

TALLINNA TEHNICAÜLIKOO
Infotehnoloogia teaduskond

Piia Carmen Milkov 232815IABB

Gisela Nüüd 233346IABB

**Kasutaja liikumisgraafikul põhineva
elektrisõiduki laadimise optimeerimislahenduse
analüüs ja edasiarendus Enefit AS näitel**

Bakalaureusetöö

Juhendaja: Liisa Jõgiste

MSc

Tallinn 2026

Autorideklaratsioon

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

Autorid: Piia Carmen Milkov, Gisela Nüüd

17.05.2026

Lühikokkuvõte

Käesoleva bakalaureusetöö eesmärgiks on analüüsida ja edasi arendada Enefit AS tellimusel valminud meeskonnaprojekti lahendust, mis optimeerib elektrisõiduki laadimisplaani kasutaja liikumisgraafiku põhjal.

Olemasoleva lahenduse täiustamiseks viisid autorid läbi kasutajauuringud elektrisõidukeid igapäevaselt kasutavate eraisikute ja ettevõtete seas. Kuna olemasolevad turulahendused ei arvesta laadimise optimeerimisel kasutaja liikumisgraafikuga, keskenduti töös rakenduse uute funktsionaalsuste loomisele ning selle konkurentsivõime hindamisele. Uuendatud lahenduse toimivust valideeriti kasutajatestide abil.

Töö käigus analüüsiti olemasolevaid turulahendusi ning kaardistati Enefit AS rakenduse edasiarenduse nõuded. Nõuete põhjal parandati meeskonnaprojektis loodud rakenduse kitsaskohad ning realiseeriti kolm uut põhifunktsionaalsust: sõiduki andmete automaatne täitmine, laadimiskulude kokkuvõte ning päevase läbisõidu määramise liugur. Ettevõtete kasutusjuhu toetamiseks loodi võimalus importida mitme sõiduki liikumisgraafikuid CSV-failina. Ehkki liikumisgraafikul põhineval laadimise optimeerimisrakendusel on potentsiaali, jääb selle sihtgrupp esialgu võrdlemisi kitsaks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 60 leheküljel, 9 peatükki, 18 joonist, 3 tabelit.

Abstract

Analysis and Development of an Electric Vehicle Charging Optimization Solution Based on User-Defined Mobility Needs on the Example of Enefit AS

The aim of this bachelor's thesis is to analyze and further develop a team project solution commissioned by Enefit AS, which optimizes an electric vehicle's charging plan based on the user's movement schedule.

To improve the existing solution, the authors conducted user studies among private individuals and companies that use electric vehicles daily. Since existing market solutions do not account for the user's movement schedule when optimizing charging, the work focused on creating new functionalities for the application and evaluating its competitiveness. The performance of the updated solution was validated using user tests.

During the work, existing market solutions were analyzed and the requirements for the further development of the Enefit AS application were mapped. Based on the requirements, the shortcomings of the application created in the team project were fixed, and three new main functionalities were implemented: automatic filling of vehicle data, a summary of charging costs, and a slider for setting daily mileage. To support corporate use cases, an option was created for importing the movement schedules of multiple vehicles as a CSV file. Although the movement schedule-based charging optimization application has potential, its target group is currently relatively narrow.

The thesis is in Estonian and contains 60 pages of text, 9 chapters, 18 figures, 3 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides
CSV	<i>Comma-separated value</i> , komadega eraldatud väärtused
Eraklient	Isik, kes laeb oma elektriautot kodus
Intervall	15-minutit
ISA	Infosüsteemide arendamine
Laadimisaken	15-minutilised järjestikku intervallid, mille ajal toimub sõiduki laadimine
Laadimisplaan	Kasutajale kuvatavad optimeeritud laadimisajad
Liikumisgraafik	Kasutaja sisestatud sündmustest formuleeruv sõidukiga läbitav graafik.
<i>localStorage</i>	Veebilehitseja kohalik mälu
LP	Lineaarne programmeerimine
OCPP	<i>Open Charge Point Protocol</i> , võimaldab laadijal suhelda erinevate süsteemidega
Persoon	Kujutletav kasutaja, kes esindab suuremat sihtgruppi
PWA	<i>Pogressive Web App</i> , progressiivne veebirakendus
SPA	<i>Single-Page Application</i> , üheleheline veebirakendus
Sündmus	Kalendrisündmus, mis koosneb järmistest parameetritest: nimetus; algus- ja lõpuaeg; algus- ja sihtkoht; kas tegemist on paindliku sündmusega; kas tegemist on korduva sündmusega, kui jah määrata nädalapäevad ja lõppkuupäev.
WIP	<i>Work In Progress</i> , pooleliolev töö
Äriklient	Ettevõtte, kelle igapäevane äritegevus tugineb osaliselt või täielikult elektrisõidukite kasutamisele

Sisukord

1	Sissejuhatus	13
1.1	Taust	13
1.2	Probleem ja eesmärk.....	14
1.3	Funktsionaalsused.....	15
1.4	Töö struktuur	15
2	Metoodika	17
2.1	Objekt	17
2.2	Tööriistad	18
2.2.1	Teegid ja raamistikud	18
2.2.2	Programmeerimiskeeled	19
2.2.3	Testimine	19
2.2.4	Välised liidesed.....	19
2.2.5	Arhitektuur ja majutuskeskkond.....	19
2.3	Tööprotsessi kirjeldus.....	20
2.3.1	Tööprotsess Enefit AS-is	20
2.3.2	Edasiarenduse metoodikad	21
2.3.3	Lõputöö arendusprotsess	22
3	Erakliendi vajaduste kaardistamine	24
3.1	Laadimisharjumuste kaardistamine	24
3.2	Funktsionaalsuste lisaväärtuse hindamine	27
3.3	Hinnang rakenduse kasutusvalmidusele	29
4	Ärikliendi vajaduste kaardistamine	30
4.1	Elektriveokite hetkeolukord Eestis	30

4.2	Ärikliendi kogemuse kaardistamine	31
4.2.1	Laadimisharjumuste kaardistamine	32
4.2.2	Järeldused	32
5	Turulahendused	33
5.1	Eraklientidele	33
5.1.1	Gridio	33
5.1.2	Terminal	34
5.1.3	Vool	34
5.1.4	Enefit Charge	35
5.1.5	Sõidukitootjad	35
5.2	Äriklientidele	36
5.2.1	Scania	36
5.2.2	Diotech Group	36
5.3	Turulahenduste analüüs	37
6	Olemasoleva lahenduse puudused	38
6.1	Optimeerimine	38
6.1.1	Eeldus, et kasutaja naaseb pärast iga sündmust koju	38
6.1.2	Sõiduaja arvestamata jätmine	38
6.1.3	Laadimisaja vale hindamine	38
6.2	Staatiline aku laetuse tase	39
6.3	Laadimisplaani salvestamine	39
7	Nõuded edasiarendusele	40
7.1	Eraklientidele mõeldud rakendus	40
7.2	Äriklientidele mõeldud rakendus	41
7.3	Kasutuslood	41
7.3.1	Kindla graafikuga elektriauto omanik, kelle sõidukulud tasub tööandja	41
7.3.2	Muutuva graafikuga elektriauto omanik	42

7.3.3	Paindliku graafikuga pistikhübriidi omanik	42
7.3.4	Ettevõtte elektrisõidukite laadimise eest vastutav isik	42
7.4	Nõuded.....	43
7.4.1	Mittefunktsionaalsed nõuded.....	43
7.4.2	Funktsionaalsed nõuded eraklientidele.....	43
7.4.3	Funktsionaalsed nõuded ärikliendile	44
8	Rakenduse edasiarendus	45
8.1	Optimeerimismeetod	45
8.1.1	Liikumise arvestamise põhimõtted	45
8.1.2	Sõiduaja arvestamine	46
8.1.3	Sõiduahelate tuvastamine	46
8.1.4	Vajamineva laetuse taseme dünaamiline arvutamine	47
8.1.5	Laadimisintervallide arvu määramine	47
8.2	Dünaamilise aku laetuse taseme modelleerimine	48
8.3	Väljade automaatne täitmine	48
8.4	Päevase läbisõidu liugur	49
8.4.1	Kasutajaliides.....	50
8.4.2	Arendusprotsess.....	51
8.5	Laadimiste ajalugu.....	52
8.5.1	Optimeeritud laadimise salvestamine	52
8.5.2	Aku dünaamilise täitumisega arvestamine	53
8.5.3	Toimunud laadimiste kinnitamine	54
8.5.4	Laadimiskulude kokkuvõte.....	55
8.6	Liikumisgraafiku importimine.....	56
8.6.1	Mallifail	56
8.6.2	Faili töötlemine	57
8.6.3	Andmete valideerimine.....	57

8.7	Progressiivne veebirakendus	58
8.8	Testkasutajate tagasiside.....	58
9	Analüüs ja järeldused.....	61
9.1	Algoritmi valik	61
9.2	Laadimisaja kasutamise eelis laadimisvõimsuse ees	62
9.3	Elektrisõidukite mudelipõhiste andmete saamine	62
9.4	Päevase läbisõidu liugur	63
9.4.1	Laadimisplaani salvestamine <i>localStorage</i> -isse	63
9.4.2	Olemasolevate funktsioonide taaskasutamine	63
9.4.3	Liuguri lehele navigeerumine.....	64
9.4.4	„Genereeri” nupu olulisus.....	64
9.5	Liikumisgraafiku importimine.....	64
9.5.1	CSV-formaadi valik.....	64
9.5.2	Brauseripoolne töötlemine.....	65
9.5.3	Valideerimise kihistatud lähenemine.....	65
9.5.4	Malli ankurelement.....	65
9.6	Progressiivne veebirakendus versus React Native	66
9.7	Realiseerimata nõuded.....	66
9.7.1	Laetustaseme hoidmise vahemik	66
9.7.2	Rakendus suunatuna ettevõttele.....	67
9.8	Edasiarenduse võimalused.....	67
9.9	Äriline kasum	68
9.10	Meeskondlik hinnang ja tööde logid	69
9.10.1	Teostatud tööde detailne kirjeldus	69
10	Kokkuvõte	73
	Kasutatud kirjandus	74

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	81
Lisa 2 – Kasutajaküsitlus	82
Lisa 3 – Tegevusdiagramm.....	96
Lisa 4 – Progressiivse veebirakenduse manifest	97
Lisa 5 – Piia Carmen Milkovi eneseanalüüs	98
Lisa 6 – Gisela Nüüd eneseanalüüs	100
Lisa 7 – Toggle logid	102

Jooniste loetelu

Joonis 1. UCD etapid.....	20
Joonis 2. Küsimustiku vastused küsimusele: “Kus Te oma autot laete?”	23
Joonis 3. Küsimustiku vastused küsimusele: “Kui palju hoolite odava hinnaga laadimisest?”	24
Joonis 4. Küsimustiku vastused küsimusele: “Kui kaua kulub Teil aega auto laadimise planeerimisele?”	24
Joonis 5. Küsimustiku vastused küsimusele: “Kui pikalt soovite näha oma laadimiste ajalugu?”	25
Joonis 6. Küsimustiku vastused küsimusele: “Kas saaksite kasu statistikast optimaalse laadimise peale kulutatud summa kohta võrreldes börsihinna keskmisega?”	26
Joonis 7. Küsimustiku vastused küsimusele: “Millised rakenduse uuendused loovad lisaväärtust?”	27
Joonis 8. Sõiduahelate loomine.	45
Joonis 9. Läbitud kilomeetrite teisendamine protsentideks.....	46
Joonis 10. Laadimisintervallide leidmine.....	46
Joonis 11. Sõiduki andmete parsimine ja vahemällu salvestamine.	48
Joonis 12. Modaalaken: kalendrisündmuse loomine või päevase läbisõidu määramine.....	49
Joonis 13. Päevase läbisõidu määramine ja laadimisplaani genereerimine.....	50
Joonis 14. Laadimisakna salvestus.	52
Joonis 15. Blokeeritud väljad laadimise ajal.	53
Joonis 16. Toimunud laadimiste kinnitamine ja optimeerimisvälise laadimise lisamine.	54
Joonis 17. Laadimiskulude nädala ja kuu kokkuvõtte vaated.	55
Joonis 18. Kasutaja tegevusdiagramm.....	95

Tabelite loetelu

Tabel 1. Auto laadimise andmed.	57
Tabel 2. Piia Carmen Milkovi töö kirjeldus.	67
Tabel 3. Gisela Nüüd töö kirjeldus.	69

1 Sissejuhatus

Käesolevas peatükis antakse ülevaade töös käsitletava rakenduse taustast ning lahendatava probleemi aktuaalsusest. Samuti mõtestatakse lahti bakalaureusetöö eesmärk ning kirjeldatakse töö struktuuri.

Lõputöö raames edasiarendatava rakenduse loomises on autorid oma panuse andnud, olles kaks lüli viieliikmelisest arendusgrupist Enefit AS-is. Käesolev töö on autorite lähenemine projekti edasiarendusele, mille käigus selgitatakse välja, kas rakendusel leidub koht ning nõudlus turul.

1.1 Taust

Keskkonnahoiule pööratakse nii meedias, poliitikas kui ka mujal aina rohkem tähelepanu. Selle kasvutrendiga käib kaasas elektrisõidukite arvu tõus Eestis, kus viimase viie aasta liitkasvumääraks on 42% [1]. Börsihinna elektripaketiga kaasnev hindade kõikumine, mis on eriti märgatav talvisel perioodil, võib muuta laadimiskulud oodatust kõrgemaks [2]. Kuluefektiivse laadimise saavutamiseks peab kasutaja arvestama samaaegselt nii elektrihindade muutumisega kui ka oma igapäevase liikumisgraafikuga, mis nõuab aku õigeaegset täitmist.

Olemasolevad nutikad laadimislahendused, nagu Enefit Charge, Gridio ja Vool, ei arvesta laadimise optimeerimisel kasutaja personaalse liikumisgraafikuga (vt Peatükk 4.1). Lisaks põhinevad need lahendused pilveteenustel, mis toob kaasa muresid andmete privaatsuse ning latentsuse osas [3].

Kirjeldatud probleemile pakkus lahenduse Enefit AS, kelle idee oli arendada rakendus, mis loob kasutajale personaliseeritud laadimisplaani, tagades kodus laadimise minimaalse kuluga. Rakendus eristub olemasolevatest lahendustest personaalsete laadimise soovitude tõttu, mis põhinevad kasutaja liikumisgraafikul.

1.2 Probleem ja eesmärk

Käesolev lõputöö põhineb Enefit AS-i tellimusel valminud rakendusel, mis genereerib kasutaja liikumisgraafiku põhjal elektrisõiduki optimaalse laadimisplaani. Rakendus toimib täielikult kasutaja seadmes, ilma pilvepõhiste lahendusteta, tagades seeläbi kasutaja andmete privaatsuse.

Rakendus eeldab, et kasutaja on eelnevalt sisestanud oma liikumisgraafiku ning laadimisega seotud tehnilised andmed. Kui tehnilised andmed tuleb sisestada vaid sõiduki esmakordsel seadistamisel, siis liikumisgraafiku haldamine on pidev tegevus, mis nõuab kasutajalt regulaarset panust. Eemaldades rakendusest liikumisgraafiku, kaob element, mis eristab rakendust teistest olemasolevatest lahendustest. Seevastu võib liikumisgraafiku sisestamine osutuda kasutajale ebamugavaks, millega kaob rakenduse üks eesmärkidest – muuta elektrisõiduki laadimise planeerimine kasutajale lihtsamaks.

Lõputöö eesmärk on analüüsida tellitud rakenduse lähenemisviisi elektrisõidukite kuluefektiivse koduse laadimise probleemile ning hinnata selle konkurentsivõimet turul. Selleks viiakse rakenduse potentsiaalsete kasutajate seas läbi küsitlus. Küsitluse tulemustest selgunud kitsaskohtade leevendamiseks arendatakse täiendustega prototüüp. Pärast tehtud täiendusi soovitakse saada vastus küsimusele: „Kas antud rakendusel on potentsiaal läbi lüüa elektriautode laadimisrakenduste turul?“ Lisaks uuritakse rakenduse potentsiaalset kasutatavust äriklientide hulgas. Eesmärgi saavutamiseks on töö jagatud järgmisteks alamülesanneteks:

- koostada küsimustik, mis selgitab välja kasutajate eelistused edasiarendusel lisatavate funktsionaalsuste osas. Küsitlus viiakse läbi börsihinna elektripaketti kasutavate täiselektrilise auto või pistikhübriidi omanike vahel;
- küsimustiku tulemuste põhjal teha rakenduses muudatused, mis tõstavad rakenduse väärtust potentsiaalsete kasutajate silmis;
- lasta täiendatud rakendust testida küsimustikus osalenud ja selleks nõusoleku andnud vastajatel;
- intervjuuerida ärikliente ning hinnata arendatava rakenduse sobivust äriklientidele;

- kliendi vajaduste ja olemasolevate lahenduste analüüsi põhjal langetada otsus, kas rakenduse ideel on potentsiaali elektrisõidukite laadimisrakenduste turul läbi lüüa.

1.3 Funktsionaalsused

Lõputöös käsitletav veebirakendus on kasutatav nii statsionaarsetel kui ka mobiilsetel seadmetel, tagades ühtse kasutajakogemuse erinevatel ekraanisuurustel. Mittefunktsionaalsete nõuete kohaselt salvestatakse ja töödeldakse kõiki andmeid lokaalselt kasutaja seadmes, tagades andmete privaatsuse ning sõltumatuse pilveteenustest.

Rakenduse funktsionaalsed nõuded hõlmavad sõiduki tehniliste andmete ning laadimisega seotud parameetrite sisestamist, sageli kasutatavate aadresside määramist, liikumisgraafiku haldamist kalendri kaudu ning optimaalse laadimisplaani ja sõiduki aku hetkeseisu vaatamist. Paindliku ajakavaga sündmuste puhul pakub süsteem soovitusi nende kohandamiseks.

Kasutaja haldab oma liikumisgraafikut kalendri kaudu, lisades nii regulaarseid kui ka ühekordseid sõite. Sisestatud andmete põhjal genereerib süsteem optimaalse laadimisplaani, mille soovituslikud laadimisajad ja hinnanguline maksumus kuvatakse kasutajale nii graafikul kui ka päevavaates.

1.4 Töö struktuur

Töö koosneb üheksast põhiosast: sissejuhatus, metoodika, era- ja ärikliendi vajaduste kaardistamine, ülevaade turulahendustest, olemasoleva lahenduse puudused, nõuded edasiarendusele, rakenduse edasiarendus ning analüüs ja järeldused.

Esimeses osas tutvustatakse töö tausta, sõnastatakse lahendatav probleem ning püstitatakse eesmärk. Teises peatükis kirjeldatakse rakendust, mida käesoleva lõputöö raames edasi arendatakse, tuuakse välja arendamisel kasutatud vahendid ning kirjeldatakse tööprotsessi. Kolmandas ja neljandas osas kaardistatakse potentsiaalsete era- ja äriklientide vajadusi. Viiendas peatükis antakse ülevaade turul olemasolevatest lahendustest ning analüüsitakse neid. Kuuendas osas tuuakse välja meeskonnaprojekti

raames valminud rakenduse kitsaskohad. Seitsmendas peatükis sõnastatakse edasiarenduse nõuded kasutajalugude ja persoonade põhjal. Kaheksandas osas antakse ülevaade realiseeritud funktsionaalsustest ning testkasutajate tagasisidest. Üheksandas peatükis analüüsitakse tulemusi ning tehakse järeldused rakenduse potentsiaali kohta elektrisõidukite laadimisrakenduste turul.

2 Metoodika

Järgnevalt on kirjeldatud tööriistad, raamistikud ning programmeerimiskeeled, mida kasutati rakenduse arendamiseks. Lisaks antakse ülevaade esimesest prototüübist ja kirjeldatakse tööprotsessi agiilse arendusmeetodiga.

2.1 Objekt

Käesolev lõputöö põhineb aine „ISA meeskonnaprojekt: tellimus" raames valminud rakendusel, mis on loodud Enefit AS-i tellimusel. Aine tulemusena arendati lokaalselt töötav veebirakendus, mis optimeerib elektrisõiduki laadimisplaani kuni 48 tunni ulatuses, tagades laadimise võimalikult väikeste kuludega.

Rakenduse põhifunktsionaalsus – optimeeritud laadimisplaani genereerimine – eeldab kolme sisendi olemasolu: elektri börsihind, kasutaja personaalne liikumisgraafik ning laadimisega seotud tehnilised andmed. Prognoositavad elektri hinnad pärinevad Elering API-st [4]. Liikumisgraafiku põhjal, mille kasutaja käsitsi sisestab koos auto ja laadija tehniliste andmetega, arvutab süsteem energiavajaduse ning genereerib optimaalse laadimisplaani. Paindliku ajakavaga sündmuste puhul pakub süsteem soovitusi nende kohandamiseks, eesmärgiga minimeerida laadimise kogukulu. Kui rakenduses kuvatav hinnanguline aku laetustase ei vasta tegelikkusele, saab kasutaja arvutatut parandada, asendades selle tegeliku väärtusega.

Bakalaureusetöö raames analüüsiti Enefit AS lähenemisviisi elektrisõidukite kuluefektiivse koduse laadimise probleemile. Rakenduse kasutajakogemuses tuvastatud kitsaskohtade leevendamiseks katsetati erinevaid täiendavaid funktsionaalsusi. Edasiarendusel keskenduti kasutajakogemuse parandamisele täiendavate funktsionaalsuste lisamise kaudu. Lõputöö raames lisatud funktsionaalsused määratleti kasutajaküsitluse tulemuste põhjal (vt Peatükk 3). Nendeks olid: sõiduki andmete automaatne sisestamine, laadimiskulude kokkuvõte ning päevase läbisõidu liugur. Mugavaks kasutajakogemuseks potentsiaalsetele äriklientidele, lisati võimalus importida liikumisgraafik CSV (ingl *Comma-Separated Values*) failina.

2.2 Tööriistad

Meeskonnaprojekti raames kasutatud arendustööriistad jäid kasutusse ka edasiarendusel. Rakenduse arendamiseks kasutati Visual Studio Code arenduskeskkonda [5], versioonihaldussüsteemi GitHub [6] ning tööprotsessi haldamiseks Atlassian Jira tarkvara [7].

2.2.1 Teegid ja raamistikud

Rakendus on realiseeritud React teegi abil. Tegemist on laialdaselt kasutatava JavaScript teegiga komponendipõhise kasutajaliidese loomiseks [8]. See võimaldab ehitada rakenduse modulaarsetest ja taaskasutatavatest osadest. Käesolevas lõputöös kasutati Reacti kogu kasutajaliidese ülesehitamiseks, dünaamiliste vaadete loomiseks ning rakenduse oleku haldamiseks.

Reacti ökosüsteemi kuuluv navigeerimisteeik React Router võimaldab luua mitmevaatelisi üheleheküljelisi rakendusi (ingl SPA – *Single Page Application*) [9]. React Routeri kasutamisega saavutati sujuv navigeerimine rakenduse funktsionaalsete vaadete vahel. Rakenduse ehitamiseks ja arendusserveri käitamiseks kasutati Vite tööriista [10], mis toetab moodulipõhist ehitusprotsessi ning võimaldab muudatuste kohest kajastumist arenduskeskkonnas (ingl HMR – *Hot Module Replacement*).

Kasutajaliidese kujundamiseks kasutati PrimeReact komponendi teeki ning PrimeIcons ikoonikogu [11]. PrimeReact sisaldab valmiskujundusega, kuid kohandatavaid komponente nagu nupud, liugurid, tekstisisestusväljad. Ühtlase visuaalse stiili tagamiseks kasutati Tailwind CSS raamistikku [12].

Elektrihindade ja optimaalsete laadimisaegade visualiseerimiseks kasutati Apache ECharts teeki [13], mis võimaldab luua dünaamilisi ja interaktiivseid graafikuid.

FullCalendar komponent [14] võimaldab kuvada kasutaja sündmuseid kalendri kuuvaates ning toetab sündmuste lisamist, muutmist ja kustutamist.

Kuupäevade ja kellaaegade töötlemiseks kasutati Moment.js teeki, mis lihtsustab ajaandmete parsimist, teisendamist ning võrdlemist.

PapaParse on laialdaselt kasutatav JavaScripti teek CSV failide parsimiseks brauserikeskkonnas [15]. Projekti raames rakendati teeki kahes kohas: kasutaja poolt üleslaetud liikumisgraafiku importimisel ning failist sõidukiandmete kogumisel. Samuti kasutati sõidukiandmete kogumisel tehisintellekti Claude [16]

2.2.2 Programmeerimiskeeled

Rakenduse koodibaas on kirjutatud TypeScriptis, mis on JavaScripti laiendus staatilise tüübisüsteemi ja kompileerimisaegsete vigade tuvastamisega [17]. See muudab koodibaasi töökindlamaks, paremini loetavaks ning lihtsamini hallatavaks.

CSS (ingl *Cascading Style Sheets*) on veebilehtede kujunduskeel, mida kasutatakse HTML-elementide välimuse, paigutuse ja visuaalse stiili määramiseks [18].

2.2.3 Testimine

Rakenduse testimiseks kasutati Jest testimisraamistikku koos React Testing Library teegiga [19], mis võimaldab kirjutada ühikteste rakenduse komponentide ja funktsioonide käitumise kontrollimiseks. Koodikvaliteedi tagamiseks kasutati ESLint staatilise analüüsi tööriista, mis tuvastab potentsiaalseid vigu ja stiilinõuete rikkumisi.

