

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Triin Mölder 179884IADB

Ärikliendi iseteeninduskeskkonna uuendamine reisiettevõtte näitel

Bakalaureusetöö

Juhendaja: Toomas Lepikult
Doktorikraad

Tallinn 2022

Autorideklatsioon

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

Autor: Triin Mölder

17.04.2022

Annotatsioon

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 64 leheküljel, 6 peatükki, 25 joonist, 1 tabelit.

Käesoleva töö eesmärk on uuendada Estravel AS ärikliendi iseteeninduskeskkonda, et ärikliendid saaksid iseseisvalt: vaadata informatsiooni oma töötajate reiseenuste kohta; eksportida soovitud raporteid formaadis .csv, .xlsx raamatupidamise jaoks; vaadata ja alla laadida reisiga seotud dokumente (arved, reisikavad, lennupiletid, voucherid ja muud vajalikud reisiga seotud dokumendid).

Selleks, et töö eesmärgid oleksid lahendatud kliendi ja ettevõtte soovidest lähtuvalt on analüüsitud ja testitud olemasoleva keskkonna funktsionaalsust, analüüsitud klientide tagasisidet praegusele keskkonnale, läbiviidud intervjuud ettevõtte kliendihaldurite ja müügidirektoriga. Kogutud ja analüüsitud info põhjal on loodud kasutajavaated.

Enne töö praktilise poolega tegelemist sai läbi viidud ka tehniline analüüs, et välja selgitada parim ja sobivaim arhitektuur ja raamistik, millega antud iseteeninduskeskkond luua.

Praktilises osas on loodud esimene versioon Estraveli ärikasutajate uuest portaalist ja kirjeldatud loodud süsteemi. Uuendatud süsteemi puhul on Frontend poolel kasutatud Axios raamistikku REST API päringute teostamiseks. Loodava süsteemi arendamiseks sai valitud TypeScript ja kasutatavaks arenduse raamistikuks sai valitud Vue.js.

Abstract

The thesis „Renewal of the Self-service Portal for the Business Customer based on the Example of Travel Agency“ is written in Estonian and contains 64 pages of text, 6 chapters, 25 figures and 1 table.

The purpose of this work is to upgrade the self-service portal of Estravel AS business customers. Business customers should be able to: view information about the travel services of their employees; export the desired reports in .csv, .xlsx for their accounting; view and download travel-related documents (invoices, itineraries, airline tickets, vouchers and other necessary travel-related documents) on their own.

In order to reach the goals of the work related to the expectations of the customers and Estravel AS, the functionality of the existing self-service portal has been analyzed and tested. Customer feedback on the current environment was analyzed, and interviews with the company's customer managers and sales director has been made. User interface views have been created based on the information collected and analyzed during the work.

Before starting with the practical side of the work, a technical analysis of an existing self-service portal and analysis of the potential new technologies to be used was carried out to find out the best and most suitable architecture and frameworks for creating a new solution.

As a result of the work also the first version of the new Estravel AS business customer self-service portal has been created and is described. The new portal was created with Vue.js framework using Typescript. The frontend was segregated from the backend services via REST API-s. Few screenshots of the new portal have been added in the annexes of the work.

Lühendite ja mõistete sõnastik

API	<i>Application Program Interface</i> , rakendustarkvara liides
CRUD	<i>Crete, Read, Update, Delete</i> , andmete muutmise meetodid: loo, loe, uuenda ja kustuta
GDPR	<i>General Data Protection Regulation</i> , isikuandmete kaitse üldmäärus
GET	HTTP päring andmete küsimiseks serverist
HTTP	<i>HyperText Transfer Protocol</i> , Hüpertexti edastamise protokoll
ID	<i>Identification</i> , isikutuvastus unikaalse koodi alusel
JavaScript	Netscape Communications Corporation'i poolt väljatöötatud prototüüpipõhine programmeerimiskeel
KM	käibemaks
MVC	<i>Model View Controller</i> , mudel vaate kontrolleri
Offline	käesoleva töö kontekstis mitte veebis tehtavad broneeringud, nt e-mail
Online	käesoleva töö kontekstis veebis tehtavad broneeringud
Responsive	veebilehtede disainimise meetod, nende renderdamiseks erinevates seadmetes vastavalt seadme akna- või ekraani suurusele, et tagada kasutajamugavus
REST	<i>Representational State Transfer</i> , oleku saatmise meetod
Scaffolding	andmebaasiga seotud koodi genereerimine
Sprint	teatud ajavahemik, mille jooksul tuleb konkreetne töö ära teha ja ülevaatamiseks valmis saada
UX	<i>user experience</i> , kasutajamugavus
UI	<i>user interface</i> , kasutajaliides

Sisukord

Sisukord	6
Jooniste loetelu	8
Tabelite loetelu	9
Sissejuhatus.....	10
1. Ülesande püstitus ja lahendusviisid	11
1.1 Taust	11
1.2 Probleemi kirjeldus	12
1.3 Töö eesmärk.....	12
1.4 Metoodika	13
2. Funktsionaalsuse ja kasutajate analüüs.....	15
2.1 Olemasoleva keskkonna funktsionaalsuse analüüs	15
2.2 Praeguse ja loodava süsteemi võrdlus	19
2.3 Loodava süsteemi soovitud funktsionaalsus.....	20
2.3.1 Funktsionaalsed nõuded.....	21
2.3.2 Loodava portaali mittefunktsionaalsed nõuded	28
2.4 Kasutajagrupid ja nende õigused	31
2.5 Kasutajate õiguste kirjeldused	32
3 Tehniline analüüs	33
3.1 Pärandsüsteemi olemus.....	33
3.2 Pärandsüsteemist ülemineku võimalused	34
3.3 Olemasoleva pärandsüsteemi kirjeldus.....	35
3.4 Tööriistade ja raamistike valik uue süsteemi jaoks	36
3.5 Erinevate JavaScripti ja Typescripti raamistike analüüs	36
3.5.1 Javascript ja Typescript	36
3.5.2 React	37
3.5.3 Angular	38
3.5.4 Vue.....	38
3.5.5 Angular või React või Vue, mida kasutada?	38
3.6 Erinevad arhitektuuri mustrid	40
3.6.1 MVC	40
3.6.2 Flux.....	41

3.6.3 Redux.....	42
3.6.4 Vuex.....	43
4 Loodava süsteemi kirjeldus	44
4.1 Valitud tehnoloogiad ja arhitektuur	44
4.2 UX ja UI	47
5 Töö tulemused	48
5.1 Tuleviku arendused.....	48
6 Kokkuvõte.....	50
Kasutatud kirjandus	52
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	54
Lisa 2 – Süsteemi kasutajaõigused	55
Lisa 3 – Kasutajaliidese vaated.....	56
Lisa 4 – Süsteemi andmemudel.....	62
Lisa 5 – Arendatud süsteemi vaated	63

Jooniste loetelu

Joonis 1. Olemasoleva portaali vaade.....	15
Joonis 2. Teenuste aruande vaade.....	17
Joonis 3. Üldkasutatavad funktsioonid kõikidele kasutajagruppidele.....	22
Joonis 4. Ärikliendi töölaua kasutajagruppide õigused	23
Joonis 5. Kasutajagruppide õigused raportite vaates enda andmete kohta	26
Joonis 6. Kasutajagruppide õigused raportite vaates teiste andmete kohta	26
Joonis 7. Raportite funktsionaalsus ja seos kasutajagruppidega	28
Joonis 8. Praeguse rakenduse üldine arhitektuur [8]	35
Joonis 9. React, Angular ja Vue raamistiketes tehtud projektide arv ajas [23]	39
Joonis 10. Model View Controller toimimise skeem [25].....	41
Joonis 11. Flux raamistiku skeem [27]	42
Joonis 12. Reflux töötamise skeem [26].....	43
Joonis 13. REST API mudel [29]	44
Joonis 14. Loodava süsteemi SwaggerHub teenuse mudeli näide	45
Joonis 15. Loodud rakenduse andmete seisu haldamise Vuex põhine arhitektuur [31].	46
Joonis 17. Süsteemi kasutajaõigused	55
Joonis 18. Töölaua kasutaja vaade.....	56
Joonis 19. Kasutajaliidese visand - Arvete raporti vaade	56
Joonis 20. Kasutajaliidese visand - Teenuse raporti arve kuupäeva põhine vaade	57
Joonis 21. Kasutajaliidese visand - Teenuse raporti teenuse kuupäeva põhine vaade ...	58
Joonis 22. Kasutajaliidese visand - Reisijate raporti arve kuupäeva põhine vaade.....	59
Joonis 23. Kasutajaliidese visand - Reisijate raporti teenuse kuupäeva põhine vaade...	60
Joonis 24. Kasutajaliidese visand - Reaside ülevaate raporti vaade.....	61
Joonis 25. Süsteemi andmemudel.....	62
Joonis 26. Loodud kasutajaliides - Arvete raporti vaade.....	63
Joonis 27. Loodud kasutajaliides – Reaside ülevaate raporti vaade.....	64

Tabelite loetelu

Tabel 1. Aspektid, mida uues süsteemis saaks paremini teha	21
--	----

Sissejuhatus

Reisimine ja reisimisega seonduv on viimastel aastatel pakkunud väga keerukaid ja pidevalt muutuvate reeglitega väljakutseid. Keerulisi ja muutuvaid reegleid on riiki sisenemisel, lennuliikluses kui ka hotellide broneerimise osas.

Kiiresti muutuvas reisimaailmas ja majanduse olukorras on vajalik reisibürool oma äriklientidele pakkuda lisandväärtust, et hoida konkureerival turul kliente. Lisaks tavapärasele reiside broneerimisele ja tellitud teenuste muutmise pändlikkusele peavad ärikliendid üha olulisemaks ülevaadet oma töötajate reiside aruandlusest.

Selleks, et ärikliendid saaksid iseseisvalt vajalikku infot kiiresti kätte, on Estravel loonud ärikasutajatele aruandluse portaali "Extranet", mis võimaldab saada ülevaadet Estravelilt tellitud teenustest, arvetest ja teenustega seotud dokumentidest.

Olemasolev ärikasutajate portaal on ajas moraalselt ja füüsiliselt vananenud ja vajab uuendamist. Antud töö teoreetilises osas on analüüsitud olemasoleva portaali puudujääke, funktsionaalsust, tehtud olemasolevate ja tulevaste kasutajalugude analüüs ning võrdlev tehniline analüüs uutest tehnoloogiatest. Praktilises osas on loodud esimene versioon Estraveli ärikasutajate uuest portaalist ja kirjeldatud loodud süsteemi.

1. Ülesande püstitus ja lahendusviisid

Järgnevas peatükis kirjeldatakse töö probleemi tausta ja püstitatakse uurimisülesanne ja esitatakse võimalikud lahendusviisid.

1.1 Taust

Käesolevas töös loodav uuendatud ärikliendi iseteeninduskeskkonna versioon on loodud reisibüroo Estravel AS äriklientide vajadustest lähtuvalt.

Estravel AS on Eesti ja Baltimaade suurim reisibüroo, mis on turismiturul tegutsenud alates 1988. aastast. Ettevõtte tugevusteks on hea klienditeenindus, ettevõtte kõrge usaldusväärsus ja eeskujulikud tehnoloogilised lahendused [1].

Estravel AS pakub parimaid reisilahendusi nii era- kui äriklientidele. Estravel müüb ja vahendab lennupileteid, reisipakette ja hotellimajutust, aga ka laeva- ja rongipileteid, reisikindlustust ning bussi- ja autorendi teenust. Lisaks Eestist väljaminevatele reisidele tegeleb Estravel sissetuleva turismi ja sisereiside vahendamise, pakkudes nii kohalikele inimestele kui väliskülastajatele võimalusi ilusa Eestimaa avastamiseks ja siin puhkamiseks. Lisaks tegeleb ka konverentside, seminaride ja ürituste korraldamisega [1].

Estravel AS teenindab nii Eesti, baltikumi kui ka teisi Euroopa Liidu riikide ärikliente. Estravel on võitnud riigihankeid ka väljaspool Eestit. Seega on oluline, et nii Eesti kui teiste riikide äriklientide vajadused ja nõuded oleksid loodava ärikliendi iseteeninduskeskkonnaga kaetud.

1.2 Probleemi kirjeldus

Estravelil on olemas oma äriklientidele mõeldud iseteeninduskeskkond juba varasemast ajast nimega "Extranet". Praegune keskkond on loodud 2000-ndate aastate esimeses pooles. Täna kasutusel olev keskkond ei vasta kõikide klientide vajadustele ja seega hetkel antud keskkonda kliendid aktiivselt ei kasuta. Klientide soovitud raportid saadetakse hetkel paralleelselt klientidele e-maili peale eelnevalt kokkulepitud ajal ja sagedusega.

Seega on vajalik klientide tänased vajadused uuesti üle vaadata ja ajakohastada. Samuti vajab ajakohastamist tehnoloogiline baas. Olemasoleval portaalil on kasutajaliides jäigalt sidestatud backend teenustega ja ei ole uuenduste tegemiseks väga paindlik. Olemasolev keskkond on loodud PHP raamistikul ja kasutab MVC mudelil.

Arvestades praegust viiruse pandeemiast ja üleüldisest julgeolekust tekkinud olukorras, soovib Estravel AS oma klientidele veelgi rohkem abiks olla ja tuge pakkuda. Täna väga kiiresti muutuva lennuliikluse ja majandus olukorras soovivad ettevõtted selget ja kiiret ülevaadet: kus nende töötajad asuvad; kui kaua kestab reis; millises hotellis peatatakse; mis on reisi maksumus ja kuidas nad saavad kiiresti vajadusel sihtkohast tagasi reisida. Lisaks vajab äriklientide raamatupidamine oma ettevõtte siseste reisikulude katmiseks vajaliku infot töötaja koodide ja projektikoodide vaates. Kuna Estravelil on kõik vajalikud andmed oma kliendibaasis ja müügisüsteemis olemas, siis saab ettevõtte pakkuda klientidele erinevate raportite ja kasutajaliideste vaadete näol lisaväärtust lisaks reiside broneerimisele ja nende muutmisele.

1.3 Töö eesmärk

Probleemi kirjeldusest lähtuvalt on töö eesmärk ajakohastada Estravel AS äriklientide iseteeninduskeskkond, kus ärikliendid saavad iseseisvalt:

- vaadata informatsiooni oma töötajate reisi teenuste kohta;
- eksportida soovitud raporteid formaadis .csv, .xlsx raamatupidamise jaoks;
- vaadata ja alla laadida reisiga seotud dokumente (arved, reisikavad, lennupiletid, voucherid ja muud vajalikud reisiga seotud dokumendid).

Oluline on, et kliendid saaksid omale sobival ajahetkel kiirelt soovitud info ise kätte, ilma, et klient peaks selleks pöörduma Estraveli poole telefoni või e-maili teel. Kuna hetkel saadetakse raportid klientidele e-maili teel, siis iseteenindust kasutades jääb ära mahukate e-mailide saatmine. Tundliku info saatmine e-maili teel on alati riskantne andmeturvalisuse vaatevinklist. Iseteenindusportaali kaudu saab andmete turvalisust ja juurdepääsu oluliselt paremini kontrollida ja kehtestada ning vastavate õigustega kliendi poolsed kontaktid saavad sealt kätte ainult neile vajaliku info, ning ei toimu andmete asjatut edasi tagasi saatmist. Lisaks raportite väljavõtmisele on iseteeninduskeskkonnas oluline ka kätte saada kõik vajalikud dokumendid (arved, reisikavad, lennupiletid, hotelli voucherid, autorendi voucherid, kindlustuspoliisid ja muud reisiks vajalikud dokumendid).

Ärikliendi iseteeninduskeskkonda on plaanitud veel mitmeid lisafunktsionaalsusi, kuid need jäävad lõputöö skoopist välja ja neid seepärast eraldi käesolevas töös põhjalikult ei kajastata.

