloud. [23] Seda võib juhtuda ka teenuse kasutaja enda vea tõttu, nagu juhtus Amazonil, kui konfiguratsioonivea tõttu lekkis Prime Video kasutusandmetega andmebaas. [24] Lisaks sellele on oht, et teenusepakkuja lõpetab järsult teenuse pakkumise ning siis puudub igasugunegi võimalus andmeid päästa.

Sobiva lahenduse leidmiseks tuleb välja arvutada, mis oleks lokaalse serveri jaoks infrastruktuuri loomise kulud ja võrrelda neid pilvemajutuse kasutamise hindadega. Arvesse tuleb võtta, kui suur on serveri liiklus ja kuidas see tulevikus areneb. Suure tõenäosusega on loogilisem kasutada hübriidi kahest põhilisest suunast, kus põhiline liiklus käib üle pilvteenuse ja lokaalses serveris on varundatud andmed.

3.4 REST API

Taustarakendusest jõuavad andmed mobiilirakendusse kasutades *Representational State Transfer* (REST) arhikektuuril põhinevat rakendusliidest. Korrektseks toimimiseks järgib REST viite kindlat ja ühte valikulist arhitektuurilist põhimõtet. Kõige olulisem on kasutajaliidese ühtsus. Päringud peavad olema üle süsteemi ühtsed ja need peaksid tagastama informatsiooni samasuguse struktuuriga. Teiseks peavad serveri ja kliendi rakendused olema täiesti eraldiseisvad üksused. Kliendi pool tohib teada ainult serveri poole ressursi aadressi. Mingil muul moel suhtlemine kahe poole vahel pole lubatud. Kolmandaks peavad päringud sisaldama kogu vajalikku informatsiooni vastuse tagastamiseks. Neljandaks peaks päringu vastuseid võimaluse korral vahemällu kirjutama, et sama päringu uuesti tegemisel ei oleks vaja serverisiseseid protseduure uuesti läbi teha. See aitaks serveri jõudlusel paraneda. Viiendaks peaks rakendusliides koosnema mitmest kihist skaleeritavuse huvides. Viimane ning valikuline punkt on see, et liideses olev kood peaks töötama ainult päringu sooritamise hetkel vastuse tagastamiseks. [25]

Enne REST-arhitektuuri loomist oli laialdasemalt kasutusel *Simple Object Access Protocol* (SOAP). REST-i ja SOAP-i suurimaks erinevuseks on andmetüüp, mille abil andmeid edastatakse. Varasem SOAP-protokoll kasutas andmete edastamiseks XML-vormingut. REST-arhitektuuri suurim tugevus on lai valik andmetüüpe andmete edastamiseks. Kõige laialdasemalt kasutatakse *JavaScript Object Notation* (JSON) ning XML-vorminguid.

3.5 Mobiilirakenduse visuaal

Töö käigus valmiva rakenduse visuaalse poole loomisel on järgitud põhiliselt Google'i loodud raamistikke. Rakenduse visuaal põhineb 2014. aastal loodud [26] Material Design suuniste järgi. Praegune versioon, Material3 (tuntud ka kui Material You), toodi välja 2021. aasta Google'i iga-aastasel arendajate konverentsil Google I/O [27]. Visuaalis on väga tähtsal kohal vaegnägijate ligipääsetavus ning kohanemine erinevatele ekraanisuurustele ja -tüüpidele. Google'i arendajate sõnul on neile väga oluline, et kasutaja saaks võimalikult suurt osa disainist isikupärastada vastavalt enda soovidele. Seetõttu on ka selles töös pandud suurt rõhku isikupärastamisele ja ligipääsetavusele.

Töö kirjutamise ajal on Material3 disainikeel saadaval ainult Androidi mobiilirakendustele mõeldud raamistikele. Material3 komponente saab kasutada Androidi disainiraamistike kasutamata kui ka kasutades Jetpack Compose'i või Flutterit. Selles töös on kasutatud Jetpack Compose disainiraamistikku.

Jetpack Compose tuli esimest korda suurema avalikkuse ette 2019. aasta Google I/O konverentsil [28], kus märgiti, et see loodi kasutajaliideste loomise lihtsustamiseks. Samal konverentsil andis Google teada, et Androidi baasil mobiilirakenduste põhiliseks arenduskeeleks saab Java asemel Kotlin [29], mida peetakse Tšehhi ettevõtte JetBrains Java edasiarenduseks [30]. Samamoodi nagu on Androidi mobiilirakendused Kotlini baasil, on ka Jetpack Compose'i raamistik Kotlini baasil. Teoreetiliselt teeb see arendajatele raamistiku kasutamisele ülemineku lihtsamaks, vältides sellega ajakulu uue keele õppimisele. Jetpack Compose'i suurim konkurent on Flutter, mille esimene täisversioon ilmus 2018. aastal [31]. Flutteri kaks suurimat eripära on keel ja kasutus. Flutter baseerub Dart keelel ning sellega saab arendada nii mobiili-, veebi- kui ka töölauarakendusi. Seevastu loodi Jetpack Compose ainult mobiilirakendusi silmas pidades.

3.6 Kaardirakenduse integratsioon

Mobiilirakendustes on kaardivaate integreerimiseks kaks populaarset varianti: Google Maps ja OpenStreetMaps (OSM). Google Maps on Google'i arendatud kaardirakendus, mis pakub palju eri funktsioone, nagu näiteks ajakohastatud liiklusandmed, navigatsioon ja tänavavaade. Lisaks on sellel ka arendajasõbralik API, mille abiga saab kerge vaevaga kaardi oma rakendusse lisada. Google Maps on arendajate hulgas väga populaarne valik tänu oma suurele kasutajaskonnale, täpsusele ja töökindlusele. Samas on selle kasutamiseks tarvis Google'i arenduskasutajat ning isegi siis on liidese tasuta kasutamiseks kindlad piirangud. Üks negatiivne külg on väga keeruline hinnapoliitika, mis võib potsentsiaalselt viia suure kasutuse puhul kulude kiire kasvuni.

Teine variant on kasutada vabavaralist alternatiivi OpenStreetMaps (OSM), mis on oma funktsionaalsuselt Google Mapsiga sarnane. See tugineb kogukonna lisatud andmetele, et kuvada ajakohast kaarti ning lisaks saab seda vastavalt kasutaja soovidele kohandada. Kuna tegu on avatud lähtekoodiga teenusega, siis selle kasutamisel pole ametlike piiranguid. Ometigi on heaks tavaks vältida suures koguses korraga tehtavaid päringuid. Seetõttu on OSM hea variant arendajatele, kes tahavd implementeerida kaarti oma rakendusse, aga soovivad vältida lisakulusid. Samas võivad OSM-i andmed olla aegunud või puudulikud Google Mapsiga võrreldes ning Androidi süsteemile implementeerimine on tunduvalt keerulisem. Google Mapsi dokumentatsioon on oluliselt põhjalikum ning suuremate võimalustega. Seevastu OSM-i puhul on saadaval väga vähe võimalusi ning suurem osa

tuleb arendajal ise implementeerida.

3.7 Juurdepääsetavus

2016. aasta lõpus rakendus Euroopa Liidu digipääsetavuse direktiiv 2016/2102 [32], mis sätestas avaliku sektori veebisaitide ja mobiilirakenduste juurdepääsetavuse. Normide kohaselt tuleb avaliku sektori teenused kavandada ja teha kättesaadavaks nii, et neid saaks kasutada ka pimedad ja vaegnägijad [33]. Töö autor puutus nende nõuetega esmakordselt kokku Tervise ja Heaolu Infosüsteemide Keskuse (TEHIK) Tervise Infosüsteemi andmevaaturi projekti [34] arendades. Projekti üheks nõudeks oli, et ka vaegnägijatel oleks võimalik rakendust kasutada. Probleemi tõstatas Eesti Pimedate Liit, kuna vaegnägijatel puudusid võrdsed võimalused kasutada riigiastutuste veebisaite [35]. Seetõttu oli ka andmevaaturi projektis tähtsal kohal vaegnägijatega arvestamine. Andmevaaturisse loodi kõrge kontrastiga kahevärviline värvipalett, kus taustaks kasutati vaheldumisi musta ja kollast värvi. Nii muutus eri komponentide eristamine tunduvalt paremaks. [36] Töös kasutatakse sellest projektist inspireeritud samalaadset süsteemi.

