

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Angelina Polštšikova 1644711ABB

**LOGISTIKAETTEVÖTTE  
TARNETEEKONDADE PIKKUSTE  
ARVUTAMISE SÜSTEEMI ANALÜÜS**

Bakalaureusetöö

Juhendaja: Erki Eessaar  
PhD

Tallinn 2019

## **Autorideklaratsioon**

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

Autor: Angelina Polštšikova

06.01.2020

## **Annotatsioon**

Käesoleva lõputöö eesmärgiks on analüüsida ja osaliselt ka loogiliselt disainida logistikaettevõtte jaoks loodava tarneteekondade pikkuste arvutamise süsteemi.

Nimetatud süsteemi ülesandeks on abistada maanteetranspordi abil kaupade tarnimist, arvutades kõik kauba tarnimise marsruudi ja kulude kalkuleerimiseks vajalikud geograafiliste asukohtade vahelised teekonnapikkused, kontrollides eelnevalt neile asukohtadele vastavate aadresside korrektsust. Selline tarkvara peab olema integreeritud olemasolevate süsteemidega ning on mõeldud logistikaettevõttes sisemiseks kasutamiseks.

Töö tulemusena valmis funktsionaalsete nõuete esitus kasutuslugudena ning mittefunktsionaalsete nõuete kirjeldus. Autor koostas süsteemi tööd kirjeldavad tegevusdiagrammid ja põhiolemitüübi seisundidiagrammi, kontseptuaalse ja loogilise disaini andmemudelid ning kasutajaliidese prototüübi. Samuti põhjendati analüüsi käigus tehtud otsuseid ning mõnede otsuste puhul selgitati, miks alternatiivide hulgast tehti just mainitud valik.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 38 leheküljel, 7 peatükki, 23 joonist, 2 tabelit.

## **Abstract**

### **Analysis of a Shipping Distance Calculation System for a Logistics Company**

The aim of the thesis is to analyze and partially logically design the Shipping Distance Calculation System for a logistics company.

The main purpose of the system is to support overland shipping by calculating all distances between geographic points that are needed for the shipping route and cost calculation. Before calculation, the system has to check the correctness of the addresses that correspond to the geographic points. The software should be integrated with existing systems and is meant for internal usage in the logistics company.

The main result of the work includes functional requirements representation by using user stories, non-functional requirements specification, activity diagrams that describe the system workflows, state machine of the main entity type, conceptual and logical design data models, and the prototype of the user interface. Decisions made during the analysis are explained and, in some cases, compared with alternatives.

The thesis is in Estonian and contains 38 pages of text, 7 chapters, 23 figures, 2 tables.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides. See määrab reeglistiku, mida kasutades saavad teised programmid kasutada rakendusliidest pakkuva programmi teenuseid.
CBM	<i>Cubic Meter</i> , kuupmeeter, kauba mahu mõõtühik.
LDM	<i>Loading Meter</i> , laadimismeeter, jooksev meeter haagise põrandapinnast täislaiuses ja -kõrguses.
PTV Group	<i>Planung Transport Verkehr AG</i> , Saksamaa ettevõtte, mis spetsialiseerub tarkvaralahendustele ja konsultatsiooniteenustele liikluse, transpordi, mobiilsuse ja logistika valdkondades.
SQL	<i>Structured Query Language</i> , relatsioonilisel mudelil põhinev populaarne andmebaasikeel, mille aluseks olev andmemudelil põhinevat andmebaasi nimetatakse SQL-andmebaasiks. Selles andmebaasikeeles saab kirjutada lauseid nii andmete käitlemiseks, andmebaasi struktuuri ja käitumise kirjeldamiseks, õiguste haldamiseks kui ka andmebaasi kasutamise tehingute juhtimiseks.
UML	<i>Unified Modelling Language</i> , unifitseeritud modelleerimiskeel. Üldotstarbeline visuaalne modelleerimiskeel, mida kasutatakse eeskätt info- ja tarkvarasüsteemide projekteerimiseks.
CASE	<i>Computer-aided software engineering software</i> , tarkvara tarkvarasüsteemide mudelipõhiseks arendamiseks ja selle arenduse automatiseerimiseks.