1.4 Metoodika

Käesolevas peatükis on kirjeldatud töö tegemisega seotud põhimõtteid, valitud meetodeid analüüsiks ja töö koostamiseks. Töökäes vajalikud metoodikad on valitud vastavalt reaalsele vajadusele sõltuvalt töö osast.

Töö esimeses osas on keskendutud olemasoleva iseteenindusportaali Extranet puuduste kaardistamisele ja uue loodava funktsionaalsuse defineerimisele. Selleks sai kasutatud järgnevat meetodeid.

- Olemasoleva portaali testimine, olemasoleva funktsionaalsuse ülevaatamine ning kirjeldamine vaatluse meetodil;
- Klientidega tehtud intervjuude analüüs;
- Intervjuud Estraveli kliendihaldurite ja müügidirektoriga.

Töö teises osas on keskendunud olemasoleva portaali tehnilisele analüüsile ja uue loodava portaali võimalike tehniliste raamistike ja meetodite analüüsile. Selleks sai kasutatud järgnevaid meetodeid.

- Intervjuu olemasoleva portaali arendajatega;
- Olemasoleva portaali tarkvara koodi ülevaatus;
- Uue portaali loomiseks võimalike raamistike ja arhitektuuri analüüs kirjanduse baasil.

Töö viimases osas on keskendunud uue portaali funktsionaalsuse ja arhitektuuri kirjeldamisele ja reaalse töötava tarkvara loomisele.

2. Funktsionaalsuse ja kasutajate analüüs

Järgnevas peatükis esitatakse olemasoleva keskkonna analüüs ning loodava süsteemi funktsionaalsed ja mittefunktsionaalsed nõuded.

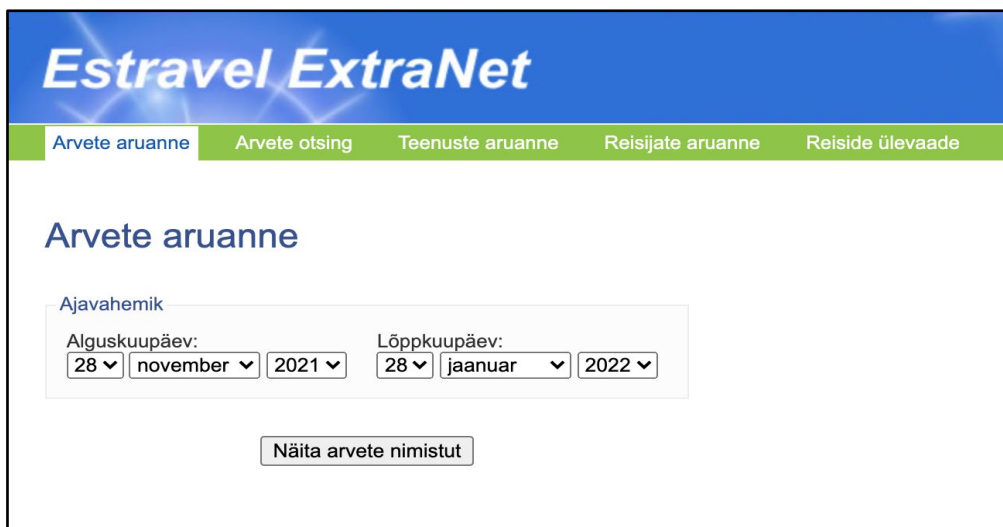
2.1 Olemasoleva keskkonna funktsionaalsuse analüüs

Süsteem on mõeldud nii äriklientide raamatupidamisele, kui ka reisijatele ja reisi tellijatele. Raamatupidajad saavad otsida erinevate kriteeriumite järgi arveid. Kliendipoolne reiside tellija kui ka reisija ise saavad portaalist otsida nendega seotud reisiteenuseid kuupäeva järgi.

Süsteemi praegune funktsionaalsus võimaldab kasutajal sisse logida kasutades kasutajanime ja parooli. Iga äriklient näeb vaid oma ettevõtte reise. Kasutajaõiguste puhul rollide eristamist hetkel ei ole. Seega kõik ärikliendi töötajad näevad kõiki antud ettevõtte reise ja dokumente, mis pole samas alati kooskõlas andmekaitse reeglitega.

Praeguses ärikliendi keskkonnas on võimalik teha viite tüüpi otsingut:

- Arvete aruanne;
- Arvete otsing;
- Teenuste aruanne;
- Reisijate aruanne;
- Reaside ülevaade.



Etravel ExtraNet

Arvete aruanne Arvete otsing Teenuste aruanne Reisijate aruanne Reaside ülevaade

Arvete aruanne

Ajavahemik

Alguskuupäev: 28 november 2021 Lõppkuupäev: 28 jaanuar 2022

Näita arvete nimistut

Joonis 1. Olemasoleva portaali vaade

Neli esimest raportit on arve ja arve kuupäeva põhised. Viimane raport on teenuse ja teenuse toimumise aja põhine.

Järgnevalt on kirjeldatud iga aruande mooduli funktsionaalsus:

Arvete aruanne

Otsingu parameetrid:

- arve kuupäev (ajavahemik).

Raporti väljastusandmed:

- kuvatakse arve number;
- arve väljastamise kuupäev;
- maksetähtaeg;
- arve summa koos KM-ga.

Muu funktsionaalsus:

- raporti eksport .xls formaadis oleva failina;
- arve failide avamine;
- võimalik on sorteerida arve numbri, arve kuupäeva, maksetähtaja ja arve summa järgi.

Arvete otsing

Otsingu parameetrid:

- reisisteenuse number.

Raporti väljastusandmed:

- kuvatakse arve number;
- arve väljastamise kuupäev;
- maksetähtaeg;
- arve summa koos KM-ga.

Muu funktsionaalsus:

- raporti eksport .xls formaadis oleva failina;
- arve failide avamine;

- võimalik sorteerida arve numbri, arve kuupäeva, maksetähtaja ja arve summa järgi.

Teenuste aruanne

Otsingu parameetrid:

- teenuse kuupäev või arve kuupäev;
- teenused;
- müügikanal;
- teenusepakkuja;
- teenuse sihtkoht (riik, linn).

Raporti väljastusandmed:

Võimalik genereerida nii kokkuvõtlik kui ka detailsem raport.

- kokkuvõtlikus raportis kuvatakse teenuste maksumus teenuse liigiti.

Vaata lähemalt

Lend	9 511.74
Auto/Buss	5 255.48
Sihtkoha teenused	0.00
Kruis	0.00
Laev	610.60
Hotell	10 927.70
Kindlustus	0.00
Muud teenused	37.00
Pakett	0.00
Rong	123.66
Reisikorraldaja pakett	0.00
Viisa	0.00
Kokku	26 466.18

Joonis 2. Teenuste aruande vaade

- detailsamal (Vaata lähemalt) raportil kuvatakse:
 - arve number;
 - teenuse kuupäev;
 - teenuse asukoha alguspunkt;
 - teenuse asukoha sihtpunkt;
 - reisija nimi, teenus koos kirjeldusega;
 - kogus;
 - summat koos KM-ga;

- teenusepakkuja nimi;
- reisikonsultandi nimi;
- reisitelliija nimi;
- müügikanal.

Muu funktsionaalsus:

- raporteid võimalik eksportida .xls formaadis oleva failina;
- arve failide avamine.

Reisijate aruanne

Otsingu parameetrid:

- teenuse kuupäev või arve kuupäev;
- teenused, müügikanal;
- reisija ees- ja perekonnanimi.

Raporti väljastusandmed:

- kuvatakse kokkuvõtvalt iga reisija reisiteenuste maksumus antud perioodil.

Muu funktsionaalsus:

- raporteid võimalik eksportida .xls formaadis oleva failina.

Reiside ülevaade

Otsingu parameetrid:

- teenuse kuupäev;
- reisija ees- ja perekonnanimi.

Raporti väljastusandmed:

- kuvatakse reisi algus ja lõpp kuu kuupäeva;
- reisija nime;
- reisi sihtkohta;
- teenuse nimetust koos teenuse pakkuja ja teenuse kirjeldusega;
- dokumendi pildi avamine.

Muu funktsionaalsus:

- raporteid võimalik eksportida .xls formaadis oleva failina.

2.2 Praeguse ja loodava süsteemi võrdlus

Praegust süsteemi analüüsisid ja olles tutvunud klientide tänaste vajadustega toon välja positiivsed ja negatiivsed aspektid praeguses süsteemis.

Iseteeninduskeskkonna raportite moodul peab kliendil võimaldama iseseisvalt kätte saada vajalikku infot. Moodulis peab olema võimalik välja võtta ja alla laadida aruandeid, arveid, reisikavasid, reisimiseks vajalikke dokumente (lennupileteid, hotelli vouchereid, autorendi vouchereid jms).

Selleks, et uus keskkond vastaks kõikidele nõuetele ja oleks tulevikku orienteeritud, analüüsisin esmalt praeguse keskkonna funktsionaalsust, tutvusin kogutud klientide tagasisidega ning viisin läbi intervjuusid ettevõtte kliendihaldurite ja müügidirektoriga. Võttes arvesse kogutud infot esitan raporti mooduli osade nõuded: funktsionaalsed ja mittefunktsionaalsed nõuded.

Positiivsed aspektid

- võimalik kliendipõhine andmete kuvamine;
- visuaalselt lihtne;
- raportid genereeruvad kiiresti;
- raport faili ekspordi võimalus;
- arve piltide avamise võimalus;
- sorteerimise võimalus;
- peale raporti genereerimist on näha ka valitud parameetrid.

Tabel 1. Aspektid, mida uues süsteemis saaks paremini teha

Negatiivsed aspektid	Uues süsteemis muutmiseks
Otsingu parameetri puudulikkus.	Lisada valikud: <i>Project code, Employee code, Account finance number, Customer reference, Passenger reference</i>
Kuupäevade valimine rippmenüüst ei ole mugav	Kuupäeva valikute lisamisel kasutada kalendrit

Raporteid saab eksportida .xlsx faili formaadist	Lisada täiendavalt juurde .pdf, .csv formaadid
Dokumentide ikoonid ebaselged	Kasutada ettevõttes kasutatavaid kujundusikoone
Dokumentide avamise nupud ei tööta osadel juhtudel	Testida, et dokumentide avamise nupud töötaksid alati
Osade raportite puhul jääb ebaselgeks raportis kuvatav info	Lisada iga raporti juurde info nupud, mille all selgitused, nt raporti kuvatud info on arve kuupäevade alusel
Kõiki teenusepakkujaid näidata ei tohi	Selekteerida teenusepakkujad
Kasutajaõiguste puhul on hetkel üks tase	Välja selgitada kasutajaõiguste vajalikkus ja tuua sisse mitu kasutajaõiguste taset
Arvete aruanne ja arvete otsingu raportid kuvavad ühte infot	Via antud raportid kokku üheks raportiks

Ettevõtte kliendihalduritega suheldes selgusid järgmised aspektid, millele uue süsteemi arendamisel tähelepanu pöörata:

- dokumentide juures *download* valik peaks võimaldama dokumente alla laadida kõik korraga ja ka üksikhaaval;
- vajalik kuvada kõiki dokumente, hetkel saab ainult lennupiletid avada;
- arvete raportis peaks saama valida, et näita ainult maksmata arveid;
- raportite avamisel anda vaikimisi ette viimase kuu arved või viimase kuu teenused;
- reise ülevaate raportis värvida erinevad reisid erinevat värvi, eristada saaks tellimusnumbri alusel.

2.3 Loodava süsteemi soovitud funktsionaalsus

Selleks, et loodav süsteem oleks kasutajale loogiline, mugav kasutada ja täidaks oma eesmärgi toon järgnevalt välja vajalikud funktsionaalsed ja mittefunktsionaalsed nõuded. Nõuete kirjapanekul on arvestatud juba olemasoleva süsteemi analüüsi ning ettevõtte kliendihaldurite ja klientide tagasisidet praegusele süsteemile.

2.3.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded (*Functional Requirements*) täpsustavad süsteemi funktsionaalsuse, vastates küsimusele, mida peab infosüsteem tegema ehk väljakirjutis süsteemi teenustest, mida süsteem pakub, kuidas süsteem reageerib antud sisendile ja kuidas käitub teatud situatsioonides, näiteks: ärinõuded, administratiivsed funktsioonid, autentimine, välised liidesed, aruandlusnõuded, arhiivinõuded ning juriidilised või regulatiivsed nõuded [2].

Funktsionaalsete nõuete kogumiseks on olemas erinevaid meetodikaid. Tutvudes erinevate materjalidega võib öelda, et enam kasutatavamad on intervjuu läbiviimine, kasutajabod, prototüüpimine, koosolek-vaatlus, ajurünnak, küsimustik. Funktsionaalsed nõuded peavad olema sõnastatud võimalikult lihtsalt, et need kõigile projekti osapooltele üheselt mõistetav oleksid.

Järgnevalt toon välja funktsionaalsed nõuded, mis on vajalikuid kõikide raportite puhul:

Tähis: F1

Nimetus: Sisselogimine.

Kirjeldus: Kasutaja peab saama sisse logida Smart-ID, Mobiili-ID, e-posti aadressi ja parooli (eelkõige kasutajad väljapool Eestit) kaudu. Kasutaja näeb ainult oma kasutajarollile ettenähtud infot ja nähtavad on ainult need menüüid, mis antud kasutajaõigustele lubatud on.

Kasutajad: Kõik kasutajagrupid.

Tähis: F2

Nimetus: Väljalogimine.

Kirjeldus: Kasutaja peab saama end portaalist välja logida.

Kasutajad: Kõik kasutajagrupid.

Tähis: F3

Nimetus: Keele valik.

Kirjeldus: Keelevalikuna on võimalus valida eesti või inglise keel.

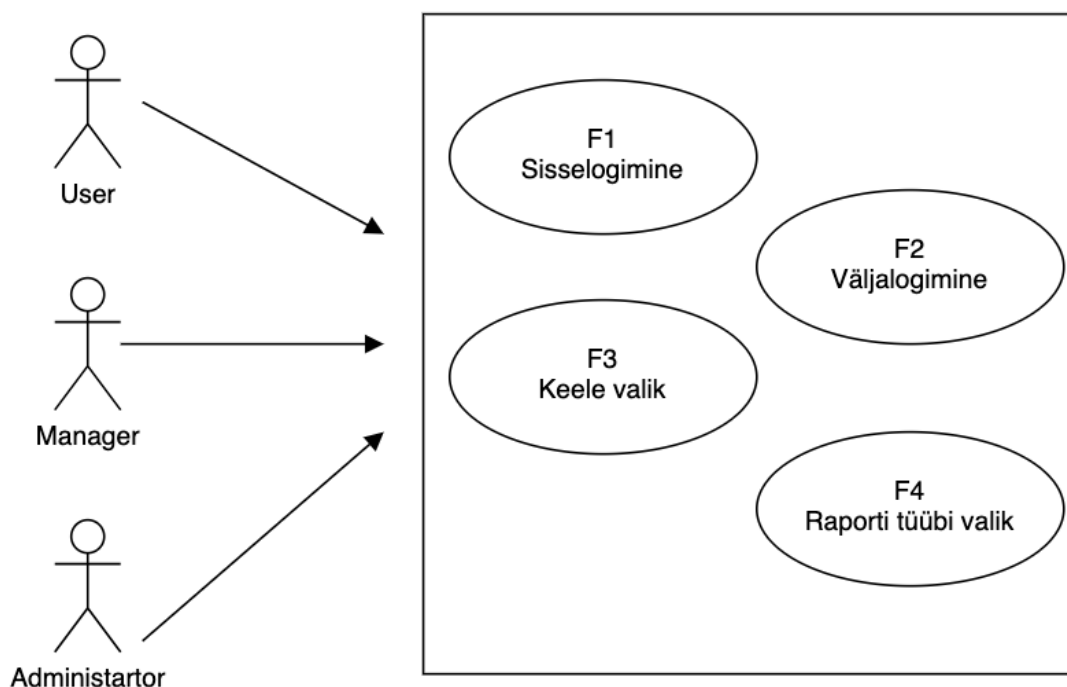
Kasutajad: Kõik kasutajagrupid.