3.8 Rakenduses kasutatavad keeled

Töö käigus valmiva rakenduse üks suurimaid eripärasid on suur kasutajaliidese keelte valik. Rakendus on lokaliseeritud eesti, inglise, vene, soome, läti, leedu, saksa ja võro keelde. Valik baseerub 2021. aasta Tallinna turismi ülevaatel, mille järgi külastasid linna enim just neid keeli kõnelevad turistid [37]. Lokalisatsioonid võimaldavad suuremal hulgal kasutajatel rakendust mugavamalt kasutada. Lokalisatsioonid on põhiliselt mõeldud potentsiaalsetele kasutajatele, kelle jaoks teeks see rakenduse kasutamise lihtsamaks. [38]

3.9 Tagasiside kogumine ja analüüs

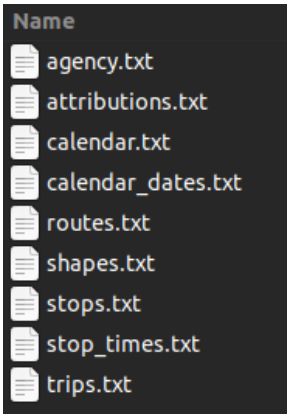
Rakenduse töö ja kasutuskogemuse kohta on analüüsiks võimalik koguda kahte tüüpi andmeid: kvantitatiivsed ja kvalitatiivsed. Adekvaatseks kvantitatiivseks analüüsiks on oluline suur küsitletavate hulk. Kvantitatiivse analüüsi eesmärgiks on panna uuritav ala numbritesse ja teha selle põhjal järeldused. See on hea viis leidmaks rakenduse kasutustrende ning nõrkasid kohtasid. Sisuliselt saab ka väikest andmehulka kvantitatiivselt analüüsida, aga see tekitab suurema mõõtevea ning ei pruugi anda nii põhjalikku tagasiside kui suurem andmehulk. Kvalitatiivne analüüs paneb rõhku kogutavate andmete kvaliteedile ja põhjalikkusele. Oluline on tagasisidest saada teada, kuidas rakendust tajutakse, mis on selle tugevad ja nõrgad küljed ning mis võiks olla teisiti. [39]

4. Arenduskäik

See peatükk kirjeldab töö käigus valmiva rakenduse arenduskäiku. Keskendutakse avaandmetega tehtud tööle, kirjeldatakse, kuidas käib taustarakenduse tegemine ning kuidas see hiljem serverisse püsti sai.

4.1 Avaandmete parsimine

Arendustööd alustades tuli autoril esiteks välja mõelda, millist ühistranspordi andmekogu kasutada. Valikus oli Tallinna Transpordiameti GTFS andmekogu, Eesti avaandmete teabevärava ja Ühistranspordiregistri avaandmed. Selle töö raames oli mõistlikum keskenduda ainult Tallinna ja Harjumaa ühistranspordi infole ehk kasutada Tallinna Transpordiameti GTFS andmekogu. Andmekogu värskendatakse kord päevas ning see koosneb üheksast CSV-formaadis failist: *agents, attributions, calendar, calendar_dates, routes, shapes, stops, stop_times* ja *trips* (Joonis 1).



Joonis 1. GTFS andmekoosseis

Kuna andmekogus olevad failid on CSV-formaadis, siis andmete parsimine on väga triviaalne toiming. Igal andmekogus oleval failil on oma kindel standardiseeritud vorming, mida on kirjeldatud Google Transit dokumentatsioonis [40]. See võimaldab andmekogu hõlpsasti vahetada, ilma et peaks mõtlema parsimise ümber tegemisele.

Näiteks GTFS andmekoosseisu kuuluva peatuste andmed (Joonis 2) koosnevad standardi järgi unikaalsest peatuse identifikaatorist, peatuse koodist, nimest ja kirjeldusest, koordinaatidest, peatuse aadressist ja asukohatüübist. Ühe regionaalse lisana on peatuste andmetesse lisatud Thorebi identifikaator. Thoreb on Rootsi ettevõte, mis paigaldas 2008. aastaks Tallinna ühissõidukitesse pardakompuutrid, mis kuvasid juhile bussiliini kulgemist

ja sõiduki paiknemist reaalajas. [41] Thorebi süsteemi võib pidada tänapäevase ajakohastatud informatsiooni eelkäijaks.

```
stop_id,stop_code,stop_name,stop_desc,stop_lat,stop_lon,stop_url,location_type,parent_station,thoreb_id
134773,5701054-1,"Laimi",,59.22155,23.67992,,,,3294
134774,5701055-1,"Laimi",,59.22143,23.67969,,,,4545
9495,5700265-1,"Nõva",,59.22309,23.68909,,,,9495
9496,5700266-1,"Nõva",,59.22284,23.68963,,,,9496
9493,5700453-1,"Vaisi",,59.21335,23.71257,,,,9493
9494,5700454-1,"Vaisi",,59.21334,23.71180,,,,9494
25469,5700338-1,"Risti",,58.99870,24.05880,,,,25469
25470,5700337-1,"Risti",,58.99850,24.05886,,,,25470
```

Joonis 2. Peatuste andmed GTFS andmekoosseisus

Andmeid parsimiseks ettevalmistades on mõistlik hoolikalt läbi mõelda, mida rakenduse kontekstis täpsemalt vaja läheb. See vähendab protsesse, mida ühe andmekoguga vaja teha on, mis omakorda vähendab ajakulu. Väga suurte andmekogude puhul on ülimalt oluline võimalikult palju üleliigset eemaldada. Seda tüüpi rakenduse puhul on jõudlus väga tähtis ning mida vähem on üleliigset, seda kiiremini rakendus töötab. Peatuste andmete puhul on olulisteks väljadeks, mida hiljem andmebaasi salvestada, peatuse unikaalne identifikaator, peatuse tekstiline nimi ja peatuse koordinaadid.

Selline protsess tuleb läbi teha iga GTFS andmekogusse kuuluva andmestikuga. Veelgi parema jõudluse saavutamiseks saab parsimise protsessid paralleliseerida. See tähendab, et iga faili parsimine pannakse käima eraldi lõimedesse ning see võimaldab mitut faili samaaegselt töödelda. See on eriti efektiivne suurte andmehulkade puhul, kus ühes lõimes tegutsedes peaks iga järgnev protsess ootama eelmise protsessi lõppu.

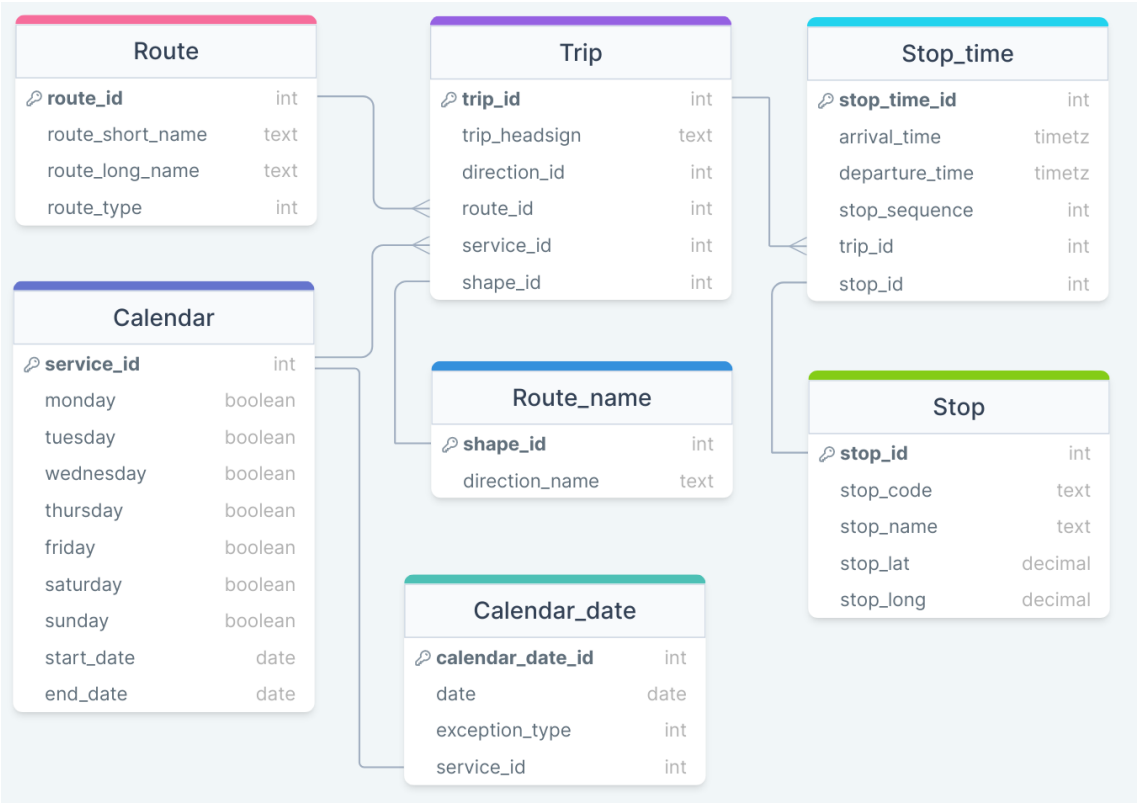
4.2 Taustarakenduse arendamine

Töö järgmises osas oli vaja välja mõelda süsteem, kuidas vajalikke ühistranspordiga seotud andmeid mõistlikult hoiustada ning kuidas tagada nendele kiire ligipääs. Andmete töötlemiseks ja suhtluseks mobiilirakendusega loodi Micronaut raamistikus mikroteenus. Mikroteenust luues oli valida kahe suurema raamistiku vahel: Spring Boot ja Micronaut. Mõlemad on oma olemuselt sarnased ja töötavad ühetaolistel põhimõtetel. Valik langes siiski Micronauti kasuks, sest selle mälukasutus on väiksem ja käivitusaeg on kiirem. [42] Taustarakendus koosneb kolmest suuremast sektsioonist: sisendandmete parsimine ja sünkroniseerimine, ajakohastatud ühisõidukite asukohtade väljunud ning parsitud andmekogu andmete väljund.