2.2.4 Välised liidesed

Elektrihindade andmed koguti Eleringi avaliku API [4] kaudu, mis vahendab Nord Pooli elektri börsihindu. Projekti raames päriti API kaudu nii jooksva- kui ka järgmise päeva 15-minutilisi elektrihindu. Saadud andmeid kuvati kasutajale dünaamilisel elektrihinnagraafikul koos hindade põhjal genereeritud optimeeritud laadimisplaaniga.

Asukohaandmete töötlemiseks kasutati Stadia Maps teenust [20] kahe erineva API kaudu. Geocoding API võimaldas aadressiotsingu automaatset täitmist kasutajale sündmuste lisamisel. Routing API kaudu arvutati sündmuste vahelised vahemaad ja sõiduajad, mida kasutatakse energiavajaduse hindamisel.

2.2.5 Arhitektuur ja majutuskeskkond

Edasirendusprotsessi lõppfaasis täiendati rakendust progressiivse veebirakenduse (ingl PWA – *Progressive Web App*) funktsionaalsusega. Tegemist on standardsetele

veebitehnoloogiatele tugineva lahendusega, mis võimaldab luua platvormiüleselt toimiva süsteemi ilma platvormispetsiifilist koodi kirjutamata [21].

Rakenduse avalikuks tegemiseks ja majutamiseks (ingl *hosting*) kasutati Microsoft Azure-i pilveteenust, täpsemalt Azure Storage Account teenuse Blob Storage funktsionaalsust [22]. Tegemist on turvalise, skaleeritava ja kuluefektiivse platvormiga staatiliste failide, sealhulgas Reactis loodud üheleheküljeliste rakenduste kuvamiseks lõppkasutajatele.

2.3 Tööprotsessi kirjeldus

Käesolevas peatükis antakse ülevaade rakenduse arendamisel kasutatud tööprotsessidest ja meetodikatest. Esmalt kirjeldatakse meeskonnaprojekti raames rakendatud arendusprotsessi, seejärel tutvustatakse edasiarenduses kasutatud meetodikaid ning lõpuks antakse ülevaade lõputöö arendusprotsessist tervikuna.

2.3.1 Tööprotsess Enefit AS-is

Arendusprotsess algas 1. oktoobril 2025 viieliikmelise tudengite arendusmeeskonnana Enefit AS-is.

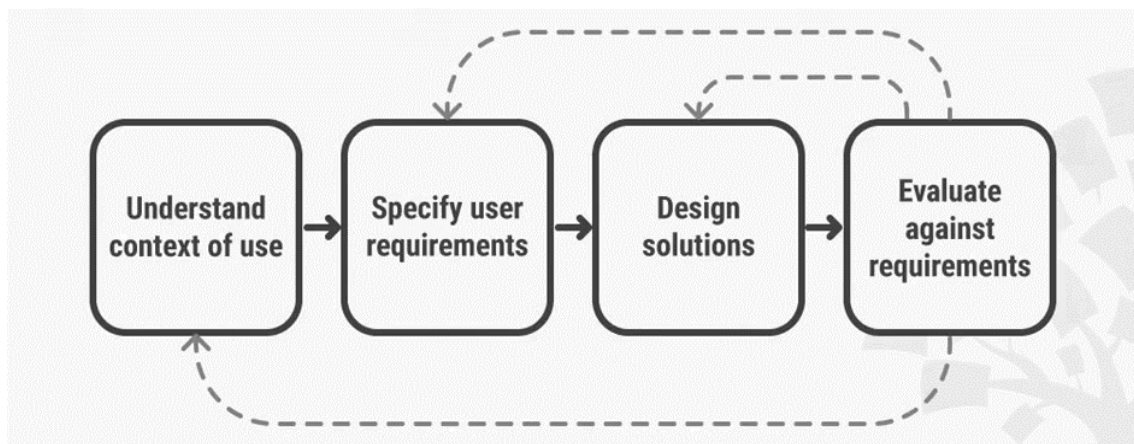
Töö käigus järgiti Kanban meetodikale omaseid protsesse. Kanbani meetodika suunab arendusmeeskondi oma töövoogu visualiseerima ning piirama poolelioleva töö hulka igas töövoos etapis [23]. Tööpäevad algasid 15-minutiliste püstijalakoosolekutega (ingl *stand-up meeting*), kus iga meeskonnaliige andis ülevaate eelmise päeva tulemustest, päeva plaanist ning võimalikest takistustest. Iga meeskonnaliige vastutas endale valitud arendusülesannete täitmise eest. Protsessi läbipaistvust toetab Kanban tahvel, mis kuvab ülesannete hetkeseisu, tõstab esile prioriteedid ning võimaldab tuvastada töövoos kitsaskohad [23].

Kõikide tsükli ülesannete valmimise järel viidi läbi demo koosolek, kus tutvustati valminud lahendust mentoritele ja tooteomanikule. Sellele järgnes *refinement* koosolek, kus tooteomaniku koostatud kasutajalood tükeldati alamülesanneteks ning lisati uute arendusülesannetena Kanban tahvlile.

Kirjeldatud arendustsükleid läbiti mitu ning iga tsükli lõppedes alustati kohe uuega. Selline töökorraldus kestis kuni 15. jaanuarini 2026, mil arendamine toimus kahel tööpäeval nädalas.

2.3.2 Edasiarenduse metoodikad

Pärast õppeaine „ISA meeskonnaprojekt: tellimus“ lõppemist rakendasid bakalaureusetöö autorid kasutajakeskse disaini meetodit (ingl UCD – *User-Centered Design*). UCD on iteratiivne protsess, mille kõigis etappides kaasatakse kasutajad aktiivselt, kogudes nende vajaduste mõistmiseks tagasisidet küsitluste ja intervjuude kaudu [24]. Iga iteratsioon koosneb reeglina neljast etapist (vt Joonis 1).



Joonis 1. UCD etapid [24].

Esimese asjana viiakse läbi kasutuskonteksti analüüs, mille eesmärk on kaardistada keskkond ja tingimused, kus kasutaja loodavat süsteemi rakendaks. Teise sammuna tuvastatakse sihtrühma vajadustest lähtuvad kasutajanõuded. Sellele järgneb disainifaas, mille käigus töötatakse välja eelnevalt defineeritud nõuetele vastavad potentsiaalsed lahendused. Hindamisetapis valideeritakse lahenduste vastavust algsetele nõuetele ja sihtrühma vajadustele. Neid etappe läbitakse tsükkliliselt seni, kuni saavutatakse eesmärgipärane lõpptulemus [24].

Samuti rakendati arendamisel komponendipõhist arendusmetoodikat (ingl CDD – *Component Driven Development*) [25]. CDD lähenemise puhul jaotatakse rakendus väiksemateks iseseisvateks komponentideks, mida on võimalik rakenduse erinevates osades korduvkasutada. Arendusprotsess algab väiksemate komponentide loomisega, millest moodustatakse terviklikud kasutajaliidese moodulid. Selline lähenemine muudab

rakenduse hõlpsamini skaleeritavaks, vähendab koodi dubleerimist ning tagab kasutajaliidese ühtse disaini ja toimimise.

Uute rakenduse vaadete arendamisel kasutati *UI-first* arenduspraktikat, kus esmalt loodi kasutajaliidese visuaalne struktuur ning seejärel lisati järk-järgult funktsionaalsus. Visuaalse struktuuri loomine enne loogikakihi realiseerimist võimaldab hinnata kasutajakogemust varajases arendusfaasis ning vähendab hilisemate ümberarenduste vajadust [26].

2.3.3 Lõputöö arendusprotsess

Lõputöö arendusprotsess koosnes kahest iteratsioonist, millest igaüks hõlmas nelja etappi: olukorra kaardistamine, kasutajatelt tagasiside kogumine, lahenduse arendamine ning tulemuste hindamine.

Esimene iteratsioon lähtus eraisikust kasutajate sihtgrupist. Õppeaine „ISA meeskonnaprojekt: tellimus“ raames valminud prototüübi kitsaskohad kaardistati kasutajaküsitluse abil (vt Peatükk 3). Küsitluse tulemuste põhjal täiendati prototüüpi (vt Peatükk 8) ning seejärel testiti täiendatud rakendust kasutajagrupiga (vt Peatükk 8.8). Saadud tagasiside põhjal, tehti järeldused edasiarenduse võimaluste kohta.

Teises iteratsioonis laiendati sihtgruppi äriklientidele. Edasiarenduse ideeks oli muuta mitme sõiduki laadimisgraafiku sisestamine mugavamaks luues CSV faili importimise võimalus, mille alusel luuakse mitmele laadijale mõeldud optimeeritud laadimisplaan. Ettevõtetega viidi läbi intervjuud, et mõista nende vajadusi ja väljakutseid, mis kaasnevad suure hulga elektrisõidukite laadimisega (vt Peatükk 4.2).

Mõlema iteratsiooni käigus loodi persoonad, ehk *arhetüüpsed kasutajad, kelle eesmärgid ja omadused esindavad suurema kasutajarühma vajadusi* [27]. Persoonade loomine aitas visualiseerida rakenduse potentsiaalseid lõppkasutajaid. Edasiarenduse planeerimiseks kaardistati olemasoleva rakenduse kitsaskohad (vt Peatükk 6) ning funktsionaalsete nõuete määramisel keskenduti persoonade vajadustele (vt Peatükk 7). Kirjeldatud lähenemist nimetatakse *As-Is To-Be* protsessiks, mille käigus kaardistatakse praegune olukord ning sõnastatakse tulevikuvision, kaasates protsessi potentsiaalsed kasutajad [28].

Kogu protsessi käigus täiendati rakenduse seniseid funktsioone: optimeerimismeetod, sõiduaja arvestamine, aku laetuse taseme muutumine ja laadimisplaani salvestamist. Kasutajakogemuse parandamiseks viidi sisse järgmised uuendused: rakendus on PWA-na lisatav seadme avaekraanile; korduvaid sündmusi on võimalik muuta korraga terve seeriana; valikuliste sündmuste korral pakub rakendus sündmuse toimumisaja nihutamist; aadresside salvestamisel kontrollitakse, et sama aadressi poleks juba varem lisatud; sõiduki mudeli valimisel täidetakse tehnilised andmed automaatselt; päevase läbisõidu liuguri abil on võimalik kiiresti laadimisplaan genereerida. Äriklientidele lisati võimalus importida liikumisgraafik CSV failina.

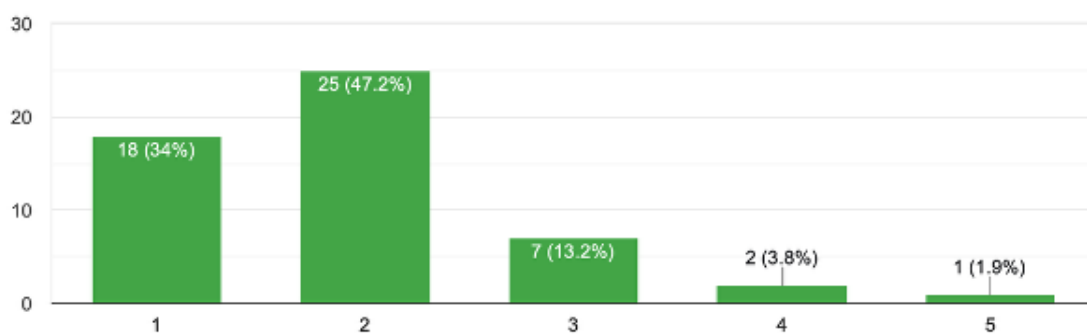
3 Erakliendi vajaduste kaardistamine

Käesoleva lõputöö raames viidi läbi küsitlus (Lisa 2) eesmärgiga kaardistada eraisikust kasutajate vajadused, ootused ning eelistused edasiarenduse puhul. Vastajateks olid börsihinna elektripaketti kasutavad pistikhübriidi või täiselektrilise auto omanikud. Kasutajauuring viidi läbi küsitluse vormis Google Forms keskkonnas. Uurimismeetod valiti eesmärgiga, olla kättesaadav võimalikult suurele sihtrühmale.

3.1 Laadimisharjumuste kaardistamine

Kliendiuuringus osales kokku 53 vastajat, kellest 42 omasid täiselektrilist auto, kaheksa pistikhübriidi ja kolm vastajat mõlemat tüüpi sõidukit.

Sooviga mõista, kas rakendus, mis on suunatud kodusele laadimisele, vastab sihtrühma tegelikele laadimisharjumustele, paluti vastajatel hinnata viie palli skaalal nende auto laadimise asukohta. Väärtus üks tähistas skaalal „Laen alati kodus” ning väärtus viis tähistas „Laen alati kodust eemal”. Tulemused näitavad, et üle 80 protsendi vastanutest laeb sõidukit peamiselt kodus, kus populaarseimateks vastusteks osutusid väärtused üks ja kaks, vastavalt tulemustega 18 ja 25 häält (vt Joonis 2).

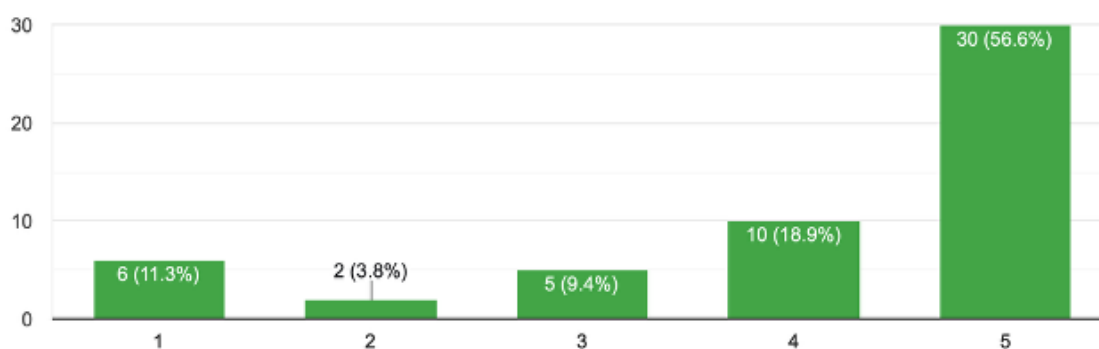


Joonis 2. Küsimustiku vastused küsimusele: “Kus Te oma autot laete?”

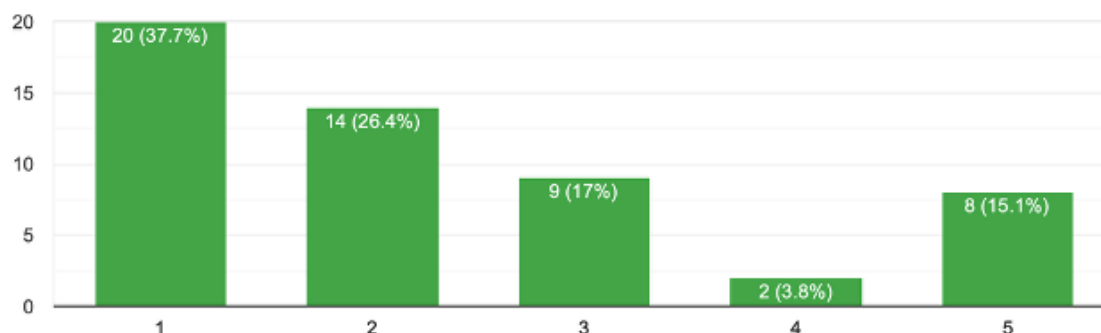
Küsitluse vastajatele, kes ei lae sõidukit alati kodus, esitati lisaküsimus, mis palus täpsustada vastuse põhjust. Enim välja toodud põhjuseks osutus fakt, et pikemaid marsruute ei ole võimalik ühe laadimisega läbida. Lisaks märkis üheksa vastajat, et töö

juures laadides on laadimiskulud madalamad ning kolm vastajat leidis, et kodune laadimine on nende jaoks liiga aeglane.

Selleks, et hinnata, kas rakenduse põhiidee – kuluefektiivne laadimine – vastab kasutajate tegelikele eelistustele, uuriti kahte omavahel seotud aspekti: kui oluliseks peavad vastajad odavat laadimishinda ning kui palju aega ollakse valmis laadimisplaani planeerimisele kulutama eesmärgiga kulusid vähendada. Küsimusele „Kui palju hoolite odava hinnaga laadimisest?“ skaalal, kus üks tähistas „Ei vaata elektri hinda enne laadimist“ ning viis „Planeerin alati laadimise kellaaega“, valis kõrgeima väärtuse 30 vastajat (vt Joonis 3). Seevastu ilmneb vastupidine suundumus laadimise planeerimisele kulutatavas ajas. Küsimusele „Kui kaua kulub Teil aega auto laadimise planeerimisele?“ valis kõige väikseima hinnangu 20 vastajat ning kõige suurema hinnangu kaheksa vastajat (vt Joonis 4). Skaala madalaim hinnang vastab väärtusele „Ei kuluta sellele üldse aega“ ning suurim hinnang „Tegemist on igapäevase tegevusega“. Tulemus viitab asjaolule, et laadimise planeerimine ei võta palju aega isegi kui huvi kulu minimeerimise vastu on kõrge.



Joonis 3. Küsimustiku vastused küsimusele: „Kui palju hoolite odava hinnaga laadimisest?“

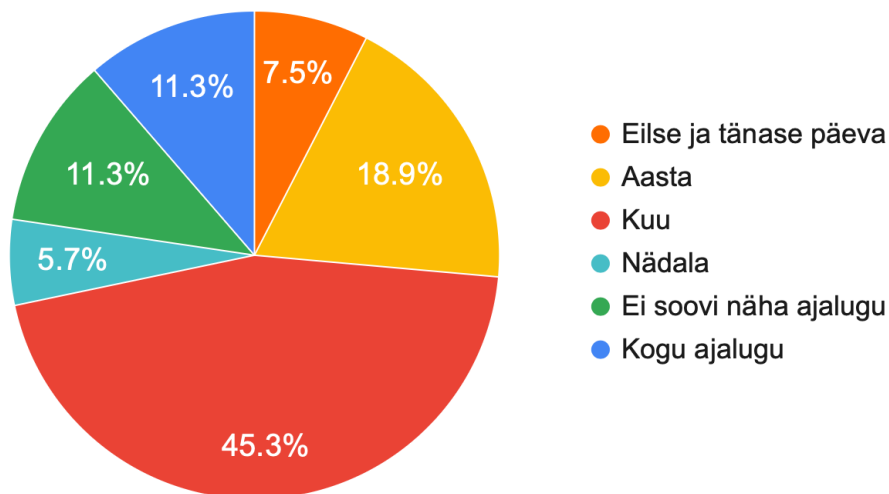


Joonis 4. Küsimustiku vastused küsimusele: „Kui kaua kulub Teil aega auto laadimise planeerimisele?“

Üheks rakenduse funktsionaalsuseks on laadimissessiooni jaotamine soodsamatele 15-minutilistele intervallidele. Küsitluses uuriti, kas kasutajad on sellise lahendusega nõus. Tulemustest selgus, et viiekümne kolmest vastajast 35 oli kulude vähendamise nimel valmis laadimist tükeldama. Selle vastu oli 18 vastajat, kes eelistab laadida järjest ilma pausideta.

Elektrisõidukite liitiumioonakude eluea pikendamiseks, on soovitatav hoida aku laetustase vahemikus 20-80% [29], mida järgib ka lõputöös käsitletava rakenduse algoritm. Küsitluses uuriti, kas kasutajad soovivad laetustaseme protsendi piire ise kohandada. 46 vastajat eelistas piire ise määrata ning seitse vastajat leidis, et see ei ole vajalik.

Laadimiskulude kokkuvõtte sobiva ajavahemiku määramiseks uuriti kasutajate eelistuste kohta laadimisajaloo ulatuse osas. Küsimustiku tulemustest selgus, et kõige rohkem saadakse kasu kuu tagusest laadimise ajaloost, seda varianti olles valinud 24 vastajat. Seitse vastajat soovib näha vähemalt aasta tagust ajalugu, kuus vastajat peab vajalikuks kogu ajalugu ja sama palju vastajaid ei soovi ajalugu üldse näha, viis vastajat soovib näha nädalast ajalugu ning neli vastajat piirdub eilse ja tänase päevaga (vt Joonis 5).



Joonis 5. Küsimustiku vastused küsimusele: “Kui pikalt soovite näha oma laadimiste ajalugu?”

Mitmed vastajad lisasid kommentaarina juurde, et aastane ajalugu on vajalik aastaaruande loomiseks. Samuti mainiti, et rakendused, mida vastajad hetkel kasutavad

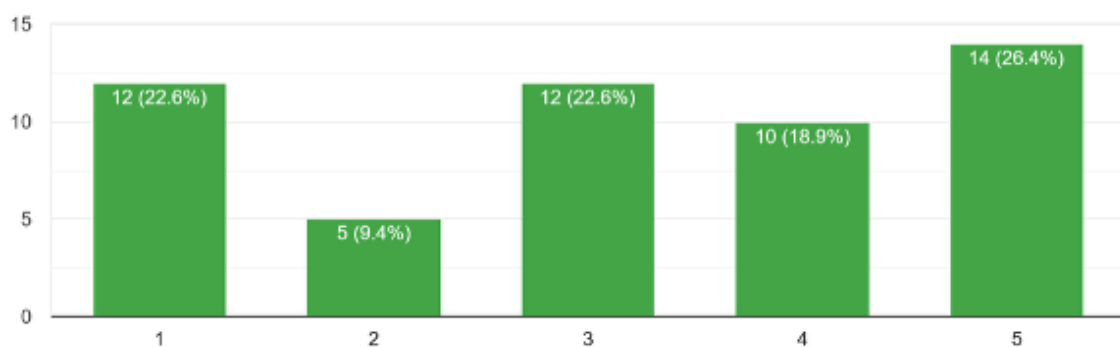
pakuvad võimalust näha kogu ajalugu. Tulemuste põhjal võeti vastu otsus säilitada viie aasta laadimisandmeid.

3.2 Funktsionaalsuste lisaväärtuse hindamine

Veendumaks, et küsimustiku täitjad saavad rakendusest ühte moodi aru, seletati detailselt selle kontseptsiooni, mida täiendasid kuvatõmmised rakendusest. Järgnevalt pakuti välja rakendust täiendavad funktsionaalsused ning uuriti, kas vastajate arvates loob kirjeldatud funktsionaalsus rakendusele lisaväärtust.

Küsimustikus esitleti kuut funktsionaalsust: laadimiskulutuste kokkuvõte, statistika optimaalse laadimisega säästetava raha kohta, automaatne autoandmete väljade täitmine automudeli valikul, ilmaprognoosil põhinev täpsem laadimisvajaduste ennustamine, liuguriga saadav ülevaade laadimise kohta ning interaktiivne päevaplaani muutmine koos laadimisplaani uuendusega. Mainitud täienduste eesmärgiks on muuta rakenduse kasutajakogemust paremaks ning seeläbi tõsta rakenduse väärtust.

Säilitades laadimisajaloo andmed on võimalik rakenduses kuvada laadimiskulutuste kokkuvõte erinevate perioodide lõikes. Seda pidas viiekümne kolmest vastajast kasulikuks 38 vastajat. Seevastu statistika osas, mis võrdleb optimaalse laadimisega saavutatud keskmist hinda börsipaketi keskmise elektri hinnaga, jagunesid arvamused viie palli skaalal ühtlaselt (vt Joonis 6).

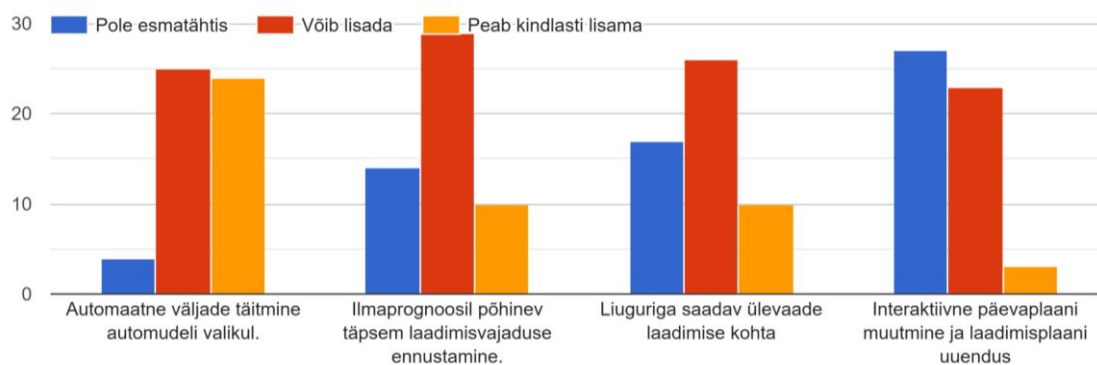


Joonis 6. Küsimustiku vastused küsimusele: “Kas saaksite kasu statistikast optimaalse laadimise peale kulutatud summa kohta võrreldes börsihinna keskmisega?”

Kasutajakogemuse parandamiseks pakuti välja järgmised täiendused, mis vähendavad andmesisestuse koormust ning täpsustavad prognoosi tulemusi:

- kasutaja ei pea enam kõiki sõiduki andmeid käsitsi sisestama, vaid saab valida rippmenüüst automudeli, mille abil täidetakse ülejäänud väljad automaatselt;
- lisades algoritmile andmed järgmise päeva ilma prognoosi kohta, oskab rakendus paremini arvestada sõiduki aku tühjenemise kiirusega ning seeläbi pakkuda täpsemat ennustust järgmise päeva vajamineva laetustaseme kohta;
- kasutaja saab liuguriga määrata eesootavate sõitude kilometraaži, mille põhjal genereeritakse optimaalne laadimisplaan ilma kalendrisse sündmusi lisamata;
- kasutaja saab katsetada kalendris määratud liikumisgraafiku muutmist. Eemaldades kalendris olevaid sündmusi, uueneb laadimisplaan vastavalt tehtud muudatusele.