Tähis: F4

Nimetus: Raporti tüübi valik.

Kirjeldus: Raportite vaadetes peab kasutaja saama valida, kas päring tehakse arve kuupäeva või reisikuupäeva järgi.

Kasutajad: Kõik kasutajagrupid.



Joonis 3. Üldkasutatavad funktsioonid kõikidele kasutajagruppidele

Tähis: F5

Nimetus: Ärikliendi töölaud.

Kirjeldus: Peale sisse logimist kuvatakse töölauda vaadet. Töölual oleval info on jooksva kuu kohta ja aluseks teenuse kuupäev.

Töölual kuvatakse järgmist infot: avatud tellimuste arv, reisijate arv, erinevate sihtkohade arv (rüüripõhine), teenuste maksumus, jooksva nädala reisijate info (reisija nimi ja reisi periood), teenuse gruppide jaotus.

Kasutajad: Kliendi poolelt: Administraator, Reisi tellija, Raamatupidaja

Estraveli poolelt: Administraator, Kliendihaldur.

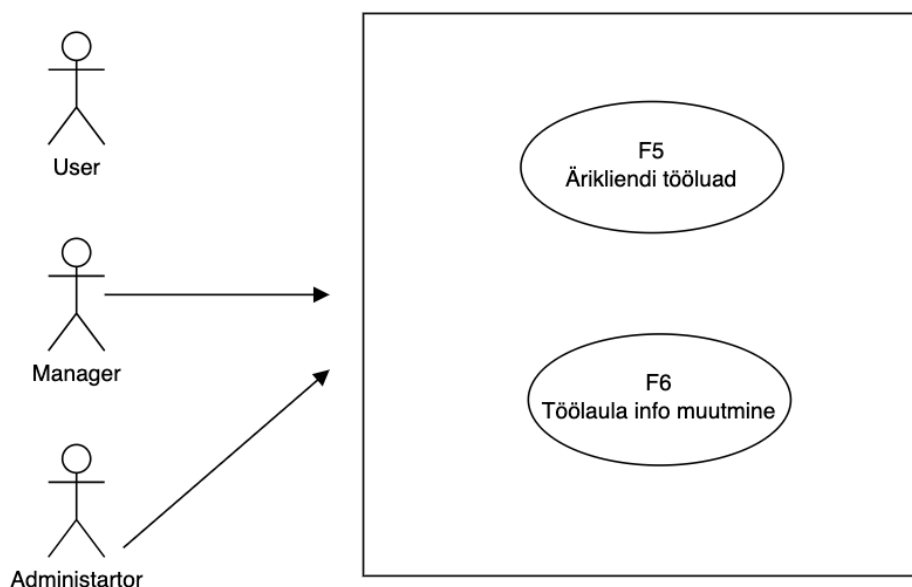
Tähis: F6

Nimetus: Töölaua info muutmine.

Kirjeldus: Kasutaja saab ise valida, et kas töölaua kuvatav info on arve kuupäeva või teenuse kuupäeva alusel. Töölaua olevat info ajaperioodi saab kasutaja ise muuta.

Kasutajad: Kliendi poolelt: Administraator, Reisi tellija, Raamatupidaja.

Estraveli poolelt: Administraator, Kliendihaldur.



Joonis 4. Äriklendi töölaua kasutajagruppide õigused

Tähis: F7

Nimetus: Arve raporti parameetrite valik.

Kirjeldus: Kasutaja peab saama ise arvete perioodi valida ja otsida arveid arve numbriga, reisija nime, reisija *reference* numbriga, kliendi *reference* numbriga või tellija nime järgi.

Arvete raportis kuvatakse järgmist infot: arve number, arve kuupäev, makse kuupäev, arve summa. Vaikimisi näidatakse kõiki arveid, kuid kasutaja saab määrata, et näidataks ainult maksmata arveid.

Sorteerimine: arve numbriga, arve kuupäeva, makse kuupäeva järgi.

Kasutajad: Kõik kasutajagrupid.

Tähis: F8

Nimetus: Teenuste raporti (arve kuupäeva alusel) parameetrite valik.

Kirjeldus: Kasutaja peab saama ise arveks tehtud teenuste perioodi valida ja otsida lisaks teenuse sihtkoha (riigi või linna), teenusepakkuja, teenuse grupi või müügikanali (online või offline) alusel.

Kasutajal peab nägema kahte vaadet: ülevaatlilik ja detailne vaade. Ülevaatlilikus raportis kuvatakse järgmist infot: teenuse nimetust ja teenuse summat (koos km-ga). Detailsel raportis kuvatakse järgmist infot: teenuse alguse kuupäev, teenuse alguspunkt, teenuse sihtpunkt, reisija nimi, teenuse nimetus koos kirjeldusega, kogus, teenuse summa, teenusepakkuja, reisikonsultant, tellija nimi ja müügikanal. Sorteerimine: teenuse kuupäeva ja reisija nime järgi.

Kasutajad: Kõik kasutajagrupid.

Tähis: F9

Nimetus: Teenuste raporti (teenuse kuupäeva alusel) parameetrite valik.

Kirjeldus: Kasutaja peab saama ise teenuste perioodi valida ja otsida lisaks teenuse sihtkoha (riigi või linna), teenusepakkuja, teenuse grupi või müügikanali (online või offline) alusel.

Kasutajal peab nägema kahte vaadet: ülevaatlilik ja detailne vaade. Ülevaatlilikus raportis kuvatakse järgmist infot: teenuse nimetust ja teenuse summat (koos km-ga). Detailsel raportis kuvatakse järgmist infot: arve number, teenuse alguse kuupäev, teenuse alguspunkt, teenuse sihtpunkt, reisija nimi, teenuse nimetus koos kirjeldusega, kogus, teenuse summa, teenusepakkuja, reisikonsultant, tellija nimi ja müügikanal. Sorteerimine: teenuse kuupäeva ja reisija nime järgi.

Kasutajad: Kõik kasutajagrupid.

Tähis: F10

Nimetus: Reisijate raporti (arve kuupäeva alusel) parameetrite valik.

Kirjeldus: Kasutaja peab saama ise arvete perioodi valida ja otsida lisaks reisija nime, reisija numbril, tellija nime, teenuse grupi või müügikanali (online või offline) alusel.

Kasutajal peab nägema kahte vaadet: ülevaatlilik ja detailne vaade. Ülevaatlilikus raportis kuvatakse järgmist infot: reisija nime ja tellimuste summat (koos km-ga). Detailisel raportis kuvatakse järgmist infot: reisija nime, reisija numbrit, arve numbrit, arve summat ja tellija nime. Sorteerimine reisija nime järgi.

Kasutajad: Kõik kasutajagrupid.

Tähis: F11

Nimetus: Reisijate raporti (teenuse kuupäeva alusel) parameetrite valik.

Kirjeldus: Kasutaja peab saama ise arvete perioodi valida ja otsida lisaks reisija nime, reisija numbrit, tellija nime, teenuse grupi või müügikanali (online või offline) alusel.

Kasutajal peab nägema kahte vaadet: ülevaatlilik ja detailne vaade. Ülevaatlilikus raportis kuvatakse järgmist infot: reisija nime ja tellimuste summat (koos km-ga). Detailisel raportis kuvatakse järgmist infot: reisija nime, reisija numbrit, arve numbrit, arve summat ja tellija nime.

Sorteerimine reisija nime järgi.

Kasutajad: Kõik kasutajagrupid.

Tähis: F12

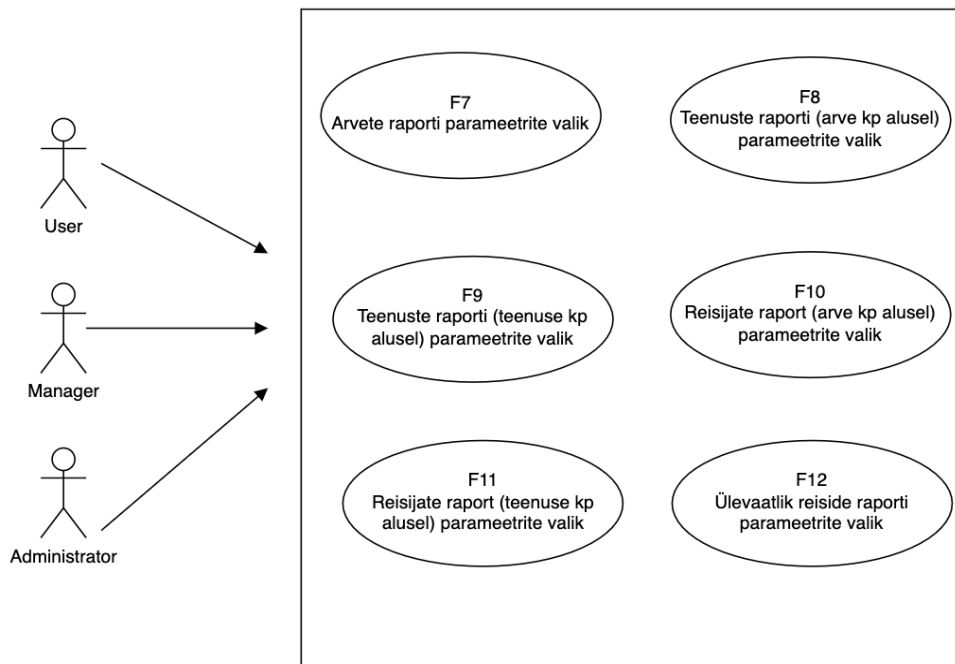
Nimetus: Ülevaatliliku reiside raporti parameetrite valik.

Kirjeldus: Kasutaja peab saama ise teenuste perioodi valida ja otsida lisaks reisija nime, reisija numbrit, tellija nime, sihtriigi, teenuse grupi või pileti numbrit alusel.

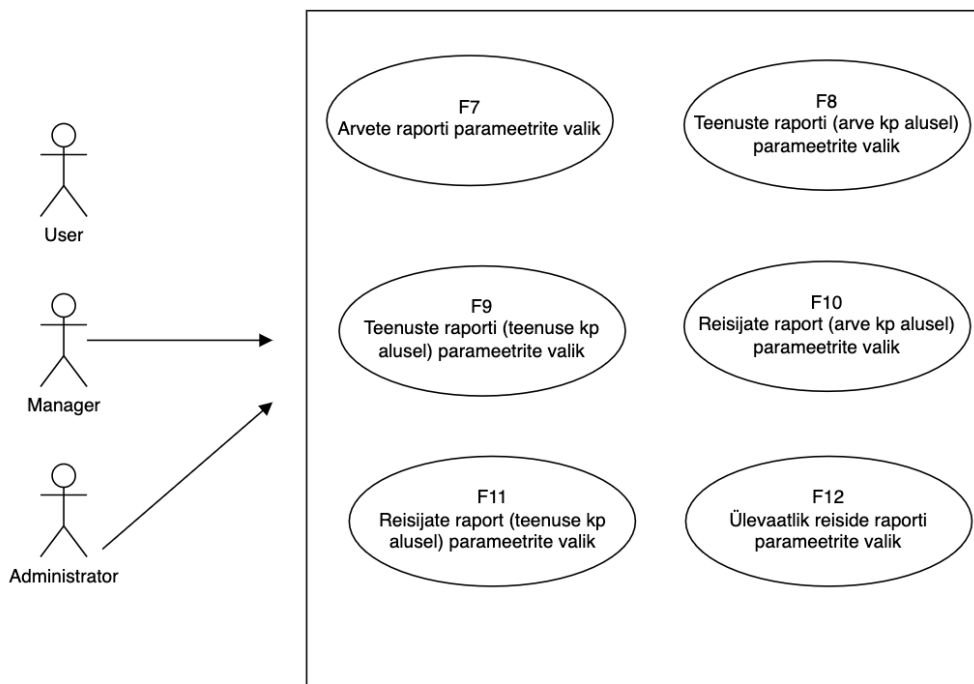
Raportis kuvatakse järgmist infot: teenuse kuupäevad, reisija nimi, sihtkoht, teenuse nimetus, dokumendid.

Sorteerimine: teenuse kuupäevade ja reisija nime järgi.

Kasutajad: Kõik kasutajagrupid.



Joonis 5. Kasutajagruppide õigused raportite vaates enda andmete kohta



Joonis 6. Kasutajagruppide õigused raportite vaates teiste andmete kohta

Tähis: F13

Nimetus: Dokumentide avamine ja allalaadimine.

Kirjeldus: Dokumendi (arved, voucherid, lennupiletid jms) avamine lingist koos allalaadimise võimalusega.

Kasutajad: Kõik kasutajagrupid.

Tähis: F14

Nimetus: Raportites lisaparaameetrite kuvamine (kõikides raportites).

Kirjeldus: Raporti parameetritena lisatakse projekti kood, eelarve kood ja finants number, kui vastava kliendi juures on see eelnevalt seadistatud.

Kasutajad: Kõik kasutajagrupid.

Tähis: F15

Nimetus: Raportite eksport.

Kirjeldus: Päringuga tehtud raportit peab kasutajal olema võimalik eksportida .xlsx, .csv ja .pdf formaadis.

Kasutajad: Kliendi poolelt: Administraator, Reisi tellija, Raamatupidaja.

Estraveli poolelt: Administraator, Kliendihaldur.

Tähis: F16

Nimetus: Raportite kirjelduse info nupud.

Kirjeldus: Info nuppude alt peab olema võimalik kasutajal lugeda selgitavat infot. Kuvatakse, kui kasutaja läheb hiirega nupu peale.

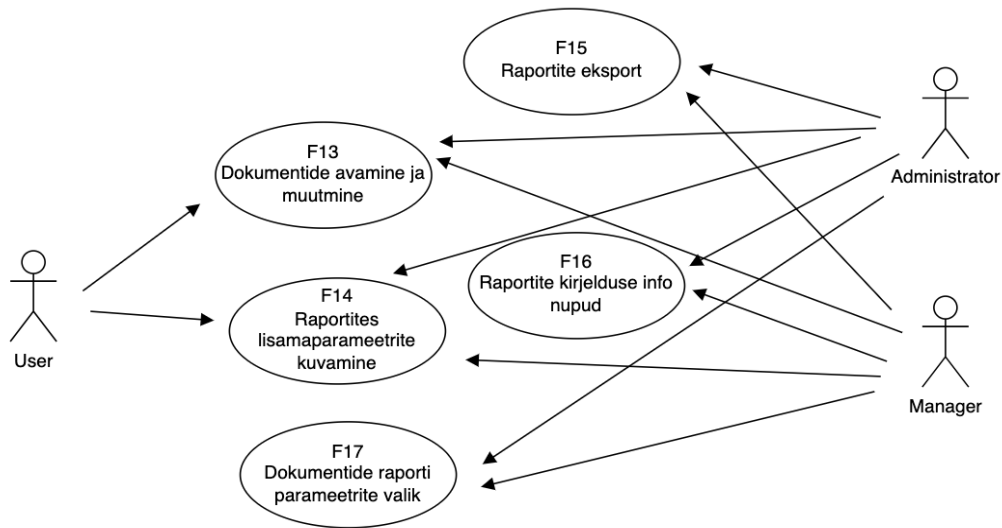
Kasutajad: Kõik kasutajagrupid.

Tähis: F17

Nimetus: Dokumentide raporti parameetrite valik.

Kirjeldus: Dokumente peab olema võimalik otsida dokumendi tüübi (arve, reisikava, lennupiletid, rongipiletid, hotelli voucher, autorendi voucher ja muude teenuse tüüpide järgi), arve numbri, teenuse broneeringu numbri alusel, reisija nime, teenuse sihtkoha, kliendi poolse projekti koodi, eelarve koodi ja finantsnumber alusel.