4.2.1 Sisendandmete parsimine ja sünkroniseerimine

Eelnevalt parsitud sisendandmete hilisemaks kastuamiseks on neid vaja kuidagi hoiustada. Selleks on kõige mõistlikum luua andmebaas, et iga hilisema päringu järel ei peaks andmeid uuesti parsima.

Parsitud andmed salvestatakse PostgreSQL andmebaasi. Selle andmebaasi loogilise andmemudeli diagramm (Joonis 3) loodi kasutades drawSQL [43] programmi. Joonisel on kujutatud andmebaasis olevad tabelid, nende väljad ja tüübid ning tabelitevahelised seosed.



Joonis 3. Taustarakenduse loogilise andmemudeli diagramm

Tabelisse "route" salvestatakse kõikide ühistranspordiliinide teekonnad. Teekondade tabeliga on üks mitmele seoses tabel "trip". Seal hoitakse kõikide liinide sõite, mis on omakorda mitu ühele seoses tabeliga "calendar", kus hoitakse nädalapäevasid, millal teenus töötab. See on omakorda üks mitmele seoses tabeliga "calendar_date", kus hoitakse erindeid teenuse töös. Liinide sõitude tabeliga on veel seoses tabel "route_name", milles on teekonna täpne nimi. Sellega on veel seotud peatuste aegade tabel "stop_time" ning peatuste aegade tabeliga on seotud peatuste tabel "stop", milles hoiustatakse peatuse nime ja koordinaate.

Mikroteenusesse on kirjutatud funktsioon, mis pärib automaatselt iga päev kell kolm öösel

Eesti aja järgi GTFS andmed, parsib need ja kirjutab seejärel andmebaasi. Eelnev andmebaasi sisu kustutatakse ja asendatakse uuendatud andmetega. Vajadusel saab andmeid manuaalselt sünkroniseerida, tehes selleks GET päring */sync* otspunkti pihta.

4.2.2 Parsitud andmete väljund

Andmebaasis olevate andmete hilisemaks kuvamiseks mobiilirakenduses on vaja need teha läbi otspunkti ligipääsetavaks. Kõige lihtsam ja ressursisäästlikum on andmebaasis olevad andmed parsida JSON-formaati. Micronaut raamistikule on sisse ehitatud Jackson teek, mille eesmärk on võimalikult automaatselt parsida andmebaasi tabelites olevad andmed JSON-formaati.


```
[
    {"tripId": 1, "serviceId": 60666, "stopId": 3894, "departureTime": [5, 28, 44]},
    {"tripId": 1, "serviceId": 60666, "stopId": 1308, "departureTime": [5, 29, 44]},
    {"tripId": 1, "serviceId": 60666, "stopId": 1278, "departureTime": [5, 31, 46]},
    {"tripId": 1, "serviceId": 60666, "stopId": 1274, "departureTime": [5, 33, 14]}
]
```


Joonis 4. Ühissõidukite peatuste aegade näidisväljund


```
[
    {
        "transportType":"tallinna-lin_tram",
        "transportRouteData":[
            {
                "routeName":"1",
                "routeLongName":"Kopli - Kadriorg",
                "directions":[
                    {
                        "directionName":"Kopli - Kadriorg",
                        "directionType":"tallinna-lin_tram_1_a-b",
                        "directionStops":[
                            {
                                "stopId":1148,
                                "stopSequence":1,
                                "stopName":"Kopli",
                                "latitude":59.46121,
                                "longitude":24.66793
                            }
                        ],
                        "trips":[21815,21816]
                    }
                ],
                "routeServices":[
                    {
                        "serviceId":60639,
                        "serviceDays":["SATURDAY"],
                        "dayExceptions":[
                            [2024,1,1]
                        ]
                    }
                ]
            }
        ]
    }
]
```


Joonis 5. Taustarakenduse parsitud andmete näidisväljund

Automatiseeritud JSON-formaadi genereerimiseks oli vaja luua uued andmemudelid, mis ühtlasi oleksid mobiilirakenduses võimalikult mugavalt kasutatavad. Võttes arvesse andmebaasis olevat suurt andmehulka, on kogu andmestik jaotatud kaheks: ühissõidukite

peatumiste ajad (Joonis 4) ja kogu ülejäänud info (Joonis 5). Peatumiste ajad on seetõttu teistest eraldi, et andmehulk on kordades suurem kui ülejäänud informatsioonil. See võimaldab teha suurema andmekogu pärimise ajal tekkiva tõrke korral muu informatsiooni sellegipoolest kätte saadavaks. Andmete protsessimise käigus esineva tõrke korral ei panda kogu rakendust seisma.

4.2.3 Ajakohastatud ühissõidukite asukohtade väljund

Läbi taustarakenduse on võimalik kuvada Tallinna ühistranspordivahendite asukohtasid reaalajas. Selleks kasutatakse Tallinna Linnavalitsuse andmekogu, mis kuvab ühistranspordivahendite asukohti reaalajas. Selle vastu päringut tehes tagastatakse CSV-formaadis informatsioon hetkel aktiivsete ühissõidukite tüüpidest, liini numbritest, koordinaatidest ja liikumissuundadest. Saadud informatsioon parsitakse JSON-kujule, mida oleks hiljem mugav mobiilirakenduses kasutada. Parsimise käigus lisatakse ka päringu tegemise aeg, et hiljem saaks vajadusel juba vananenud andmed kõrvale jätta.

4.3 Taustarakenduse serveris jooksutamine

Töö üks aeganõudvamaid ja keerulisemaid protsesse oli taustarakenduse serveris toimima saamine ja muutmine avalikult kättesaadavaks. Meetodeid probleemi lahendamiseks on mitmeid. Kõigepealt oli vaja leida koht, kuhu see taustarakendus käima panna. Autori esialgne idee oli kasutada juba olemasolevat veebiserverit. Sellel lahendusel oli kahjuks kaks suurt viga. Esiteks polnud serveri riistvara suure kasutajate arvu juures piisavalt võimekas. Seda probleemi annaks lahendada komponentide välja vahetamisega, aga see oleks kujunenud rahaliselt liiga kulukaks. Teiseks suureks probleemiks oli avalik ligipääs sellele serverile. Selle tarvis oleks vaja osta interneti teenusepakkujalt staatilise IP-aadressi teenus ja avada läbi ruuteri pordid, et oleks võimalik vabalt otspunktidele päringuid teha. Töö praeguse staadiumi juures läheks see aga üpris kiirelt üpris kulukaks, mistõttu oli mõistlikum kasutada mõnd pilvteenusepakkujat.

Valik langes Kamatera [44] kasuks, sest nad pakkusid väga paindliku ja taskukohast teenust, võimaldades vajaminevaid teenuseid väga täpselt kasutuse järgi konfigureerida. Seadistuse käigus sai valida protsessorite hulka, mälu ja vahemälu suurust ning millist tarkvara soovitakse kasutada. Kõik edasine oli juba sama lihtne nagu kohaliku serveri kasutamisel.

Rakenduse lihtsamaks kasutamiseks ja haldamiseks on kasutatud Dockeri platvormi. Rakenduse Dockeri konteineris jooksutamine tagab, et rakendus toimib samamoodi olenemata

masinast, millel konteinerit jooksutatakse. [45] Lisaks sellele on kasutatud Docker Compose [46] tööriista, mille abil saab ühe või mitu Dockeri konteinerit ühe käsklusega käima panna. See välistab vigasid erinevate rakenduse osade konfiguratsioonis ja tagab, et kõik teenused käivituksid alati ühtemoodi. Docker Compose'i abil käivitatakse rakendus järkjärgult ning järgmine komponent ei hakka enne tööle, kui eelmine pole käima läinud. Vea tekkimisel rakenduse käivitamine tühistatakse.