## Sisukord

1 Sissejuhatus .....	9
2 Metoodika.....	11
2.1 Lõputöö tegemise metoodika.....	11
2.2 Arendusmetoodika.....	12
2.2.1 Arendusprotsessi kirjeldus.....	13
2.2.2 Erinevused klassikalisest Scrumist.....	14
2.2.3 Autori tööpanus projektis .....	15
3 Funktsionaalsed ja mittefunktsionaalsed nõuded süsteemile .....	16
3.1 Mittefunktsionaalsed nõuded.....	16
3.2 Funktsionaalsete nõuete kirjeldus kasutuslugudena .....	17
3.3 Olulised otsused.....	22
4 Detailanalüüs .....	25
4.1 Protsessid.....	25
4.1.1 Üldprotsess .....	26
4.1.2 Automaatne töötlemine .....	27
4.1.3 Seostamine geograafilise asukohaga PTV abil.....	28
4.1.4 Käsitsi parandamine .....	29
4.1.5 Esialgse ja parandatud aadressi versioonide töötlemine .....	30
4.2 Aadressi seisundidiagramm .....	32
4.3 Kontseptuaalne andmemudel.....	33
5 Disain.....	36
5.1 Loogilise disaini andmemudel.....	36
5.2 Kasutajaliidese prototüüp .....	37
6 Olulisemad järeldused .....	44
7 Kokkuvõte .....	46
Kasutatud kirjandus .....	47
Lisa 1 – Jõudluse testimise tulemused.....	49

## Jooniste loetelu

Joonis 1. Värvide kasutamine tegevusdiagrammidel. ....	26
Joonis 2. Üldprotsess. ....	26
Joonis 3. Automaatse töötlemise alamprotsess. ....	27
Joonis 4. PTV abil geograafilise asukohaga seostamise alamprotsess. ....	28
Joonis 5. Käsitsi parandamise alamprotsess. ....	29
Joonis 6. Parandatud aadressi töötlemise alamprotsess. ....	30
Joonis 7. Esialgse aadressi töötlemise alamprotsess. ....	30
Joonis 8. Aadressi seisundidiagramm. ....	32
Joonis 9. Kontseptuaalne andmemudel. ....	33
Joonis 10. Loogilise disaini andmemudel. ....	36
Joonis 11. Sisselogimisvaade. ....	37
Joonis 12. Parandamist vajavate aadresside nimekirja vaade. ....	38
Joonis 13. Redigeerimisvaade. ....	38
Joonis 14. Geograafilise asukoha valimise vaade. ....	39
Joonis 15. Geograafilise asukoha valiku kinnitamise vaade. ....	39
Joonis 16. Eduka aadressi geograafilise asukohaga seostamise vaade (variant 1). ....	40
Joonis 17. Eduka aadressi geograafilise asukohaga seostamise vaade (variant 2). ....	40
Joonis 18. Ebaõnnestunud geograafilise asukohaga seostamise vaade. ....	41
Joonis 19. Kehtetuks aadressi märgistamise kinnitamise vaade. ....	41
Joonis 20. Eduka kehtetuks aadressi märgistamise vaade. ....	42
Joonis 21. Kehtetu aadresside nimekirja vaade. ....	42
Joonis 22. Aadressi taastamise kinnitamise vaade. ....	43
Joonis 23. Eduka aadressi taastamise vaade. ....	43

## **Tabelite loetelu**

Tabel 1. Olemitüüpide sõnalised kirjeldused. ....	33
Tabel 2. Atribuutide sõnalised kirjeldused. ....	34



# 1 Sissejuhatus

Bakalaureusetöö teemaks on analüüsida logistikaettevõtte tarneteekondade pikkuste arvutamise süsteemi, kogudes selle kohta nõudeid, koostades süsteemi detailset toimimist kirjeldavaid mudeleid ning jõudes välja esmase andmebaasi disaini ning prototüübini. Autori ülesandeks IT ettevõttes analüütikuna töötades on seda süsteemi analüüsida ning ta on tarkvara loova arendustiimi liige. Antud projekti tellijaks on rahvusvaheline logistikaettevõtte. Konfidentsiaalsuse säilitamiseks ei mainita lõputöös selle ettevõtte nime.

Analüüsitava süsteemi ülesandeks on abistada maanteetranspordi abil kaupade tarnimist. Kaubad tarnitakse kaubasaatjalt kauba saajani veoautodega ja tihti niimoodi, et üks ja sama veoauto tarnib mitme saatja kaubad korraga ehk teostab osakoorma vedu. Selleks, et teha igale kliendile korrektne hinnapakumine, peab tarneteenust pakkuv logistikaettevõtte arvutama tarnimiskulud. Praeguse süsteemi järgi arvestatakse osakoorma veo puhul transpordikulud vastavalt kauba kaalu või mahu näitajale, ehk kasutades LDM või CBM laadimisühikuid. Paraku ei arvesta aga antud arvutusmeetod tarnimismarsruutide pikkuseid. See tähendab, et kui näiteks ühe tellimuse mahalaadimise koht asub veo alguspunktile lähedamal ja tellitud kaup võtab seitse LDMi veoauto laadimispinnast ning teise tellimuse mahalaadimise koht asub veo alguspunktist palju kaugemal ja võtab viis LDMi, siis peaks esimese tarne tellija maksma tarne eest rohkem, sest LDMi väärtus on suurem. Tarnete erinevad teekonnapiikkused ei mängi aga sellisel hinnal arvutamisel mingit rolli. Selline arvutusmeetod on objektiivselt vale. Protsessi teeb keerulisemaks ka see, et reaalsuses tarnitakse väga harva kaup saatjalt otse kauba saajani. Enamasti liigub kaup läbi erinevate depoode.