Kasutajad: Kliendi poolelt: Administraator, Reisi tellija, Raamatupidaja. Estraveli poolelt: Administraator, Kliendihaldur.



Joonis 7. Raportite funktsionaalsus ja seos kasutajagruppidega

2.3.2 Loodava portaali mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded vastavad küsimusele "Kuidas tarkvara peab vajalikke funktsioone täitma?"[3].

Järgnevalt vaatame, mis on ärikliendi iseteeninduskeskkonna mittefunktsionaalsed nõuded.

Tähis: MF1

Nimetus: Raportid selged ja arusaadavad.

Kirjeldus: Raportite vaated peavad olema võimalikult lihtsad ja arusaadavad ning kooskõlas tänapäeva UX parimate praktikatega.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF2

Nimetus: Disain.

Kirjeldus: Ärikliendi iseteeninduskeskkonna erinevad vaated peavad kokku minema Estraveli üldise kujundusega.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF3

Nimetus: Andmete ajakohasus.

Kirjeldus: Infopäringute tegemisel on oluline, et andmed oleksid ajakohased. Kuvatakse kuni 5 aastat vanu andmeid ja hilisemalt eelmise kuupäevaga sisestatud andmeid.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF4

Nimetus: Vastavus GDPR-ile.

Kirjeldus: On oluline, et iseteeninduskeskkond vastaks isikuandmete kaitse üldmäärusele ja vastaks kõikidele privaatsuse nõuetele.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF5

Nimetus: Kasutajagruppide õiguseid.

Kirjeldus: On oluline, et päringute tegemisel arvestab süsteem kasutaja rollidega, sest sellest sõltub, milliseid andmeid antud kasutaja näeb.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF6

Nimetus: Süsteemi jõudlus.

Kirjeldus: Kui päringu genereerimise nuppu vajutada, siis peaks raporti genereerimine aega võtma vähem kui 5 sekundit. Kui päringu koostamine võtab aega, siis peab süsteem andma teavituse, et päringut koostatakse.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF7

Nimetus: Päringu mahu piiramine.

Kirjeldus: Süsteem peab arvestama tehtavate päringute andmemahte ja kui päring on liiga suur, siis peab sellest kasutajat teavitama.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF8

Nimetus: Versioonihaldus.

Kirjeldus: Rakenduse arendamisel tuleb kasutada versioonihaldust, et iga uus versioon oleks versioneeritud.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF9

Nimetus: Testimine.

Kirjeldus: Rakenduse kõik üleantavad versioonid peavad enne toodangusse minemist olema testitud.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF10

Nimetus: Maksimaalne kasutajate arv.

Kirjeldus: Kuna süsteemi kasutavad erinevad kliendid ja süsteemi kasutajate arv võib olla erinevatel ajahetkedel väga erinev, siis süsteemi jõudlus ei tohi langeda, kui kasutajate arv on 100 või rohkem.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF11

Nimetus: Kasutajaliides peab olema *responsive*.

Kirjeldus: Kasutajaliidese disain peab kohalduma erinevate ekraanisuurustega ehk olema *responsive*.

Kasutajad: Kõik kasutajagrupid.

Tähis: MF12

Nimetus: Andmeväljade kohustuslikkus.

Kirjeldus: Kohustuslikud andmeväljad peavad kasutajale selgelt arusaadavad olema, nt tähistatud tärniga (*).

Kasutajad: Kõik kasutajagrupid.

2.4 Kasutajagrupid ja nende õigused

Käesolevas peatükis kirjeldatakse süsteemi erinevate eesmärkidega kasutajaid, mis kuuluvad vastavatesse kasutajagruppidesse. Vastavalt kasutajagruppidele saab seejärel määratleda süsteemi kasutajarollid ja nende õigused. Kasutajagruppe on kahte tüüpi: kliendi poolsed kasutajad ja Estraveli poolsed kasutajad.

Kliendi poolsed kasutajad

Administraator - Kliendi poolsetele kasutajatele antavad süsteemi ligipääsude haldajad määratletakse kliendi ja Estraveli vahelises lepingus. Administraator kinnitab süsteemis antud kasutajad.

Reisija - teenuse tarbija ehk kes kasutab reisiteenuseid. Reisija soovib oma broneeritud reisiteenuste kohta infot nii ajas etteulatuvalt kui ka tagasiulatuvalt. Reisija soovib alla laadida erinevaid reisidokumente (piletid, voucherid, kinnitused, arved jms).

Reisi tellija - Kliendi poolne reisiteenuste tellija, kellele on lepingukohaselt antud õigus ettevõtte nimel reise tellida ja neid kinnitada. Olenevalt kliendi sisemisest töökorraldusest haldab reisi tellija oma ettevõtte reise ja vaatab, et kõik aruanded oleksid korras ja sisaldaksid vajalikku infot (reisijate unikaalsed numbrid, projekti koodid, finants). Vajadusel võtab jooksvalt välja infot oma reisijate kohta. Kui on maailma mõned piirkonnas valitseb ohu olukord või on vaja mõnel muul põhjusel kiiresti välja selekteerida reisijad, kes viibivad antud ajahetkel antud piirkonnas, siis seda teeb reisi tellija.

Raamatupidaja - Kliendi poolne raamatupidaja soovib süsteemist välja võtta erinevate perioodide arveid, samuti soovib ta ka otsida konkreetset arvet tellimuse numbri või reisija nime või teenuse grupi järgi. Raamatupidaja ülesanne on ka kontrollida, et arved oleksid makstud ja vajadusel saaks välja võtta maksmata arvete listi. Sõltuvalt kliendi vajadustest vajavad ka osad raamatupidajad erinevaid teenuse raporteid koos kliendi ja reisija numbritega, projekti ja eelarve koodidega. Raamatupidaja soovib ka arve pilte süsteemist avada.

Estraveli poolsed kasutajad

Administraator - Estraveli poolsete kasutajate ringi haldab Estraveli administraator.

Kliendihaldur - Estraveli poolne kliendihaldur, kes on määratud antud kliendi lepingute ja teeninduse juhtimisega tegelema. Kinnitab vastavalt kliendi ja Estraveli lepingule kliendipoolsed kasutajad ja määrab neile kasutajaõigused.

Lähtuvalt kasutajate kirjeldustest saab jagada kasutajaõigused järgmistesse kasutajagruppidesse: *User*, *Manager*, *Superuser*. Lisas 2 on toodud kasutajagruppide õigused erinevates süsteemi vaadetes.

2.5 Kasutajate õiguste kirjeldused

Õiguste jagamisel liigsete peensusteni ei ole mindud ja on vaadatud suuremat plaani ning arvesse on võetud ka GDPR-ist lähtuvaid nõudeid. Täpsem kasutajaõiguste väljavõte on esitatud käesoleva töö Lisas 2. Kui kasutaja logib süsteemi sisse, siis kontrollitakse, millised on konkreetse kasutaja õigused. Kui kasutajal puudub ligipääs antud toimingutele, siis kuvatakse kasutajale, et tal puuduvad vastavad õigused. Raportite süsteem on eelkõige vaatamisõigustega, seega on ka kasutajaõiguste tabelis CRUD meetodist võetud ainult READ ehk siis lugemisõigus.

CRUD meetod pakub API-de puhul nelja põhitüüpi funktsioone. Mudel peab suutma andmeid luua (CREATE), lugeda (READ), värskendada (UPDATE) ja kustutada (DELETE) [4].

3 Tehniline analüüs

Käesolevas peatükis kirjeldatakse pärandüsteemi olemust üldiselt ja antakse detailsem ülevaade olemasolevast süsteemist. Samuti analüüsitakse JavaScripti ja Typescripti raamistike ning erinevaid arhitektuuri musterid.

3.1 Pärandüsteemi olemus

Iga tarkvara, ükskõik kui hea ta on, ei ole aastate möödudes enam nii hea ja ajakohane. Üldjuhul enamik IT-süsteeme vajavad aja möödudes kaasajastamist. Kindlasti tekib küsimus, et mille järgi otsustada, et kas süsteem on vaja ümber kirjutada või mitte.

Esmalt toon välja pärandüsteemi mõiste. Pärandüsteem on vana või aegunud süsteem, tehnoloogia või tarkvararakendus, mida organisatsioon kasutab jätkuvalt, kuna see täidab endiselt funktsioone, mida ta algselt täitis. Üldjuhul ei ole pärandüsteemidel enam tuge ega hooldust ning nende kasv on piiratud ja nad ei rahulda enam organisatsiooni tehnoloogilisi vajadusi [5].

Pärandüsteemi ei ole aga alati määratletud selle vanuse järgi. Süsteemi peetakse pärandvaraks, kuna see võib olla tingitud toetuse puudumisest või suutmatusest täita ettevõtte või organisatsiooni vajadusi. Teiseks oluliseks küljeks on süsteemi hooldamiseks minev rahaline kulu. Hooldamise all mõtleme eelkõige: süsteemi tuge ja hooldust, turvalisust, vastavus nõuetele, ettevõtte võimaluste kaotamine turul, ettevõtte efektiivne toimimine [6].

Pärandüsteemid on tavaliselt nii koodibaasi kui ka funktsionaalsuse poolest üsna suured. Arvestades nende monoliitsust, ei saa süsteemi mooduleid muuta ega asendada nii kergesti. Väiksemgi uuendus või muudatus võib kaasa tuua mitu konflikti kogu süsteemis. Seega nõuab iga pärandüsteemi muudatus või värskendamine aega ja vaeva, millest kumbki pole odav [6].

Sarnaselt tarkvara endaga muutub ka selle aluseks oleva infrastruktuuri hooldamine raskemaks ja kulukamaks. Pärandsüsteemid nõuavad sageli spetsiifilist tehnilist keskkonda, sealhulgas riistvara. Seega on infrastruktuuri ülalpidamise kulutused tänapäevaste pilvepõhiste lahendustega võrreldes kõrged [6].

Selleks, et hakata pärandsüsteemi uuendama on mõistlik läbi viia olemasoleva süsteemi analüüs ja järgida mõnd hindamise raamistikku. Olulisemad punktid, millel tuleks peatuda on tehnoloogiate analüüs, arhitektuuri audit, koodi ülevaade, UI/UX ülevaade, jõudluskatsed, praegused nõuded ja võimalused tulevaseks kasvuks [6].

3.2 Pärandsüsteemist ülemineku võimalused

Kui pärandsüsteemi analüüsi tulemused näitavad, et antud süsteemiga on mõistlik ette võtta muudatused, siis üleminekuks on järgmised võimalused.

Täielikult uuele süsteemile üleminek

Loogiline lahendus on asendada pärandsüsteemid täielikult uue platvormiga. Loomulikult pole see lihtne valik. Protsess on süsteemi keerukuse tõttu äärmiselt aeganõudev ning enne uue süsteemi paigaldamist tuleb välja töötada keerukas ja läbimõeldud strateegia. Vastust vajavad olulised küsimused, näiteks mida peaks uus süsteem tegelikult suutma ja millist arhitektuuri selleks vaja on [7]?

Järkjärguline üleminek uuele süsteemile

Selle asemel, et kogu süsteem korraga välja vahetada, saab rakendada järkjärgulist üleminekut moodulite kaupa. See võimaldab ka kasutajatel omandada uue süsteemiga esmased kogemused ja anda tagasisidet [7].

Lugesdes erinevaid pärandsüsteemi ümberkirjutamisega seotud artikleid, siis kõik rõhutavad, et uuele süsteemile üleminekul tuleb kaasata kogu ettevõtte mitte ainult projektimeeskond. Uuele süsteemile ülemineku käigus tuleb kaasa tuua ka ettevõtte kultuur, juhtimisstiil ja kõige olulisem, et kogu meeskond mõistaks uuele süsteemile ülemineku vajadust.

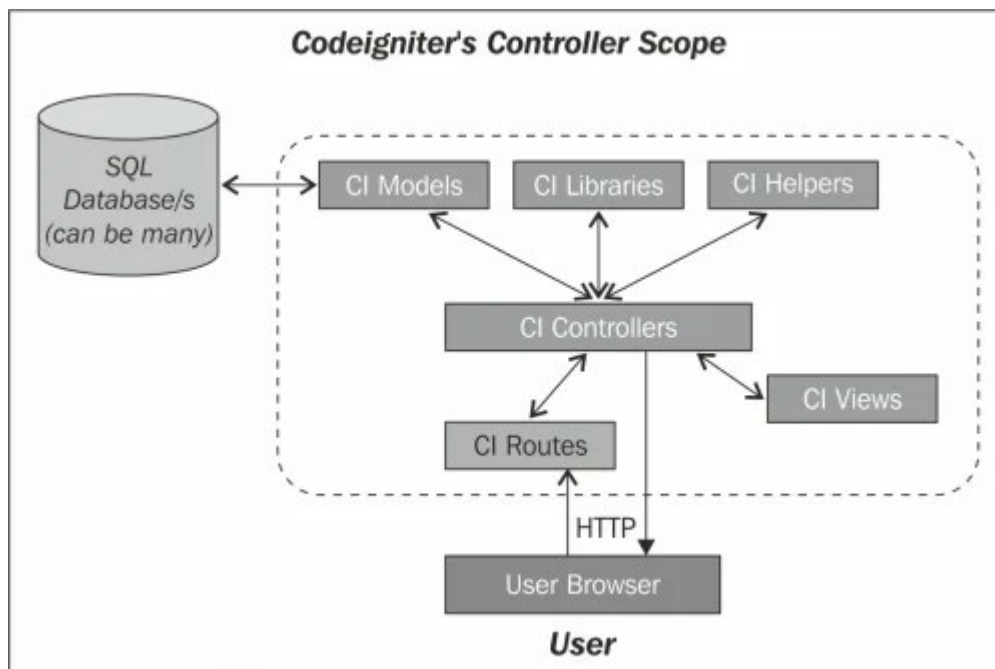
3.3 Olemasoleva pärandüsteemi kirjeldus

Olemasolev Estraveli “Extranet” portaal on loodud php raamistikul. Olemasolev portaal on loodud Model View Controller (MVC) arhitektuuril. Kasutusel on CodeIgniter raamistik, et lihtsustada tarkvaraarendust ja kontrollereite, vaadete ja mudelite genereerimist.

Süsteem suhtleb otse “Extranet” MySQL andmebaasiga. Andmemudelid, vaated ja kontrollid on automaatselt genereeritud Scaffolding meetodil, et teostada efektiivselt andmebaasi CRUD operatsioone.

Genereeritud kontrollid, vaated ja mudelid on seotud arvete, teenuste ja reisijate raportite infoga. Süsteem võimaldab kuvada raporteid arvete kuupäevade järgi. Olemasoleva süsteemi külge on loodud ka võimalus teha päringuid ka läbi REST API-ide.

Kasutusele on võetud ka JSON-RPC PHP raamistik JSON tüüpi päringute paremaks haldamiseks. Järgnevalt on toodud olemasoleva lahenduse üldine arhitektuur.



Joonis 8. Praeguse rakenduse üldine arhitektuur [8]

3.4 Tööriistade ja raamistike valik uue süsteemi jaoks

Käesoleva töö tegemiseks vajalike tööriistade ja raamistike valimisel on autor lähtunud eelkõige õpingu perioodil tuttavaks saanud tööriistadest ja raamistikest ning on lähtunud parimatest praktikatest ja tänapäevastest tarkvara arenduste meetoditest.

Backend teenuste REST API-de kirjeldamisel on kasutatud SwaggerHub online keskkonda, mis võimaldab kirjeldada REST API teenuseid vastavalt OpenAPI (OAS) spetsifikatsioonile [9], [10]. Defineerida saab päringu tüüpi, parameetreid, andmetüüpe, kirjeldusi jne. API-de testimisel on kasutatud Postman desktopi põhist rakendust [11].