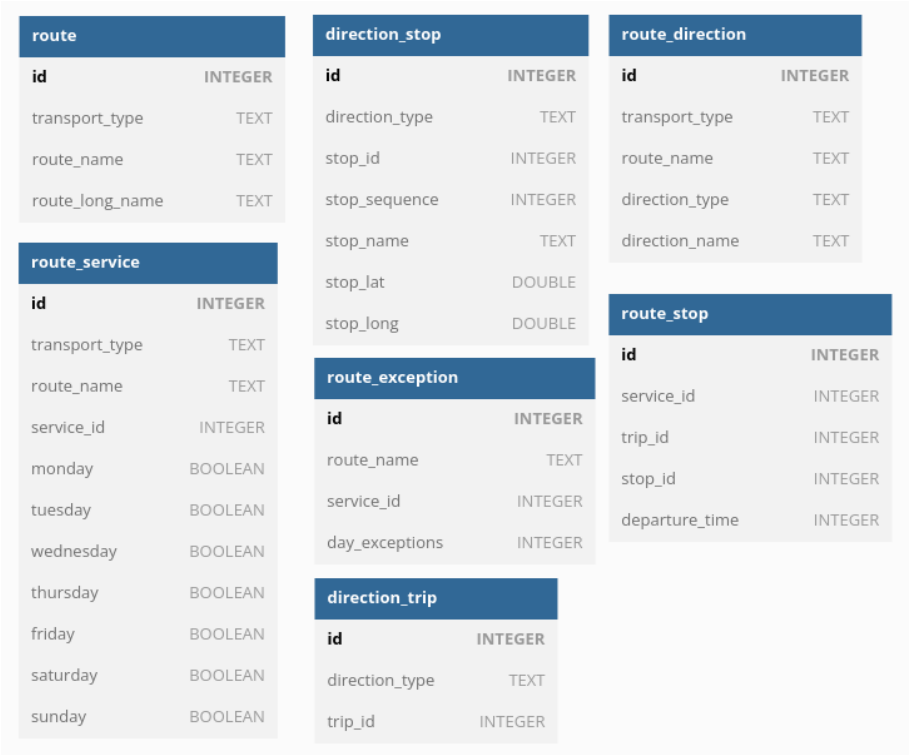
Taustarakenduse piisavaks toimimiseks oleks sellest täiesti piisanud, aga tulevikuperspektiivist ja kasutusmugavuse huvides oli otstarbekas siduda server mõne domeeniga. Domeeni registreerimise teenuseid on nii Eestis kui ka välismaal väga palju. Suurimad välismaised teenused on GoDaddy, Namecheap, Tucows ja Google [47]. Eestis on kõige tuntumad registripidajad Zone ja Veebimajutus. Pole tähtsust, läbi millise teenuse domeen registreerida, kuna registripidajate teenused on sarnased ning erinevad ainult hinna poolest. Domeen registreeriti Veebimajutuse kaudu ning serveri aadressi sidumine domeeninga oli väga triviaalne.

Parema turvalisuse tagamiseks kasutatakse serveri ja kasutaja vahel pöördproksit. Pöördproksi eesmärk on vahendada rakenduse ja kasutajavahelist suhtlust. Kasutaja tehtud päring jõuab proksisse, mis suunab selle õige rakenduse poole. Saadud vastus jõuab tagasi proksisse, mis suunab selle kasutajale edasi. See aitab kaitsta rakendust teenustõkestusrünnakute vastu, sest rakenduse IP-aadressi pole päringu sooritajal võimalik leida. Nähtaval on ainult pöördproksi aadress. Lisaks saab seda kasutada andmete vahemällu salvestamiseks, et vältida ressursimahukate toimingute liigset tegemist. [48]

4.4 Mobiilirakenduse arendamine

Mobiilirakenduse arenduse alguses tuli paika panna rakenduse disainikeel ja raamistik. Disainikeele osas langes valik kiiresti Material Design'i peale, sest see on väga põhjalikult dokumenteeritud [49]. See valik lihtsustas raamistiku valikut märkimisväärselt. Material Designi komponentide teek on saadaval kolmel raamistikul: MDC-Android, Jetpack Compose ja Flutter. Valik langes Jetpack Compose'ile, sest selle dokumentatsioon [50] on kõige põhjalikum ja see tundus esmapilgul kõige lihtsamini kasutatav.

Järgmise etapina tuli hakata looma erinevaid vaateid mikroteenusest tulevate andmete kuvamiseks. Jetpack Compose'i raamistik võimaldab väga lihtsasti luua erinevaid väiksemaid komponente, mida hiljem rakenduse vaadetes ära kasutada. Peale vajalike vaadete loomist oli vaja välja mõelda, kuidas hoiustada andmeid lokaalselt, et oleks võimalik rakendust ka võrguühenduseta kasutada. Nii suure andmehulga jaoks oli mõistlik luua lokaalne andmebaas vastavalt mikroteenusest tulnud andmetele.



Joonis 6. Rakenduse andmebaasi füüsiline andmemudel

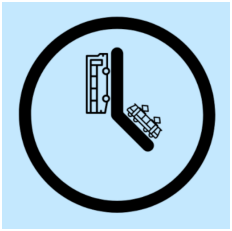
Andmebaas loodi SQLite teeki kasutades. SQLite'i eripäraks on andmebaaside loomine tavamällu, mistõttu pole vaja luua eraldi protsessi andmebaasi jaoks. Lisaks võtab see rakenduses väga vähe mälu, mistõttu on see Android rakendustes väga populaarne andmete hoiustamise viis. Loodud andmebaas (Joonis 6) võimaldab jätkata rakenduse kasutamist ka võrguühenduseta. Rakenduse andmebaasi uuendatakse automaatselt 24 tunni möödudes pärast viimast uuendamist avakuvale minnes. Sätete lehel on võimalik andmete uuendamine manuaalselt välja kutsuda. Rakendusse implementeeriti paralleelselt ka tume teema, et parendada rakenduse kasutamist pimedas.

Järgmine suur samm oli rakenduse lokalisserimise implementeerimine. Android rakendustes saab lisada erinevate lokalisatsioonide jaoks faile, kus on kirjeldatud sõne ressursid muudes keeltes kui seda rakenduse põhikeel. Esialgu valis rakendus lokalisatsiooni vastaval seadme keelele ning selle puudumisel kasutati eesti keelt. Hiljem lisati sätetesse võimalus valida rakenduse keel.

5. Tulemused

Töö tulemusel valminud mobiilirakendus toetab Android operatsioonisüsteemi versiooni 8.0 ja uuemaid, mida statistika [51] kohaselt kasutab 93,3% Android seadmetest. Minimaalne versioon sai valitud seetõttu, et see on esimene suurem arengusuuna muutus alates verisoonist 5.0.

Rakenduse logo (Joonis 7) kujutab kella, mille seieritel on buss ja tramm. Kuna rakenduse idee on kuvada ühistranspordi sõiduplaane ja paiknemisi, siis sobivad kell ja ühisõidukid seda ideaalselt sümboliseerima. Rakenduse keel sõltub esmasel käivitamisel seadme Android süsteemi keelesättest. Kui süsteemi keelt pole rakenduses saadaval, siis on rakenduse vaikimisi keeleks eesti keel. Dokumendis on illustratsioonideks kasutatud kuvatõmmiseid rakenduse eestikeelsest versioonist.



Joonis 7. Rakenduse ikoon

Rakendus on jagatud seitsmeks põhiliseks vaateks: liinide, peatuste, sõiduplaanide, teekonna, sätete, kaardivaade ja avavaade. Iga vaate funktsionaalsust ja omadusi on kirjeldatud järgnevates alapeatükkides.

5.1 Avavaade

Rakenduse avamisel kuvatakse kasutajale avavaade (Joonis 8). Võrguühenduse olemasolul tehakse rakenduse esmasel käivitamisel päring mikroteenusele, et täita andmebaas võrguühenduseta kasutamiseks andmetega. Päring tehakse ka siis, kui olemasolevad andmed on vanemad kui 24 tundi. Muudes olukordades päringut ei tehta.

Avavaate põhiosa koosneb kuuest nupust, mis suunavad kasutaja edasi valitud transporditüübi liinide vaatesse. Nupule vajutades on edasised vaated värvilahenduselt sama värvi nagu nupp. Vaate alumises paremas nurgas nähtaval veel kaks nuppu. Ülemine nupp, millel on hammasratta ikoon, avab sätete vaate. Selle all olev nupp tekstiga "TAGASISIDE", avab tagasisidevormi. Sinna saab kasutaja märkida oma arvamuse rakendusest,

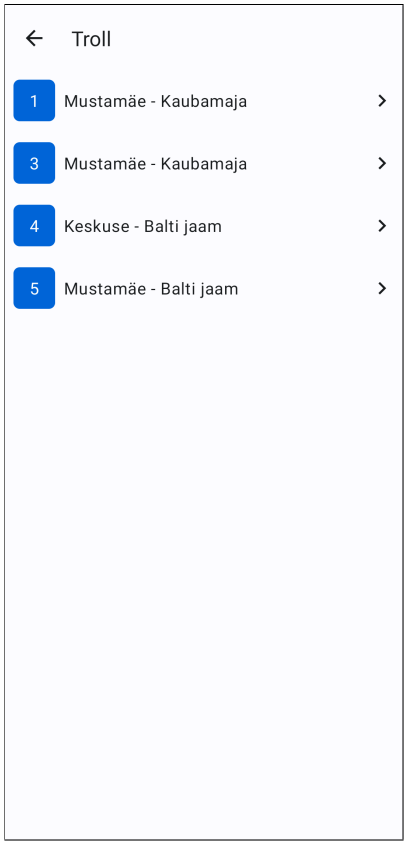
jätta vabas vormis tagasisidet ning parandus- ja täiendussoovitusi.