Tulemuste põhjal (vt Joonis 7) peeti kõige olulisemaks täienduseks sõiduki andmete automaatset täitmist. Ilmaprognoosi integreerimist algoritmi ning kilometraaži määramiseks mõeldud liugurit hinnati samuti lisaväärtust loovateks uuendusteks. Interaktiivset päevaplaani muutmise funktsiooni ei peetud esmatähtsaks edasiarenduseks.



Joonis 7. Küsimustiku vastused küsimusele: “Millised rakenduse uuendused loovad lisaväärtust?”

3.3 Hinnang rakenduse kasutusvalmidusele

Pärast esialgse rakenduse detailset tutvustust, küsiti vastajatelt, kas nad kasutaksid kirjeldatud rakendust. Küsimustikus osalenutest vastas jaatavalt 11, eitavalt 32 ning neutraalselt kümme vastajat. Küsimusele, kas vastajad kasutaksid rakendust koos lisandustega, vastas jaatavalt 19, eitavalt 25 ning neutraalselt kaheksa vastajat. Tulemused näitavad, et pakutud täiendused suurendasid küll rakenduse atraktiivsust, kuid enamik vastajaid jäi siiski eitavale seisukohale.

Vabateksti vastustes esines mitmeid kriitilisi kommentaare rakenduse keerukuse kohta. Peamise nõrkusena toodi esile liikumisgraafiku sisestamise ajakulu. Madalama päevase läbisõiduga vastajad mainisid, et laevad enda sõidukit vaid paar korda nädalas, mistõttu pole laadimise planeerimine ajakulukas tegevus.

Lisaks toodi välja, et elektriautode laadimise optimeerimiseks on turul juba mitmeid lahendusi, mis vastavad kasutajate vajadustele. Nimetatud said järgmised lahendused: Gridio, Home Assistant, Vool, Terminal ja SolaX.

4 Ärikliendi vajaduste kaardistamine

Küsitluse tulemused ei viidanud arendatava rakenduse laiaulatuslikule vajadusele. Sellest tulenevalt keskendusid käesoleva lõputöö autorid sihtrühma määratlemisele, kelle jaoks liikumisgraafiku funktsionaalsus võiks laadimise planeerimisel enim väärtust luua.

Autorite hinnangul osutub kalendrisündmuste sisestamise funktsioon kasulikuks olukorras, kus sõidukite liikumisgraafik on ette planeeritud ning ootamatuid muudatusi esineb harva. Elektrisõidukite hulka kuuluvad lisaks sõiduautodele ka kaubaautod ning raskeveokid.

Autorite hüpoteesi kohaselt võiksid ettevõtted, kes koostavad elektrisõidukitele täpseid liikumisgraafikuid, saada kasu rakendusest, mis aitab planeerida sõidukite laadimist ning optimeerida laadimiskulusid. Hüpoteesi kontrollimiseks viidi läbi intervjuud elektrisõidukeid kasutavate ettevõtetega.

4.1 Elektriveokite hetkeolukord Eestis

Eestis esineva elektriveokite laadimistaristu nõrkuse [30] tõttu toimub nende laadimine enamasti ettevõtte enda laadimisjaamas. Riigisisese transpordi korral on elektriveokite akumaht piisavalt suur, et läbida üks töövahetus ilma vahelaadimiseta. Seda kinnitab ka täiselektrilisi raskeveokeid kasutav firma A. Le Coq'i logistika- ja ostudirektor Erki Lehist: „Meie kogemus elektriveoki kasutamisel on osutunud väga positiivseks. Erinevalt esialgsetest hinnangutest ei nõudnud sõiduki kasutusele võtmine jaotuslogistikas mingeid ümberkorraldusi ja sellega saab efektiivselt teha kõiki tarneid. Sõiduk on end heast küljest näidanud ka talvisel ajal. Kuna veok liigub eelnevalt planeeritud marsruutidel ja naaseb baasi, piisab laadimisest oma territooriumil“ [31].

Elektriveokite laialdase kasutuselevõtu üheks suurimaks piduriks on nende laadimine. Ettevõtted otsivad aktiivselt uusi võimalusi, mis aitaksid elektriveokite laadimist planeerida. Näiteks elektriveokite tootja Volvo Trucks on oma koduleheküljel

küsimusele „Kas pakute lahendusi veokite laadimiseks?“ vastanud „Volvo uurib pidevalt uusi lahendusi ja koostöövõimalusi, mis on seotud meie sõidukite laadimisega“ [32].

Elektrisõidukeid ärilistel eesmärkidel kasutavate ettevõtete puhul osutub laadimise ajastamine eraisikuga võrreldes keerulisemaks ülesandeks. See on tingitud fikseeritud tööaegadest ning asjaolust, et tuleb korraga hallata mitut laadijat ja sõidukit. Uue põlvkonna elektriveokid võimaldavad laadida aku 20 protsendilt 80 protsendini ühe tunniga. See võimaldab läbida kuni 470 km [33], laiendades nende seinseid kasutusvõimalusi.

Tulevikus ei ole elektriveokite kasutuselevõtu peamiseks piiranguks enam tehnilised näitajad, vaid laadimis- ja logistikaprotsesside täpne planeerimine. Arvestades juhtidele ette nähtud 45-minutilist kohustuslikku puhkepausi [34], on laadimise integreerimine tööpäeva teostatav ilma pikemate seisakuteta.

4.2 Ärikliendi kogemuse kaardistamine

Intervjuudesse kaasati ettevõtted, kes kasutavad elektrisõidukeid igapäevatöös erinevatel eesmärkidel, esindades seeläbi erinevaid ärikasutuse kasutusjuhte.

Raskeveokitega tehtava kaubaveo esindajana kaasati A. Le Coq, kes soetas Baltikumi esimese täiselektrilise raskeveoki 2022. aastal [35]. 2026. aastaks oli ettevõtte autopargis neli täiselektrilist veokit [36].

Pakiveo- ja logistikasektorit esindas DPD, kelle autopargis oli 2026. aasta seisuga 97 elektrikaubikut, millega tarniti ja koguti 2025. aastal 52% kõikidest pakkidest. Ettevõtte eesmärk on asendada 2030. aastaks 95% senistest pakiveosõidukitest elektrikaubikutega [37].

Kliendikülastusteks kasutava autopargi näitena kaasati Telia, kes alustas oma autopargi elektrifitseerimist 2020. aastal eesmärgiga saavutada 2030. aastaks süsinikuneutraalsus [38].

Äriklientidega suhtlemise käigus kaardistati nende senine laadimiskogemus ning sellega seotud kitsaskohad. AS A. Le Coq energiatõhususe spetsialistiga peeti kirjavahetus e-posti teel. DPD operatsioonijuhi ja Telia haldusjuhiga viidi läbi intervjuud.

4.2.1 Laadimisharjumuste kaardistamine

Läbiviidud intervjuude ja kirjavahetuse põhjal saab ettevõtete hetkeolukorda elektrisõidukite laadimise vaates kirjeldada järgmiselt:

- laadimispaigad on paigutatud veoringidesse ning nende võimsus on piiratud;
- kasutusel on fikseeritud elektripakett;
- hädaolukordade puhul kasutatakse kiirlaadijaid.

A. Le Coqi puhul sõltub sõidukite liikumisgraafik sõiduki otstarbest. Tootmise ja logistikakeskuse vaheline elektriveok sõidab kindla marsruudi ja ajavahemiku järgi ning laadimine on toimub juhtide lõunapausil. Jaotusvedude graafik on seevastu muutlikum – kuigi sõidukid alustavad tööd hommikuti samal ajal, sõltub päevase sõiduringi ulatus individuaalsest tellimusest. Nende sõitude jaoks, paneb veosekava kokku vedusid korraldava koostööpartneri logistik.

Ettevõtetes DPD ja Telia, ei ole sõidukite järgmise päeva liikumisgraafik üldjuhul ette teada, mistõttu toimub laadimine peamiselt töövälisel ajal.

4.2.2 Järeldused

Vestlused näitavad, et sõidukite liikumisgraafik oleneb nii ettevõtte äritegevuse iseloomust kui ka sõidukite kasutusotstarbest. Rakenduse kalendripõhine lähenemine sobib ettevõtetele, kelle sõidukite järgmise päeva läbitav vahemaa on teada hiljemalt eelmise päeva õhtuks. Vastasel juhul ei leia rakendus ettevõtete poolset kasutust. Kõik intervjuueeritud ettevõtted kasutavad fikseeritud elektripaketti, mistõttu ei paku börsihindadel põhinev laadimise optimeerimine neile rahalist kasu.

5 Turulahendused

Edasiarenduse suundade määratlemiseks on oluline mõista turul olemasolevaid lahendusi. Käesolevas peatükis antakse ülevaade Eestis konkureerivate era- ja äriklientidele mõeldud lahenduste funktsionaalsustest ning analüüsitakse neid.

Börsihindasid saab jälgida selleks loodud veebilehtede kaudu, nagu näiteks Elektrikell [39] ja Elektrihind [40]. Mõlemad kuvavad Nord Pool turuhinda graafikuna ning toovad esile soodsaimate hindadega ajavahemikud. Nende abil saab kasutaja teha teadliku otsuse, millal sõidukit laadida. Vaatamata nende veebilehtede lihtsusele, leidub turul mitmeid alternatiivseid lahendusi, mis keskenduvad sõidukite laadimisele, eesmärgiga vähendada elektriautode laadimiskulu. Iga lahendus läheneb probleemile eriviisil, mis eristab neid teistest konkurentidest.

5.1 Eraklientidele

Erakliendile suunatud laadimislahendused jagunevad laadija- ja sõidukipõhisteks. Järgnevalt on kirjeldatud populaarsemaid teenusepakkujaid, keda mainiti kasutajaküsitluses või on saanud laiemat meediakajastust.

5.1.1 Gridio

Gridio 2.0 OÜ arendatud rakendus Gridio [41] võimaldab ühendumist kasutaja päikesepaneelidega ning planeerida laadimist päikeseenergia toodangu järgi. Rakendus ühendub sõiduki tootja pilveteenusega, edastades sõidukile käske laadimise alustamiseks ja peatamiseks, et saavutada määratud ajaks soovitud laetustase. Rakendus laeb sõidukit kasutaja määratud ajavahemikus – alates auto laadijaga ühendamisest kuni soovitud lahkumisajani – eelistades madalaimate elektrihindadega tunde. Sellele lisaks pakub Gridio võimalust määrata maksimaalne lubatud elektrihind laadimiseks.

Rakendus võrdleb kohese laadimise kulu optimeeritud laadimisega saavutatud kuluga ning kuvab kasutajale igapäevase kokkuhoiu koos ülevaatega sõiduki energiatarbimisest. Gridio nutika laadimise funktsionaalsus on hetkel kättesaadav Tesla,

Volkswagen ID., Škoda, Renault, Hyundai, Kia, Audi, Cupra, SEAT ja BMW elektrisõidukite omanikele [42].

5.1.2 Terminal

Terminali mobiilirakendus Terminal Energia [43] juhib sõiduki laadimist Terminali Easee kodulaadija kaudu, pakkudes targaks laadimiseks kolme strateegiat.

Esimene strateegia – laadimine ajalise kestuse alusel – on kättesaadav kõikidele elektriautodele ja pistikhübriididele. Selle strateegia puhul määrab kasutaja vajaliku laadimise kestuse tundides.

Targa laadimise ülejäänud funktsioonide kasutamiseks kasutab Terminal kolmanda osapoolena Gridio pakutavat auto ühendamise teenust. Seega on ülejäänud kaks strateegiat saadaval üksnes Gridio poolt toetatavatele sõidukitele. Teine strateegia võimaldab kasutajal määrata aku laetuse taseme, milleni sõidukit laetakse. Kolmas strateegia on laadimine elektrihinna järgi, mis võimaldab lisaks aku miinimumtaseme määramisele seada ka elektrihinna piirmäära. Olukorras, kus elektrihind jääb alla määratud piirmäära, jätkatakse laadimist minimaalsest laetuse tasemest kõrgemale.

Targa laadimise alustamisel tehakse esmalt testlaadimine, mille käigus määratakse, millise võimsusega on võimalik hetkel sõidukit laadida. Laadimise käigus on võimalik seadistusi muuta, mille korral korrigeeritakse laadimisplaani koheselt.

Terminal on ainus lahendus, mis võimaldab laadida sõidukit börsihinna alusel ka fikseeritud elektripaketi kasutamisel. Lisaks elektrisõiduki laadimisele sisaldab Terminali rakendus üldise elektritarbimise jälgimist.

5.1.3 Vool

Voolu mobiilirakendus [44] juhib sõiduki laadimist kodulaadija kaudu. Lisaks Voolu enda laadijatele on rakendus loodud ka teiste OCPP-ühilduvusega (ingl *Open Charge Point Protocol*) laadijate jaoks. Rakendus tuvastab automaatselt madalaima börsihinna ning laeb sõiduki kasutaja määratud ajaks soovitud laetuse tasemeni. Samuti saab kasutaja laadimist distantsilt alustada ja lõpetada. Lisaks on võimalik seadistada korduv tark laadimine, mille puhul määratakse laadimise parameetrid vaid üks kord. Rakendus

kuvab ülevaate toimunud laadimistest, sealhulgas kogukulud, nutika laadimisega saavutatud säästu protsent ning keskmise kWh hinna.

2026. aasta veebruaris kaotas Vool ajutiselt sõiduki andmetel põhinevad laadimisrežiimid [45]. Võib eeldada, et see on tingitud sõidukitootjate API-ga liidestamise kõrgetest kuludest.

Näiteks Tesla Fleet API kasutab alates 2025. aastast tasustatud mudelit [46], kus kulu sõltub päringu tüübist ning sagedusest. Minimaalse päringute arvuga – 150 000 *streaming signal*’it, 1000 käsku Tesla seadmele, 500 andmepäringut ja 50 sõiduki äratamist – ulatub ühe sõiduki kuukulu 4 USD-ni (3.42 €) käibemaksuta [47]. Päringute arv sõltub peamiselt andmete uuendamissagedusest, jälgitavate andmeväljade arvust ning sõiduki olekust, pilveteenuse kulud kumuleeruvad suure kasutajaskonna korral kiiresti.

5.1.4 Enefit Charge

Enefit Charge mobiilirakendus [48] tagab kuluefektiivse laadimise nii kodus kui tööl. Enefiti rakendus, mis ühildub Enefiti laadijatega, pakub erinevaid laadimisviise: tark, ajastatud ja lihtne laadimine. Targa laadimise korral peab kasutaja määrama laetuse soovitud taseme ning kellaaja, milleks laadimine peab olema lõpetatud. Ajastatud laadimise korral ei võeta börsihinda arvesse ning laadimine toimub kasutaja sisestatud algus- ja lõpuaja kohaselt. Lihtne laadimine alustab auto laadimist koheselt ning lõpetab kui aku on täis laetud.

Rakenduses saab luua erinevaid targa laadimise profiile, mille alusel laetakse sõiduk määratud ajaks soovitud tasemeni. Enefiti rakendusega saab ühendada kõik Enode’i nimekirjas olevad sõidukid [49]. Nimekirjast puuduva sõiduki andmed tuleb sisestada käsitsi, et rakendus saaks vajalikud arvutused teostada.

5.1.5 Sõidukitootjad

Laadimise planeerimiseks pakuvad lahendusi ka sõidukitootjad, järgnevalt on kirjeldatud kahe tootja lähenemisviise.

Kia võimaldab laadimise planeerimist sõiduki pardasüsteemi kaudu [50]. Ühendades sõiduki laadijaga, saab kasutaja määrata aku laadimise algusaja ning tipptunnivälise

laadimise sätteid. Tipptunnivälise laadimise puhul on võimalik valida kahe variandi vahel: laadimine toimub ainult tipptunnivälisel ajal, mille korral ei pruugi aku saavutada täislaetuvust, või jätkatakse laadimist ka pärast tipptunnivälise perioodi möödumist, et tagada aku täislaetavus.

Tesla rakendus juhib laadimist Tesla laadija (ingl *Wall Connector*) kaudu [51], millega saab laadida lisaks Teslale ka teiste tootjate sõidukeid. Laadija on vaikimisi seadistatud eelistama tipptunnivälist laadimist, kuid vajadusel saab rakenduses määrata laadimise ajalisekestuse. Teiste tootjate sõidukite laadimisel peab kontrollima, et sõiduki enda laadimise ajastamise seaded oleksid kooskõlas Tesla laadija seadistustega.

5.2 Äriklientidele

Äriklientidele suunatud laadimislahendusi pakuvad nii sõidukitootjad kui ka energiaturul tegutsevad ettevõtted. Järgnevalt on kirjeldatud A. Le Coqi raskeveokiparki kuuluva Scania [35] ja DPD laadimisjaamu haldava Diotech Group-i [52] lahendusi. Ülejäänud, siin käsitlemata lahendused pakuvad äriklientidele üldjoontes sarnast funktsionaalsust.

5.2.1 Scania

Scania on loonud enda elektriveokitega ühilduva rakenduse [53], mis kuvab teavet nii sõidukite kui ka laadimise kohta. Rakendus võimaldab sõidupargi halduril jälgida elektrisõidukite laadimisolekut, aku laetuse taset ning muid tehnilisi andmeid. Samuti kuvatakse elektrisõidukite asukohad reaajas kaardivaates.

Rakendus saadab reaajas teavitusi laadimise lõppemise ja võimalike katkestuste kohta, võimaldades laadimisprotsessi pidevalt jälgida. Kogu laadimistegevusest saab visualiseeritud ülevaate ning vastavaid aruandeid on võimalik eksportida. Lisaks võimaldab rakendus määrata kavandatud marsruutide põhjal optimaalsed laadimiskohad ja -ajad, lihtsustades sellega juurdepääsu avalikule laadimistaristule [54].

5.2.2 Diotech Group

Diotech on elektriautode laadijate tootja VOOL ametlik paigalduspartner, pakkudes elektrisõidukite laadimisseadmeid koos tarkvaralahendustega [55]. Ettevõtte

terviklahendus hõlmab objekti tehniliste võimaluste kaardistamist ning laadimislahenduse projekteerimist.

Samuti pakutakse laadimist tervikteenusena, mis sisaldab laadimistaristu paigaldust, kliendikoolitusi ning hooldus- ja käiduteenuseid. Laadimise haldusplatvorm võimaldab reaajas jälgida sõiduki olekut, kas see on vooluvõrku ühendatud, laeb või on sõidus. Lisaks kuvatakse platvormil sõiduki läbisõitu ja praegust asukohta. Süsteemist saab ülevaate nii kogu sõidupargi kui ka üksiku sõiduki energiatarbimisest kilovatt-tundides ja eurodes erinevate ajavahemike lõikes [56]. Eri kokkuleppel pakub ettevõtte ka CO₂ kvoodihaldust, mis võimaldab kliendil teenida täiendavat tulu [55].

5.3 Turulahenduste analüüs

Ühelgi kirjeldatud lahendusel ei ole funktsionaalsust, mis jälgiks kasutaja sõiduki liikumisgraafikut. Arvestades, et elektriauto ei vaja, sõltuvalt liikumisharjumustest, igapäevast laadimist, võib käesolevas lõputöös käsitletava rakenduse liikumisgraafiku funktsionaalsus osutada üleliigseks. Seda on näha ka olemasolevate lahenduste puhul, mis võimaldavad kasutajal määrata üksnes soovitud aku laetuse taseme ning kellaaja, milleks laadimine peab olema lõpetatud.

Äriklientidele mõeldud lahendused keskenduvad peamiselt laadimisvõimsuse jaotamisele, laadimistaristu haldamisele ning sõidukite jälgimisele. Lõputöös arendatavale ideele – laadimise optimeerimine liikumisgraafiku alusel – on kõige sarnasem Scania lahendus, mis planeerib laadimist pikkadel sõitudel. Samas tuleb arvestada, et Eestis on vahemaad lühikesed ja seetõttu laevad ärikliendid oma sõidukeid üldjuhul ettevõtte enda territooriumil.

6 Olemasoleva lahenduse puudused

Käesolevas peatükis tuuakse välja meeskonnaprojekti raames valminud rakenduse analüüsimisel leitud kitsaskohad.

6.1 Optimeerimine

Enne edasiarendust valminud laadimisplaani optimeerimislahendus rakendas ahnet algoritmi, mis valis Elering API-st pärinevate 15-minutiliste intervallide hulgast odavaima hinnaga ajavahemikud. Kuigi algoritm jäi edasiarendusel samaks, tuvastati kolm sisulist puudujääki, mis vähendasid optimeerimise täpsust ning usaldusväärsust.

6.1.1 Eeldus, et kasutaja naaseb pärast iga sündmust koju

Esialgne lahendus käsitles iga sündmuste vahelist ajavahemikku potentsiaalse laadimisaknana, eeldades, et kasutaja naaseb pärast igat sündmust koju. Eeldus ei vasta reaalsele kasutusjuhtudele, kus kasutaja võib külastada mitmeid asukohti ilma vahepeal koju naasmata.

6.1.2 Sõiduaja arvestamata jätmine

Esialgne lahendus ei võtnud arvesse sündmusele sõitmiseks kuluvat aega. Seetõttu võis optimeerija valida laadimiseks aja, mis lõpetab laadimise vahetult enne sündmuse algust. Tulemusena ei saavuta sõiduki aku planeeritud sõiduks piisavat laetuse taset.

6.1.3 Laadimisaja vale hindamine

Esialgne algoritm leidis vajaminevate laadimisintervallide arvu sõidu kilometraaži, aku mahu ja keskmise energiakulu põhjal, kasutades laadimiskiirust konstantse suurusena. Katsetamisel selgus, et see meetod annab tegelikust laadimisajast optimistlikumaid tulemusi, kuna ei arvesta laadimise efektiivsust mõjutavaid tegureid.

6.2 Staatiline aku laetuse tase

Süsteem ei arvestanud sõidu käigus kuluvat energiat ega laadimisest tulenevat aku laetuse taseme (ingl SoC – *State of Charge*) tõusu. Sellest tulenevalt kuvati rakenduse avalehel vananenud aku laetuse taset. Probleem kandus edasi rakenduse tulemustesse, kus kasutati optimeerimisel sõiduki esmakordsel seadistamisel sisestatud algväärtust. Selline lähenemine muutis laadimisplaani ebausaldusväärseks olukorras, kus kasutajal on päevas mitu sündmust, kuna SoC muutust ei arvestatud jooksvalt.

6.3 Laadimisplaani salvestamine

Rakenduse esialgne versioon salvestas käesoleva päeva optimeeritud laadimisplaani veebilehitseja kohalikku mälli (ingl *localStorage*) võtmega `charging_plan`. Antud päeva kogu laadimisplaani kirjutati tervikuna üle järgmiste rakenduses toimuvate muudatuste korral:

- sündmuse lisamine, muutmine, kustutamine;
- sõiduki või laadija tehniliste andmete muutmine;
- uute börsihindade lisandumine.

7 Nõuded edasiarendusele

Eraklientide küsitlusest ega äriklientide intervjuudest ei selgunud otsest vajadust arendatava rakenduse järele. Siiski näevad käesoleva lõputöö autorid potentsiaali lahenduses, mis tugineb sõidukite liikumisgraafikule.

Järgnevalt on kirjeldatud autorite visiooni nii era- kui ka äriklientide rakenduseks, kasutuslood ja edasiarenduse nõuded.

7.1 Eraklientidele mõeldud rakendus

Küsitluse vastuste põhjal võib järeldada, et seadmes lokaalselt toimival rakendusel, mis eeldab andmete käsitsi sisestamist, on Eesti turul piiratud klientuur. Olukorda muudab keerulisemaks elektrisõidukite väike osakaal. 2026. aasta 1. aprilli seisuga on Eestis registreeritud 10 675 täiselektrilist ja 3 824 pistikhübriidsõidukit, mis moodustavad 2,2% Eesti kogu sõidukipargist [1]. Samuti leidub turul juba piisavalt lahendusi, mis katavad eraklientide peamised kasutusvajadused.

Pistikhübriidide omanike laadimiskäitumist mõjutavad oluliselt elektri hind ning potentsiaalne rahaline sääst. Kuigi eeldatakse, et pistikhübriidi laetakse kord päevas, jätab osa kasutajaid sõiduki regulaarselt vooluvõrku ühendamata. Sellist käitumist soodustavad kõrgemad koduse elektri hinnad ja vähem rahaline kokkuhoid võrreldes bensiiniga sõitmisega [57]. Seega võiks laadimiskulude minimeerimine suurendada oodatavat säästu ning motiveerida pistikhübriidide omanikke oma sõidukeid sagedamini laadima.

Küsitluse vabateksti väljas mainisid väiksema päevase läbisõiduga vastajad, et nad laevad oma sõidukit vaid paar korda nädalas, mistõttu pole laadimise planeerimine nende jaoks ajakulukas tegevus. Sellest võib järeldada, et rakendus pakuks suuremat väärtust eelkõige pistikhübriidi omanikele, kes peavad väiksema akumahu tõttu sõidukit sagedamini laadima, et maksimeerida elektri toel sõitmist.

7.2 Äriklientidele mõeldud rakendus

Autorite hinnangul võiksid rakendusest enim kasu saada ettevõtted, kelle sõidukite liikumisgraafikud on ette teada. Suurimat väärtust pakuks lahendus ettevõtetele, kelle sõidud on lühemad, pikemate pausidega või kelle kasutuses on kiirlaadijad. Kirjeldatud juhtudel võimaldaks arendatav rakendus laadimisprotsessi tõhusamalt optimeerida.