Kasutaja vaadete disainimisel on kasutatud diagrams.net online rakendust [12].

Front-endi arendamise keskkonnaks valisin tasuta ja vabavaralise Visual Studio Code programmi [13].

Dokumentatsiooni koostamisel on kasutatud Google Docsi ja koopiade halduseks Google Drive keskkonda.

3.5 Erinevate JavaScripti ja Typescripti raamistike analüüs

3.5.1 Javascript ja Typescript

JavaScripti raamistik on JavaScripti koodi teekide kogu, mis pakub veebiarendajale korduvate programmeerimise toimingute jaoks eelnevalt kirjutatud koodi [14].

JavaScript on kõrgetasemeline, sageli “*just-in-time*” kompileeritud keel, mis vastab ECMAScript standardile. ECMAScript (or ES) on Javascripti standard, mis on mõeldud veebilehtede omavahelise ühilduvuse tagamiseks erinevates veebibrauserites [15].

Kui JavaScript töötati välja, tutvustas JavaScripti arendusmeeskond JavaScripti kliendipoolse programmeerimis keelena. Kuid kui arendajad JavaScripti kasutama hakkasid said arendajad nad aru, et JavaScripti saab kasutada ka serveripoolse

programmeerimiskeelena. JavaScripti kasvades, muutus JavaScripti kood keeruliseks ja raskeks. Seetõttu ei suutnud JavaScript isegi täielikult täita objektorienteeritud programmeerimiskeele nõuet. See takistab senini JavaScripti kui serveripoolse tehnoloogia edukust suurettevõtete tasemel [16].

Seejärel töötas arendusmeeskond välja TypeScripti, et see lõhe ületada. TypeScript on ECMAScript 6 (ES6) versioon ja on tagasiühilduv ECMAScript 5-ga (st JavaScriptiga) standardiga [17], [18].

Tänapäeval tuntuimad ja enim kasutatavad JavaScripti ja Typescripti raamistikud on React, Vue ja Angular [19]. Nende põhilised erinevused seisnevad selles, kuidas nad on koostatud, millist süntaksit kasutatavad ja millised tööriistad nendega seotud on. Ei ole olemas ühte parimat, vaid sobiv raamistik tuleb valida vastavalt vajadustele, oskustele ja erinevatele väikestele nüanssidele tuginedes. Tihti peale saavadki määravaks väikesed detailid, aga detailid on samuti olulised. Järgnevalt teen väikese võrdluse eelmainitud raamistike kohta, et valida oma praktilisele töö osale sobiv raamistik.

3.5.2 React

React on JavaScripti teek peaauglikult “*single-page*” tüüpi kasutajaliideste või mobiilirakenduste loomiseks. React teek on välja arendatud ettevõtte Meta (endine Facebook) poolt, mille esmane versioon anti välja 2013. aastal. React muudab interaktiivsete kasutajaliideste loomise kergeks. Rakenduse iga oleku jaoks saab kujundada lihtsaid vaateid ning React värskendab ja renderdab andmete muutumisel tõhusalt õigeid komponente [20].

Reacti kasutatakse põhiliselt baasteegina, kelle ülesandeks on seisu haldamine ja selle renderdamine, seega rakenduste loomiseks on soovitav kasutada täiendavaid teeke. React-i kasutatakse tihti peale koos JSX-ga (Javascript XML), mis on Javascripti keele laiendus ja mis sarnaneb HTML-ile ja mis võimaldab komponentide struktureeritud renderdamist, mis on paljudele arendajatele tuttav [20].

3.5.3 Angular

Angular (tuntud ka kui Angular 2+) on Typescripti põhine vaba ja avatud lähtekoodiga veebirakenduste arendamise raamistik. Angular on arendatud ettevõtte Google poolt. Esimene versioon Angular raamistikust oli arendatud Jaavascripti baasil aastal 2010. Hiljem kirjutas sama meeskond selle ümber Typescripti aastal 2016. Angular raamistikku kasutatakse samuti põhiliselt kasutajaliideste arenduseks. Angular kasutab komponente, mis on defineeritud “NgMoodul”-itena. Angular NgModules erinevad JavaScripti (ES2015) moodulitest natuke, kuid täiendavad neid. Komponentid kasutavad teenuseid “services”. Teenuseid saab lisada komponentidesse sõltuvustena “Dependency Incection”, muutes koodi modulaarseks, korduvkasutatavaks ja tõhusaks [21].

3.5.4 Vue

Vue on avatud lähtekoodiga Typescriptil põhinev veebirakenduste arendamise raamistik, mis on loodud Evan You poolt aastal 2004, kes varasemalt töötas Google AngularJs meeskonnas. Peale Google meeskonnast lahkumist otsustas ta luua oma raamistiku korjates välja enda arust parima osa AngularJs-st. Vue põhiline fookus on vaadete kihis, kuid sisaldab ka keerukamat funktsionaalsust nagu *routing* ja *state* haldus. Vue võimaldab tavapärasest HTML-i rikastada [22].

3.5.5 Angular või React või Vue, mida kasutada?

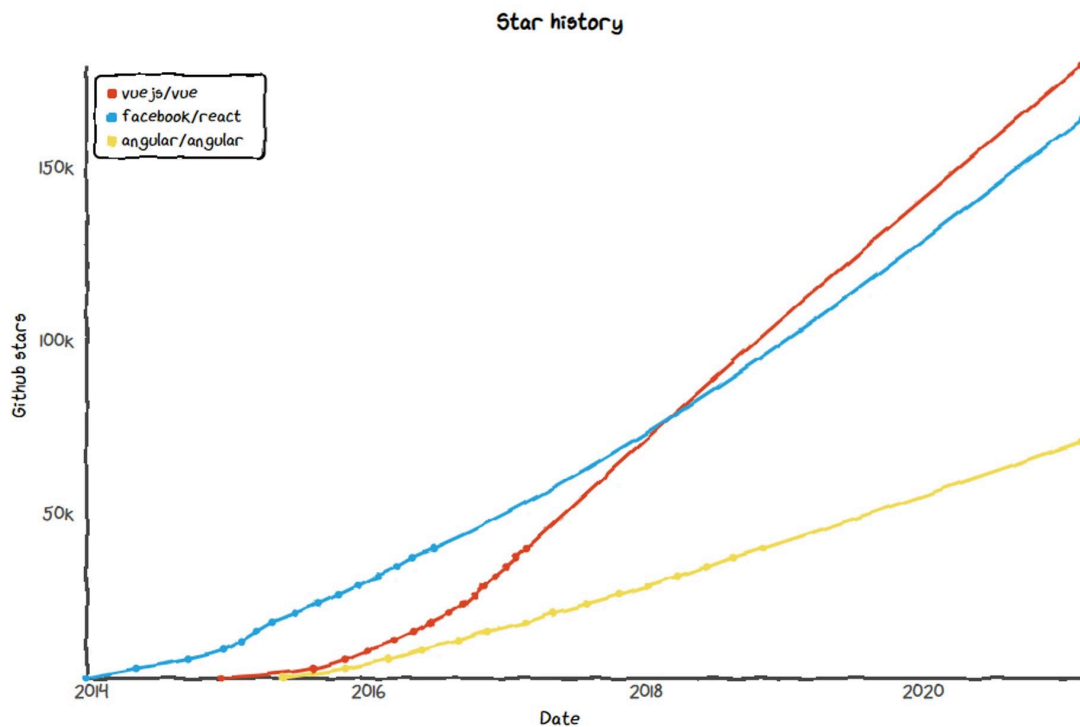
Veel paar aastat tagasi arutlesid arendajad peamiselt selle üle, kas nad peaksid oma projektide jaoks kasutama Angular või Reacti. Kuid viimase paari aasta jooksul oleme näinud huvi kasvu kolmanda raamistiku nimega Vue.js vastu.

Angular ja React raamistiku mõlemat saab hästi kasutada mobiili- ja veebirakenduste jaoks, kuid Angular on üldiselt parem keerukamate rakenduste jaoks. React nõuab sageli lisamoduleid ja komponente, mis hoiab põhiteegi väikesena, kuid tähendab, et väliste tööriistade kaasamine nõuab lisatööd. Angular on seevastu rohkem täielikum teek, mis ei vaja palju lisasid. React sobib rohkem kesktaseme ja edasijõudnutele JavaScripti

arendajatele, kes on tuttavad ES6 ja uuemate kontseptsioonidega, samas kui Angular on sobilik arendajatele, kes tunnevad ka TypeScripti [23], [24].

Vue sobib üldiselt paremini väiksematele ja vähem keerukatele rakendustele ning seda on Reactiga ja Angulariga võrreldes lihtsam nullist õppida. Vue-d võib olla lihtsam uutesse või olemasolevatesse projektidesse integreerida ja paljud tunnevad, et selle HTML-mallide kasutamine koos JSX-iga on eelis. Vue on selge valik vähem kogenud arendajatele ja Angular on eelistatud neile, kes töötavad suuremate rakendustega. Suur teek nagu Angular nõuab rohkem hoolsust, et olla uuendustega kursis ja Vue on selle osas vähem nõudlik [23], [24].

Järgnevalt on toodud GitHub projektide arv ajas, mis kasutavad React, Angular ja Vue raamistikke [23].



Joonis 9. React, Angular ja Vue raamistiktes tehtud projektide arv ajas [23]

3.6 Erinevad arhitektuuri mustrid

3.6.1 MVC

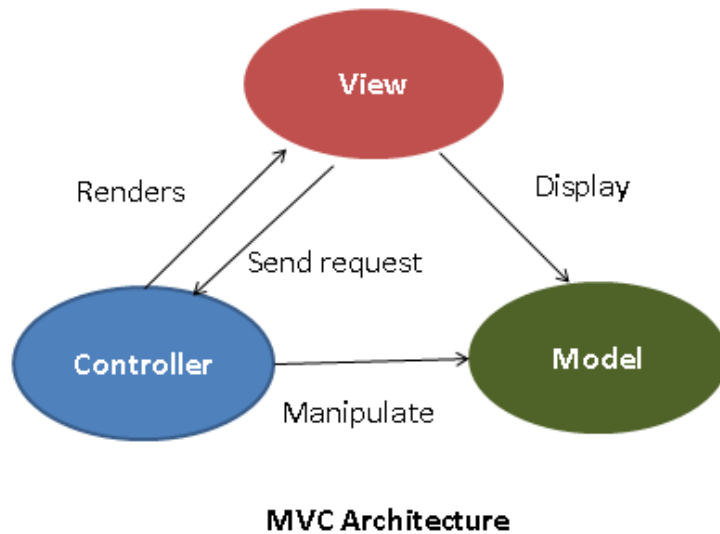
Model View Controller (MVC) arhitektuuri muster on olnud pikalt kasutusel arenduses (alates 1976 aastast) ja on tuntud oma kolme kihilise arhitektuuri poolest:

- Mudel (Model): säilitab rakenduse andmed ja peamised komponendid;
- Vaade (View): kuvab mudeli kasutajaliidese vaated;
- Kontroller (Controller): toimib liidesena vaate ja mudeli komponentide vahel [25].

MVC peamine idee seisneb selles, et koodi igal osal on eesmärk ja need eesmärgid on erinevad. Osa koodi sisaldab rakenduse andmeid, osa koodist muudab rakenduse välimust ja osa koodist juhib rakenduse toimimist.

MVC mudeli erinevatel osade ülesanded on järgmised:

- Mudel: mudeli kood kirjeldab tavaliselt reaalseid asju. Mudel sisaldada üldjuhul rakenduse andmeid ja olulisi rakenduse jaoks vajalikke komponente. Näiteks kui teha Ülesande rakendus, määratleks mudeli kood, mis on "ülesanne" ja mis on "loend", kuna need on ülesande rakenduste peamised komponendid;
- Vaade: Vaated koosnevad funktsioonidest, mis suhtlevad vahetult kasutajaga. See on osa koodist, mis muudab teie rakenduse ilusaks ja määrab muul viisil, kuidas kasutaja seda näeb ja rakendusega suhtleb;
- Kontroller: Kontrolleri kood toimib mudeli ja vaate vahelise siduva lülina, võttes vastu kasutaja sisendi ja otsustades, mida sellega teha. See on rakenduse aju ning ta seob omavahel kokku mudeli ja vaate. Kui kontroller saab kasutajalt päringu, kasutab see sobivat mudelit ja vaadet ning genereerib vastuse, saates selle kasutajale tagasi [25], [26].

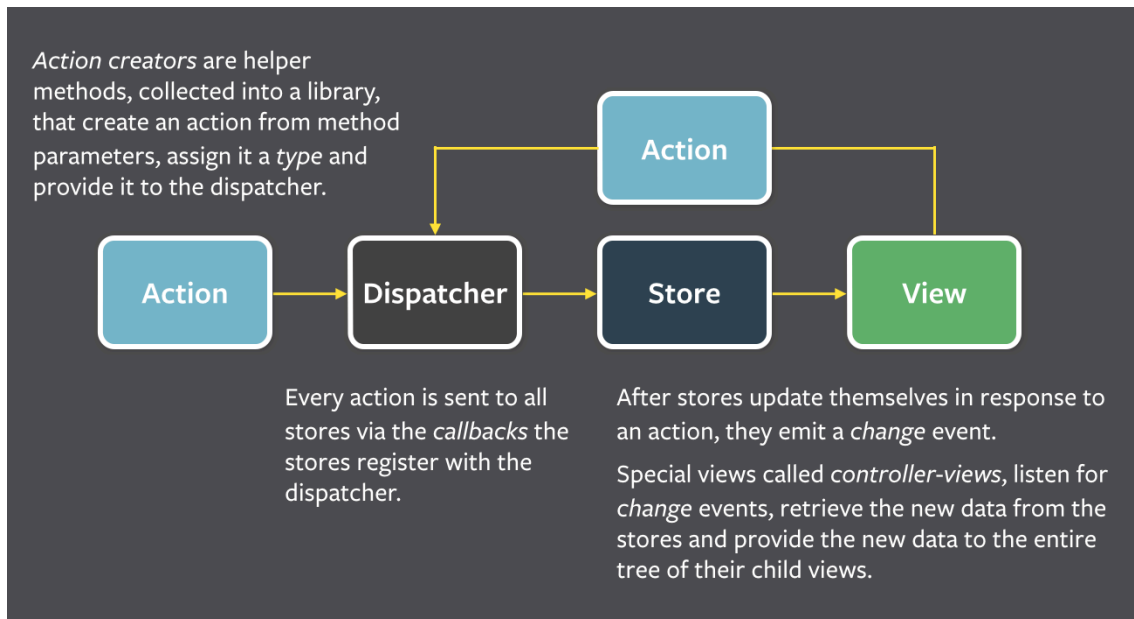


Joonis 10. Model View Controller toimimise skeem [25]

3.6.2 Flux

Facebooki arendajad seisid silmitsi probleemidega oma MVC-rakenduse skaleerimisel MVC arhitektuuri ebastabiilsuse ja keerukuse tõttu. Nad otsustasid teha süsteemis mõned olulised muudatused ja lasid välja Fluxi arhitektuuri alternatiivina MVC arhitektuurile. Flux on mõeldud kliendipoolsete veebirakenduste loomiseks. See täiendab Reacti vaate komponente, kasutades ühesuunalist andmevoogu. See on pigem muster kui formaalne raamistik. Fluxi arhitektuur põhineb järgmistel komponentidel:

- Hoidla (Store): rakenduse oleku ja loogika konteiner;
- Toiming (Action): andmete edastamise dispatcher-ile;
- Vaade (View): sama, mis vaade MVC arhitektuuris;
- Dispetšer (Dispatcher): koordineerib kaupluste toiminguid ja värskendusi [27].