Joonis 8. Rakenduse avavaade

5.2 Liinide vaade

Pärast transporditüübi valimist liigub rakendus edasi liinide vaatesse (Joonis 9). Selles vaates kuvatakse kasutajale andmete olemasolul loetelu transporditüübi liinidest. Andmete puudumisel kuvatakse ainult ülemine navigatsiooniriba. Vaate ülemises servas on riba, mille vasakus servas asub nupp eelmisele vaatele liikumiseks. Selle kõrval on tekst hetkel valitud transporditüübiga.

Navigatsiooniriba all kuvatakse hetkel valitud transporditüübi liinide andmed. Liini rea vasakus servas kuvatakse liini number. Selle kõrval on liini alg- ja lõpp-peatusest koosnev liini nimi ning ikoon, mis viitab sellele, et vajutus viib edasi järgmisesse vaatesse.



Joonis 9. Rakenduse liinide vaade

5.3 Peatuste vaade

Pärast liini valimist liigub rakendus peatuste vaatesse (Joonis 10). Peatuste vaade koosneb kahest osast: ülmenine navigatsiooniriba ja peatuste loetelu. Navigatsiooniriba vasakus servas asub nupp eelmisele vaatele liikumiseks. Selle kõrval asub tekst liini numbriga ning liini nimega. Teksti kõrval asub veel kaks nuppu. Vasakpoolne nupp avab liini kaardivaate ning parempoolne nupp muudab liini suunda. Kui liinil on täpselt kaks suunda, siis toimub vahetus otsekohe. Rohkemate suundade olemasolul avaneb aken, kus saab valida, millist liini soovitakse näha.

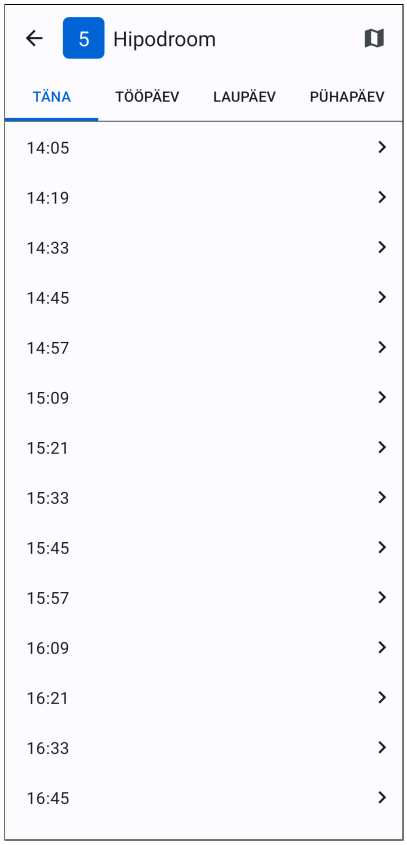
Navigatsiooniriba all asub loetelu liini peatustest. Paksus kirjas on märgitud liini esimene ja viimane peatus. Loetelu reas on kirjas liini nimetus ning selle kõrval on ikoon, mis viitab järgmise vaate avamisele.

Joonis 10. Rakenduse peatuste vaade

5.4 Sõiduplaanide vaade

Peatuse valimise järgsel avaneb sõiduplaanide vaade (Joonis 11). Navigatsiooniribal astseb vasakus servas eelmisele vaatele tagasi liikumise nupp. Selle kõrval on tekst liini numbri ja peatuse nimega. Riba paremas servas on kaardivaate nupp, mis avab peatuse asukoha kaardil.

Navigatsiooniriba all paikneb veel üks riba vähemalt ühe sektsiooniga. Esimene sektsioon on alati sama päeva järgmised väljumised. Väljumiste puudumisel kuvatakse tekst "Täna rohkem väljumisi ei ole". Selle sektsiooni kõrval võib asuda nädalapäeva nimega või üldise nimega sektsioone, mille all kuvatakse kõik peatumisest väljumise kellaajad. Näiteks sektsioon "TÖÖPÄEV" all kuvatakse kõik peatusest väljumised igal tööpäeval. Kellaaegade kõrval on ikoon, mis viitab järgmisele vaatele.



Joonis 11. Rakenduse sõiduplaanide vaate

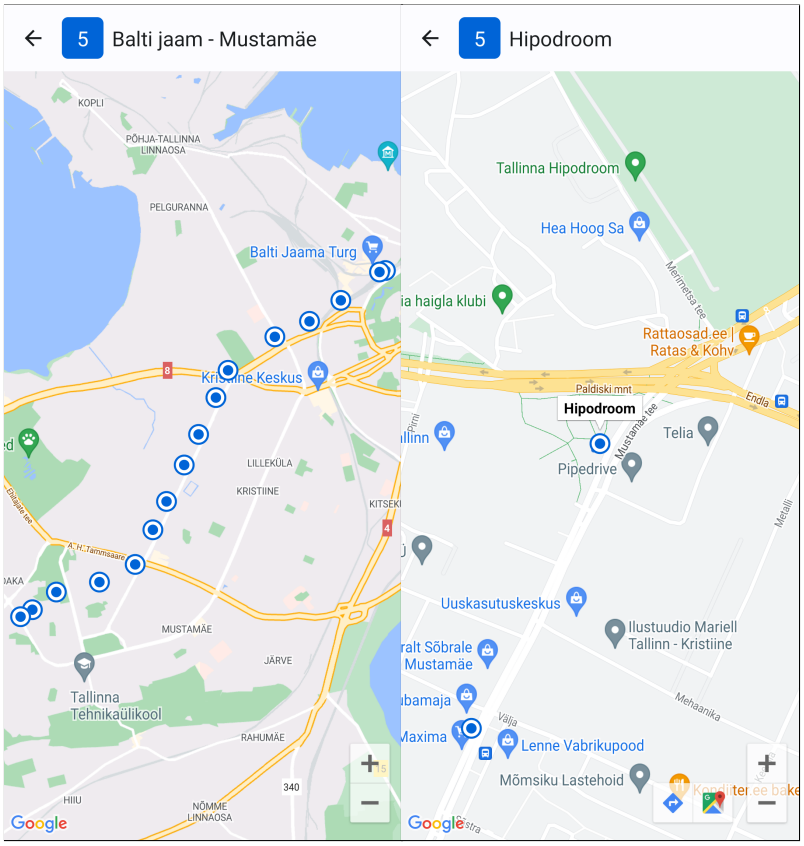
5.5 Teekonna vaade

Peatusest väljumise kellaajale vajutades avatakse liini teekonna vaade (Joonis 12). Vaates kujutatakse ühe kindla reisi jõudmisi liini peatustesse. Liini alg- ja lõpp-peatused on märgitud poolpaksus kirjas ja hetkel valitud peatus on märgitud paksus kirjas. Peatuste nimedest vasakul kuvatakse ühissõiduki peatusesse jõudmise kellaaega.

Joonis 12. Rakenduse teekonna vaade

5.6 Kaardivaade

Kaardivaateid on rakenduses kahte tüüpi (Joonis 13). Esiteks on liini teekonna vaade, kus kuvatakse kaardil korraga terve liini peatuste asukohad. Teiseks on peatuse kaardivaade, kus suurendatakse vaade avatud peatuse asukohani. Mõlemaid kaarte saab suurendada ja vähendada ning ka pöörata. Peatuse ikoonile vajutades avaneb kaardil selle peatuse nimi ning veel kaks nuppu. Mõlemad nupud avavad Google Maps rakenduse. Vasakpoolne tekitab juhised selle peatuseni ning parempoolne avab lihtsalt peatuse asukoha Google Maps rakenduses.