Antud probleemi lahendamiseks luuakse rakendus – tarneteekondade pikkuste kalkulaator, mis võimaldaks arvutada kogu kauba tarnimise marsruudi ja kulude kalkuleerimiseks vajalikud geograafiliste asukohtade vahelised teekonnapiikkused, kontrollides eelnevalt neile asukohtadele vastavate aadresside korrektsust. Selline tarkvara peab olema integreeritud olemasolevate süsteemidega ning on mõeldud

logistikaettevõtte sisemiseks kasutamiseks. Tulevikus plaanib ettevõtte ehitada uue kulude arvutamise ja aruannete koostamise süsteemi, mis võtaks vajalikud teekonnapikkused otse antud lõputöös kirjeldatava tarneteekondade pikkuste kalkulaatori andmebaasist.

## 2 Metoodika

See peatükk kirjeldab lõputöö metoodikat ning arendusmetoodikat, mida arendustiim kasutab oma töö tegemiseks. Kuna tööd tehakse rühmatööna ja autor on rühma üks liige, siis peatüki viimases alamosas on välja toodud, milline täpselt oli tema tööpanus kogu selles projektis. Arendus toimub projektina, kuid peale selle lõppu on vajalik tarkvara edasine hooldamine.

### 2.1 Lõputöö tegemise metoodika

Valitud metoodikaks on disaini tegevusuuring (ingl *Action Design Research*) [1]. Samuti lähtutakse töös kontekstipõhise uuringu (ingl *Context-driven Research*) ja koostööl põhineva uuringu (ingl *Colaborative Research*) põhimõtetest [2]. Arenduse jooksul suhtleb lõputöö autor pidevalt teiste arendustiimi liikmete, rüsinameistri, tarkvara tellija ning tarkvara võimalike kasutajatega. Saades nendelt tagasisidet, täiendab ta vastavalt analüüsi, võrdleb alternatiive ning koostöös arendustiimiga korrigeerib loodavat lahendust. Samuti kasutab autor nõuete leidmiseks ettevõtte sisemist dokumentatsiooni [3]. Disainiotsuste vastuvõtmisel arvestab autor tellija ettevõtte vajaduste ja integreeritavate süsteemide omapäradega, mis määravad konteksti uue lahenduse loomiseks. Disaini tegevusuuringu viimase faasina kirjeldab autor olulisemaid järeldusi, milleni ta seda projekti tehes jõudis.

Agiilsete arendusmetoodikate põhimõtetest lähtudes kirjeldatakse funktsionaalsed nõuded süsteemile kasutuslugudena [4] – [6], mis annavad nõuete esitamiseks vajalikku paindlikust.

Nõuete analüüsimisprotsessil lähtutakse Adžic'i [6] väljatoodud printsiipidest:

- lahenduse skoobi tuletamine reaalistest eesmärkidest ja vajadustest,
- nõuete täpsustamine koostöös teiste osapooltega,
- illustreerivate näidete kasutamine nõuete täpsustustes,
- täiendatud nõuete viimistlemine,

- arendatava süsteemi pidev valideerimine,
- ajakohase „elava“ dokumentatsiooni pidev täiendamine.

Süsteemi nõuete, oodatava struktuuri ja käitumise visualiseerimiseks kliendile ja arendustiimile otsustas lõputöö autor kasutada UML-i [7], [8]. UML-mudelite koostamiseks valis autor CASE-vahendina tarkvara Enterprise Architect [9], sest see toetab kõiki projekti jaoks vajalikke mudelite tüüpe. Samuti on autoril palju kogemusi antud tarkvara kasutamisega.