Joonis 11. Flux raamistiku skeem [27]

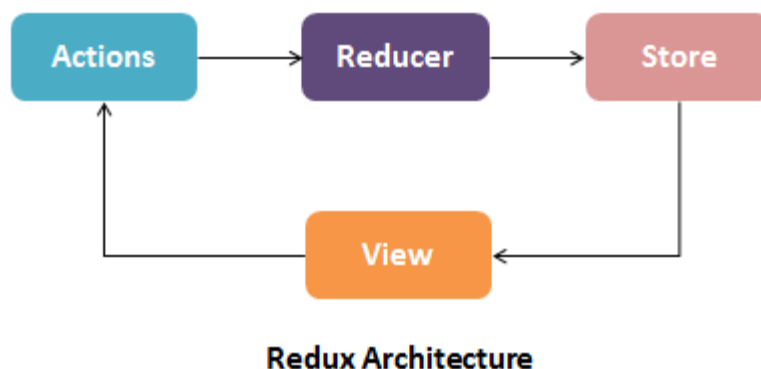
Kui kasutaja millelgi klõpsab, siis loob vaate (*view*) toiminguid (*actions*). Toiming (*action*) võib luua uusi andmeid ja saata need dispetšerile (*dispatcher*). Seejärel saadab dispetšer tegevuse tulemuse vastavasse hoidlasse (*store*). Hoidla (*store*) värskendab olekut tulemuse põhjal ja saadab vaatele uuenduse [27].

3.6.3 Redux

Aastal 2015 aga leidis Dan Abramov, et see arhitektuur võiks olla oluliselt lihtsam. Selle tulemusel arendasid Dan Abramov ja Andrew Clark Reduxi.

Redux on teek, mis järgib Fluxi ideed, kuid hoopis teistsugusel viisil. Reduxi arhitektuur tutvustab mõned uued komponendid:

- Reduktor (*Reducer*): loogika, mis otsustab, kuidas andmed muutuvad funktsioonides;
- Tsentraliseeritud hoidla (*Centralized Store*): sisaldab oleku objekti, mis hoiab kogu rakenduse olekut [27].



Joonis 12. Reduxi töötamise skeem [26]

Fluxi ja Reduxi peamine erinevus seisneb selles, et Flux sisaldab rakenduse kohta mitut hoidlat, kuid Redux sisaldab ühte hoidlat rakenduse kohta. Selle asemel, et paigutada olekuteavet mitmesse rakenduse hoidlasse, hoiab Redux kõike rakenduse ühes piirkonnas. See teeb rakenduse haldamise oluliselt lihtsamaks [27].

3.6.4 Vuex

Vuex tekkis sarnaselt Reduxiga. Umbes pool aastat pärast Reduxi kasutuselevõttu, 2016. aasta alguses, algas töö Vuexi kallal. Vuex põhineb samuti Flux mustriks, kuid nagu ka Reduxis, ei järgi see täielikult Flux mustrit. Samamoodi nagu Reduxi puhul on ka Vuex puhul "ühe tõeallika" põhimõte - üks rakendus, üks hoidla (*store*). Vuexil pole ka dispetšerit (*dispatcher*), mis on Fluxis olemas. Nii Redux kui ka Vuex tegelevad oleku muutmisega, mitte nii, nagu Flux eeldab. Mõlemal juhul kandub dispetšeri vastutus hoidlale (*store*). Vuex ja Redux põhiline erinevus seisneb selles, et Redux kasutab reduktooreid (*Reducers*), siis Vuex kasutab mutatsioone (*mutations*). Reduxi olek on alati muutumatu, samas kui Vuexis on andmete muutmiseks ainus viis poe poolt mutatsiooni läbiviimine. Mutatsioone on natuke lihtsam kasutada, kui reduktooreid [28].

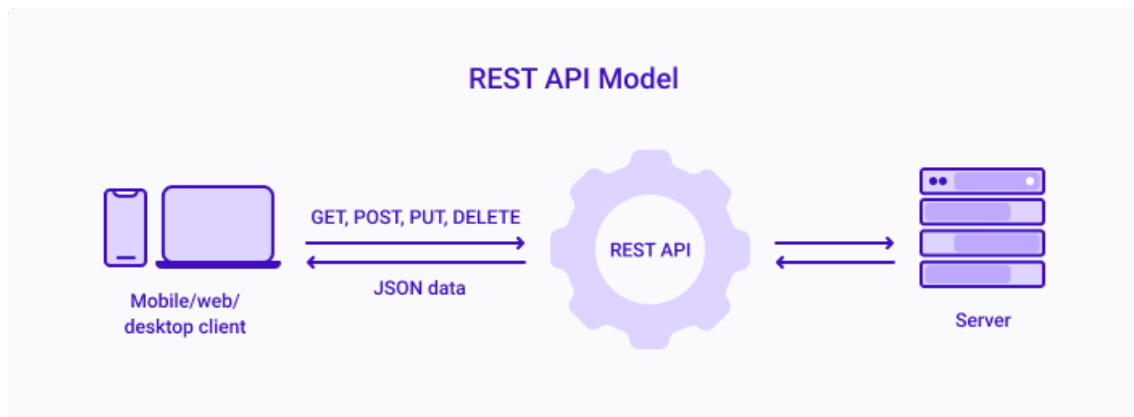
4 Loodava süsteemi kirjeldus

Järgnevas peatükis antakse ülevaade uue süsteemi loomiseks valitud tehnoloogiatest ja arhitektuurist.

4.1 Valitud tehnoloogiad ja arhitektuur

Uue loodava Estraveli “Extranet” portaali puhul sai otsustatud täiesti uue süsteemi loomise kasuks, et teha mõned olulised muudatused nii süsteemi arhitektuuris, programmeerimiskeeles, kui ka funktsionaalsuses.

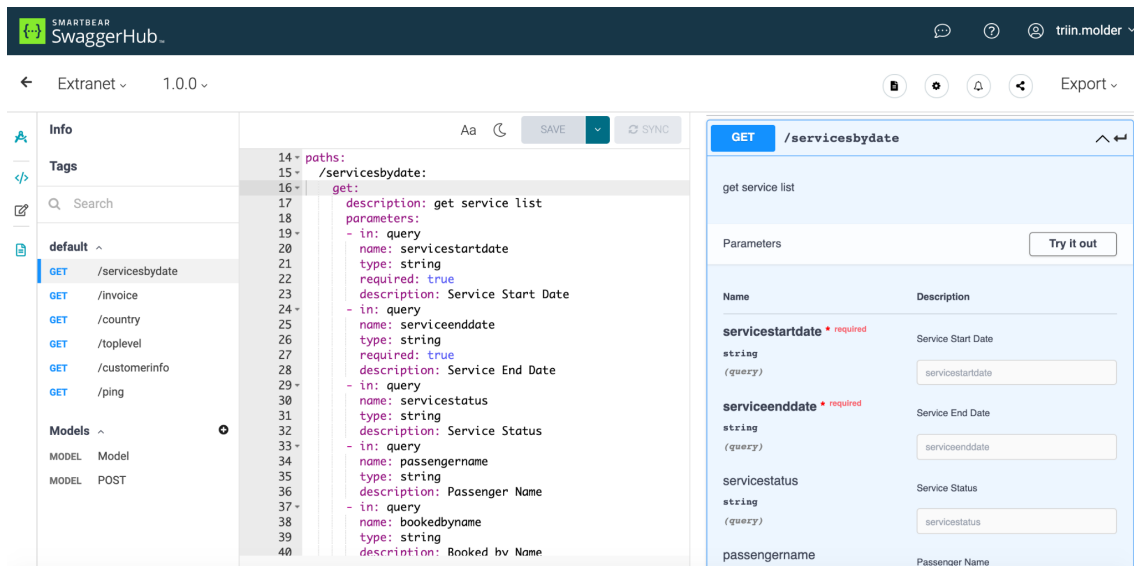
Esiteks sai viidud lahti front-end backend teenustest kasutades selleks REST API põhiste teenuste integratsiooni. See võimaldab arendada kasutajaliidest eraldiseisvalt ülejäänud süsteemist ja võimaldab vajadusel täiendavaid või uusi kasutajaliideseid samadele teenustele (näiteks mobiilirakendus).



Joonis 13. REST API mudel [29]

Backend teenuste jaoks sai defineeritud vajalikud REST API teenused kasutades SwaggerHub portaali, mis võimaldab defineerida vajalikud teenused OpenAPI spetsifikatsiooni kohaselt ja võimaldab loodud API-sid ka kohe testida. Backend teenuste arendaja lõi antud spetsifikatsioonide järgi vajalikud teenused. Süsteemi kasutajaliides suhtleb backend teenustega läbi turvaliste HTTPS põhise REST API-de. Teenuste testimiseks sai kasutatud Postman rakendust.

Järgnevalt on toodud näide SwaggerHub keskkonnast, kus vajalikud teenuste mudelid kirjeldatud on.



Joonis 14. Loodava süsteemi SwaggerHub teenuse mudeli näide

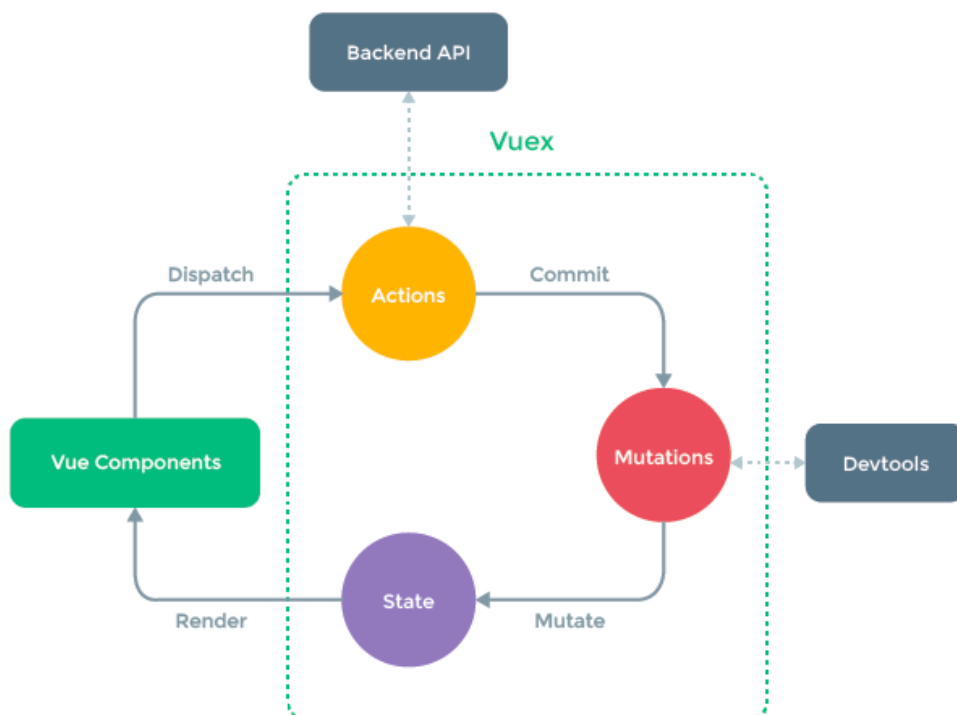
Frontend poolel sai kasutusele võetud Axios raamistik REST API päringute teostamiseks. Axios põhineb javascriptil ja on loodud spetsiaalselt HTTP teenuste hõlpsamaks kasutamiseks [30]. Axios võimaldab ära defineerida päringu ja päringule oodatava vastuse struktuurid väga mugavalt. Vajalikud teenused on jagatud vastavalt funktsionaalsusele eraldi failidesse.

Teenuste kasutamiseks peavad kasutajad ennast autentima. Kasutajate sisse ja välja logimiseks sai loodud eraldi komponent “Identity”, mis hakkab haldama kasutajate sisse ja välja logimise staatust, kasutajate registreerimist, nende rolle ja õiguseid süsteemis.

Loodava süsteemi arendamiseks sai valitud TypeScript (ECMAScript 6), mis on ka tagasiühilduv ECMAScript 5-ga (st JavaScriptiga) standardiga. TypeScript kasuks otsustamise tingis asjaolu, et TypeScript on nii öelda tugevalt tüübitud programmeerimiskeel, mis teeb arendamise lihtsamaks ja võimaldab vältida nõrgalt tüübitud keelega võrreldes arendusvigu, kuna andmetüübid on rangelt/konkreetselt defineeritud ja võimalike vigade tegemine on väiksem. Kasutatavad andmestruktuurid sai defineeritud eraldi domeeni objektidena Typescript abil. Loodud andmestruktuurid ja nende vahelised seosed on toodud töö Lisas 4.

Kasutatavaks arenduse raamistikuks sai valitud Vue.js, kuna see sobib paremini väiksematele ja vähem keerukatele rakendustele ja kuna olen arenduses veel algaja, siis oli seda lihtsam õppida, kui Reacti või Angulari.

Koos Vue.js raamistikuga sai valitud ka Vuex rakenduse seisude haldamise raamistik. Nagu eelnevalt kirjeldatud, siis on rakenduses vaid üks hoidla (*store*), mille olekute muutmine käib läbi mutatsioonide. Järgnevalt on toodud ülevaade, kuidas Vuex loodud rakenduses andmete statusseid haldab.



Joonis 15. Loodud rakenduse andmete seisude haldamise Vuex põhine arhitektuur [31]

Erinevad kasutaja vaated on loodud eraldi vaadetena ja nende vahel navigeerimiseks on kasutusel "Router", mis suunab kasutajad õigesse vaatesse, kui kasutaja vajutab vastava lingi peale [32].

Järgmises peatükis on toodud ülevaade loodud kasutajaliideste vaadetest, mis rakenduse erinevaid funktsioone täidavad.

Rakenduse kogu lähtekood koos versioonihaldusega on üles laetud GitLab keskkonda.

4.2 UX ja UI

Enne reaalselt kasutajaliidese loomist front-endis sai tehtud uue süsteemi kasutajaliidese vaated. Selleks sai kasutatud Draw.io programmi. Loodud kasutajaliidese vaated said läbi vaadatud kõikide kasutajate gruppidega ning saadud tagasiside põhjal sai tehtud täiendavad parandused. Antud kasutajaliidese loomisel sai arvesse võetud UX (user experience) ning UI (user interface) parimaid praktikaid, milleks on eelkõige:

- mõista, kes on süsteemi kasutajad ja mis on nende eesmärgid antud süsteemis;
- jälgida enamlevinud mustreid, millega kasutajad on harjunud. Näiteks erinevate elementide harjumuspärased asukohad;
- tasub jälgida, et klikke lõpptegevuse tegemiseks oleks võimalikult vähe;
- jälgida brändi üldist disaini, seda nii logo, kirjastiili, värvilahendustega;
- kui on võimalik, siis tasub vaikeväärtused määratleda, et kasutaja jõuaks oodatud tulemuseni kiiremini;
- info erinevate toimingute kohta, et kasutaja saaks aru, mis etapis toimin on [33].

Arvesse võttes erinevaid parimaid UX ja UI praktikaid sai loodud süsteemi kasutajaliideste vaated, mis on toodud Lisas 3.

5 Töö tulemused

Käesolev töö andis eelkõige tööautorile endale kasuliku ja edaspidiseks tööks vajalikud kogemused, kuidas toimib kogu protsess olles ise nii analüütik, arendaja kui testija.

Käesoleva töö raames on valminud kasutajaliidese vaated, mida on võimalik kasutada ka hiljem keskkonna edasiarendamisel. Lisaks on välja selgitatud põhjaliku analüüsi ja võrdluse tulemusel uue süsteemi loomiseks parim tehniline raamistik ja tööriistad.

Välja sai selgitatud koostöös kliendi, kliendihalduritega erinevad kasutajagrupid ja nende õigused iseteeninduskeskkonnas.