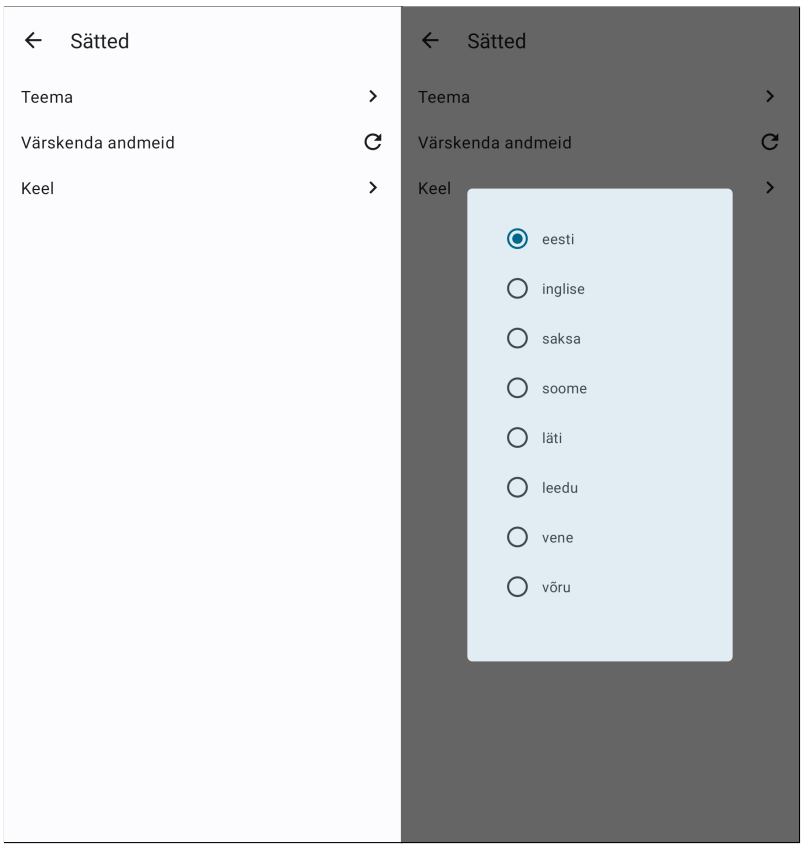


(a) (b)

Joonis 13. Rakenduse (a) teekonna ja (b) peatuse kaardivaade

5.7 Sätete vaade

Sätete vaatesse (Joonis 14) pääseb rakenduses avavaatest. Vaates on võimalik nupuvajutusel vahetada rakenduse teemat, värskendada andmeid ning vahetada rakenduse keelt. Rakenduse keele nupule vajutades avaneb loetelu keelevalikutest. Mõnel teises keelel vajutades muutub rakenduse lokalisatsioon. Rakenduse teema ja keelevalik on püsivad andmed. See tähendab, et ka rakenduse sulgemisel jäävad valitud sätted püsima. Muudatused kaovad ainult rakenduse andmete kustutamisel.



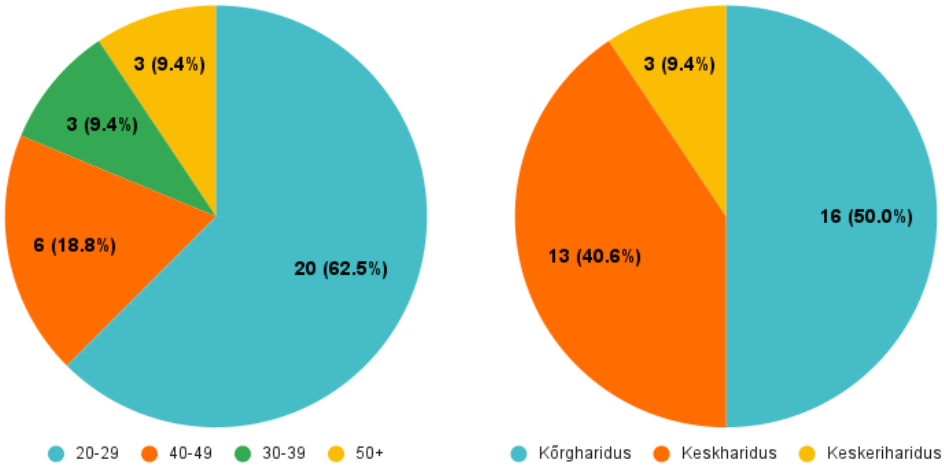
(a) (b)

Joonis 14. Rakenduse (a) sätete ja (b) keelevaliku vaade

5.8 Rakenduse valideerimine lõppkasutajatega

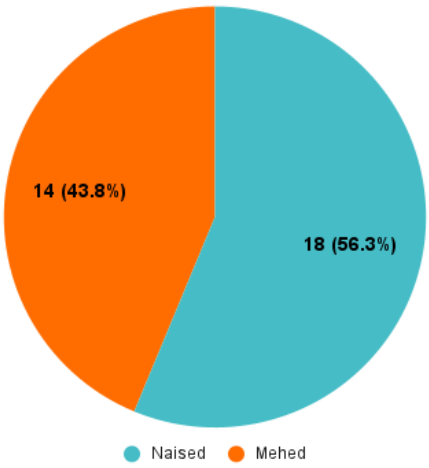
Rakenduse valideerimiseks ja saadud tagasiside analüüsiks koostati tagasisideküsimustik, mida oli võimalik kasutajatel rakenduse kasutamisel täita. Küsimustikus koguti kasutajate sugu, vanusevahemikke, nende haridustaset ja vabas tekstis tagasiside rakenduse positiivsete ja negatiivsete külgede ning vajalike täienduste kohta. Saadud vastused on kvalitatiivselt analüüsitavad. Rakendust testis arenduse käigus 32 eri kasutajat.

Kõige suurem osa küsitlusele vastajatest jäi vanusevahemikku 20–29 aastat, kokku 20 inimest (Joonis 15). Teistest vanusevahemikest oli suurim 40–49 aastased, kokku kuus vastajat. 30–39 aastaste ja vanematest kui 50 aastaste hulgast vastas küsitlusele mõlemast vahemikust kolm inimest. Haridustaseme poolest jagunesid vastajad sisuliselt kahte leeri: kõrg- ja keskharidus (Joonis 15). Kõrgharidusega vastajaid oli 16, mis oli täpselt 50% vastajatest. Vastanutest 13-l oli keskharidus ning ülejäänul kolmel vastajal oli keskeriharidus. Küsitlusele vastas 18 naist ja 14 meest (Joonis 16).



(a) (b)

Joonis 15. Testijate vanusevahemikud (a) ja haridustase (b)



Joonis 16. Testijate sugu

Suurem osa saadud tagasisidest oli väga positiivne. Enim märgiti vastustes positiivse küljena ära rakenduse kasutuskiirus ja lihtsus. Nooremad kasutajad tõid välja, et kasutajaliides on väga puhta ja loogilise ülesehitusega. Vanemad kasutajad märkisid positiivseks küljeks, et suurema kirjasuurusega ei lähe rakendus katki ega midagi paigast ära.

Peale esimest suuremat tagasiside saamist toodi välja ka mõned rakenduse nõrgemad küljed ja probleemid, mis esinesid testimise käigus. Rakenduse esimesel käivitamisel ilmnes mõne telefoniga probleem, kus mõningaid andmeid oli lokaalses andmebaasis dubleeritud. Mõne transporditüübi alla tekkis samasid liine kaks korda ning liinidel oli iga peatust kaks korda. See omakorda tegi peatuse valimise natuke katki, aga samas

ei jooksutanud rakendust kokku. Probleemi lahendas andmete uuesti värskendamine seadest. Ühel kasutajal tekkis olukord, kus rakendus ütles, et puudub serveriga ühendus, kuigi kellelgi teisel sellist probleemi ei tekkinud. Lõpuks tuli välja, et kasutaja töökoha võrgusätted keelasid turvasertifikaadita otspunktidele päringute tegemise. Arenduse käigus puudus mikroteenuse otspunktil vajalik turvasertifikaat, mis lisati arenudse lõppfaasis.

Tagasisidest tuli välja paar ideed, mida saaks kasutada rakenduse edasiseks arendamiseks. Esiteks märgiti, et liine võiks saada lemmikutesse lisada. See on täiesti loogiline ja kasulik asi, mida rakendusse implementeerida. Arvestades sellega, kui palju ühistranspordi liine on Tallinnas ja Harjumaal, siis see teeks rakenduse kasutamise märkimisväärselt lihtsamaks. Seda eriti olukordades, kus põhiliselt kasutataksegi ainult paari liini, siis pole vaja mitme vaate vahel liigelda.

Tagasisides mainiti veel, et võiks olla liinide otsing. See on samuti väga mõistlik idee, sest see aitaks vähendada erinevate vaadete vahel liikumist, mis omakorda kiirendaks sõiduplaanide leidmise protsessi. Selle saaks tõenäoliselt implementeerida koos eelnevalt mainitud liinide lemmikuks lisamisega.