Analüüsi käigus uuritakse tarnearuannete süsteemi andmebaasis olevaid aadresse, mida hakatakse saatma loodava teekonnapikkuste kalkulaatorile. Selle ülesande täitmiseks andmebaasist andmete lugemiseks valis autor Oracle SQL Developer tarkvara [10]. Valiku esimene põhjendus on, et andmebaas on tehtud Oracle andmebaasisüsteemis. Nimetatud vahendit arendab Oracle ise, see on tasuta ja võimekas ning autoril on selle kasutamise kogemus. Kasutajaliidese visualiseerimiseks luuakse prototüüp Figma [11] tarkvaraga. Autor valis antud tarkvara, sest sellega saab kiiresti ja mugavalt joonistada realistlikke kasutajaliidese vaateid ning teha prototüübi dünaamiliseks – elementide peale vajutades saab liikuda vaadete vahel nagu päris veebilehel.

## **2.2 Arendusmetoodika**

Tarneteekondade pikkuste arvutamise süsteemi loomise projekt on väga dünaamiline, pidevalt muutuva skoobi ja nõuetega. Sellest lähtudes valiti arendusmetoodikaks Scrum [12], mis võimaldab kohaneda muudatustega käigu pealt. Antud valik tehti projekti arendustiimi ja rüsinameistri poolt, arvestades ka asjaolu, et käesolevas IT-ettevõttes on Scrum laialdaselt kasutatud ja arendustiimi liikmed on selle metoodikaga harjunud. Alternatiivina kaaluti teist agiilset arendusmetoodikat Kanban [13], kuid valiti siiski Scrum, sest tänu oma struktuurile eeldab see tihedamat suhtlemist arendustiimi liikmete vahel, mis suurendab projekti läbipaistvust ja tõhustab koostööd. Scrum on paindmetoodika, mida iseloomustab tarkvara arendamine iteratiivselt (arenduse pisitsükleid kasutades) ja iga iteratsiooniga täienduste või uue osa lisamine. Scrum raamistik aitab efektiivselt lahendada keerulisi ja adaptiivseid tarkvaraarenduse probleeme ning keskendub projektijuhtimisele, kus on raske plaane pikaks ajaks ette teha [12]. Arvestades käesoleva IT-ettevõtte struktuuri, kliendi soovide ja projekti omapäradega, esinevad kasutatavas arendusmetoodikas „klassikalisest“ Scrumist mõned

erinevused, mida tutvustatakse jaotises 2.2.2. Kuna Scrumi teooria on enamasti ingliskeelne, on antud töö parema loetavuse eesmärgil mõisted tõlgitud eesti keelde, tuginedes uurimis- ja arendusfirma Cybernetica poolt väljatöötatud standardipõhise tarkvaratehnika sõnastikule [14].

Käesoleva projektis esinevad järgmised raamistiku ette nähtud rollid.

Tooteomanik (ingl *Product Owner*) [12] – see roll kuulub projekti eest vastutavale logistikaettevõtte esindajale, kes tunneb väga hästi oma ettevõtte vajadusi loodava lahenduse suhtes ning on küsimuste korral esmane infoallikas.

Rüsinameister (ingl *Scrum Master*) [12] – rüsinameistri ülesanded täidab töö teostava IT-ettevõtte sisemine projektijuht. Vajadusel saab selle rolli projektijuhi asemel osaliselt endale võtta ka analüütik.

Arendustiimi liige (ingl *Development Team Member*) [12] – projekti arendustiim koosneb arhitektist, arendajast ja analüütikust. Kuigi igal arendustiimi liikmel on oma eriala, mis määrab projekti valdkonna, millesse ta kõige rohkem panustab, vastutab projekti edukuse eest kogu arendustiim tervikuna, mitte vastutust osadeks jaotades. Antud töö autor on arendustiimi liige kuid ühtlasi on talle antud analüütiku vastutus, mille täpsemat sisu tutvustatakse jaotises 2.2.3.

### **2.2.1 Arendusprotsessi kirjeldus**

Analüütik kogub nõuded süsteemile, lisab need projekti tööjärge eeposte (ingl *Epics*) [15] ja kasutuslugudena (ingl *User Stories*) [4] – [6] ning viib nende põhjal läbi detailanalüüsi. Seejärel annavad arendajad iga kasutusloo jaoks ajahinnangu plaanispokkeri (ingl *Planning Poker*) [16] meetodil ning vajadusel tükeldavad kasutuslood väiksemateks osadeks ehk ülesanneteks.