Töö praktilise poole pealt valmis Estravel AS uue iseteenindusportaaali baaslahendus koos kahe raportimooduliga. Lahenduse arhitektuur ja loodud baasmoodulid seavad aluse tulevaseks lisamoodulite kiireks arenduseks, et olemasolev süsteem vahetada välja uue ja moodsama vastu. Raporti moodulitest sai valmis tehtud üks raport arvete kohta, mida on võimalik vastavalt funktsionaalsete nõuete kirjeldusele erinevate filtritega piiritleda ja teine raport on reise ülevaate kohta, mida samuti on võimalik piiritleda erinevate otsigu kitsendustega. Raporteid on võimalik ka alla laadida .xlsx ja .csv formaadis. Arveid on võimalik avada .pdf kujul.

Kõik töö eesmärgiks seatud eesmärgid said saavutatud arvestades samas, et tehtud töö on vaid üks osa suuremast kliendi iseteenindussüsteemist.

5.1 Tuleviku arendused

Kuna antud töö on üks väike osa suuremast projektist, siis tuleviku arendused moodustavad kogu projektist väga olulise osa. Järgnevalt tuleb läbi viia lõppkasutajatega kasutusmugavuse testid, et saada konstruktiivset tagasisidet ja teha vajadusel täiendused. Seejärel saab edasi liikuda järgmiste raportite Backend API-de loomise ja täiendavate raportimoodulite arendusega, kus tuleb arvestada ka visuaale, mis järgmiste raportite jaoks said antud töös loodud.

Lisaks on vajalik luua igapäeva töölaua moodul juba olemasolevate visuaalide põhjal.

Käesoleva töö valmimise jooksul on hetkel jäänud õhku küsimus, kas eraldi dokumentide vaade põhimenüüna on vajalik või on see mõistlik tõsta raportite alla ja teha eraldi otsing. Eelmainitud punkt vajab aga ettevõtte sisemist arutelu.

Samuti on veel teisi teemasid, mis vajavad ettevõtte sisemist otsustamist, kuidas kogu süsteemiga edasi liikuda. Näiteks üks põhiküsimusi, kas ja millisel määral omavahele siduda erakliendi ja ärikliendi iseteeninduse keskkonnad.

6 Kokkuvõte

Käesolevas töös püstitatud ülesandeks oli parendada praegust ärikliendi süsteemi kahel olulisel eesmärgil. Esiteks, et ärikliendid saaksid iseseisvalt vajalikku infot Estravelilt tellitud teenustest, arvetest ja teenustega seotud dokumentidest ning neid ka iseseisvalt alla laadida. Teine oluline aspekt oli viia süsteem üle uuemale tehnoloogiale, et hiljem seda ka teiste moodulitega integreerida ja täiendada.

Püstitatud eesmärkide edukaks lahendamiseks analüüsisin olemasolevat keskkonda. Keskkonna analüüsimisel testisin keskkonda manuaalselt vaadates üle olemasoleva funktsionaalsuse. Klientidelt kogutud tagasiside analüüsimise ning kliendihaldurite ja müügidirektorilt saadud ettepanekute põhjal disainisin uue süsteemi vaated arvestades defineeritud funktsionaalseid ja mittefunktsionaalseid nõudeid.

Kasutajagruppide ja kasutajaõiguste defineerimisel arvestasin nii kliendi kui Estraveli poolsete kasutajatega. Kliendipoolseid kasutajagruppe on neli: Administraator, Reisija, Reisi tellija, Raamatupidaja. Estraveli poolseid kasutajagruppe on kaks: Administraator ja Kliendihaldur.

Antud töö ajalises mahus hõlmas suures osas analüütiline analüüs, mis on nii käesoleva töö vaatepunktist kui ka kogu kliendi iseteeninduskeskkonna tulevikuvaatest väga oluline. Analüütilisele poolele järgnes tehniline analüüs, et valida sobivad tööriistad ja raamistikud, et teostada töö praktiline osa. Kasutajaliidese poolele sai kasutusele võetud Vue.js. Koos Vue.js raamistikuga sai valitud ka Vuex rakenduse seisude haldamise raamistik.

Töö praktilise poole pealt valmis uus äriklientide iseteenindus portaal koos kahe raporti mooduliga, mida on võimalik kliendil juba kasutada. Loodud said siis üks raport arvete ja teine reiside kohta.

Käesolev töö annab võimaluse ettevõttel edasi liikuda uue ärikliendi portaali arendamisega. Uus ärikliendikeskkond on parem võrreldes vana keskkonnaga kaasaegse

tehnoloogia ja kasutaja jaoks selgema funktsionaalsuse poolest, mida ühes või teises raportis andmete kuvamisel aluseks on võetud.

Töö kõik eesmärgid said saavutatud ja loodud süsteem on hea vundament Estravel AS uuest äriklientide iseteenindussüsteemist.

Kasutatud kirjandus

- [1] Estravel, *Meist*. [Online]. Loetud aadressil: <https://www.estravel.ee/meist/> Kasutatud: 06.03.2021
- [2] K. Pohl, *Requirements Engineering. Fundamentals, Principles, and Techniques*. Springer, 2010
- [3] J. Tepandi, *Tarkvara protsessid ja kvaliteet*, 2020
- [4] Codecademy, *What is CRUD?*. [Online]. Loetud aadressil: <https://www.codecademy.com/article/what-is-crud> Kasutatud: 27.02.2022
- [5] StackScale, *What is a legacy system?*, 2021. [Online]. Loetud aadressil: <https://www.stackscale.com/blog/legacy-systems/> Kasutatud: 05.02.2022
- [6] Altexsoft, *Legacy System Modernization: How to Transform the Enterprise for Digital Future*. [Online]. Loetud aadressil: <https://www.altexsoft.com/whitepapers/legacy-system-modernization-how-to-transform-the-enterprise-for-digital-future/> Kasutatud: 05.02.2022
- [7] Fintech & Finance News, *Migrating from legacy to new banking systems*, 2021. [Online]. Loetud aadressil: <https://ffnews.com/newsarticle/migrating-from-legacy-to-new-banking-systems/> Kasutatud: 05.02.2022
- [8] Packt, *Scope of the CI controller*. [Online]. Loetud aadressil: <https://subscription.packtpub.com/book/all-books/9781849694704/3/ch03lvl1sec20/scope-of-the-ci-controller> Kasutatud: 26.03.2022
- [9] Swagger, *SwaggerHub*. [Online]. Loetud aadressil: <https://swagger.io/tools/swaggerhub/> Kasutatud: 28.01.2022
- [10] Swagger, *OpenAPI Specification*. [Online]. Loetud aadressil: <https://swagger.io/specification/> Kasutatud: 28.01.2022
- [11] Postman, *What is Postman?*. [Online]. Loetud aadressil: <https://www.postman.com/> Kasutatud: 28.01.2022
- [12] Diagrams.net, *Diagrams.net Integrations*. [Online]. Loetud aadressil: <https://www.diagrams.net/> Kasutatud: 28.01.2022
- [13] Visual Studio (version 17.1), Microsoft. [Online]. Loetud aadressil: <https://visualstudio.microsoft.com/> Kasutatud: 28.01.2022
- [14] General Assembly, *What is a Javascript framework?*, 2021. [Online]. Loetud aadressil: <https://generalassemb.ly/blog/what-is-a-javascript-framework/> Kasutatud: 05.02.2022
- [15] ACM DL Digital Library, *JavaScript: the first 20 years*, 2020. [Online]. Loetud aadressil: <https://dl.acm.org/doi/10.1145/3386327> Kasutatud: 05.02.2022
- [16] Mdn web docs, *What is JavaScript?*, 2022. [Online]. Loetud aadressil: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript Kasutatud: 05.02.2022
- [17] Ars Technica, *Microsoft TypeScript: the JavaScript we need, or a solution looking for a problem?*. [Online]. Loetud aadressil: <https://arstechnica.com/information-technology/2012/10/microsoft-typescript-the-javascript-we-need-or-a-solution-looking-for-a-problem/> Kasutatud: 05.02.2022

- [18] TypeScript, *What is TypeScript?*. [Online]. Loetud addressil: <https://www.typescriptlang.org/> Kasutatud: 05.02.2022
- [19] Technostacks, *Best JavaScript Frameworks For Robust Web Development*, 2021. [Online]. Loetud addressil: <https://technostacks.com/blog/best-javascript-frameworks> Kasutatud: 05.02.2022
- [20] React, *React*. [Online]. Loetud addressil: <https://reactjs.org/> Kasutatud: 05.02.2022
- [21] Angular, *Introduction to Angular concepts*, 2022. [Online]. Loetud addressil: <https://angular.io/guide/architecture> Kasutatud: 05.02.2022
- [22] Vue.js, *What is Vue?*. [Online]. Loetud addressil: <https://vuejs.org/v2/guide/> Kasutatud: 05.05.2022
- [23] Codeinwp, *Angular vs React vs Vue: Which Framework to Choose*, 2021. [Online]. Loetud addressil: <https://www.codeinwp.com/blog/angular-vs-vue-vs-react> Kasutatud: 05.02.2022
- [24] Brainhub, *Vue vs React for Team Leaders*, 2021. [Online]. Loetud addressil: <https://brainhub.eu/library/vue-vs-react/> Kasutatud: 05.02.2022
- [25] Codecademy, *MVC: Model, View, Controller*. [Online]. Loetud addressil: <https://www.codecademy.com/article/mvc> Kasutatud: 06.02.2022
- [26] Clarion Technologies, *MVC vs Flux vs Redux – The Real Differences*. [Online]. Loetud addressil: <https://www.clariontech.com/blog/mvc-vs-flux-vs-redux-the-real-differences> Kasutatud: 06.02.2022
- [27] Medium, *Flux vs MVC Design Pattern*, 2018. [Online]. Loetud addressil: <https://madasamy.medium.com/flux-vs-mvc-design-pattern-de134dfaa12b> Kasutatud: 20.03.2022
- [28] Merixstudio, *Vuex vs. Redux - similarities and differences*, 2020. [Online]. Loetud addressil: <https://www.merixstudio.com/blog/vuex-vs-redux/> Kasutatud: 06.02.2022
- [29] Merixstudio, *What is a REST API?*, 2021. [Online]. Loetud addressil: <https://stallionemma88.medium.com/what-is-a-rest-api-a8a382975378> Kasutatud: 26.03.2022
- [30] Axios, *What is Axios?*. [Online]. Loetud addressil: 26.03.2022
<https://github.com/axios/axios> Kasutatud: 26.03.2022
- [31] Vuex, *What is Vuex?*. [Online]. Loetud addressil: <https://vuex.vuejs.org/> Kasutatud: 20.03.2022
- [32] Vue Router, *Vue Router*. [Online]. Loetud addressil: <https://router.vuejs.org/guide/> Kasutatud: 26.03.2022
- [33] Studio, *Design Consistency Guide: Best Practices for UI and UX Designers*. [Online]. Loetud addressil: <https://www.uxpin.com/studio/blog/guide-design-consistency-best-practices-ui-ux-designers/> Kasutatud: 26.03.2022

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Triin Mõlder

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

24.04.2022

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

Lisa 2 – Süsteemi kasutajaõigused

Kasutaja	Töölaud		Kasutajagrupp
	Vaatamine	Kuupäevade muutmine	
Kliendi administraator	jah	jah	Administrator
Reisi tellija	jah	jah	Manager
Reisija	ei	ei	User
Kliendi raamatupidaja	jah	jah	Manager
Estraveli administraator	jah	jah	Manager
Estraveli kliendihaldur	jah	jah	Administrator

Kasutaja	Reisijate raport (arve kp alusel)		Reisijate raport (teenuse kp alusel)		Reiside ülevaade		Kasutajagrupp
	Enda andmed	Teiste andmed	Enda andmed	Teiste andmed	Enda andmed	Teiste andmed	
Kliendi administraator	jah	jah	jah	jah	jah	jah	Administrator
Reisi tellija	jah	jah	jah	jah	jah	jah	Manager
Reisija	jah	ei	jah	ei	jah	ei	User
Kliendi raamatupidaja	jah	jah	jah	jah	jah	jah	Manager
Estraveli administraator	jah	jah	jah	jah	jah	jah	Administrator
Estraveli kliendihaldur	jah	jah	jah	jah	jah	jah	Manager

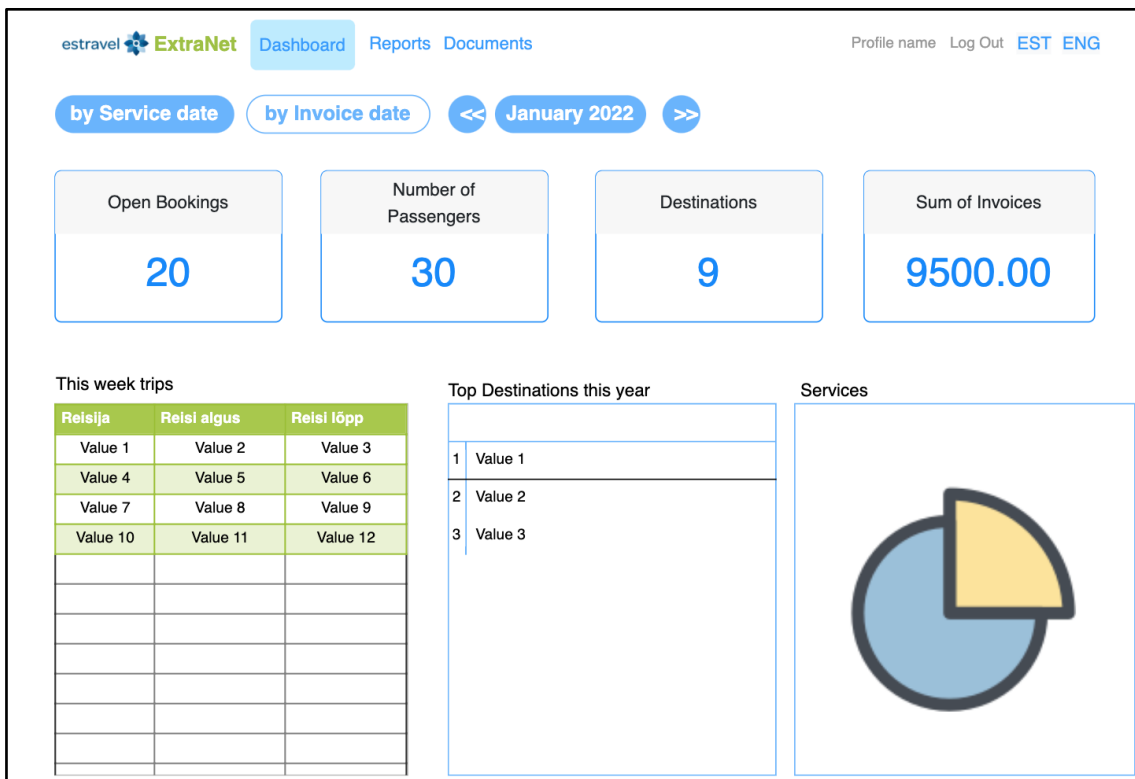
Kasutaja	Raportid						Kasutajagrupp
	Arve raport		Teenuse raport (arve kp alusel)		Teenuse raport (teenuse kp alusel)		
	Enda andmed	Teiste andmed	Enda andmed	Teiste andmed	Enda andmed	Teiste andmed	
Kliendi administraator	jah	jah	jah	jah	jah	jah	Administrator
Reisi tellija	jah	jah	jah	jah	jah	jah	Manager
Reisija	jah	ei	jah	ei	jah	ei	User
Kliendi raamatupidaja	jah	jah	jah	jah	jah	jah	Manager
Estraveli administraator	jah	jah	jah	jah	jah	jah	Administrator
Estraveli kliendihaldur	jah	jah	jah	jah	jah	jah	Manager