Tulevikuperspektiivis võiks tõenäoliselt rakenduse andmekogu kolida Tallinna Transpordiameti andmekogult Eesti Transpordiameti andmekogule. See võimaldaks rakendusse lisada terve Eesti ühistranspordi sõiduplaanid ja ei peaks piirduma ainult Tallinna ja Harjumaaga. Mikroteenuses tähendaks see tõenäoliselt üsna minimaalseid muudatusi, sest mõlemad andmekogu põhinevad GTFS süsteemil. Mobiilirakenduses ilmselt piisaks avavaate ringi tegemisest. Praeguste staatiliste nuppude asmel tuleks tekitada vastavalt linnale või regioonile dünaamiliselt genereerivad nupud.

6. Kokkuvõte

Bakalaureusetöö eesmärgiks oli luua mobiilirakendus Tallinna ja Harjumaa ühistranspordi andmete kuvamiseks koos mikroteenusega, mis töötleb andmeid ja võimaldab neid mobiilirakenduses kasutada.

Töö tulemusena valmis Android operatsioonisüsteemil töötav mobiilirakendus, mis suudab kuvada ühistranspordi sõiduplaane. Sõiduplaanide kuvamine on kiire, sest andmed salvestatakse lokaalselt ning neid uuendatakse kord päevas või soovi korral varem. Lisaks valmis Micronaut raamistikul mikroteenus, mis töötleb avaandmetest kogutud informatsiooni rakenduse sujuvaks tööks vajalikule kujule. Mikroteenus värskendab kord päevas andmeid ning eemaladab aegunud andmed enda andmebaasist.

Töö lõppfaasis sai hulk kasutajaid rakendust proovida, et avastada võimalikke vigasid ning parendada funktsionaalsust ja mugavust. Tagasiside rakendusele oli võrdlemisi positiivne. Tehti ka mitu täiendusettepanekut, mis lähevad tulevikus kindlasti arendusse. Loodud rakendus on avaldatud Google Play Store'i keskkonnas ning leitav nime alt Transport Info.

Rakenduse edasisel arendamisel võiks lisada juurde kasutust mugavamaks tegevaid funktsioone nagu näiteks liinide lemmikutesse lisamine, otsing ja Tallinna Transpordiameti ühistranspordiuudiste näitamine rakenduse avakuval.

Kasutatud kirjandus

[1] H. Keerutaja. "Trafi – asendamatu äpp ühistranspordiga liiklejale." (2017), [Online]. Loetud aadressil: https://tehnika.postimees.ee/4295995/trafi-asendamatu-app-uhistranspordiga-liiklejale (kasutatud 04/16/2023).

[2] "Kuues tallinna bussipeatuses hakkasid tööle elektroonilised tablood." (2012), [Online]. Loetud aadressil: https://www.err.ee/338297/kuues-tallinna-bussipeatuses-hakkasid-toole-elektroonilised-tablood (kasutatud 04/16/2023).

[3] "Tallinn plaanib 50 ühissõiduki peatusesse elektroonilisi infotabloosid." (2017), [Online]. Loetud aadressil: https://www.err.ee/840355/tallinn-plaanib-50-uhissoiduki-peatusesse-elektroonilisi-infotabloosid (kasutatud 04/16/2023).

[4] *Ühistranspordiregistri avaandmete spetsifikatsioon*, Maanteeamet, 2020.

[5] "Kkk – kuidas toimib reisiplaneerija peatus.ee?" Transpordiamet. (2022), [Online]. Loetud aadressil: https://www.transpordiamet.ee/kuidas-toimib-reisiplaneerija (kasutatud 04/20/2023).

[6] "Tallinn kavandab ühissõidukite reaalajas jälgimise süsteemi," BNS. (2008), [Online]. Loetud aadressil: https://www.postimees.ee/1804699/tallinn-kavandab-uhissoidukite-reaalajas-jalgimise-susteemi (kasutatud 04/11/2023).

[7] "Autorile viitamine + jagamine samadel tingimustel 3.0 jurisdiktsiooniga sidumata (cc by-sa 3.0)." (2023), [Online]. Loetud aadressil: https://creativecommons.org/licenses/by-sa/3.0/deed.et (kasutatud 04/26/2023).

[8] *Nptg and naptan schema guide*, NaPTAN NPTG, 2012. [Online]. Loetud aadressil: http://naptan.dft.gov.uk/naptan/schema/2.4/doc/NaPTANSchemaGuide-2.4-v0.57.pdf (kasutatud 04/18/2023).

[9] "Our open data," Transport for London. (2023), [Online]. Loetud aadressil: https://tfl.gov.uk/info-for/open-data-users/our-open-data (kasutatud 04/18/2023).

[10] "Öbb - beam mich heim, scotty!" Konsument. (2009), [Online]. Loetud aadressil: https://konsument.at/auto-transport/oebb (kasutatud 04/18/2023).

[11] "Öbb scotty," ÖBB. (2023), [Online]. Loetud aadressil: https://www.oebb.at/en/fahrplan/fahrplanauskunft/scottymobil (kasutatud 04/18/2023).

[12] M. Akhromieieva. "Db introduces real-time seat occupancy displays on first routes in regional transport." (2023), [Online]. Loetud aadressil: https://www.railtarget.eu/technologies-and-infrastructure/db-introduces-realtime-seat-occupancy-displays-on-first-routes-in-regional-transport-4281.html (kasutatud 04/18/2023).

[13] "Sõiduplaanid." (2023), [Online]. Loetud aadressil: https://transport.tallinn.ee (kasutatud 04/26/2023).

[14] "Eesti avaandmete teabevärav." (2022), [Online]. Loetud aadressil: https://avaandmed.eesti.ee (kasutatud 04/16/2023).

[15] "Tallinna avaandmed." (2022), [Online]. Loetud aadressil: https://web.archive.org/web/20220819142325/https://avaandmed.tallinn.ee/ (kasutatud 04/16/2023).

[16] "Tallinna ühistranspordi peatused ja marsruudid." (2023), [Online]. Loetud aadressil: https://avaandmed.eesti.ee/datasets/tallinna-uhistranspordi-peatused-ja-marsruudid (kasutatud 04/16/2023).

[17] "Tallinna ühistranspordi peatused ja marsruudid." (2023), [Online]. Loetud aadressil: https://avaandmed.eesti.ee/datasets/uhistranspordivahendite-asukohad-reaalajas (kasutatud 04/16/2023).

[18] "Ühistranspordi infosüsteem," Transpordiamet. (2022), [Online]. Loetud aadressil: https://www.transpordiamet.ee/uhistranspordi-infosusteem (kasutatud 04/20/2023).

[19] "General transit feed specification - background," General Transit Feed Specification. (2023), [Online]. Loetud aadressil: https://gtfs.org/background/ (kasutatud 04/22/2023).

[20] "What is git?" Atlassian. (2023), [Online]. Loetud aadressil: https://www.atlassian.com/git/tutorials/what-is-git (kasutatud 04/12/2023).

[21] E. Enietan. "Alternatives to git." (2022), [Online]. Loetud aadressil: https://dev.to/niza/alternatives-to-git-4p6m (kasutatud 04/12/2023).

[22] R. Bailey. "Overview of redundant systems." (2020), [Online]. Loetud aadressil: https://www.atlantic.net/dedicated-server-hosting/overview-of-redundant-systems/ (kasutatud 04/13/2023).

[23] S. Zheng. “Hackers claim theft of police info in china’s largest data leak.” (2022), [Online]. Loetud aadressil: https://www.bloomberg.com/news/articles/2022-07-04/hackers-claim-theft-of-police-info-in-china-s-largest-data-leak (kasutatud 04/13/2023).

[24] Z. Whittaker. “Amazon accidentally exposed an internal server packed with prime video viewing habits.” (2022), [Online]. Loetud aadressil: https://techcrunch.com/2022/10/27/amazon-prime-video-server-exposed/ (kasutatud 04/13/2023).

[25] “What is a rest api?” IBM. (2023), [Online]. Loetud aadressil: https://www.ibm.com/topics/rest-apis (kasutatud 04/13/2023).

[26] B. Oliveira. “Google i/o 2014 app source code now available.” (2014), [Online]. Loetud aadressil: https://android-developers.googleblog.com/2014/07/google-io-2014-app-source-code-now.html (kasutatud 04/15/2023).

[27] “Unveiling material you,” Material Design Blog. (2021), [Online]. Loetud aadressil: https://material.io/blog/announcing-material-you (kasutatud 04/15/2023).

[28] C. Haase. “Google i/o 2019: Empowering developers to build the best experiences on android + play.” (2019), [Online]. Loetud aadressil: https://android-developers.googleblog.com/2019/05/google-io-2019-empowering-developers-to-build-experiences-on-Android-Play.html (kasutatud 04/16/2023).

[29] “Android’s kotlin-first approach,” Google Developers. (2023), [Online]. Loetud aadressil: https://developer.android.com/kotlin/first (kasutatud 04/16/2023).

[30] “Corporate overview,” Jetbrains. (2023), [Online]. Loetud aadressil: https://resources.jetbrains.com/storage/products/jetbrains/docs/jetbrains_corporate_overview.pdf (kasutatud 04/17/2023).