Vesteldud ettevõtete hulgas ei leidunud ühtegi, kes optimeeriks sõidukite laadimist elektrihinna alusel. Fikseeritud elektripaketi tõttu laetakse sõidukite akud töövälisel ajal üldjuhul maksimaalselt täis. Olukord võiks muutuda juhul, kui ettevõtted hakkaksid kasutama elektritootjate pakutavaid lahendusi, mis võimaldavad laadida sõidukeid börsihinna alusel. Näiteks pakub Terminal AS võimalust „Ole kodus fiks, aga lae börsiga“ [58]. Kui sellist võimalust pakutaks ka ettevõtetele, omaks käesolev lahendus ettevõtetele suuremat praktilist väärtust.

7.3 Kasutuslood

Kasutuslugudes on rakendatud autorite loodud persoonasid, kelle kujundamisel tugineti nii kasutajaküsitluse tulemustele kui ka loetud kirjandusele, millele on selles töös viidatud. Persoonade loomisel peeti eelkõige silmas liikumisgraafiku funktsionaalsust edasiarendatavas rakenduses. Järgnevalt antakse ülevaade loodud persoonadest ja nende põhivajadustest.

7.3.1 Kindla graafikuga elektriauto omanik, kelle sõidukulud tasub tööandja

Elektriauto omanikuna, kellel on kindel iganädalane liikumisgraafik, soovin sisestada oma igapäevased liikumised nii, et ei peaks laadimisplaani saamiseks rutiinseid sõite korduvalt sisestama.

Aktsepteerimiskriteeriumid:

- mugav korduvate sündmuste sisestamine, mida saab kasutaja vajadusel muuta ning kustutada;
- kasutaja saab vaadata uuendatud laadimisplaani nii börsihinna graafikul kui ka kalendripõhises päevavaates;

- laadimiskulude kokkuvõte, mida saaks esitada tööandjale.

7.3.2 Muutuva graafikuga elektriauto omanik

Elektriauto omanikuna, kelle liikumisgraafik on ebaregulaarne, ei soovi ma olla kohustatud sisestama kindlat liikumisgraafikut. Saades teada oma liikumiskava, soovin kiiresti sisestada läbitava vahemaa, et saada selle põhjal optimeeritud laadimisplaani.

Aktsepteerimiskriteerium:

- rakendus omab alternatiivset võimalust laadimisplaani kiireks genereerimiseks.

7.3.3 Paindliku graafikuga pistikhübriidi omanik

Pistikhübriidi omanikuna, kellel on paindlik liikumisgraafik, soovin, et rakendus optimeeriks laadimisplaani vastavalt minu liikumisvajadustele ning pakuks võimaluse korral välja sündmuse nihutamist.

Aktsepteerimiskriteeriumid:

- rakendus peaks leidma viise laadimiskulude vähendamiseks, tehes selleks ettepaneku nihutada sündmust sama päeva piires ja/või jätta sündmus täielikult ära;
- sündmuse nihutamise ettepaneku korral kuvab rakendus uue kellaaja, millele sündmus soovitatakse üle viia;
- ettepaneku juures peaks olema kuvatud nõustumisel säästetav summa;
- ettepanekuga nõustumisel, kohandatakse laadimisplaani vastavalt muudatusele.

7.3.4 Ettevõtte elektrisõidukite laadimise eest vastutav isik

Ettevõtte poolse laadimise eest vastutava isikuna soovin sisestada kõikide sõidukiparki kuuluvate elektrisõidukite liikumigradikud, mille alusel koostatakse optimeeritud laadimisplaani. Optimeerimise tulemusena leitakse iga sõiduki jaoks laadimisajavahemik, mis minimeerib sõidukite laadimise kogukulu.

Aktsepteerimiskriteeriumid:

- rakendus optimeerib mitme elektrisõiduki laadimist mitme laadija vahel;
- rakendus optimeerib mitme elektrisõiduki laadimist mitme laadija vahel;
- laadimisi saab filtreerida sõiduki- ja laadijapõhiselt;
- mitme sõiduki liikumisgraafiku sisestamine on kasutajasõbralik.

7.4 Nõuded

Järgnevalt on sõnastatud mittefunktsionaalsed nõuded ja lisatavad funktsionaalsed nõuded, mis rahuldavad nii era- kui ka äriklientide vajadusi.

7.4.1 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded kujunesid Enefit AS tootemaniku ja arendusgrupi mentorite esialgsetest nõuetest, neid edasiarendusel ei muudetud.

- lokaalne – kõik andmed salvestatakse ja töödeldakse lokaalselt kasutaja seadmes, mis tagab rakenduse sõltumatuse pilvest ning andmete privaatsuse;
- skaleeritav – rakendust on võimalik edasi arendada ilma terviklikkust kahjustamata;
- platvormiülene kasutatavus – rakendus peab olema kasutatav nii veebis kui ka mobiiliseadmetes, pakkudes seejuures mugavat kasutajaliidest.

7.4.2 Funktsionaalsed nõuded eraklientidele

Rakendusse lisatavad funktsionaalsed nõuded kujunesid bakalaureusetöö autorite visioonist, küsitluse edasiarenduse populaarsematest valikutest ja loodud persoonadest. Valitud nõuded said valideeritud autorite praktika poolsete mentorite poolt.

- väljade automaatne täitmine – kasutaja ei pea sõiduki tehnilisi andmeid käsitsi sisestama. Aku maht, keskmine energiakulu ning aku täislaadimisele kuluv aeg täidetakse auto mudeli valimisel automaatselt;

- laadimise kogukulud – toimunud laadimistest on kokkuvõtlik lehekülg, kus kasutajale kuvatakse valitud perioodi, kas nädala, kuu või aasta laadimise kogukulud ning vastavalt päevane, nädalane või kuune keskmine kulu;
- päevase läbisõidu liugur – avalehel on kasutajal võimalus valida, kas loob kalendrisündmuse või sisestab liuguriga eeldatava läbitava kilometraaži koos laadimise lõppajaga. Kasutaja sisestatud andmete põhjal genereeritakse optimaalne laadimisplaan, mille kasutaja peab kinnitama;
- laetustaseme hoidmise vahemik – kasutaja seadetes on võimalik määrata vaikeväärtuse (20-80%) asemel kasutajal endal sobilik aku laetustaseme hoidmise vahemik, mille põhjal genereeritakse optimaalne laadimisplaan;
- väljade automaatne täitmine – kasutaja ei pea sõiduki tehnilisi andmeid käsitsi sisestama. Automudeli valimisel täidetakse automaatselt aku maht, keskmine energiakulu ning aku täislaadimisele kuluv aeg.

7.4.3 Funktsionaalsed nõuded ärikliendile

Ehkki ükski küsitletud ettevõtte ei näinud laadimise planeerimisel liikumisgraafiku funktsionaalsusest otsest kasu, soovisid käesoleva lõputöö autorid siiski katsetada selle rakendamist mitme sõiduki puhul. Arvestades asjaolu, et paljudes majapidamistes on tänapäeval rohkem kui üks auto, võib mitme sõiduki liikumisgraafiku jälgimine ja arvessevõtmine luua lisaväärtust ka erakliendile suunatud laadimiserakenduses.

- laadimise optimeerimine – kasutaja saab sisestada oma autopargis olevad sõidukid ja nende liikumisgraafikud, mille põhjal arvutatakse optimeeritud laadimisplaan arvestades koormusjuhtimist mitme laadija vahel;
- liikumisgraafiku importimine – kasutajal on võimalik importida mitme sõiduki liikumisgraafik CSV failina, mille põhjal luuakse optimaalne laadimisplaan;
- liikumisgraafiku sisestamine – rakenduse poolt oodatava formaadi saamiseks saab kasutaja alla laadida õige struktuuriga CSV faili.

8 Rakenduse edasiarendus

Valminud funktsionaalsete nõuete põhjal teostati rakenduse edasiarendus, mille käigus lahendati ka eelnevalt tuvastatud kitsaskohad. Rakenduse teostatud funktsionaalsustest annab ülevaate kasutaja tegevusdiagramm (vt Lisa 3). Järgnevalt kirjeldatakse süvitsi edasiarenduse käigus realiseeritud funktsionaalsusi.

8.1 Optimeerimismeetod

Edasiarenduse käigus täiendati laadimisaegade leidmise optimeerimismeetodit, mis koosneb neljast järjestikust sammust. Esmalt laiendatakse iga sündmuse ajavahemikku vastavalt sündmuste vahel liikumiseks kuluva ajale. Laiendatud sündmuste põhjal moodustatakse sõiduahelad. Ahelate alusel leitakse laadimiseks sobivad ajavahemikud koos igaühe jaoks vajamineva laetustasemega. Leitud ajavahemike seast valitakse ahne algoritmi abil odavaima hinnaga laadimisintervallid. Järgnevates alampeatükkides kirjeldatakse iga sammu teostust täpsemalt.

8.1.1 Liikumise arvestamise põhimõtted

Esialgses rakenduses eeldati, et kasutaja naaseb pärast iga sündmust koju. Edasiarenduses asendati see lihtsustus realistlikuma liikumisloogikaga, mis arvestab kasutaja tegelikke liikumismustreid. Loogika lähtub sündmuse lisamisel määratud algus- ja sihtpunktist ning kasutaja koduaadressist. Edasiarenduse vältel kehtivad järgmised eeldused:

- igal sündmusel liigutakse sündmuse alguspunktist sihtpunkti;
- päeva viimase sündmuse lõppedes naaseb kasutaja koju;
- järjestikuste sündmuste korral kontrollitakse, kas esimese sündmuse sihtpunkt ühtib järgmise sündmuse alguspunktiga. Ühtivuse korral eeldatakse, et kasutaja liigub otse ühelt sündmuselt teisele, naasmata vahepeal koju. Erinevuse korral

eeldatakse, et kasutaja naaseb kahe sündmuse vahel koju ning lahkub sealt järgmise sündmuse alguspunkti suunas.

Kirjeldatud loogikale tuginevad järgmised funktsionaalsused: sündmuste laiendamine sõitudeks kuluvate aegadega (vt Peatükk 8.1.2), sõiduahelate moodustamine (vt Peatükk 8.1.3), aku laetuse taseme leidmine (vt Peatükk 8.2) ning lõpuks optimaalse laadimisplaani genereerimine.

8.1.2 Sõiduaja arvestamine

Edasiarenduses lisati funktsioon, mis laiendab iga sündmuse ajavahemikku vastavalt peatükis 8.1.1 kirjeldatud liikumisloogikale (vt Peatükk 8.1.1). Iga sündmuse algusajale lisatakse aeg sündmuse algpunktist sihtpunkti jõudmiseks. Järjestikuste sündmuste korral, kus kasutaja naaseb sündmuste vahel koju, lisatakse esimese sündmuse lõpuajale koju sõiduks kuluv aeg ning järgmise sündmuse algusajale lisandub kodust sündmuse sihtpunkti sõitmise ajakulu. Päeva viimase sündmuse lõpuajale lisatakse koju sõiduks kuluv aeg.

Sõiduajad päritakse Stadia Maps Routing API kaudu, kasutades sisendina sündmuste ja kodus aadressi koordinaate.

8.1.3 Sõiduahelate tuvastamine

Sõiduahelate tuvastamiseks lisati funktsioon, mis grupeerib järjestikused sündmused ahelateks, lähtudes peatükis 8.1.1 kirjeldatud liikumisloogikast (vt Peatükk 8.1.1). Ahelasse kuulumist kontrollitakse Joonisel 8 oleva koodiga:

```
const chainBreaks =  
  !next ||  
  !locationsMatch(event.locationTo, next.locationFrom) ||  
  locationsMatch(next.locationFrom, homeAddress);
```

Joonis 8. Sõiduahelate loomine.

Ahela tuvastamine võimaldab optimeerijal leida laadimiseks sobivad ajavahemikud, mille jooksul kasutaja on kodus ning saab sõidukit laadida.

Ahela kilometraaž teisendatakse aku protsentideks Joonisel 9 kujutatud valemi abil:

```
const energyConsumedKwh = (distanceKm / 100) * energyConsumption;  
const pctConsumed = (energyConsumedKwh / batteryCapacity) * 100;
```

Joonis 9. Läbitud kilomeetrite teisendamine protsentideks.

8.1.4 Vajamineva laetuse taseme dünaamiline arvutamine

Esialgses lahenduses eeldati, et iga laadimistsükli eesmärk on saavutada fikseeritud 80% laetuse tase. Lisaks ei arvestatud aku tegelikku hetkelaetust enne iga laadimisintervalli valimist. Seetõttu jäeti arvesse võtmata nii eelneva laadimisega lisandunud kui ka sõitmisega kadunud laetustaseme muutused. Edasiarenduses arvutatakse iga vaba ajavahemiku jaoks eraldi vajaminev laetuse tase, lähtudes sellele vahetult järgneva sõiduahela energiavajadusest ning aku hetkeväärtusest.

Arvutatud energiavajadusele lisatakse 20-protsendine varu, mis tagab piisava laetustaseme ootamatute olukordade korral ning pikendab aku eluiga, vältides laadimist madalal laetustasemel [29].

8.1.5 Laadimisintervallide arvu määramine

Vajaliku intervallide arvu leidmiseks kasutati laadimisvõimsuse asemel sõiduki tegelikku laadimisega. Selline lähenemine väldib liialt optimistlike tulemuste saamist, kuna arvutus põhineb sõiduki tegelikul täislaadimise ajal, mitte staatilisel laadimisvõimsusel, mis võib laadimise käigus muutuda.

Vajalike 15-minutiliste laadimisintervallide arvu leidmiseks kasutatakse Joonisel 10 näidatud valemit:

```
const intervalsNeeded = Math.min(  
  Math.ceil(((SoCToAdd / 100) * timeToFullChargeInMinutes) / 15),  
  intervals.length,  
);
```

Joonis 10. Laadimisintervallide leidmine.

Soovitud laetustaseme (SoCToAdd) ja täislaadimise aja (timeToFullChargeInMinutes) korrutis annab soovitud laetustaseme saavutamiseks kuluva laadimisaja minutites. Elektrihinna optimeerimisel kasutatakse 15-minutilisi intervale, mistõttu tuleb intervallide arvu saamiseks eelnev tulemus jagada viieteistkümnega. Tulemuse ümardamine ülespoole tagab, et ka osaliselt vajalik intervall arvestatakse tervikliku

laadimisperioodina. Lisaks piiratakse tulemust olemasolevate hinnaintervallide arvuga, vältimaks olukorda, kus algoritm prooviks kasutada rohkem intervalle, kui on tegelikult saadaval.

8.2 Dünaamilise aku laetuse taseme modelleerimine

Edasiarenduse oluline tehniline muudatus oli aku laetuse taseme dünaamiline muutumine. Uus CalculateSoCPlan funktsioon arvutab iga 15-minutilise intervalli lõikes SoC väärtuse muutuse järgmiselt:

- kui intervall kuulub optimaalsesse laadimisaknasse, lisatakse SoC-ile laadimise käigus kogutav protsent;
- sõitmise ajal lahutatakse SoC-ist vastav energiakulu, mis arvutatakse läbitud distantssi, sõiduki keskmise energiakulu ning aku mahu põhjal.

Funktsioon tagastab massiivi SoCPlan, kus iga kirje sisaldab vastava ajahetke laetustaseme väärtust.

Selleks, et laetuse tase uueneks kasutajaliideses automaatselt, lisati funktsioon, mis leiab SoCPlan-ist järgmise SoC muutumise aja ning seab vastavaks ajaks taimeri. Muutuse saabumisel uuendatakse *localStorage* väärtusi ning seadistatakse järgmine taimer. Uus taimer seatakse iga kord, kui laetustaseme väärtust mõjutavad andmed muutuvad. Käimasolev taimer tühistatakse ka siis, kui kasutaja sulgeb vaate.

8.3 Väljade automaatne täitmine

Kasutajaküsitluse tulemusel osutus sõiduki andmete automaatne täitmine kõige oodatumaks lisafunktsionaalsuseks (vt Peatükk 3.2). Esialgses lahenduses pidi kasutaja sisestama kõik sõiduki tehnilised andmed käsitsi. Ülesanne muutis rakenduse esmase seadistamise raskemaks kui kasutaja ei teadnud sõiduki tehnilisi näitajaid peast.

Sõiduki tehniliste andmete kogumiseks kasutati tehisintellekti abi. Claude-1 [16] paluti koguda EVDB veebilehel [59] leiduvate elektrisõidukite andmed ning tagastada need Excel failina. Vastusena saadi 600 elektrisõiduki andmed, mis hõlmasid sõiduki

modelit, aku mahtu (kWh), keskmist energiakulu (kWh/100 km) ning nullist sajani laadimiseks kuluvat aega 11 kW laadimiskiirusel.

CSV faili lugemiseks kasutati rakenduses PapaParse teeki. Rakenduse käivitumisel parsitakse faili sisu ning tulemused salvestatakse vahemällu, et vältida kasutajaliidese uuenduse korral korduvat faili pärimist (vt Joonis 11).

```
let cachedVehicleData: VehicleDataInput[] | null = null;

async function getVehicleData(): Promise<VehicleDataInput[]> {
  if (cachedVehicleData) return cachedVehicleData;
  cachedVehicleData = await parseFileContent();
  return cachedVehicleData;
}
```

Joonis 11. Sõiduki andmete parsimine ja vahemällu salvestamine.

Sõiduki andmete määramise lehel kuvatakse kasutajale rippmenüü, mis sisaldab endas automudelite loendit. Mudelit saab valida rippmenüüst kui ka otsida nime järgi. Olles mudeli valinud, täidetakse automaatselt kolm välja: aku maht, keskmine energiakulu ning aku täislaadimisele kuluv aeg. Kasutajale säilib võimalus täidetud väärtusi käsitsi muuta, mis on vajalik juhtudel, kus automaatselt täidetud väljad ei vasta tegelikele väärtustele või andmebaas ei sisalda andmeid kasutaja sõiduki kohta. Seda eriti väärtuste puhul, mis on mõjutatavad mitmetest faktoritest. Nendeks võivad olla nii laadimise kiirus kui ka keskmine energiakulu, mis oleneb suures osas juhi sõidustiilist. Samuti võib sõiduki tegelik laadimiskiirus tulenevalt laadimise tingimustest erineda.

Kirjeldatud lahendus vähendab sõiduki esmase seadistamise ajakulu ning langetab rakendusega alustamise barjääri, eriti nende kasutajate jaoks, kellel puudub ülevaade oma sõiduki täpsetest tehnilistest näitajatest. Seevastu tuleb arvestada, et hetke sõidukite andmete saamise viis ei täida andmete tervikluse nõuet. EVDB veebilehe andmete uuendamisel, ei uuendata rakenduse Excel failina sisestatud andmeid. Tulevase täisväärtusliku lahenduse jaoks oleks soovitatav andmete hankimiseks kasutada ametlikku andmeallikat.

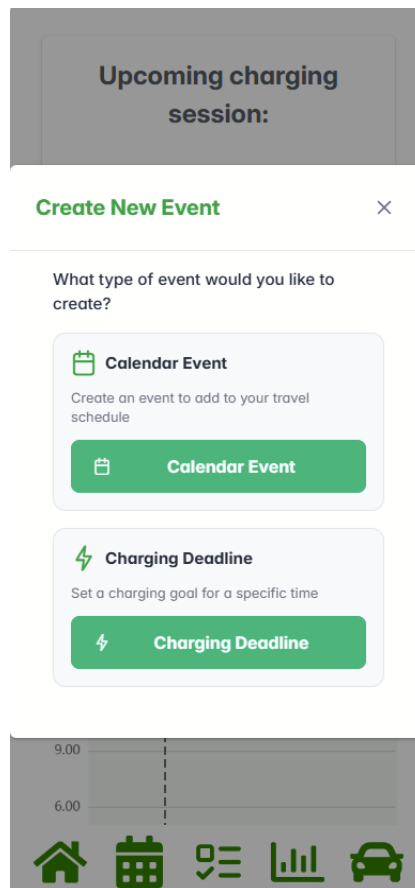
8.4 Päevase läbisõidu liugur

Kasutajaküsitluse tulemusel selgus, et kolmas populaarseim täiendus oleks päevase läbisõidu liugur (vt Peatükk 3.2). Liuguri on suunatud eelkõige kasutusjuhule, kus

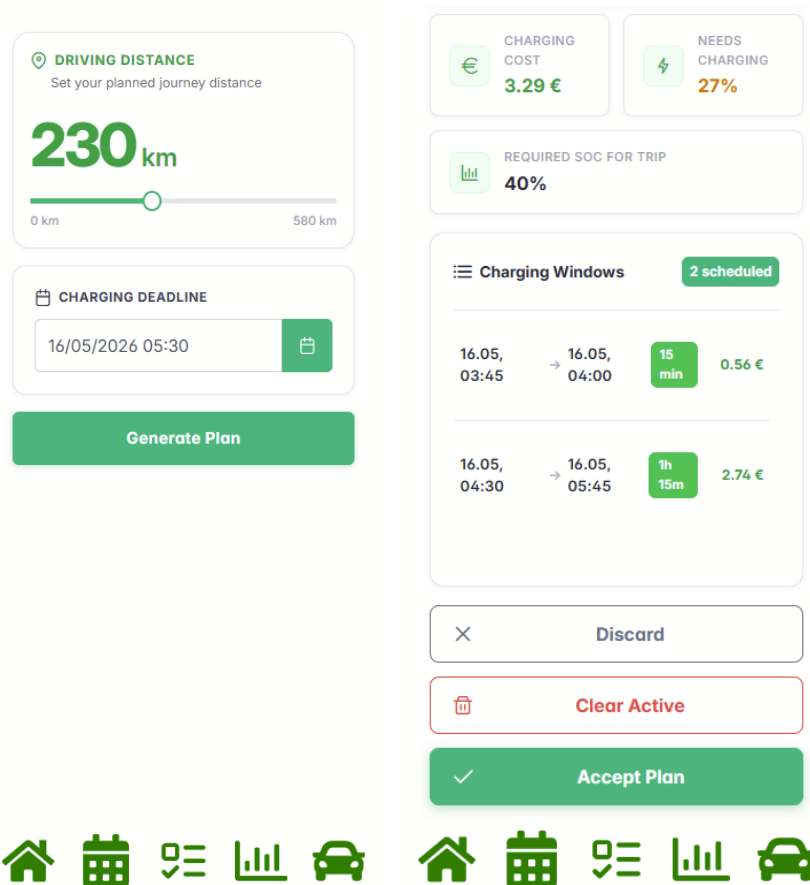
liikumisgraafik on ebaregulaarne või muutub lühikese etteteatamisajaga (vt Peatükk 7.3.2).

8.4.1 Kasutajaliides

Liuguri lehele pääseb rakenduse avalehelt „Lisa sündmus” nupu kaudu. Sellele vajutades avaneb modaalaken, kus kasutaja valib kalendrisündmuse loomise ning päevase läbisõidu määramise vahel (vt Joonis 12). Liuguri lehel on kuvatud distantsti valimise liugur ning väli laadimise tähtaja määramiseks (vt Joonis 13). Liuguri maksimaalne väärtus arvutatakse sõiduki aku mahu ja keskmise energiakulu põhjal. Pärast plaani genereerimist kuvatakse kasutajale järgmised tulemused: hinnanguline laadimiskulu, vajaminev aku laetustase, täiendava laadimise vajadus ning optimaalsed laadimisajad (vt Joonis 13).



Joonis 12. Modaalaken: kalendrisündmuse loomine või päevase läbisõidu määramine.



Joonis 13. Päevase läbisõidu määramine ja laadimisplaani genereerimine.

8.4.2 Arendusprotsess

Nii liuguri optimaalsete laadimisaegade genereerimiseks kui ka aku laetustaseme arvutamiseks kasutati samu funktsioone, mida rakendatakse kalendrisündmuste töötlemisel. Olemasoleva koodi muutmise vältimiseks salvestati liuguri laadimisplaan eraldi *localStorage* võtmesse `slider_charging_plan`, samas kui kalendrisündmuste plaan jäi võtmesse `charging_plan`. Avalehe loogika kontrollib käivitumisel, kas `slider_charging_plan` on täidetud ning kasutab selle olemasolu korral seda nii tulevaste ja hetkel käivate laadimiste *widget*-is, hinnagraafikul kui ka päevavaates.

Päevavaates lahendati probleem, kus kalendrisündmused ja liuguri laadimisajad kuvati samaaegselt. Liuguri plaan ei põhine kalendrisündmustel, mistõttu tekitas nende samaaegne kuvamine segadust ning elementide kattumise riski. Lahendusena peidetakse liuguri plaani kasutamisel päevavaates olevad kalendripõhised sündmused.

Liuguri lehele lisati nupp aktiivse liuguri plaani tühistamiseks. Selle toiminguga kustutatakse `slider_charging_plan` ning rakendus pöördub tagasi kalendrisündmustel põhineva plaani juurde, mis genereeritakse uuesti ajakohaste andmete põhjal.

Arendamise käigus käigus leiti juhtum, kus aktiivse liuguri plaani korral genereeriti uus plaan, kuid selle jõustamisest sooviti loobuda. Seoses sellega, et uue plaani genereerimisel kirjutati `slider_charging_plan` uute andmetega üle, kadus eelmine aktiivne plaan. Lahendusena võeti kasutusele ajutine *localStorage* võti `slider_charging_plan_preview`, kuhu salvestatakse genereeritud, kuid veel kinnitamata plaan. Kasutaja kinnitamisel kirjutatakse sisu `slider_charging_plan`-i üle ning ajutine kirje kustutatakse. Tühistamisel kustutatakse üksnes ajutine kirje, muutmata aktiivset liuguri plaani.