Kasutaja	Dokumendid		Kasutajagrupp
	Enda andmed	Teiste andmed	
Kliendi administraator	jah	jah	Administrator
Reisi tellija	jah	jah	Manager
Reisija	jah	ei	User
Kliendi raamatupidaja	jah	jah	Manager
Estraveli administraator	jah	jah	Administrator
Estraveli kliendihaldur	jah	jah	Manager

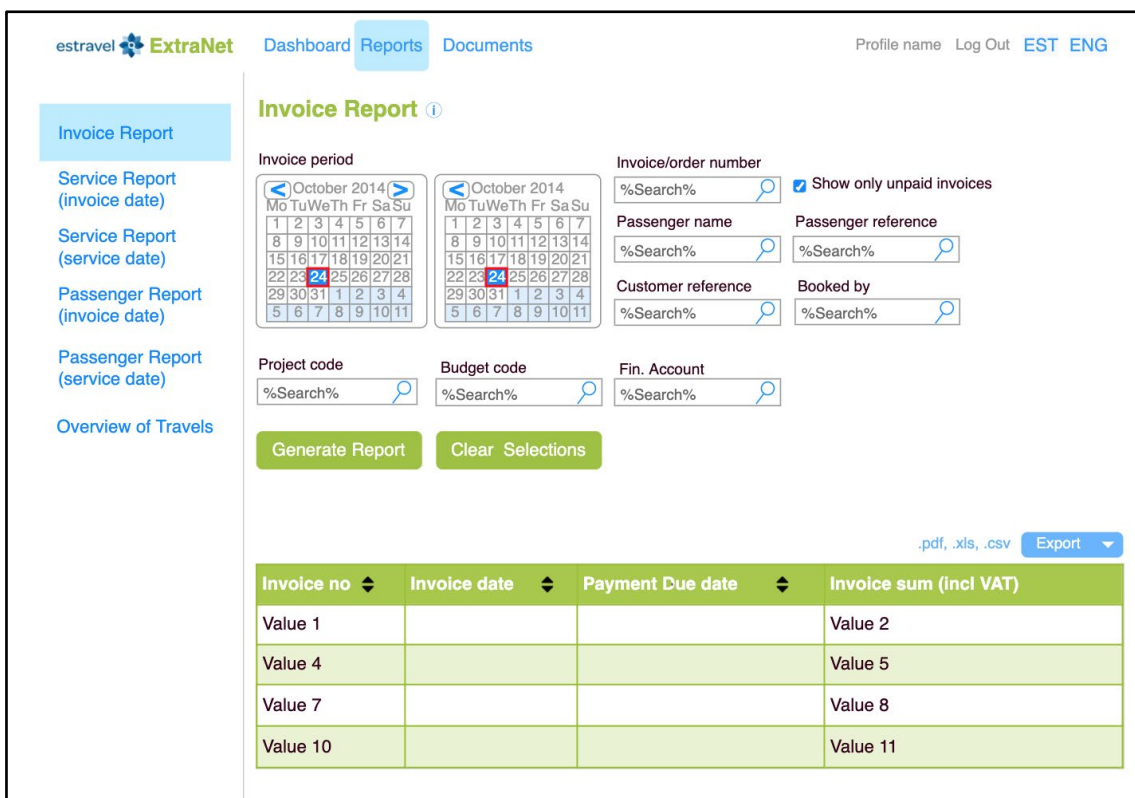
Kasutaja	Kasutajate lisamine	Kasutajagrupp
Kliendi administraator	jah	Administrator
Reisi tellija	ei	Manager
Reisija	ei	User
Kliendi raamatupidaja	ei	Manager
Estraveli administraator	jah	Administrator
Estraveli kliendihaldur	jah	Administrator

Joonis 17. Süsteemi kasutajaõigused

Lisa 3 – Kasutajaliidese vaated



Joonis 18. Töölaua kasutaja vaade



Joonis 19. Kasutajaliidese visand - Arvete raporti vaade

estavel **ExtraNet** Dashboard Reports Documents Profile name Log Out EST ENG

Invoice Report

Service report by invoice date

Invoice period:

Destination country:

Destination city:

Service Provider:

Service group: Select All Air Air

Sales Channels: Offline Online

Budget code: %Search%

Project code: %Search%

Customer reference: %Search%

Fin. Account: %Search%

[Generate Report](#) [Clear Selections](#)

[View details](#) [.pdf](#) [.xls](#) [.csv](#) [Export](#)

Service	Invoice sum (incl VAT)
Value 1	Value 2
Value 4	Value 5
Value 7	Value 8
Value 10	Value 11

[Details view](#) [.pdf](#) [.xls](#) [.csv](#) [Export](#)

Invoice no	Service date	Departure	Destination	Passenger name	Service (service group, product, description)	Qty	Sum (incl VAT)	Service Provider	Travel Consultant	Booked by	Sales chan
Value 1			Value 2	Value 3							
Value 4			Value 5	Value 6							
Value 7			Value 8	Value 9							
Value 10			Value 11	Value 12							

Joonis 20. Kasutajaliidese visand - Teenuse raporti arve kuupäeva põhine vaade

[estravel](#) [ExtraNet](#)

[Profile name](#) [Log Out](#) [EST](#) [ENG](#)

[Dashboard](#) [Reports](#) [Documents](#)

Service report by service date

[Invoice Report](#)
[Service Report \(invoice date\)](#)
[Service Report \(service date\)](#)
[Passenger Report \(invoice date\)](#)
[Passenger Report \(service date\)](#)
[Overview of Travels](#)

Invoice period:

Destination country: **Service group:** Select All Air Air

Destination city: **Sales Channels:** Offline Online

Service Provider: **Budget code:** **Fin. Account:**

Customer reference: **Project code:** **Generate Report** **Clear Selections**

View details [.pdf](#) [.xls](#) [.csv](#) [Export](#)

Service	Invoice sum (incl VAT)
Value 1	Value 2
Value 4	Value 5
Value 7	Value 8
Value 10	Value 11

Details view [.pdf](#) [.xls](#) [.csv](#) [Export](#)

Invoice no	Service date	Departure	Destination	Passenger name	Service (service group, product, description)	Qty	Sum (incl VAT)	Service Provider	Travel Consultant	Booked by	Sales chan
Value 1			Value 2	Value 3							
Value 4			Value 5	Value 6							
Value 7			Value 8	Value 9							
Value 10			Value 11	Value 12							

Joonis 21. Kasutajaliidese visand - Teenuse raporti teenuse kuupäeva põhine vaade

estavel **ExtraNet** Dashboard **Reports** Documents Profile name Log Out EST ENG

Invoice Report

Service Report (invoice date)

Service Report (service date)

Passenger Report (invoice date)

Passenger Report (service date)

Overview of Travels

Passenger report by invoice date ¹

Invoice period

October 2014 Text

Mo	Tu	We	Th	Fr	Sa	Su
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	6	7	8	9	10	11

October 2014

Mo	Tu	We	Th	Fr	Sa	Su
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	6	7	8	9	10	11

Service group

Select All

Air

Air

Sales Channels

Offline Online

Passenger name

Passenger reference no

Booker name

Customer reference

Project code

Budget code

Fin. Account

.pdf, .xls, .csv

Passenger name	Invoice sum (incl VAT)
Value 1	Value 2
Value 4	Value 5
Value 7	Value 8
Value 10	Value 11

Details view .pdf, .xls, .csv

Passenger name	Passenger reference no	Invoice no	Invoice sum (incl VAT)	Booked by
Value 1			Value 2	Value 3
Value 4			Value 5	Value 6
Value 7			Value 8	Value 9
Value 10			Value 11	Value 12

Joonis 22. Kasutajaliidese visand - Reisijate raporti arve kuupäeva põhine vaade

estavel **ExtraNet** Dashboard **Reports** Documents Profile name Log Out EST ENG

Invoice Report

Service Report (invoice date)

Service Report (service date)

Passenger Report (invoice date)

Passenger Report (service date)

Overview of Travels

Passenger report by Service date ?

Invoice period

October 2014

Mo	Tu	We	Th	Fr	Sa	Su
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	6	7	8	9	10	11

October 2014

Mo	Tu	We	Th	Fr	Sa	Su
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	6	7	8	9	10	11

Service group

Select All

Air

Air

Sales Channels

Offline Online

Passenger name

Passenger reference no

Booker name

Customer reference

Project code

Budget code

Fin. Account

Generate Report **Clear Selections**


View details .pdf, .xls, .csv **Export**

Passenger name	Invoice sum (incl VAT)
Value 1	Value 2
Value 4	Value 5
Value 7	Value 8
Value 10	Value 11

Details view

Passenger name	Passenger reference no	Invoice no	Invoice sum (Incl VAT)	Booked by
Value 1			Value 2	Value 3
Value 4			Value 5	Value 6
Value 7			Value 8	Value 9
Value 10			Value 11	Value 12

Joonis 23. Kasutajaliidese visand - Reisijate raporti teenuse kuupäeva põhine vaade

estavel  **ExtraNet** [Dashboard](#) [Reports](#) [Documents](#) Profile name Log Out [EST](#) [ENG](#)

[Invoice Report](#)

[Service Report \(invoice date\)](#)

[Service Report \(service date\)](#)

[Passenger Report \(invoice date\)](#)

[Passenger Report \(service date\)](#)

[Overview of Travels](#)

Overview of travels ①

Service date

October 2014

Mo	Tu	We	Th	Fr	Sa	Su
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	6	7	8	9	10	11

October 2014

Mo	Tu	We	Th	Fr	Sa	Su
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	6	7	8	9	10	11

Destination country

Small select ▼

Service group

Select All

Air

Air

Ticket number

🔍

Passenger name

🔍

Passenger reference no

🔍

Booker name

🔍

Customer reference

🔍

Project code

🔍

Budget code

🔍

Fin. Account

🔍

Generate Report

Clear Selections

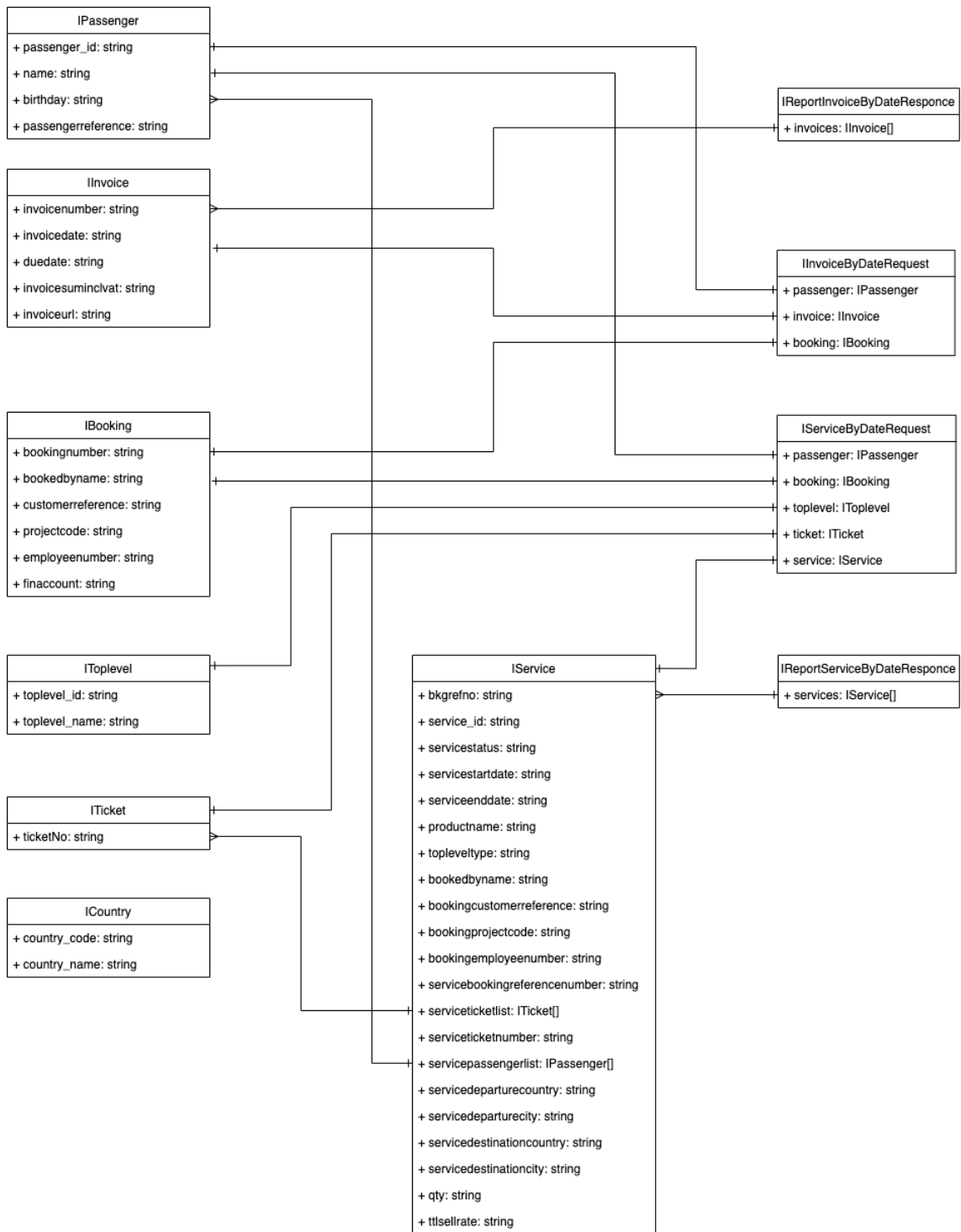
.pdf, .xls, .csv

Export ▼

Service start date	Service end date	Passenger name	Destination	Service	Documents
Value 2	Value 3				
Value 5	Value 6				
Value 8	Value 9				
Value 11	Value 12				


Joonis 24. Kasutajaliidese visand - Reaside ülevaate raporti vaade

Lisa 4 – Süsteemi andmemudel



Joonis 25. Süsteemi andmemudel

Lisa 5 – Arendatud süsteemi vaated



[Invoice Report](#)
[Overview of Travels](#)

[Dashboard](#)
[Reports](#)
[Documents](#)

[TrinTest](#)
[Logout](#)
[EST](#)
[ENG](#)

Invoice Report

Invoice Period Start*

Invoice Period End*

Invoice/Order Number

Payment Status

More parameters ▾

Generate Report
Clear Selection

Download

Invoice no	Invoice date	Payment Due date	Invoice sum (incl VAT)
I2275229-01	2022-02-11	2022-02-11	381.8
I2275226-01	2022-02-07	2022-02-07	201.8
I2275225-01	2022-02-07	2022-02-07	131.8

Joonis 26. Loodud kasutajaliides - Arvete raporti vaade

Overview of Travels

Invoice Report
[Overview of Travels](#)

Service Startdate: Service Enddate: Destination Country:

Service Group: Select all, Air, Car, Cruise, Ferry, Hotel, Insurance, City, Other, Rail, Package, TO package, Visa

More parameters
 Passenger Name: Employee Code:
 Ticket Number: Project Code:

Passenger	Service startdate	Service enddate	Departure	Destination	Service group	Service	Qty	Sum	Booked by	Employee No.	Project No.	Ticket No.	Booking No.
TRIIN MÖLDER	2022-02-14	2022-02-14	Estonia, Tallinn	Finland, Helsinki	Air	Flight ticket	1	140	test			67789876543234567	2275232
TRIIN MÖLDER	2022-02-14	2022-02-14	Estonia, Tallinn	Finland, Helsinki	Air	Flight ticket	1	120	test			101-000000	2275231
TRIIN MÖLDER	2022-02-14	2022-02-14	Estonia, Tallinn	Finland, Helsinki	Air	Flight ticket	1	120	test			101-000000	2275230
TRIIN MÖLDER	2022-02-16	2022-02-16	Estonia, Tallinn	Poland, Warsaw	Air	Flight ticket	1	164.99	Mati Karu			112-124567890	2275227

Joonis 27. Loodud kasutajaliides – Reaside ülevaate raporti vaade