[31] “Flutter faq,” Flutter. (2023), [Online]. Loetud aadressil: https://docs.flutter.dev/resources/faq (kasutatud 04/17/2023).

[32] “Web accessibility,” European Commission. (2023), [Online]. Loetud aadressil: https://digital-strategy.ec.europa.eu/en/policies/web-accessibility (kasutatud 04/08/2023).

[33] “Pimedad ja vaegnägijad,” European Commission. (2020), [Online]. Loetud aadressil: https://what-europe-does-for-me.eu/et/portal/2/C16 (kasutatud 04/08/2023).

[34] "Tervise infosüsteem." (2023), [Online]. Loetud aadressil: https://www.tehik.ee/tervise-infosusteem (kasutatud 04/23/2023).

[35] R. Liive. "Pimedate liit ja tehik hakkavad otsima lahendust e-teenustes esinevatele probleemidele." (2021), [Online]. Loetud aadressil: https://tervise.geenius.ee/rubriik/uudis/pimedate-liit-ja-tehik-hakkavad-otsima-lahendust-e-teenustes-esinevatele-probleemidele/ (kasutatud 04/08/2023).

[36] "Check text and background for sufficient color contrast," Deque University. (2023), [Online]. Loetud aadressil: https://dequeuniversity.com/tips/color-contrast (kasutatud 04/08/2023).

[37] K. Alamets. "Tallinna turism 2021: Linnakülastajad." (2021), [Online]. Loetud aadressil: https://file.visittallinn.ee/7sw8ci/turismiaaasta-2021-slaidid.pdf (kasutatud 04/16/2023).

[38] "Why app localization is good for mobile app success," Ulatus. (2022), [Online]. Loetud aadressil: https://www.ulatus.com/translation-blog/why-app-localization-is-good-for-mobile-app-success/ (kasutatud 04/17/2023).

[39] R. Costa. "Quantitative vs qualitative testing: Validating our steps." (2020), [Online]. Loetud aadressil: https://www.justinmind.com/blog/qualitative-quantitative-testing/ (kasutatud 05/03/2023).

[40] "Static transit." (2023), [Online]. Loetud aadressil: https://developers.google.com/transit/gtfs/reference (kasutatud 04/26/2023).

[41] T. Laiksoo, "Tallinna ühistransport muutub kasutajasõbralikumaks," *Keskkonnatehnika*, vol. 2, pp. 15–16, 2007.

[42] "Micronaut vs springboot: Which one is better?" Espeo. (2022), [Online]. Loetud aadressil: https://espeo.eu/blog/micronaut-vs-springboot/ (kasutatud 04/18/2023).

[43] "Beautiful database diagrams." (2023), [Online]. Loetud aadressil: https://drawsql.app (kasutatud 05/18/2023).

[44] "Company profile." (2023), [Online]. Loetud aadressil: https://www.kamatera.com/Company_Profile (kasutatud 05/04/2023).

[45] "Why docker," Docker. (2023), [Online]. Loetud aadressil: https://www.docker.com/why-docker/ (kasutatud 04/21/2023).

[46] "Docker compose overview." (2023), [Online]. Loetud aadressil: https://docs.docker.com/compose/ (kasutatud 05/05/2023).

[47] C. Preusler. “20 largest domain registrars in the world (may 2023).” (2023), [Online]. Loetud aadressil: https://www.hostingadvice.com/how-to/largest-domain-registrars/ (kasutatud 04/21/2023).

[48] “What is a reverse proxy?” Cloudflare. (2023), [Online]. Loetud aadressil: https://www.cloudflare.com/learning/cdn/glossary/reverse-proxy/ (kasutatud 04/21/2023).

[49] “Material design.” (2023), [Online]. Loetud aadressil: https://m3.material.io (kasutatud 05/05/2023).

[50] “Get started with jetpack compose.” (2023), [Online]. Loetud aadressil: https://developer.android.com/jetpack/compose/documentation (kasutatud 05/07/2023).

[51] “Android api levels.” (2023), [Online]. Loetud aadressil: https://apilevels.com (kasutatud 05/11/2023).

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina Risto Reiljan

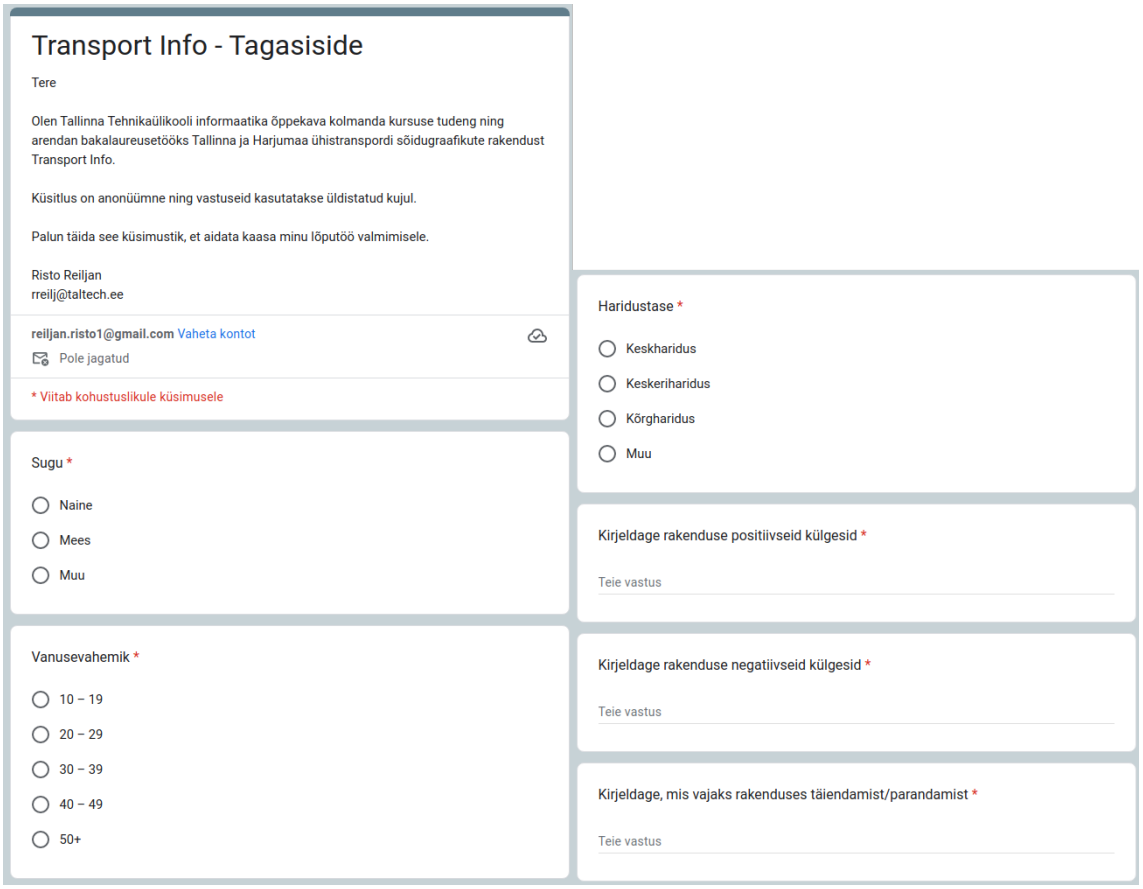
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Mobiilirakendus ühistranspordiandmete haldamiseks ja vaatamiseks", mille juhendaja on Elli Valla .
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu 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.

22.05.2023

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

Lisa 2 – QR kood rakenduse leidmiseks Google Play Store keskkonnast

Lisa 3 – Rakenduse tagasiside vorm



Transport Info - Tagasiside

Tere

Olen Tallinna Tehnikaülikooli informaatika õppekava kolmanda kursuse tudeng ning arendan bakalaureusetööks Tallinna ja Harjumaa ühistranspordi sõidugraafikute rakendust Transport Info.

Küsitlus on anonüümne ning vastuseid kasutatakse üldistatud kujul.

Palun täida see küsimustik, et aidata kaasa minu lõputöö valmimisele.

Risto Reiljan
rreilj@taltech.ee

reiljan.risto1@gmail.com Vaheta kontot

Pole jagatud

* Viitab kohustuslikule küsimusele

Sugu *

- Naine
- Mees
- Muu

Vanusevahemik *

- 10 – 19
- 20 – 29
- 30 – 39
- 40 – 49
- 50+

Haridustase *

- Keskharidus
- Keskeriharidus
- Kõrgharidus
- Muu

Kirjeldage rakenduse positiivseid külgesid *

Teie vastus

Kirjeldage rakenduse negatiivseid külgesid *

Teie vastus

Kirjeldage, mis vajaks rakenduses täiendamist/parandamist *

Teie vastus