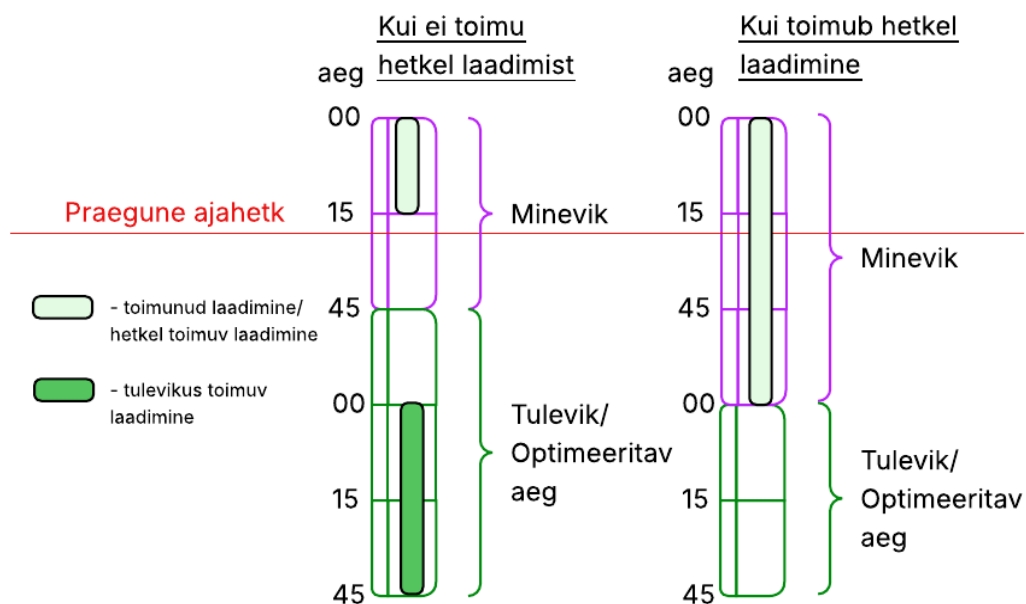
Aku laetustaseme plaan realiseeriti olemasoleva `CalculateSoCPlan` funktsiooni eeskujul (vt Peatükk 8.2), kuid ei sisaldanud mitme sündmuse haldamise loogikat. Laadimisplaanist tulenevad aku laetustaseme muutused salvestatakse olemasolevasse andmestruktuuri. Kuna kõik rakenduse komponendid kasutavad sama struktuuri, ei vajanud ülejäänud rakendus täiendavaid muudatusi.

8.5 Laadimiste ajalugu

Kasutajaküsitluse tulemused näitasid, et ligikaudu 72% vastanutest peab kasulikuks laadimisajalool põhinevat perioodilist kulude kokkuvõtet (vt Peatükk 3.2). Vastava info kuvamiseks tuli esmalt tagada toimunud laadimisakende salvestamine. Andmete talletamine algab automaatselt pärast laadimisakna käivitumist *localStorage* võtmesse `past_charging_times`.

8.5.1 Optimeeritud laadimise salvestamine

Tagamaks, et optimeerimine ei kirjutaks hetkel toimuvat laadimist üle, loodi funktsioon, mis tuvastab laadimise hetkeseisu. Kui laadimist parajasti ei toimu, alustatakse optimeerimist käesoleva ajahetke intervalli lõpust. Kui aga laadimine juba käib, optimeeritakse laadimisaegu alates käimasoleva laadimisakna intervalli lõpust (vt Joonis 14).

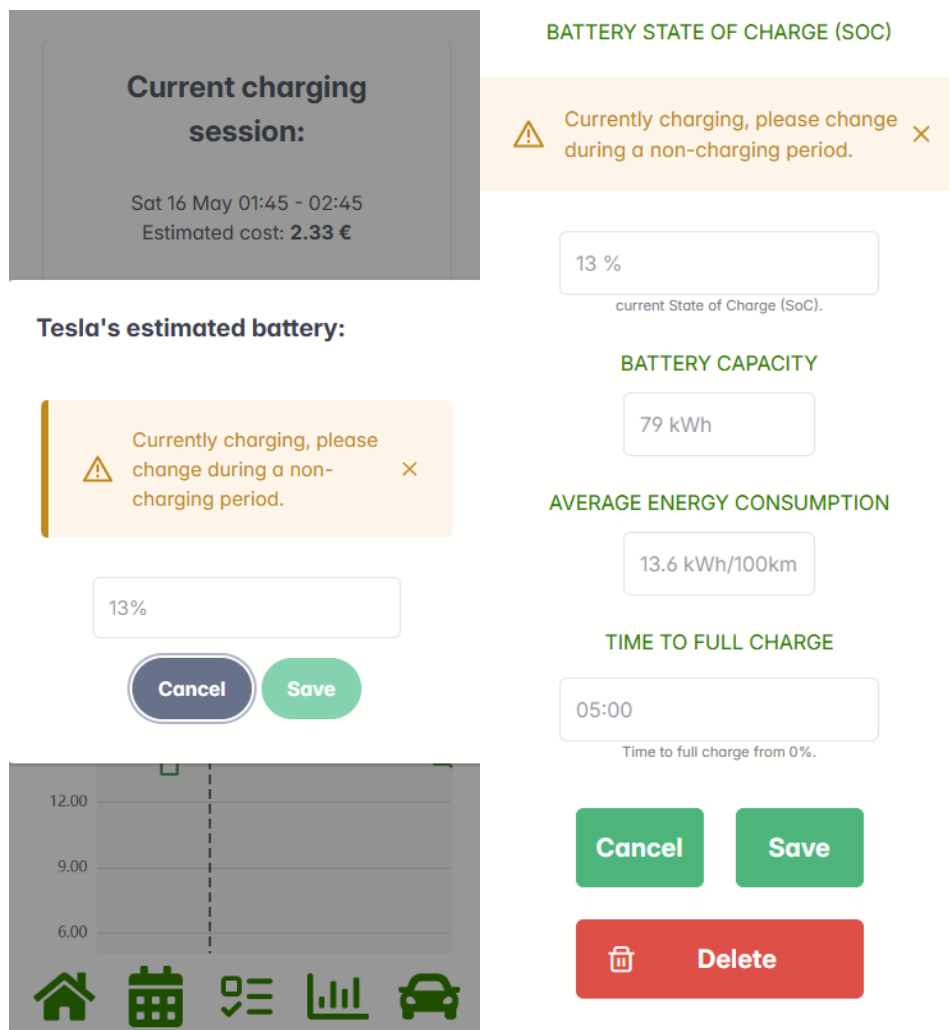


Joonis 14. Laadimisakna salvestus.

Hetkel käiva laadimise ajal kontrollitakse uue sündmuse lisamisel või olemasoleva muutmisel, kas sündmus kattub aktiivse laadimisaknaga. Kattuvuse tuvastamisel jagatakse laadimisaken uuesti intervallideks ning sellest eemaldatakse intervallid, mille täitmiseni laadimisprotsess veel jõudnud ei ole.

8.5.2 Aku dünaamilise täitumisega arvestamine

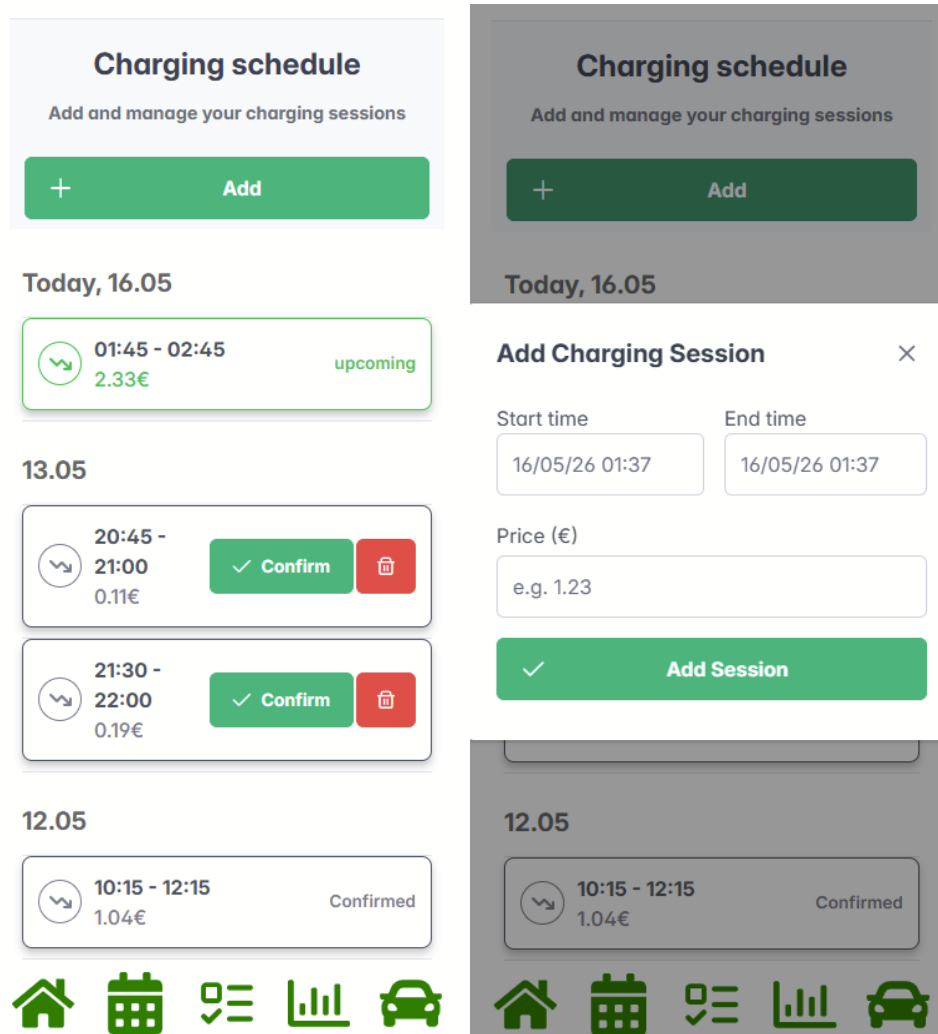
Arvestada tuli ka erijuhtumiga, kus käimasoleva laadimise ajal muudetakse sõiduki või laadija andmeid, mis käivitab laadimisplaani ümberarvutamise. Sellised muudatused ei pruugi toimuda täpselt intervalli alguses, vaid võivad aset leida intervalli kestel. Selleks, et säilitada laadimise käigus lisanduva aku laetuse taseme korrektne arvutamine, blokeeriti kasutaja jaoks laadimist mõjutavate väljade muutmine aktiivse laadimise ajal (vt Joonis 15).



Joonis 15. Blokeeritud väljad laadimise ajal.

8.5.3 Toimunud laadimiste kinnitamine

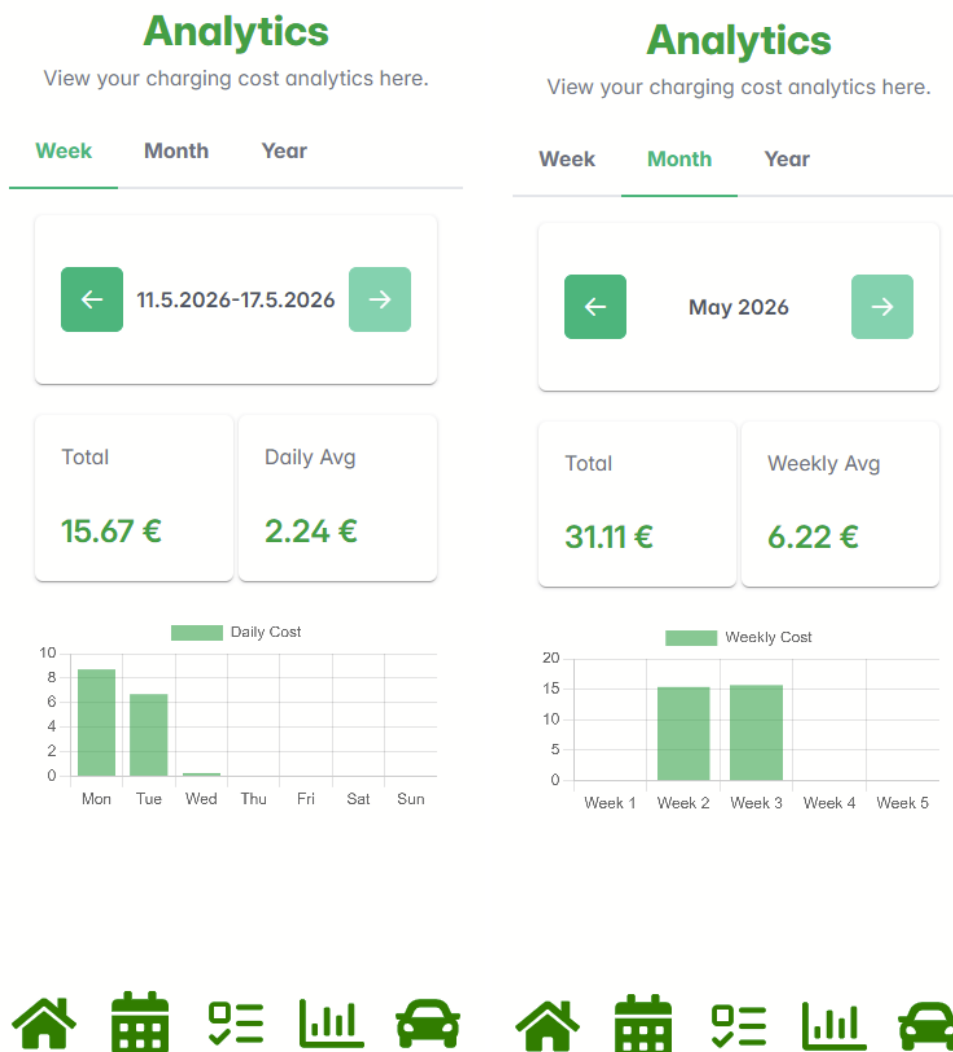
Kuna rakendus ei ole ühendatud kasutaja sõiduki ega laadijaga, tuli rakendusse lisada võimalus tegelikult toimunud laadimiste kinnitamiseks (vt Joonis 16). Lisaks optimeeritud laadimisplaanis kuvatud laadimisaegadele on kasutajal võimalik sisestada ka laadimisi, mis toimusid väljaspool optimeeritud laadimisintervalle (vt Joonis 16).



Joonis 16. Toimunud laadimiste kinnitamine ja optimeerimisvälise laadimise lisamine.

8.5.4 Laadimiskulude kokkuvõte

Seadmesse salvestatud laadimisajaloo põhjal kuvatakse rakenduses laadimiste kokkuvõtteid nädala, kuu ja aasta lõikes, mis on jaotatud eraldi vahekaartidele (vt Joonis 17). Kõikidel nendel vahekaartidel on kasutajal võimalik noolte abil liikuda käesolevast perioodist ajas tagasi, näiteks vaadata eelmise nädala andmeid. Valitud vahekaardil kuvatakse vaadeldava perioodi kogukulud ning keskmine kulu vastavalt kas päeva, nädala või kuu kohta. Juhul kui vaadeldava perioodi sisse jääb kuu- või aastavahetus, ei kaasata eelmise kuu või aasta andmeid selle perioodi analüüsi.



Joonis 17. Laadimiskulude nädala ja kuu kokkuvõtte vaated.

8.6 Liikumisgraafiku importimine

Ärikliendi kasutusjuhu toetamiseks (vt Peatükk 7.3.4) lisati rakendusse võimalus importida mitme sõiduki liikumisgraafik CSV-failina. Funktsionaalsus on suunatud kasutajatele, kes haldavad mitut sõidukit ning kelle liikumisgraafik on ette teada ja struktureeritud kujul olemas.

8.6.1 Mallifail

Liikumisgraafiku importimiseks vajalikud elemendid asuvad rakenduse kalendrilehel. Lehel on kuvatud kaks nuppu: malli allalaadimise nupp ning faili üleslaadimise nupp.

Õige andmestruktuuri soodustamiseks pakutakse kasutajale korrektse vormistusega allalaaditavat CSV faili malli, mis sisaldab nelja kohustuslikku veergu: sõidu algusaeg, lõpuaeg, läbitud vahemaa kilomeetrites ning sõiduki registreerimisnumber. Ajad sisestatakse formaadis YYYY-MM-DD hh:mm.

Allalaadimise nupu vajutamisel loob rakendus taustal ajutine HTML lingi, mis suunab kasutaja brauseri faili aadressile ning käivitab automaatselt allalaadimise. Ajutine link eemaldatakse kohe, kui allalaadimine on alanud.

8.6.2 Faili töötlemine

Faili üleslaadimiseks kasutatakse PrimeReact FileUpload komponenti, mis seadistati aktsepteerima üksnes CSV faile. Faili sisu parsimiseks kasutatakse PapaParse teeki.

CSV failides kasutatakse veergude eraldamiseks sõltuvalt kasutaja arvuti seadistatud regioonist, erinevaid kirjavahemärke. Selle probleemi lahendamiseks lisati funktsioon, mis tuvastab sisestatud faili päise alusel, kasutatava kirjavahemärgi. See edastatakse PapaParse teegile faili parsimiseks.

Sarnaselt lahendati ka kümnendkoha eraldaja probleem – liastud funktsioon teisendab enne andmete edasist käsitlemist, koma kasutavad arvud punkti kasutavateks. Mõlemad funktsioonid tagavad, et parsitud andmed vastavad rakenduse eeldatavale andmestruktuurile sõltumata kasutaja operatsioonisüsteemi lokaalidest.

Pärast parsimist määratakse faili veergude nimed rakenduse sisemistele võtmetele. Loetud kuupäevad teisendatakse Unix-ajatempliteks, mis tagab ühtse aja andmeformaadi kogu rakenduse ulatuses. Töödeldud andmed salvestatakse *localStorage*-sse.

8.6.3 Andmete valideerimine

Faili valideerimine toimub viies järjestikusel etapil, mis on realiseeritud järgmiste funktsioonidega:

- `ValidateFileType` – kontrollib, et üleslaaditud fail vastab CSV formaadile;
- `ValidateHeaders` – kontrollib kõigi nelja kohustusliku päise olemasolu;

- `ValidateDataTypes` – kontrollib aja väljade kuupäevaformaadi vastavust ning andmetüüpide korrektsust;
- `ValidateTiming` – kontrollib, et sõidu algusaeg on enne lõpuaega ning mõlemad ajad on tulevikus;
- `ValidateOverlap` – kontrollib, et sama registreerimisnumbriga sõidukil ei esine ajaliselt kattuvaid sõite. Kattuvuste tuvastamiseks rühmitatakse sõidud sõiduki kaupa ning sorteeritakse algusaja järgi, misjärel kontrollitakse iga järgmise sõidu algusaega eelneva lõpuajaga.

Valideerimise ebaõnnestumisel kuvatakse kasutajale PrimeReact Toast komponendi kaudu veateade ning faili üleslaadimise komponent tühjendatakse, et kasutaja saaks esitada parandatud faili. Eduka üleslaadimise korral kuvatakse teade õnnestumisest.

8.7 Progressiivne veebirakendus

Algses Enefit AS tellimuses sooviti rakenduse konverteerimist React Native teeki [60], kuid arenduse algusfaasis loobus tootetootja sellest nõudest. Meeskonnaprojekti keskenduti seejärel React teegis loodavate komponentide sobimisele statsionaarsetel kui ka mobiilsetel seadmetel.

Kasutajakogemuse parandamiseks muudeti Reactil ja TypeScriptil põhinev veebirakendus progressiivseks veebirakenduseks (PWA). Selleks lisati *Web App* manifest (vt Lisa 4) ja minimaalne *Service Worker* [21]. Kuigi algselt ei kavandatud ega arendatud süsteemi PWA-na, võimaldas see hilisem lisandus pakkuda nutiseadmes allalaetavale rakendusele sarnast kasutajakogemust. Selle muudatuse tulemusena saab kasutaja lisada rakenduse oma seadme kodulekraanile, mis teeb selle igapäevase kasutamise oluliselt kiiremaks ja mugavamaks.

8.8 Testkasutajate tagasiside

Täiendatud rakendust testiti seitsme kasutajaga, kes olid osalenud eelnevalt läbi viidud küsitluses (vt Peatükk 3). Testimise eesmärk oli hinnata täiendatud rakenduse kasutatavust ning koguda tagasisidet edasiarenduse võimaluste kohta.

8.10.1 Liuguri täpsuse hindamine

Testimisel ilmnis olukord, kus päevase läbisõidu liuguril (vt Peatükk 8.4) kuvatav maksimaalne lubatud läbisõit ei vastanud kasutaja sõiduki tegelikule ulatusele. Kasutaja oli sisestanud sõidukiandmetesse kasutatava aku mahu – mitte maksimaalse, vaid sõidukitootja poolt kasutatavaks lubatud mahu – ning armatuuril kuvatava keskmise energiakulu, mis oli madalam kui tootja veebilehel märgitud väärtus. Kuna rakendus arvutab maksimaalse läbisõidu aku mahu ja keskmise energiakulu põhjal, kuvati liuguril tegelikust suurem väärtus – 53 km, kuigi kasutaja pistikhübriidi tegelik maksimaalne elektriline läbisõit on 45 km.

Kasutaja valis siiski liuguri maksimaalse väärtuse, soovides aku täis laadida. Rakendus määras soovitud laetustaseme saavutamiseks vajalikuks laadida viiel 15-minutilisel intervallil. Testimisel jälgiti aku läbisõidu ulatust iga intervalli järel (vt Tabel 1).

Tabel 1. Auto laadimise andmed.

Intervall	Läbisõidu ulatus	Laadimisaeg 100%-ni
0	19 km	1 h 35 min
1	26 km	1 h 15 min
2	37 km	55 min
3	42 km	20 min
4	44 km	20 min
5	45 km	0 min

Testimine näitas, et vaatamata liuguril kuvatavale ebatäpsele maksimaalsele läbisõidule ennustas rakendus vajaminevate intervallide arvu korrektselt – viie intervalli järel oli sõiduk täis laetud ning tegelik maksimaalne läbisõit saavutatud. See kinnitab, et laadimisaja põhine intervallide arvu arvutamine (vt Peatükk 8.1.5) toimib korrektselt ka juhul, kui kasutaja sisestatud energiakulu erineb tootja andmetest.

Liuguril kuvatava vale maksimaalväärtuse põhjuseks on kasutaja sisestatud keskmise energiakulu ja tootja andmete vaheline erinevus. Kuna rakendus kasutab maksimaalse läbisõidu arvutamiseks kasutaja sisestatud energiakulu, kuid aku maht pärineb tootja

andmetest, toob nende väärtuste ebakõla kaasa ebatäpse tulemuse. Täisväärtusliku lahenduse puhul tuleks kaaluda, kas maksimaalse läbisõidu arvutamisel kasutada tootja standardset energiakulu väärtust või kasutaja sisestatud väärtust.

8.10.2 Tagasiside funktsionaalsustele

Testimisel osales ka Volvo pistikhübriidi kasutaja. Volvol on loodud rakendus, mille põhifunktsiooniks on laadimise alustamine ning lõpetamine läbi rakenduse. Sellest tulenevalt on kasutaja sunnitud jälgima elektri hindu eraldi veebilehe kaudu. Kasutajale ei valmistanud kalendri funktsionaalsuse kasutamine raskusi ja märkis, et kasutaks rakendust igapäevaselt juhul, kui see võimaldaks laadimist otse rakenduse kaudu alustada ja lõpetada.

Testimise käigus pakuti välja erinevate kasutajate poolt järgmised edasiarenduse ideed: sündmuse algusaja sisestamisel automaatne lõpuaja uuendamine, tihedamini läbitavate marsruutide salvestamise võimalus, plaanitaval teekonnal laadimisvõimaluste olemasolu kuvamine ning sündmuste lisamine häälkäskluse abil.

8.10.3 Üldised järeldused

Seitsmest testijast kolm leidsid, et kasutaksid rakendust igapäevaselt, kuna rutiinne liikumisgraafik muudab kalendri täitmise hoomatavaks. Kõik kolm on pistikhübriidi omanikud, kes laevad sõidukit igapäevaselt. Ülejäänud neli testijat pidasid aga kalendri täitmist liialt aeganõudvaks. Kõik testijad kiitsid sõidukiandmete automaatse täitmise funktsionaalsust, mis lihtsustas rakenduse esmast seadistamist. Laadimisaegade optimeerimise puhul avaldas positiivset muljet sündmusele sõiduks kuluva aja arvestamine ning analüüsileht andis hea ülevaate kogukuludest.

9 Analüüs ja järeldused

Praegu on elektrisõidukite laadimist optimeerivad lahendused kasutajate tegelikest liikumisvajadustest sageli eraldatud. Olemasolevad lahendused teevad laadimisotsuseid kas sõiduki või laadijaga suhtlevate teenuste kaudu või kasutaja poolt käsitsi sisestatud parameetrite põhjal.

Käesoleva lõputöö autorid arendasid edasi meeskonnatööna loodud rakendust, mis salvestab ja töötleb andmeid lokaalselt kasutaja seadmes. Lõputöö peamine eesmärk oli välja selgitada, kas kõnealusel rakendusel on turupotentsiaali. Selleks viidi läbi era- ja äriklientide küsitlused ning analüüsiti meeskonnaprojekti raames valminud rakenduse puudusi. Töö käigus realiseeriti kümnest püstitatud nõudest kaheksa, mille tulemusi valideeriti sihtrühma seas korraldatud kasutajatestimisega.

Järgnevates alapeatükkides analüüsitakse valminud rakendust ja realiseerimata jäänud nõudeid ning esitatakse võimalikud suunad edasiseks arenduseks. Lisaks hinnatakse lahenduse ärilist potentsiaali ning antakse arenduslogide põhjal ülevaade autorite individuaalsest panusest projekti.

9.1 Algoritmi valik

Ahne algoritm ei taga optimaalseimat tulemust [61], mistõttu soovitasid Enefit AS-i mentorid kasutada lineaarprogrammeerimist (LP). Sellega alustas tööd üks meeskonna praktikantidest, kelle lõputöö teemaks oli erinevate optimeerimisalgoritmide võrdlemine käesoleva rakenduse kontekstis [62].

Vaatamata LP matemaatilisele optimaalsusele ei vastanud praktikandi loodud lahendus rakenduse nõuetele, mistõttu loobuti selle kasutuselevõtust. Aku laetuse tase vähenes lineaarselt sündmuse algusest lõpuni, arvestamata tegelikku sõiduaega. Samuti ei võetud arvesse koju sõitmiseks vajaminevat energiat. Lisaks oli laadimise ülempiiriks seatud 80%, mis välistab pikemateks sõitudeks vajaliku laetuse taseme saavutamise.

Teaduskirjanduses on leitud, et LP ja ahne algoritmi lahendused saavutavad laadimiskulu minimeerimise osas sarnase tulemuse. Erinevus ilmneb peamiselt arvutuskooormuses. Samuti leitud, et ahnete algoritmide arvutuslik efektiivsus on eriti sobiv ressursipiiranguga seadmete jaoks, mis ei saa kasutada pilvepõhist töötlust [63]. Rakendus töötab täielikult lokaalselt kasutaja seadmes, mistõttu on arvutuslik efektiivsus oluline kriteerium algoritmi valikul.

Eeltoodud põhjustel otsustasid autorid täiustada esialgset ahnet algoritmi, lahendades tuvastatud puudujäägid.

9.2 Laadimisaja kasutamise eelis laadimisvõimsuse ees

Peatükis 8.1.5 kirjeldatud lähenemise põhjendamiseks tuuakse näide Audi Q6 SUV e-tron quattro andmete põhjal. Aku maht 94,9 kWh ning laadimisaeg nullist saja prosendini kiirusel 7.4 kW on 15 tundi. [64]

Staatilise laadimisvõimsuse põhjal arvutatud laadimisaeg:

Laadimisaeg tundides = aku maht (kWh) / laadimiskiirus (kW)

Laadimisaeg tundides = 94.9 kWh / 7.4 kW = 12 tundi ja 48 minutit

Valem [kui leiame siis lisada viide] annab tulemuseks 12 tundi 48 minutit, kuid tegelik laadimisaeg on 15 tundi. Erinevus on tingitud laadimist mõjutavatest teguritest, nagu välistemperatuur, laadimisefektiivsuse langus kõrgel laetuse tasemel ja laadija ning aku vanusest tulenev võimsuse kõikumine [29].

Toodud näide illustreerib, miks staatiline laadimisvõimsus ei peegelda tegelikku laadimisaega ning põhjendab tegeliku laadimisaja kasutamist intervallide arvu arvutamisel.

9.3 Elektrisõidukite mudelipõhiste andmete saamine

Esmalt uuriti avalikke elektrisõidukite andmeid pakkuvaid API-teenuseid. Selgus, et kõik sobivad teenused, pakuvad andmeid üksnes tasulise API kaudu, mis ei sobinud prototüübi arenduse konteksti. Alternatiivse andmeallikana kasutati EVDB veebilehte

[59], mis sisaldab põhjalikku ülevaadet elektrisõidukite tehnilistest näitajatest. Katsetati andmete automaatset kogumist kasutades veebikraapimise (ingl *web scraping*) meetodil, kuid katse ebaõnnestus. Veebileht blokeeris automaatsed päringud, et vältida ligipääsu andmetele, mis on kättesaadavad tasuta API kaudu.

Tagasilöökide tõttu pöörduti sõiduki tehniliste andmete kogumiseks tehisintellekti poole. Tööriistaks valiti Claude, kuna see suudab töödelda mahukaid veebilehti tervikuna ning samal ajal meeles pidada kasutaja antud käsku.

Mudeli valimisel automaatselt kuvatav laadimisaeg arvutatakse 11 kW laadimiskiiruse põhjal. See väärtus valiti vaikeväärtuseks, kuna see jääb kõige levinuma vahelduvvoolu laadimistüübi (Level 2 AC) piiride – 7,4 kW ja 22 kW – vahele [65].

9.4 Päevase läbisõidu liugur

9.4.1 Laadimisplaani salvestamine *localStorage*-isse

Esialgne idee oli salvestada liuguri laadimisplaani samale *localStorage* võtmele, kus hoitakse kalendrisündmuste plaani. Laadimisplaanide eristamiseks oleks lisatud plaani tüüpi tähistav väli. Sellisel juhul peaks iga plaani lugev komponent kontrollima välja väärtust, mis suurendab komponentide vahelist sõltuvust ning muudab tulevase arenduse keerulisemaks.

Eraldi võtmete kasutamine järgib ühe vastutuse põhimõtet [66], mille kohaselt esindab iga võti ühte selgelt piiritletud andmekogumit. See tagab, et komponent, mis vajab üksnes kalendrisündmuste laadimisplaani, ei pääse kunagi ligi liuguri plaanile.

9.4.2 Olemasolevate funktsioonide taaskasutamine

Arendamisel järgitud puhta koodi põhimõtted tegid liuguri ühendamise olemasoleva loogikaga vaevatuks. Optimaalsete laadimisaegade genereerimiseks taaskasutati kalendrisündmuste laadimisaegade leidmise funktsioone. Sisenditena anti funktsioonile ajavahemik optimeerija käivitamise hetkest kuni kasutaja määratud tähtajani ning tühi kalendrisündmuste massiiv. See lähenemine välistab koodi kordumist ning tagab, et liuguri tulemused põhinevad samadel arvutustel, mida kasutatakse kalendrisündmuste laadimisaegade optimeerimiseks.

Aku laetustaseme muutumise plaani arvutamiseks realiseeriti eraldi funktsioon, mis kasutab olemasolevat laadimise lisamise loogikat. Liuguri kasutusjuhul puuduvad konkreetsed kalendrisündmused, mistõttu lähtutakse energiakulu arvutamisel üksnes kasutaja sisestatud kilometraažist. Laadimise tähtajal vähendatakse aku laetustaseme väärtust läbisõidust tuleneva energiakulu võrra.

9.4.3 Liuguri lehele navigeerumine

Liuguri lehele ligipääsu kujundamisel kaaluti mitmeid variante: eraldi nupp avalehel, navigeerimisriba laiendamine ning olemasoleva nupu kohandamine. Uue nupu lisamine avalehele oleks suurendanud kognitiivset koormust, samuti oleks navigeerimisriba laiendamine muutnud rakenduse struktuuri keerulisemaks. Lõpliku lahendusena täiendati olemasolevat „Lisa sündmus“ nuppu modaalaknaga, kus kasutaja valib soovitud toimingu.

9.4.4 „Genereeri“ nupu olulisus

Liuguri lehele lisati nupp „Genereeri“, vältimaks olukorda, kus plaani arvutamine käivituks reaalselt iga liuguri liigutusega. Liuguri komponendi väärtused muutuvad lohistamisel kiiresti, mistõttu reaalselt arvutus põhjustaks tarbetut arvutuskooormust ja halvendaks kasutajakogemust. Nupu vajutamine eraldab andmete sisestamise arvutuse käivitamisest, mis on eriti oluline kohalikul töötavas rakenduses, kus kõik arvutused toimuvad kasutaja seadmes.

9.5 Liikumisgraafiku importimine

Päevase läbisõidu liugur on suunatud kasutusjuhule, kus kasutaja liikumisgraafik ei ole ette teada ning sündmuste lisamine kalendrisse ei ole otstarbekas (vt Peatükk 8.4). Järgnevalt põhjendatakse liuguri arendamisel tehtud tähtsamaid disaini- ja arhitektuuriotsuseid.

9.5.1 CSV-formaadi valik

Liikumisgraafiku importimiseks valiti CSV formaat, mis on laialdaselt kasutatav ning toetatav suuremate tabelarvutustarkvarade poolt, sealhulgas Microsoft Excel, Google

Sheets ja LibreOffice Calc. JSON- või XML-formaadiga on CSV-faile võimalik luua ja muuta ilma eriliste tehniliste teadmisteta, mis muudab selle lihtsasti kasutatavaks.

9.5.2 Brauseripoolne töötlemine

Faili töötlemine toimub täielikult brauseri poolel, ilma serveripäringuteta, mis on kooskõlas rakenduse lokaalse andmetöötamise põhimõttega (vt Peatükk 7.2.1). See on saavutatud kasutades PapaParse teeki, mis on loodud brauserikeskkonnas töötamiseks. Suurte failide puhul töötleb PapaParse faili sisu osade kaupa, mistõttu ei pea brauser kogu faili ühe korraga mälli laadima. Tänu sellele töötab rakendus sujuvalt ka mahukate sõidugraafikute korral.

9.5.3 Valideerimise kihistatud lähenemine

Valideerimise järgib kihilise struktuuri lähenemist: esmalt kontrollitakse faili tüüpi, seejärel struktuuri, andmetüüpe, sõidu lõpu ja algusaega ning lõpuks kattuvusi. Iga järgmine etapp eeldab, et eelmine on edukalt läbitud. Selline järjestus tagab täpse veateate, mis muudab probleemi tuvastamise kasutaja jaoks lihtsamaks.

Failitüübi valideerimisel kombineeriti MIME tüübi ja faililaiendi kontroll. Üksnes MIME tüübile tuginemisel võib esineda probleeme, kuna sama CSV fail võib sõltuvalt kasutaja operatsioonisüsteemist ja tarkvarast omada erinevat MIME tüüpi – näiteks Excelist üleslaetud CSV fail võib omada tüüpi vnd.ms-excel mitte csv. Faililaiendi kontrolli lisamine tagab, et rakendus aktsepteerib CSV faile sõltumata nende MIME-tüübist.

9.5.4 Malli ankurelement

Mallifaili allalaadimiseks valiti brauseri poolel töötav lahendus, mis ei nõua serveripoolset loogikat. See on kooskõlas rakenduse lokaalse andmetöötamise põhimõttega ning väldib vajadust failiserveri järele. Alternatiivina oleks võimalik mallifail serverist dünaamiliselt genereerida, kuid see oleks vastuolus nimetatud nõudega ning lisaks ebavajalikku keerukust.

9.6 Progressiivne veebirakendus versus React Native

Kasutajakogemuse parandamiseks uuriti võimalust konverteerida olemasolev Reacti veebirakendus React Native'i rakenduseks, mis oli algselt ka tooteomaniku soov. Teiste arendajate varasemate arenduskogemuste põhjal hinnati see lahendus ajamahukaks, kuna Reacti ja React Native'i komponendid on küll struktuurilt sarnased, kuid kõik olemasolevad Reacti komponendid tuleb React Native'i jaoks ümber kirjutada [60].

Rakenduse muutmine PWA-ks osutus seevastu kiireks lahenduseks. Kuigi PWA-de jõudlus ei ole piisav keerukamate rakenduste arendamise jaoks, sobib see käesoleva rakenduse tagasihoidlikemate nõuete puhul [67]. Lisaks võimaldaks PWA edasiarendusel lisada kasutajale *push notification*-id näiteks laadimisakna alguse või lõpu kohta.

9.7 Realiseerimata nõuded

Arendusprotsessi käigus jäi kümnest püstitatud nõudest realiseerimata kaks. Üks nõue osutus arenduse käigus vastuoluliseks süsteemi optimeerimismeetodiga ning teise realiseerimiseni ei jõutud ajaliselt.

9.7.1 Laetustaseme hoidmise vahemik

Optimeerimisel leitakse sõiduahela läbimiseks vajaminev laetuse tase, millele lisatakse 20-protsendiline varu. Saadud tulemuselt lahutatakse sõiduki hetke laetuse tase ning arvutatakse vajaminevate laadimisintervallide arv.

Laetustaseme ülemise piiri määramise funktsionaalsust ei peetud vajalikuks realiseerida, kuna optimeerimine lähtub otseselt teekonna läbimiseks vajalikust energiast. Kasutaja poolt seatud range ülemine piirmäär võib teatud juhtudel takistada teekonnaks vajaliku laetuse saavutamist. Näiteks kui laadimisintervalli lõpuks on laetustase 70% ning järgmise intervalliga tõuseks see 81%-ni, jätkaks 80% ülemine piirmäär järgmise intervalli ära. Sellisel juhul piirduks aku laetuse tase vaid 70%-ga. Täpse laetustasemeni laadimine oleks saavutatav edasiarenduses, kus laadimisintervallid leitakse 15-minutiliste intervallide asemel minutitäpsusega.

9.7.2 Rakendus suunatuna ettevõttele

Valminud rakenduses ei ole mitme sõiduki sisestatud sõidugraafikud veel ühendatud optimaalse laadimisplaani genereerimisega. Kuigi andmed salvestatakse *localStorage*-isse, ei kaasata neid hetkel optimeerimisprotsessi. Ärikliendile suunatud täisväärtusliku lahenduse tagamiseks peab rakendus toetama laadimisplaanide paralleelset genereerimist mitmele sõidukile, arvestades sealjuures nende prioriteete (vt Peatükk 7.4.3). Samuti tuleb arvestada rohkem kui ühe laadijaga. Tuleks luua vaade, mis kuvab laadimisplaanid laadijate kaupa, näidates laadimisajavahemikke ning nendega seotud sõidukeid.

9.8 Edasiarenduse võimalused

Edasiarenduses tuleks arvestada erinevate kasutajate vajadustega ning keskenduda konkreetsele sihtrühmale. Sellele aitab kaasa tihe koostöö igapäevaselt elektriautosid kasutavate eraisikute või ettevõtetega on oluline suunatus tulevikule, arvestades, et elektriautod muutuvad võimsamaks ning laadijad kiiremaks. Samuti tuleb silmas pidada alates 2027. aastast jõustuvat kohustust, mille kohaselt peavad erakasutuseks paigaldatavad laadijad olema V2G valmidusega [68].

Kuna kõik elektriauto omanikud ei pea laadimisele igapäevaselt mõtlema, võib liikumisgraafiku kalendrisse sisestamine tunduda neile ebavajalik. Juhtudel, kus laadimine ei toimu iga päev, võiks rakendus kuvada prognoositud hindu, sest Nord Pooli börsihinnad on praegu ette teada vaid kuni kaheks päevaks. Elektri hindade mõjutavatest teguritest ja prognoosimise täpsusest on antud põhjalik ülevaade magistritöös „Nordpooli elektri hinna analüüs ja ennustumudeli arendamine“. Sellest tööst nähtub, et Eesti turu hindade mõjutavad kõige enam Soome, Baltimaade, Norra ja Rootsi elektri hinnad [69]. Prognooside toel saaks auto laadimist planeerida mitu päeva ette, lähtudes kasutaja tegelikust liikumisvajadusest.

Ekslikult arvatakse, et ilm on peamine elektri hindade kujunemise tegur, kuid tegelikkuses mõjutab ilmastik oluliselt elektrisõiduki energiatarbimist. Paljud sõidukitootjad prognoosivad läbisõitu ajaloolise energiatarbimise põhjal, mis ei võta arvesse sõidutingimuste muutumist. Täpsema tulemuse saavutamiseks tuleks aga arvestada nii ilmastikuolude, liikluse kui ka juhi sõidustiiliga [70]. Lisaks aku

laetustaseme (SoC) täpsemale prognoosimisele võimaldaks ilma integreerimine täiustada ka optimeerimisalgoritmi, arvutades sõiduki keskmist energiakulu vastavalt välistemperatuurile – sellist lahendust kasutab ka Computernik OÜ arendatud ja hallatav veebiplatvorm energiakulu kuvamiseks [71]. Kasutades Stadia Maps-i tasuta versiooni, on võimalik algoritmi lisada ka reaalse liiklusolude mõju, seda nii sihtkohta sõitmisel kui ka tagasiteel [72].

Kasutusmugavuse parandamiseks saaks kasutada *Edge-Based* GPS jälgimist [73], et koguda reaajas andmeid kasutaja liikumisharjumuste kohta. Tulevaste sõitude prognoosimiseks on võimalik rakendada TensorFlow Lite raamistikku [74]. Antud lahendus võimaldaks masinõppemudeleid jooksutada lokaalselt kasutaja seadmes, tagades asukohaandmete privaatsuse ning kasutaja ei peaks oma sõite käsitsi kalendrisse sisestama.

Ettevõtetele suunatud edasiarendust tuleks laiendada selliselt, et rakendus toetaks mitme laadija ja sõiduki samaaegset haldamist (vt Peatükk 9.7.2). Rakendus peab suutma arvestada sõidukite liikumisvajadustega ning teostama laadijate vahel elektrivõimsuse reaajas jaotamist.

9.9 Äriline kasum

Teadaolevalt ei ole varem loodud sellist rakendust, mis optimeeriks laadimist otseselt kasutaja liikumisplaani põhjal. Kuna loodud rakendus ei kogu sõiduki ja liikumisvajaduse andmeid automaatselt, sõltub optimeerimise edukus kasutaja sisestatud infost. Kasutajatega läbiviidud testimise tulemused kinnitavad, et rakenduse praktiline väärtus sõltub paljuski inimese iganädalasest rutiinist ja päevaplaanist. Loodud lahenduse peamine eelis turul olevate alternatiivide ees seisneb auto andmetel põhinevas arvutuskäigus. Erinevalt olemasolevatest turulahendustest, kus sõidukiga sidumise võimalus on piiratud teatud automudelitega, võimaldab loodud rakendust kasutada mis tahes sõidukiga.

Kasutusmugavuse tagamiseks peaks rakendus olema ühendatav, kas kasutaja laadijaga või autoga, et teada mis on auto hetkene laetustase ning millal toimub laadimine. See eemaldaks vajaduse käsitsi andmete sisestamiseks, kuid rakendus muutuks pilveteenustest sõltuvaks. Auto tootja API-ga liidestamine on kulukas (vt Peatükk

5.1.3). Kasumit saaks teenida pakkudes lahendust koos Enefit AS elektrilepingu ja kodulaadijaga.

9.10 Meeskondlik hinnang ja tööde logid

Projekti eesmärgiks oli analüüsida meeskonnaprojektina valminud rakendust ning edasiarenduste põhjal langetada otsus, kas rakendusel on turupotentsiaali. Nelja kuu jooksul saavutati eesmärk omavahelise pideva koostööna, mis hõlmas olemasoleva rakenduse, turulahenduste ja potentsiaalsete klientide analüüsimist ning rakenduse arendamist.

Lõputöö koostamise käigus muutus selle fookus, kui tekkis idee pakkuda rakendust ka äriklientidele. Veebist kogutud teabe põhjal tundus, et tegemist on palju lubava suunaga. Paraku ei õnnestunud hetkel leida sobivat äriklientide sihtrühma, kes vastaks rakenduse kasutamise eeldustele: kasutaks elektribörsi paketti ning teaks ette järgmisel päeval läbitavat kilometraaži. Sellest hoolimata andis protsess väärtusliku kogemuse potentsiaalsete klientidega suhtlemisel.

Töö autorid hindavad üksteise panust töösse samaväärselt. Lõputööle kulutatud aeg logiti veebirakenduses Toggl Time Tracker, mille alusel tehti kahepeale tööd kokku 861 tundi (vt Lisa 7).

9.10.1 Teostatud tööde detailne kirjeldus

Tabelid 2 ja 3 kirjeldavad käesoleva lõputöö autorite 26.01.2026-17.05.2026 tehtud tööd nädalate kaupa.

Tabel 2. Piia Carmen Milkovi töö kirjeldus.

Nädal	Kirjeldus
26.01-01.02	Kasutaja lisatud sündmused ei saa kattuda.
02.02-08.02	Laetustaseme muutmise UI komponendi parandamine; <i>LocalStorage</i> -s andmete muutuse korral muutuvad ka andmed esilehel.
09.02-15.02	Date formati parandamine; Salvestatud aadresside uuendamine <i>localStorage</i> -s.

Tabel 2. Piia Carmen Milkovi töö kirjeldus.

Nädal	Kirjeldus
16.02-22.02	Lõputöö idee tutvustamine juhendajale; Lisatava funktsionaalsuse mõtlemine.
23.02-01.03	Koodi refaktoreerimine. Kalendris saab sündmuse lisada kui vajutatakse päeva kasti peale.
02.03-08.03	Esilehel uueneb hindade- ja laadimisgraafik automaatselt kui sõiduki SoC-i uuendatakse.
09.03-15.03	Lõputöö küsimustiku koostamine.
16.03-22.03	Asukoha algkoht ja sihtkoht ei saa olla sama väärtus sündmustel; Kalendripäevavaatest sündmuse kustutamine uuendab laadimisplaani.
23.03-29.03	Meeskonnaliikme poolt loodud SoC_plan-iga tutvumine; Sündmuseid saab muuta kui need toimuvad praegusest hetkest edasi, käimasolevaid või möödunud saab ainult kustutada.
30.03-05.04	Kasutajaküsitluse vastuste analüüsimine + kõne juhendajaga; Laadimiskulude kokkuvõtte lehekülje komponentide planeerimine; Lõputöö dokumendi kirjutamine.
06.04-12.04	Laadimiskulude kokkuvõtte lehekülje komponentide sobitamine erinevas suurustes ekraanile; Eestis elektriveokite kasutuse uurimine; Lõputöö dokumendi kirjutamine.
13.04-19.04	DPD operatsioonijuhiga kõne; Laadimiste minevikku salvestamise loogika väljamõlemine; Lõputöö dokumendi kirjutamine.
20.04-26.04	Telia haldusjuhiga kõne; Laadimise mineviku salvestamine; SoC_plani ja mineviku laadimiste ühendamine; Mineviku kuvamine hindade- ja laadimisgraafikul; Laadimiskulude analüüs võtab toimunud laadimiste andmeid; Lõputöö dokumendi kirjutamine.
27.04-03.05	Nuppude blokeerimine kui laadimine käib; Sündmuse lisamisel/muutmisel hetkel käimasoleva laadimisaknaga kattumise parandamine; Meeskonnaliikme poolt loodud <i>Going Home</i> funktsionaalsuse testimine; Automaatsete uuenduste parandamine avalehel; Lõputöö dokumendi kirjutamine.

Tabel 2. Piia Carmen Milkovi töö kirjeldus.

Nädal	Kirjeldus
04.05-10.05	Laadimiskulude kokkuvõtte leheküljel toimuvate laadimiste keskmiste ja kokku arvutamine; PWA versiooni loomine; Ajaloos ja Gisela loodud SoC loogika ühendamine; Elektrihindade prognoosimise uurimine; Rakenduse avalikuks tegemine Azure-ga; Lõputöö dokumendi kirjutamine.
11.05-17.05	Slaider loogika ühendamine ajaloo ja toimunud laadimistega; Toimunud laadimiste kinnitamine+kustutamine ja optimeeritud laadimisplaanist väljaspool laadimiste lisamine; Tegevusdiagrammi loomine; Kasutaja poolse testimise läbiviimine; Lõputöö dokumendi kirjutamine.

Tabel 3. Gisela Nüüd töö kirjeldus.

Nädal	Kirjeldus
26.01-01.02	Olemasoleva aku laetustaseme koodiga tutvumine ning toimimise testimine; Esiialgse SoCPlan-i loomine, mis kuvab SoC väärtust iga teadaoleva elektrihinna ajahetke jaoks.
02.02-08.02	Avastatud probleem SoCPlan-i loogikaga: kui kasutaja muudab SoC väärtust käsitsi, uuendatakse esimest SoCPlan kirjet. Kui esimese kirje ja praeguse ajahetke vahel on toimunud laadimine, näitab rakendus hetkest SoC väärtust valesti; SoCPlan-i esimene kirje vastab käimasoleva intervalli ajatemplile; Avastatud probleem säilib.
09.02-15.02	SoCPlan-i esimene kirje vastab käimasolevast intervallist ühe intervalli võrra varasemale ajatemplile; Lõputöö küsimustiku koostamine.
16.02-22.02	Laetud protsent lisandub laadimise algusele järgneval ajatemplil; SoCPlan arvestab sõitmise kaasneva energiakuluga; SoC väärtus uueneb laadimisel või sõitmisel automaatselt ilma lehe uuendamiseteta.
23.02-01.03	Optimeerimismeetodite uurimine ning LP-põhise lähenemise katsetamine ning tingimuste defineerimine; Ahne algoritmi kasuks valimine ning selle täiustamisega jätkamine.
02.03-08.03	Liikumisloogika põhimõtete loomine: alguspunkti sihtpunkti liikumine, koju naasmise loogika ning järjestikuste sündmuste sihtpunktide kontroll;

Tabel 3. Gisela Nüüd töö kirjeldus.

Nädal	Kirjeldus
09.03-15.03	Funktsiooni lisamine sündmuste ajavahemike laiendamiseks vastavalt liikumisloogikale; Stadia Maps Routing API kasutamine sõiduaegade arvutamiseks.
16.03-22.03	SoCPlani arvutamise loogika muutmine vastavalt uuele liikumisloogikale; Korduvkasutatavate funktsioonide loomine SoC lisamiseks ja vähendamiseks.
23.03-29.03	Sõiduahelate loomine, kuna ilma sõiduahelateta optimeerimine ei andnud korrektseid tulemusi.
30.03-05.04	Sõiduahelate loogika ühendamine optimeerijaga: ahela kilometraaž teisendatakse aku protsentideks ning optimeerija leiab laadimiseks sobivad ajavahemikud, mille jooksul kasutaja on kodus; Lõputöö kirjutamine.
06.04-12.04	Korrektsete SoC väärtuste lisamine optimeerijasse: vajaminev laetuse tase arvutatakse iga vaba ajavahemiku jaoks, millele lisatakse 20-protsendiline varu; Lõputöö kirjutamine.
13.04-19.04	Intervallide arv arvutatakse sõiduki tegeliku täislaadimise aja põhjal; Lõputöö kirjutamine.
20.04-26.04	CSV malli loomine ärikliendi liikumisandmete importimiseks; Malli allalaadimise funktsionaalsuse lisamine; Üles laetud CSV faili valideerimine ning töötlemine. Lõputöö kirjutamine.
27.04-03.05	Sõiduki väljade automaatne täitmine EVDB andmete põhjal: algselt proovides <i>web scraper</i> -iga, hiljem Claude'lt saadud CSV-ga; Lõputöö kirjutamine.
04.05-10.05	Päevase läbisõidu liuguri arendamine; Lõputöö kirjutamine.
11.05-17.05	Lõputöö kirjutamine.

10 Kokkuvõte

Elektrisõidukite populaarsuse kasvu ja kõrgete elektrihindade taustal puudub olemasolevate laadimislahenduste seas rakendus, mis arvestaks laadimise optimeerimisel kasutaja personaalse liikumisgraafikuga.

Bakalaureusetöö eesmärgiks oli analüüsida Enefit AS-i tellimusel valminud laadimisrakendust, arendada edasi selle puudujääke ning hinnata lahenduse konkurentsivõimet turul. Edasiarenduse suundade määratlemiseks viidi läbi kasutajaküsitlus 53 eraisiku seas ning intervjuud kolme ettevõttega. Arenduses lähtuti kasutajakeskse disaini põhimõtetest, ning läbiti kaks iteratsiooni – esimene keskendus eraisikust kasutajatele, teine äriklientidele.

Küsitluse tulemuste põhjal lisati rakendusse kolm uut funktsionaalsust: sõiduki andmete automaatne täitmine, laadimiskulude kokkuvõte ning päevase läbisõidu liugur. Ärikliendi kasutusjuhu toetamiseks lisati mitme sõiduki liikumisgraafiku importimine CSV-failina. Loodud lahenduste turupotentsiaali hindamiseks testiti täiendatud rakendust kasutajagrupiga.

Kasutajate tagasiside põhjal selgus, et optimeeritud laadimisrakendusel on turul potentsiaal, kuid selle sihtrühm on hetkel suhteliselt kitsas. Kõige enam võiks lahendus pakkuda väärtust kindla igapäevase liikumisgraafikuga pistikhübriidide omanikele. Samuti järeldati, et uute funktsioonide toel on rakendusel selge perspektiiv laieneda äriklientide suunas. Pikemas vaates ja jätkuva edasiarenduse korral on tegemist tööriistaga, mis aitab transpordikulusid tõhusalt optimeerida.

Kasutatud kirjandus

- [1] „Elektriautode statistika Eestis,“ Elektriauto.ee, 2026. [Võrgumaterjal]. Loetud aadressil: <https://elektriauto.ee/statistika/eesti> (Kasutatud 2. mai 2026).
- [2] „Elektrihinna mõjurid,“ Elektrituru käsiraamat, Elering AS. [Võrgumaterjal]. Loetud aadressil: <https://elering.ee/elektrituru-kasiraamat/7-kauplemine-avatud-elektriturul/72-elektrihinna-mojurid> (Kasutatud 2. mai 2026).
- [3] „Difference between Edge Computing and Cloud Computing,“ GeeksforGeeks, 2025. [Võrgumaterjal]. Loetud aadressil: <https://www.geeksforgeeks.org/cloud-computing/difference-between-edge-computing-and-cloud-computing/> (Kasutatud 3. mai 2026).
- [4] „Elering dashboard API documentation,“ Elering AS. [Võrgumaterjal]. Loetud aadressil: <https://dashboard.elering.ee/assets/swagger-ui/index.html#/nps-controller/getPrice> (Kasutatud 2. mai 2026).
- [5] „Visual Studio Code“. [Võrgumaterjal]. Loetud aadressil: <https://code.visualstudio.com/> (Kasutatud 11. mai 2026).
- [6] „Git hub“. [Võrgumaterjal]. Loetud aadressil: <https://github.com/> (Kasutatud 11. mai 2026).
- [7] „Jira“. [Võrgumaterjal]. Loetud aadressil: <https://www.atlassian.com/software/jira> (Kasutatud 11. mai 2026).
- [8] „React“. [Võrgumaterjal]. Loetud aadressil: <https://react.dev/> (Kasutatud 11. mai 2026).
- [9] „React Router“. [Võrgumaterjal]. Loetud aadressil: <https://reactrouter.com/> (Kasutatud 11. mai 2026).
- [10] „Vite“. [Võrgumaterjal]. Loetud aadressil: <https://vite.dev/> (Kasutatud 11. mai 2026).
- [11] „PrimeReact“. [Võrgumaterjal]. Loetud aadressil: <https://primereact.org/installation/> (Kasutatud 11. mai 2026).
- [12] „tailwindcss“. [Võrgumaterjal]. Loetud aadressil: <https://tailwindcss.com/> (Kasutatud 11. mai 2026).
- [13] „Apache ECharts“. [Võrgumaterjal]. Loetud aadressil:

- <https://echarts.apache.org/en/index.html> (Kasutatud 11. mai 2026).
- [14] „FullCalendar“. [Võrgumaterjal]. Loetud aadressil: <https://fullcalendar.io/> (Kasutatud 11. mai 2026).
- [15] „Papa Parse,“ Papa Parse by Matt Holt. [Võrgumaterjal]. Loetud aadressil: <https://www.papaparse.com/> (Kasutatud 11. mai 2026).
- [16] Anthropic. Claude Sonnet 4.6 [Suurkeelemudel]. Loetud aadressil: <https://claude.ai/> (Kasutatud 10. mai 2026).
- [17] „TypeScript“. [Võrgumaterjal]. Loetud aadressil: <https://www.typescriptlang.org/> (Kasutatud 11. mai 2026).
- [18] „CSS: Cascading Style Sheets,“ mozilla.org. [Võrgumaterjal]. Loetud aadressil: <https://developer.mozilla.org/en-US/docs/Web/CSS> (Kasutatud 11. mai 2026).
- [19] „React Testing Library,“ [Võrgumaterjal]. Loetud aadressil: <https://testing-library.com/docs/react-testing-library/intro/> (Kasutatud 11. mai 2026).
- [20] „Stadia Maps“. [Võrgumaterjal]. Loetud aadressil: <https://stadiamaps.com/> (Kasutatud 11. mai 2026).
- [21] „Progressive web apps,“ mozilla.org. [Võrgumaterjal]. Loetud aadressil: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps (Kasutatud 11. mai 2026).
- [22] „Tutorial: Access Azure services from a .NET web app,“ Microsoft, 2026. [Võrgumaterjal]. Loetud aadressil: <https://learn.microsoft.com/en-us/azure/app-service/scenario-secure-app-access-storage?tabs=azure-portal> (Kasutatud 11. mai 2026).
- [23] M. O. Ahmad, J. Markkula, ja M. Oivo, „Kanban in software development: A systematic literature review,“ *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, Santander, Hispaania, 2013, pp. 9-16, doi: 10.1109/SEAA.2013.28.
- [24] „What is User Centered Design (UCD)?“ IxDF - Interaction Design Foundation, 2016. [Võrgumaterjal]. Loetud aadressil: <https://ixdf.org/literature/topics/user-centered-design> (Kasutatud 4. mai 2026).
- [25] „What are some advantages of Component Driven Development ?“ GeeksforGeeks, 2025. [Võrgumaterjal]. Loetud aadressil: <https://www.geeksforgeeks.org/reactjs/what-are-some-advantages-of-component-driven-development/> (Kasutatud 13. mai 2026).
- [26] K. Olsen et al., „Usability Testing and the System Usability Scale Effectiveness Assessment on Different Sensing Devices of Prototype and Live Web System

- Counterpart,“ *Sensors*, 2025. [Võrgumaterjal]. Loetud aadressil: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12845781/> (Kasutatud 13. mai 2026).
- [27] „persoona,“ Justiits- ja Digiministeerium, Kirjeldus. [Võrgumaterjal]. Loetud aadressil: <https://digiriik.eesti.ee/tooriist/persoona> (Kasutatud 7. mai 2025).
- [28] S. Munagavalasa, „Process Flows–As-Is Versus To-Be,“ raamatust *The Salesforce Business Analysis Playbook - Second Edition*, Packt Publishing, 2nd ed., 2026, ch. 4. [Võrgumaterjal]. Loetud aadressil: <https://learning.oreilly.com> (Kasutatud 13. mai 2026).
- [29] E. D. Kostopoulos, G. C. Spyropoulos, and J. K. Kaldellis, „Real-world study for the optimal charging of electric vehicles,“ *Energy Reports*, vol. 6, pp. 418-426, 2020. [Võrgumaterjal]. Loetud aadressil: <https://www.sciencedirect.com/science/article/pii/S2352484719310911> (Kasutatud 2. mai 2026).
- [30] M.-E. Torjus, „Ettevõtjatel avaneb võimalus saada toetust elektriveokite laadimistaristu rajamiseks“, Kliimaministeerium, 2024. [Võrgumaterjal]. Loetud aadressil: <https://www.kliimaministeerium.ee/uudised/ettevotjatel-avaneb-voimalus-saada-toetust-elektriveokite-laadimistaristu-rajamiseks> (Kasutatud 7. mai 2026).
- [31] „A. Le Coqi kogemus: kus ja millisteks veduteks saab Eestis kasutada elektrilisi veokeid?“, Logistikauudised, 2026. [Võrgumaterjal]. Loetud aadressil: <https://www.logistikauudised.ee/uudised/2026/03/13/a-le-coqi-kogemus-kus-ja-millisteks-vedudeks-saab-eestis-kasutada-elektrilisi-veokeid> (Kasutatud 2. aprill 2026).
- [32] „Akud ja laadimine“, Volvo Trucks. [Võrgumaterjal]. Loetud aadressil: <https://www.volvotrucks.ee/et-ee/trucks/electric/faq/faq-batteries-charging.html#accordion-abc447bebc-item-9fa1a653a8> (Kasutatud 2. aprill 2026).
- [33] „Volvo toob turule uues elektriveokid – sõiduulatusega kuni 700 km,“ Volvo Trucks. [Võrgumaterjal]. Loetud aadressil: <https://www.volvotrucks.ee/et-ee/news/press-releases/2026/apr/volvo-launches-new-electric-trucks---with-ranges-up-to-700-km.html> (Kasutatud 7. mai 2026).
- [34] Vabariigi Valitsus. (20. veebruar 2001). Mootorsõidukijuhi sõidu- ja puhkeaja kestuse ning arvestuse kord. [Võrgumaterjal]. Loetud aadressil: <https://www.riigiteataja.ee/akt/26860> (Kasutatud 7. mai 2026).
- [35] „A. Le Coq ja Scania esitlesid Baltikumi esimest elektriveokit,“ A. Le Coq, 2022. [Võrgumaterjal]. Loetud aadressil: <https://www.alecoq.ee/a-le-coq-ja-scania-esitlesid-baltikumi-esimest-elektriveokit/> (Kasutatud 7. mai 2026).

- [36] „Mida oleme teinud süsiniku jalajälje vähendamiseks“, A. Le Coq. [Võrgumaterjal]. Loetud aadressil: <https://www.alecoq.ee/kliimamuutustega-kohandumine-ja-nende-leevendamine/> (Kasutatud 7. mai 2026).
- [37] „Keskkond“, DPD. [Võrgumaterjal]. Loetud aadressil: <https://www.dpd.com/ee/et/jatkusuutlikkus/keskkond/> (Kasutatud 7. mai 2026).
- [38] „Telia vahetab oma autopargi elektriautode vastu“, Telia, 2020. [Võrgumaterjal]. Loetud aadressil: <https://www.telia.ee/uudised/telia-vahetab-oma-autopargi-elektriautode-vastu/> (Kasutatud 7. mai 2026).
- [39] „elektrikell.ee“. [Võrgumaterjal]. Loetud aadressil: <https://www.elektrikell.ee/> (Kasutatud 3. mai 2026).
- [40] „Elektrihind“. [Võrgumaterjal]. Loetud aadressil: <https://elektrihind.ee/borsihind/> (Kasutatud 3. mai 2026).
- [41] „Gridio“, Gridio 2.0 OÜ. [Võrgumaterjal]. Loetud aadressil: <https://www.gridio.io/et> (Kasutatud 29. märts 2026).
- [42] „Milliseid sõidukeid Gridio toetab?“, Gridio 2.0 OÜ. [Võrgumaterjal]. Loetud aadressil: <https://www.gridio.io/et/faq> (Kasutatud 1. mai 2026).
- [43] „Terminal energia“, AS Terminal. [Võrgumaterjal]. Loetud aadressil: <https://terminalenergia.ee/elekter/tarklaadimine/> (Kasutatud 30. aprill 2026).
- [44] „Vool“, MultiCharge OÜ. [Võrgumaterjal]. Loetud aadressil: <https://www.vool.com/et/tooted/vool-app/> (Kasutatud 1. aprill 2026).
- [45] VOOL, vastus kasutaja Facebook-i postitusele, 23. jaanuar 2026. [Võrgumaterjal]. Loetud aadressil: https://www.facebook.com/groups/elektriautodeestis/posts/4332041370361541/?comment_id=4332059350359743 (Kasutatud 1. aprill 2026).
- [46] „2024-11-27: Pay-per-use pricing“, Tesla, 2024. [Võrgumaterjal]. Loetud aadressil: <https://developer.tesla.com/docs/fleet-api/announcements#2024-11-27-pay-per-use-pricing> (Kasutatud 4. mai 2026).
- [47] „Fleet API“, Tesla. [Võrgumaterjal]. Loetud aadressil: <https://developer.tesla.com/> (Kasutatud 4. mai 2026).
- [48] „Enefit Charge“, Enefit. [Võrgumaterjal]. Loetud aadressil: <https://www.enefit.ee/et/abi/elektriautode-laadimine/enefit-charge> (Kasutatud 29. märts 2026).
- [49] „Brands“, Enode Developers. [Võrgumaterjal]. Loetud aadressil: <https://developers.enode.com/brands> (Kasutatud 2. mai 2026).

- [50] „Laadimine ja kliima,“ KIA AUTO AS. [Võrgumaterjal]. Loetud aadressil: https://ownersmanual.kia.com/docview/webhelp/doc/58f234d2-86d4-4a91-929d-f6cf9ad0b39e/topics/chapter1_3_3.html (Kasutatud 6. mai 2026).
- [51] „Tesla App for Wall Connector,“ Tesla. [Võrgumaterjal]. Loetud aadressil: <https://www.tesla.com/support/charging/wall-connector/tesla-app> TESLA (Kasutatud 6. mai 2026).
- [52] „DPD Eesti ja Diotech Groupiga rohelisema tuleviku poole!“ Alexela AS, 2025. [Võrgumaterjal]. Loetud aadressil: <https://www.alexela.ee/et/uudis/elektrilaadijad/dpd-est-i-ja-diotech-groupiga-rohelisema-tuleviku-poole> (Kasutatud 9. mai 2026).
- [53] „Scania Fleet,“ Scania Eesti. [Võrgumaterjal]. Loetud aadressil: <https://www.scania.com/ee/et/home/services/my-scania/scania-fleet.html> (Kasutatud 8. mai 2026).
- [54] „Laadimisteenused,“ Scania Eesti. [Võrgumaterjal]. Loetud aadressil: <https://www.scania.com/ee/et/home/services/digital-services/charging-services.html> (Kasutatud 9. mai 2026).
- [55] „Elektriautode laadimislahendused,“ Diotech. [Võrgumaterjal]. Loetud aadressil: <https://www.diotech.ee/elektriautode-laadimislahendused/> (Kasutatud 8. mai 2026).
- [56] „Hoia oma autoparki laetuna ja liikumises,“ MultiCharge OÜ. [Võrgumaterjal]. Loetud aadressil: <https://www.vool.com/et/lahendused/autopargi-omanikele/> (Kasutatud 9. mai 2026).
- [57] D. Chakraborty, S. Hardman, ja G. Tal, “Why do some consumers not charge their plug-in hybrid vehicles? Evidence from Californian plug-in hybrid owners” *Environmental Research Letters*, 15(8), 2020. [Võrgumaterjal]. Loetud aadressil: <https://iopscience.iop.org/article/10.1088/1748-9326/ab8ca5/pdf> (Kasutatud 2. mai 2026).
- [58] „Terminal ole kodus fiks, aga lae börsiga“ AS Terminal. [Võrgumaterjal]. Loetud aadressil: <https://terminalenergia.ee/elektter/kampaania/> (Kasutatud 2. mai 2026).
- [59] „Electric Vehicle Database,“ EV Database. [Võrgumaterjal]. Loetud aadressil: <https://ev-database.org/> (Kasutatud 12. mai 2026)
- [60] W. Bergaoui, „Transitioning from React.js to React Native,“ DEV Community, 2024. [Võrgumaterjal]. Loetud aadressil: https://dev.to/wafa_bergaoui/transitioning-from-reactjs-to-react-native-4i6b (Kasutatud 11. mai 2026).

- [61] J. Bang-Jensen, G. Gutin ja A. Yeo, „When the greedy algorithm fails,“ *Discrete Optimization*, pp. 121-127. [Võrgumaterjal]. Loetud aadressil: <https://www.sciencedirect.com/science/article/pii/S1572528604000222?via%3Dihub> (Kasutatud 15. mai 2026)
- [62] M. Gubanov, „Targa elektrisõidukite laadimisabilise ChargeMate rakenduse optimeerimismootori ja kasutajaliidese parendamine,“ TalTech Press, Tallinn, 2026.
- [63] Z. Han, K. Grolinger, M. Capretz, „Scheduling Electric Vehicle Charging for Grid Load Balancing“. [Võrgumaterjal]. Loetud aadressil: https://www.eng.uwo.ca/electrical/faculty/grolinger_k/docs/SchedulingEVCharging1.pdf (Kasutatud 15. mai 2026).
- [64] „Q6 SUV e-tron quattro,“ Audi. [Võrgumaterjal]. Loetud aadressil: <https://www.audi.co.uk/en/models/q6-e-tron/q6-e-tron/layers/battery-and-range/> (Kasutatud 15. mai 2026).
- [65] „mobire,“ Mobire Eesti AS. [Võrgumaterjal]. Loetud aadressil: <https://mobire.ee/koik/elektriauto-laadimine-kodus/> (Kasutatud 15. mai 2026).
- [66] Robert C. Martin „Design Principles and Design Patterns,“ (2000). [Võrgumaterjal]. Loetud aadressil: staff.cs.utu.fi/~jounsmmed/doos_06/material/DesignPrinciplesAndPatterns.pdf (Kasutatud 15. mai 2026).
- [67] A. Nilsson, „Performance and feature support of Progressive Web Applications : A performance and available feature comparison between Progressive Web Applications, React Native applications and native iOS applications,“ Dissertation, 2022. [Võrgumaterjal]. Loetud aadressil: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1653815&dswid=9275> (Kasutatud 13. mai 2026).
- [68] Euroopa Komisjon, „Komisjoni delegeeritud määrus (EL) 2025/656, Lisa II punkt 2.1.3“ Euroopa Liidu Teataja, L 2025/656, 2. aprill 2025. [Võrgumaterjal]. Loetud aadressil: https://eur-lex.europa.eu/legal-content/ET/TXT/HTML/?uri=OJ:L_202500656 (Kasutatud 16. mai 2026).
- [69] K. T. Kama, „Nordpooli elektrihinna analüüs ja ennustusmudeli arendamine,“ TalTech Press, Tallinn, 2020.
- [70] I. Miri, A. Fotouhi, ja N. Ewin, Electric vehicle energy consumption modelling and estimation—A case study, *Int J Energy Res*, 2021, vol 45, pp. 501–520, doi: <https://doi.org/10.1002/er.5700>.

- [71] „Elektriauto.ee,“ Computernik OÜ. [Võrgumaterjal]. Loetud aadressil: <https://elektriauto.ee/> (Kasutatud 15. mai 2026).
- [72] „Traffic-Influenced Routing,“ Stadia Maps. [Võrgumaterjal]. Loetud aadressil: <https://stadiamaps.com/products/routing-navigation/traffic-influenced-routing/> (Kasutatud 15. mai 2026).
- [73] M. V. Klymenko ja A. M. Striuk, „Design and implementation of an edgecomputing-based GPS tracking system,“ Journal of Edge Computing, 2023, 2(2), pp. 175-189. [Võrgumaterjal]. Loetud aadressil: <https://acnsci.org/journal/index.php/jec/article/view/634/678> (Kasutatud 15. mai 2026).
- [74] „TensorFlow lite“. [Võrgumaterjal]. Loetud aadressil: Loetud aadressil: <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite> (Kasutatud 15. mai 2026).

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Meie, Piia Carmen Milkov ja Gisela Nüüd

- 1 Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Kasutaja liikumisgraafikul põhineva elektrisõiduki laadimise optimeerimislahenduse analüüs ja edasiarendus Enefit AS näitel“, mille juhendaja on Liisa Jõgiste
 - 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 Oleme teadlikud, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autoritele.
- 3 Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

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 tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Kasutajaküsitlus

Elektriautode optimaalse laadimise planeerimise rakenduse edasiarendus

Meie oleme TalTech'i äriinfotehnoloogia bakalaureuseõppe tudengid Piia Carmen Milkov ja Gisela Nüüd.

Bakalaureusetöö raames soovime edasi arendada elektriautode kodus laadimise rakendust, mis genereerib optimeeritud laadimisplaani, arvestades kasutaja liikumisvajadusi.

Lõputöö jaoks viime läbi **küsitluse elektriauto omanike hulgas, kellel on börsihinna elektripakett.**

Oleksime väga tänulikud, kui aitaksite meil paremini mõista kasutaja soove rakenduse edasi arendamiseks ning leiaksite 10-20 minutit küsitluse täitmiseks.

Vastamine on anonüümne ning kogutud andmeid kasutatakse üldistatud kujul üksnes bakalaureusetöö eesmärgil.

Elektriauto kasutaja

Kas Te omate pistikhübrid või elektriautot? *

☐

 Pistikhübrid

☐

 Elektriauto

☐

 Other: _____

Kus Te oma autot laete? *

Alati kodus

1

☐

2

☐

3

☐

4

☐

5

☐

Alati avalikus kohas (kodust eemal)

Kui Te **ei lae** autot **alati kodus**, siis palun kirjeldage lühidalt, mis on selle põhjuseks.

- ☐ Kodus puudub laadimisvõimalus
- ☐ Töö juures saab soodsamalt laadida
- ☐ Kodune laadimine on liiga aeglane
- ☐ Koduse elektrisüsteemi võimsus on piiratud
- ☐ Other: _____

Mitut sõidukit Te laete kodus? *

- ☐ Ei lae kodus
- ☐ Ainult ühte
- ☐ Rohkem kui ühte

Kui vastasite eelmisele küsimusele jaatavalt, siis mis laadijat Te omate/rendite?

- ☐ Kiirlaadija
- ☐ Tavalaadija
- ☐ V2G (võrku tagasi laadimise võimalusega)

Kui palju hoolite odava hinnaga laadimisest? *

- | | | | | | | |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|--|
| | 1 | 2 | 3 | 4 | 5 | |
| Ei vaata elektri hinda enne laadima panemist | ☐ | ☐ | ☐ | ☐ | ☐ | Planeerin alati auto laadimise kellaaega |

Kui kaua kulub Teil aega auto laadimise planeerimisele? *

- | | | | | | | |
|------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Ei kuluta sellele üldse aega | ☐ | ☐ | ☐ | ☐ | ☐ | Tegemist on igapäevase tegevusega |

Tutvustame Teile meie poolt valmis arendatud rakendust

Palun lugege läbi järgmised lõigud, mis kirjeldavad meie juba loodud rakendust, et oskaksite hiljem küsimustele vastata.

Kasutaja poolt sisestatud liikumisvajaduste ja sõiduki tehniliste andmete alusel genereerib algoritm optimeeritud laadimisgraafiku. Tulemusena kuvatakse Teile soovituslikud laadimisajad ning seeläbi laadimiskulud ning säästetav summa.

Rakenduse avalehel kuvatakse järgmist laadimise aega koos maksumusega.

Samuti nupud, millega saab

- soovi korral rakenduse poolt kalkuleeritud laadimistaset (protsenti) muuta/täpsustada;
- lisada sündmust kalendrisse;
- liikuda lehele, kus kuvatakse planeeritavad laadimissessioonid.

Graafikul kuvatakse kasutaja liikumisvajadusega arvestavaid laadimisaegu ning selle all kasutaja päevaplaani.

Current charging session:

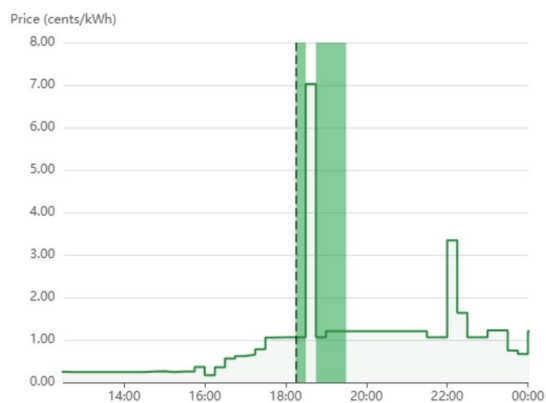
Fri 13 March 18:15 - 18:30
Estimated cost: 0.02 €

Tesla's estimated battery: 60%

Add event

Generate

Price Optimal charging times

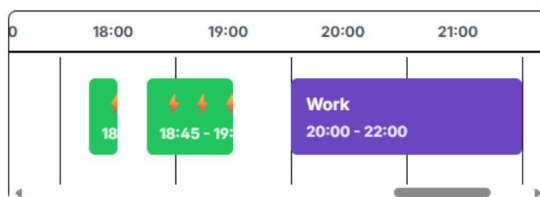


Today: Friday, March 13

Mandatory

Optional

Charge



Andmete sisestamine. Rakenduse esmasel avamisel tuleb kasutajal täita küsimustik saamaks vajalikud andmed laadimise jaoks. Andmeid saab hiljem ka muuta.

Home information

HOME ADDRESS

Pikk tn 26, Tallinn

CHARGER TYPE

V1G

[My addresses](#)

Vehicle information

VEHICLE TYPE

Car

MODEL

Tesla

INITIAL BATTERY STATE OF CHARGE

77 %

BATTERY CAPACITY

kWh

AVERAGE ENERGY CONSUMPTION (WLTP)

208 kWh/100km

CHARGING AND DISCHARGING SPEED

kW

kW

TIME TO FULL CHARGE

hh:mm

[Cancel](#)

[Save](#)

Delete

Rakendus on muudetud kasutajasõbralikumaks lubades kasutajal määrata **enim kasutatavad aadressid**, et neid sündmuse sisestamisel kiiresti valida.

Frequently Used Addresses ×

Home

Tallinna Tehnikaülikool, Tallinn, Harju, Estonia

Jõusaal

Audentese Spordikeskus, Tallinn, Harju, Estonia



+ Add Address

Liikumisgraafiku lisamiseks peab kasutaja käsitsi sisestama kalendrisse oma sündmuste graafiku. Sündmuse lisamisel tuleb kirjeldada

- sündmuse asukoht (kust sündmusele liigutakse ja kus sündmus toimub);
- kestvus
- täpsustada, kas pärast antud sündmust liigutakse koju.

Samuti saab kasutaja määrata sündmuse kui *Optional*, mis tähendab, et antud sündmuse toimumise aeg ei ole kellaajaliselt range. Näiteks võivad sellisteks sündmusteks osutada poodi minek, jõusaalis käik jne.

Sündmuse korduvaks (*Repeats*) määramine on kirjeldatud järgmises lõigus.

Create Event

Title:

Starts **Ends**

Optional Event ☒

Location

Start typing and select an address from the dropdown.

Going home ☒

From:

Tallinna Tehnikaülikool, Tallinn, Harju
TalTech Sports Hall, Tallinn, Harju, Es
TalTech Landmark, Tallinn, Harju, Est
Mets Taltech SOC, Tallinn, Harju, Est

Repeats ☒

Tuesday, Wednesday

☐ Monday
☒ Tuesday
☒ Wednesday
☐ Thursday

End repeat:

< March 2026 >

Mo	Tu	We	Th	Fr	Sa	Su
23	24	25	26	27	28	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Location

Start typing and select an address from the dropdown.

Going home ☒

From:

Tallinna Tehnikaülikool, Tallinn, Harju
TalTech Sports Hall, Tallinn, Harju, Es
TalTech Landmark, Tallinn, Harju, Est
Mets Taltech SOC, Tallinn, Harju, Est

Repeats ☒

Tuesday, Wednesday

☐ Monday
☒ Tuesday
☒ Wednesday
☐ Thursday

End repeat:

< March 2026 >

Mo	Tu	We	Th	Fr	Sa	Su
23	24	25	26	27	28	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Kalendri vaade korduvstundmustega. Kasutajal on võimalus määrata sündmus korduvalt toimuvana. Näiteks kui Teil on esmaspäevast reedeni kella 9'st 5'ni töö, siis ei pea Te seda korduvalt sisestama, vaid saate valida rippmenüüst päevad, millal antud sündmus kordub. Samuti tuleb kalendris määrata kuupäev, millal kordumine lõppeb. Muidugi saab kasutaja pärast sündmuse loomist seda vajadusel hiljem muuta või kustutada.

March 2026

today

<

>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
23	24	25	26	27	28	1
2	3	4	5	6	7	8
Töö	Töö	Töö	Töö	Töö		
9	10	11	12	13	14	15
Töö	Töö	Töö	Töö	Töö		
	Jõusaal	Jõusaal				
16	17	18	19	20	21	22
Töö	Töö	Töö	Töö	Töö		
	Jõusaal	Jõusaal				
23	24	25	26	27	28	29
Töö	Töö	Töö	Töö	Töö		
	Jõusaal	Jõusaal				
30	31	1	2	3	4	5
Töö	Töö	Töö	Töö	Töö		
	Jõusaal					

Add Event

Märkides sündmuse *Optional*, hakkab algoritm otsima võimalusi, kuidas **sündmuse aega muutes oleks võimalik raha säästa**. Teile võidakse teha ettepanek nihutada sündmust sama päeva piires või jätta sündmus täielikult ära. Soovituste vastuvõtmine ei ole kohustuslik ning kehtima jääb Teie poolt algselt määratud aeg.

Alternative optional events

Accept desired changes, otherwise ignore them.

Jõusaal

Original time

21:00 - 23:00

 Total benefit

0.23€

Cancel Event

Jõusaal

Original time

21:00 - 23:00

Suggested time

21:45 - 23:45

 Total benefit

0.01€

Accept suggestion

Kasutaja saab ülevaate oma järgmistest **planeeritavatest laadimissessioonidest** eraldi lehel, kus on kuvatud ka iga sessiooni maksumus.

Optimal charging & discharging schedule

TODAY



01:00 – 01:45
Charging

completed
0.21 €



02:45 – 03:00
Charging

completed
0.10 €



03:45 – 04:00
Charging

in progress
0.11 €



06:45 – 07:15
Charging

upcoming
0.34 €

TOMORROW



23:00 – 00:00
Charging

upcoming
1.02 €

Kas kirjeldatud rakendus lihtsustaks teie igapäeva elu? *

1 2 3 4 5

Rakendus ei tooks
lisaväärtust

☐ ☐ ☐ ☐ ☐

Rakendus kuvab väärtuslikku
informatsiooni

Kas Te kasutaks kirjeldatud rakendust? *

1 2 3 4 5

Ei

☐ ☐ ☐ ☐ ☐

Jah

Meie poolt pakutavad rakenduse edasiarenduse ideed.

Järgnevalt pakume välja täiendavaid funktsionaalsuseid rakenduse edasiarenduseks. Soovime, et vastaksite kas Teie arvates tooks antud funktsionaalsus lisaväärtust hetkesele rakendusele.

Kui pikalt sooviksite näha oma laadimiste ajalugu? *

- ☐ Ei soovi näha ajalugu
- ☐ Tänapäeva
- ☐ Eilse ja tänapäeva
- ☐ Nädala ajalugu
- ☐ Kuu ajalugu
- ☐ Other: _____

Soovituslik on hoida auto laetuse tase vahemikus 20-80%. Kas sooviksite need alumise ja ülemise laetuse piirid ise määrata? *

- ☐ Jah
- ☐ Ei

Kas Te saaksite kasu statistikast laadimise peale kulutatud summa kohta? Näiteks ühel kuul laadimisele kulutatud summa. *

1 2 3 4 5

See info ei oma mulle tähtsust

☐ ☐ ☐ ☐ ☐

Tegemist on väärtusliku informatsiooniga

Kas Te saaksite kasu statistikast optimaalse laadimise peale kulutatud summa kohta? Näiteks ühe kuu laadimise keskmine börsihind vs börsi kuu keskmine elektri hind. *

1 2 3 4 5

See info ei oma mulle tähtsust

☐ ☐ ☐ ☐ ☐

Tegemist on väärtusliku informatsiooniga

Kas eelistate laadida sõidukit järjest või olete nõus laadimist ajaliselt jaotama, et vähendada kulusid? Küsimus tuleneb asjaolust, et elektri börsihind võib 15-minutiliste intervallide jooksul märkimisväärselt kõikuda. Laadimisele pauside jätmine võimaldab seetõttu kasutada soodsama hinnaga perioode ning vähendada kogukulu. *

Näiteks võib jaotatud laadimisgraafik välja näha järgmiselt:

15:00 - 15:15

15:45 - 16:15

16:30 - 16:45

☐ Eelistan laadida järjest

☐ Olen nõus laadimist jaotama

Lisada funktsionaalsus, mille puhul kasutaja ei pea kõiki auto andmeid ise käsitsi ^{*} sisestama, vaid valib automudeli rippmenüüst ja ülejäänud väljad täidetakse automaatselt rakenduse poolt ise. Vajadusel saab kasutaja andmeid muuta või täiendada.

- ☐ See funktsionaalsus muudaks rakenduse oluliselt paremaks
- ☐ See funktsionaalsus ei muudaks rakendust paremaks

Hetkel saab kasutaja soovi korral rakenduse poolt kalkuleeritud laadimistaset ^{*} (protsenti) muuta/täpsustada. See omadus on lisatud põhjusel, et rakendus ei oska täpselt arvestada lisateguritega, mis aku taset mõjutavad. Näiteks on nendeks teguriteks ummikud ning konditsioneeride kasutamine. Kui aga lisaksime algoritmile andmed järgmise päeva ilma prognoosi kohta, oskaks rakendus paremini arvestada Teie sõiduki aku tühjenemise kiirusega ning seeläbi pakkuda täpsemat ennustust järgmise päeva vajamineva laetustaseme kohta.

- ☐ See funktsionaalsus muudaks rakenduse oluliselt paremaks
- ☐ See funktsionaalsus ei muudaks rakendust paremaks

Kiireks visuaalseks ülevaateks järgmise päeva laadimise kohta on lisatud liugur, mille abil saab kasutaja määrata järgmise päeval sõidetava kilometraaži. Antud lahendus välistab vajaduse lisada sündmuseid kalendrisse, kuid võimaldab siiski genereerida järgmiseks päevaks optimaalse laadimise. Vastavalt sisestatud väärtusele, kuvab rakendus:

- laadimisest tuleneva hinna,
- laadimisega saavutatava aku taseme,
- mitu protsenti on vaja akut laadida, et saavutada vajalik tase,
- algoritmi poolt leitud optimaalseim ajavahemik laadimise toimumiseks

TOMORROW'S TRIP

Charging Calculator

⚡ LIVE

Distance

47 km



5 km

200 km

Charging Cost

1.22 €

Required SoC

42%

Needs Charging

+10%

Best Window

03:00–04:00

Battery State

32% → 42%



0%

Target: 42%

100%

SPOT PRICES — TOMORROW



■ Cheapest hours

■ Other hours

☐ See funktsionaalsus muudaks rakendust oluliselt paremaks

☐ See funktsionaalsus ei muudaks rakendust paremaks

Kasutaja saab katsetada kalendris määratud liikumisgraafikuga. Vajutades sündmusele saab kasutaja lisada või eemaldada selle oma päevaplaanist. Lisades sündmuse graafikusse, rakendus uuendab laadimisplaani vastavalt tehtud muudatusele.

*

Which events tomorrow?

Tap an event to include or exclude it from your plan

☒

Work Meeting
09:00 · 12 km one-way

Required

☐

Kids' Practice
15:30 · 8 km one-way

Tap to add

☒

Grocery Run
17:00 · 5 km one-way

Selected

☐

Gym
18:30 · 14 km one-way

Tap to add

☐

Friends Dinner
20:00 · 22 km one-way

Tap to add

 **OPTIMISED CHARGING PLAN**

Total distance

34 km

Charging cost

1.09 €

Best charging window

03:00 – 04:00



Required battery level

37%

- ☐ See funktsionaalsus muudaks rakendust oluliselt paremaks
- ☐ See funktsionaalsus ei muudaks rakendust paremaks

Kui vastasite kahele eelmisele küsimusele jaatavalt, palun täpsustage, kumb uuendus pakub Teie hinnangul suuremat väärtust.

- ☐ Liuguriga saadav ülevaade laadimise kohta, mis eemaldab vajaduse sündmuseid kalendrisse lisada
- ☐ Interaktiivne sisestatud päevaplaani muutmine ja laadimisplaani uuendus

Milliseid uuendusi peate kasulikuks? *

	Pole esmatähtis	Võib lisada	Peab kindlasti lisama
Automaatne väljade täitmine automudeli valikul.	☐	☐	☐
Ilmaprognoosil põhinev täpsem laadimisvajaduse ennustamine.	☐	☐	☐
Liuguriga saadav ülevaade laadimise kohta	☐	☐	☐
Interaktiivne päevaplaani muutmine ja laadimisplaani uuendus	☐	☐	☐

Kas Te kasutaks rakendust koos eelnevalt kirjeldatud lisandustega? *

☐ Ei
 ☐ 1
 ☐ 2
 ☐ 3
 ☐ 4
 ☐ 5
 ☐ Jah

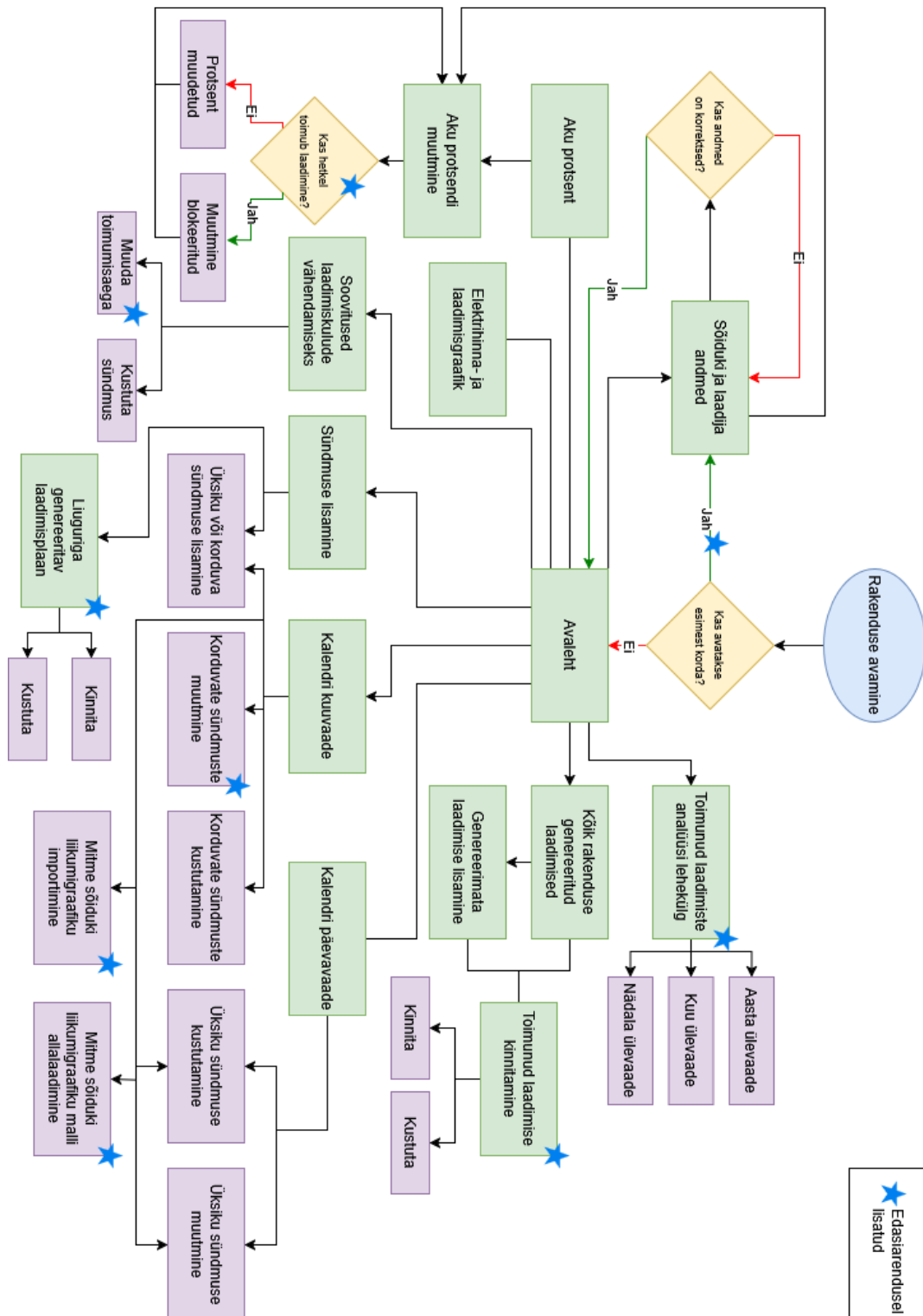
Kui Teil tekkis endal mõtteid, milliseid uuendusi võiks rakendus veel omada, siis palun kirjeldage neid nii täpselt kui oskate.

Your answer

Sooviksime pärast edasiarenduste elluviimist saada Teie tagasisidet. Tagasiside andmine hõlmab kuni kahe nädala rakenduse testimist, millele järgneb küsitluse täitmine. Palume Teil edasiseks suhtluseks sisestada oma e-posti aadress.

Your answer

Lisa 3 – Tegevusdiagramm



Joonis 18. Kasutaja tegevusdiagramm.

Lisa 4 – Progressiivse veebirakenduse manifest

```
{
  "name": "ChargeMate",
  "short_name": "ChargeMate",
  "description": "Smart EV charging planner.",
  "start_url": "/",
  "scope": "/",
  "display": "standalone",
  "background_color": "white",
  "theme_color": "#16a34a",
  "icons": [
    {
      "src": "/pwa-192.png",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "any"
    },
    {
      "src": "/pwa-512.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "any"
    },
    {
      "src": "/pwa-512-maskable.png",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "maskable"
    }
  ]
}
```

Lisa 5 – Piia Carmen Milkovi eneseanalüüs

Minu peamine panus projekti oli laadimiskulude kokkuvõtte lehekülje loomine, toimunud laadimiste kinnitamise funktsionaalsuse arendamine ja lahenduse muutmine progressiivseks veebirakenduseks (PWA).

Laadimiskulude kokkuvõtte kuvamiseks tuli muuta andmete salvestamise loogikat, sest varasem meeskonnatööna valminud süsteem kirjutas sisendandmete muutumisel kogu päeva plaani üle. Lõpliku lahendusena salvestatakse laadimise alguses kogu laadimisaken eraldi *localStorage* võtmesse `past_charging_times` ning aktiivse laadimise ajaks lukustatakse auto ja laadija andmete muutmine. Lisaks lisasin uue sündmuse lisamisel või muutmisel aegade kattuvuse kontrolli, kui sündmus kattub aktiivse laadimisega, jaotatakse laadimine uuesti intervallideks ja kattuv osa eemaldatakse.

Kuna rakendusel puudub võimalus otseselt laadija või autoga ühenduda, lõin süsteemi, mis võimaldab kasutajal plaanitud laadimisi ise kinnitada ja kustutada. Samuti saab kasutaja käsitsi lisada laadimisi, mis on toimunud väljaspool salvestatud plaani. Lahendus tagab, et laadimiskulude kokkuvõtte lehel kuvatakse ainult reaalselt toimunud laadimistel põhinevaid andmeid.

Lisaks kaardistasin era- ja äriklientide vajadusi, koostades küsitlust ning viies läbi intervjuusid, et selgitada välja rakenduse edasiarendamise suunad ning potentsiaal turul. Arenduse algfaasis parandasin ka mitmeid väiksemaid puudusi, mis otseselt rakenduse põhifunktsionaalsust ei mõjutanud, kuid parandasid oluliselt kasutajakogemust. Näiteks lisasin piirangud, et salvestatud aadresse ei saaks topelt sisestada ning sündmusele sõitmise algus- ja sihtkoht ei tohi kattuda, samuti lõin võimaluse muuta korduvaid sündmusi terve seeria kaupa. Tagantjärele analüüsid leian, et oleksime võinud rohkem aega panustada automaattestimise juurutamisele, mis oleks koodimuudatuste järel vähendanud manuaalse testimise mahtu.

Üheks olulisemaks väljakutseks osutus tehniliste ressursside adekvaatne hindamine. Kuigi tooteomanik soovis algselt viia rakenduse üle React Native'ile, selgus minu analüüsist, et kõikide komponentide ümberkirjutamine oleks ebamõistlikult ajamahukas.

Otsus luua alternatiivina progressiivne veebirakendus (PWA) andis mulle väärtusliku õppetunni, kuidas langetada ärikontekstis pragmaatilisi valikuid, mis toovad tulemuse kiiremini ja efektiivsemalt.

Projekti arendades mõistsin, et koodi kirjutamine on vaid üks osa protsessist ning sama tähtsal kohal on meeskonnasisene koostöö ja kasutajate vajaduste mõistmine. Üheks õpetlikumaks hetkeks kujunes liikumisgraafiku funktsionaalsuse arendamine, mis ei saanud algselt oodatud sooja vastuvõttu. Tagasisidet koos mentoritega analüüsisides õppisin olulise tootearenduse põhitõe, isegi kui esmane tagasiside toob esile ainult kitsaskohti, tuleb osata näha lahenduse pikaajalist potentsiaali ning seda argumenteeritult kaitsta.

Lisa 6 – Gisela Nüüd eneseanalüüs

Minu peamiseks vastutusalaks kujunes optimeerimismeetodi arendamine ning selle sidumine rakenduse ülejäänud osadega. Töö algas realistliku liikumisloogika loomisega. Esialgses lahenduses eeldati, et kasutaja naaseb pärast iga sündmust koju, mis enamasti ei vasta tegelikkusele. Töötasin välja loogika, mis tuvastab järjestikuste sündmuste vahelised seosed ning arvutab selle põhjal tegelikud laadimiseks sobivad ajavahemikud. Selleks rakendasin sõiduahelate tuvastamise põhimõtet, mis grupeerib järjestikused sündmused ahelateks ning leiab nende kumulatiivse energiavajaduse. Sõiduaegade arvutamiseks kasutasin Stadia Maps Routing API-t, mis tagastab sisestatud koordinaatide põhjal teekonna läbimise kestuse.

Samuti lisasin loogika, mis tagas vajaduspõhise laadimise. Esialgses lahenduses laeti aku alati 80%-ni, sõltumata sõidu tegelikust energiavajadusest ning aku laetuse tasemest. Uues lahenduses arvutatakse eraldi iga vaba ajavahemiku jaoks vajaminev laetuse tase. Arvutus lähtub sellele ajavahemikule järgneva sõiduahela energiavajadusest ning aku laetuse tasemest ajavahemiku alguses.

Paralleelselt optimeerimistööga lisasin loogika, mis jälgib aku laetustaseme muutumist, arvestades nii laadimist kui ka sõitmist. Laetustaseme muutumisel uuendatakse kasutajaliides automaatselt, ilma et kasutaja peaks rakendust käsitsi värskendama.

Eraklientide kasutusmugavuse parandamiseks lisasin päevase läbisõidu liuguri, mis on suunatud kasutajatele, kelle liikumisgraafik on muutlik. Liuguri loogika realiseerimiseks taaskasutati olemasolevaid optimeerimis- ja laetustaseme arvutamise funktsioone. Äriklientide vajaduste toetamiseks lisasin rakendusse võimaluse importida mitme sõiduki liikumisgraafik CSV failina. Ülesanne hõlmas faili valideerimist, parsimist ning andmete teisendamist.

Projekti teostamisel sain rakendada varasematest kogemustest saadud teadmisi. Sõiduahelate moodustamine tuletas meelde varasemat projekti, kus tuli leida optimaalseid marsruute planeetide vahel reisimiseks koos võimalike vahepeatustega. Varasem kogemus sarnase probleemiga kiirendas probleemile lahenduse leidmist.

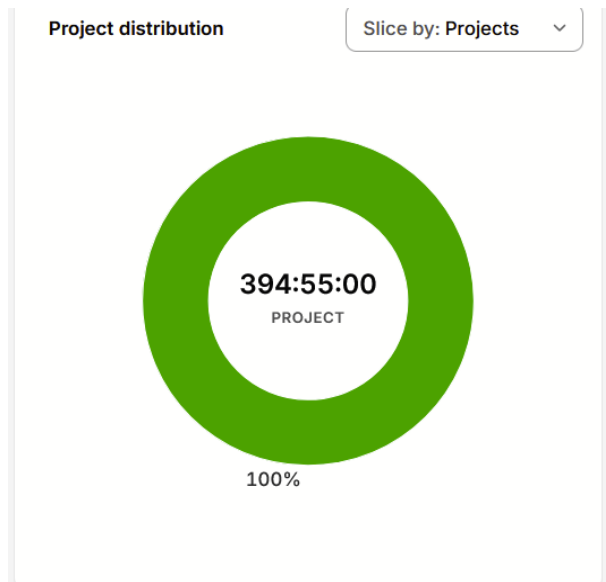
Suuremate ülesannete lahendamisel aitas kaasa oma mõtete kirja panemine ning probleemi jaotamine väiksemateks sammudeks. Esmalt sõnastasin probleemi, leidsin koodist selle juure, kaardistasin mõjutatud osad ning seejärel otsisin võimalikke lahendusi.

Projekti vältel sain palju uusi teadmisi elektrisõidukite ja nende laadimise kohta. Vestlused nii Enefit AS-i kui ka erinevate ettevõtete spetsialistidega elektrisõidukite kasutamise ja laadimisega seotud väljakutsete teemal olid äärmiselt huvitavad. Eriti väärtuslikuks pean võimalust käsitleda teemat nii kliendi kui ka teenusepakkuja vaatenurgast.

Suureks abiks oli varasemalt läbitud aine “Algoritmid ja andmestruktuurid”, mis andis vajalikud teadmised nii aine algoritmi kui ka lineaarprogrammeerimise toimimisest ja nende puudustest.

Lisa 7 – Toggle logid

Piia Carmen Milkov



Gisela Nüüd