Arendusprotsess põhineb kahenädalastel sprintidel [12]. Lõputöö esitamise hetkeks oli läbi viidud kuus sprinti. Iga sprinti alguses toimub plaaniskoosolek (ingl *Planning Meeting*) [12], mida mugavuse pärast ühendatakse tavaliselt eelmise sprinti läbivaatusega (ingl *Sprint Review*) [12]. Esialgu arutatakse eelmise sprinti tulemusi ning arendustiim demonstreerib arendatud süsteemi osa tooteomanikele. Kui eelmine sprint on lõpetatud, arutatakse tööjärjes olevaid kasutuslugusid. Tooteomanik seab kasutuslood tähtsuse järjekorda ning seejärel otsustatakse, millised kasutuslood

võetakse uude sprinti. Kui tööjärjes esinevad tarkvaradefektid (ingl *Bugs*), siis tegeletakse nendega tavaliselt esimeses järjekorras. Arendusprotsessi käigus viiakse iga tööpäeva alguses läbi lühike püstijalakoosolek (ingl *Stand-up Meeting*) [12], mille käigus selgitab iga arendustiimi liige oma töö edenemist.

Sprindi jooksul peab iga arendusülesanne läbima viis etappi [3]:

- Tegemata (ingl *To-do*) – arendajad valivad endale ülesanded.
- Tegemisel (ingl *In Progress*) – arendus käib.
- Koodi läbivaatus (ingl *Code Review*) – arendaja poolt valmis kirjutatud kood vaadatakse läbi arendustiimi teise arendaja poolt, vajadusel seda parandades.
- Vastuvõtmine (ingl *Acceptance*) – arendatud süsteemi osa testitakse ning veendutakse, et see vastab vastuvõtukriteeriumitele.
- Valmis (ingl *Done*) – kood on mestitud *master* harusse ning on valmis juurutamiseks.

Iga sprindi lõpus viiakse läbi tagasivaate koosolek (ingl *Retrospective*) [12], et arutada ja analüüsida eelmise sprindi tulemusi. Tagasivaate koosolekutel kasutatakse enamasti purjepaadi (ingl *Sailboat*) tehnikat [17].

### **2.2.2 Erinevused klassikalisest Scrumist**

Klassikaline Scrum ei erista eri rolle arendustiimi sees, vaid eristab ainult raamistikus ettenähtud tootemaniku, rüsinameistri ning arendustiimi liikme rolle [12]. Käesoleva projekti puhul on iga osaleja oma eriala spetsialist (nt analüütik, arhitekt, arendaja jne), kes võtab endale projekti raames Scrumis ettenähtud rolli, kuid kannab ühtlasi ka oma spetsialisti rolli. Seega tööprotsessis keskenduvad osalejad oma eriala ülesannetele. Antud lähenemine on tingitud käesoleva ettevõtte struktuurist ja asjaolust, et üks spetsialist võib olla kaasatud samal ajal mitmetesse projektidesse.

Teine kõrvalekalle klassikalisest Scrumist seisneb selles, et käesolevas projektis kuulub analüütiku ülesannete hulka lahendust kirjeldavate detailsete mudelite koostamine ja ülesjoonistamine CASE vahendiga, mis pole nii detailselt ja suures mahus tavaliselt Scrumile iseloomulik. Antud lähenemisel on mitu põhjust. Esiteks peetakse kliendi

ettevõttes väga oluliseks korraliku ja detailse dokumentatsiooni koostamist. See võimaldab pikas plaanis arendusprotsessi uusi inimesi lihtsalt kaasata ning vajadusel loodud lahendust taaskasutada. Teiseks ei kirjelda tööjärjes olevad kasutuslood süsteemi nõudeid piisavalt detailselt, mudelid aga täpsustavad puuduvaid detaile. Lisaks on mudelid efektiivne ja mugav viis nõuete ja pakutud lahenduste visualiseerimiseks, mis on suureks abiks analüütikule teiste arendustiimi liikmete ja klientettevõtte esindajatega läbirääkimistel.

### **2.2.3 Autori tööpanus projektis**

Autor osaleb arendusprotsessis analüütikuna arendustiimi koosseisus. Enne projekti ametlikku algust on analüütiku ülesandeks koguda vajalikku taustainformatsiooni optimaalse tarkvaralahenduse loomiseks. Seda võib võtta kui osa sprindist 0, kirjeldatud allikas [18], kuigi see võttis aega kauem kui kolm nädalat. Projekti jooksul kogub analüütik nõudeid süsteemile ning lisab need projekti tööjärge eeposte ja kasutuslugude kujul. Nõuded andmebaasile väljendatakse kontseptuaalse andmemudelina. Nõuete allikateks on e-kirjad, telefonikõned, koosolekud, andmete uuring jne. Projekti käigus viib ta läbi nende nõuete põhjaliku analüüsi, tagamaks, et iga järgmise sprindi alguseks on arendajatel olemas kindlad ülesanded ning kogu vajalik informatsioon nende ülesannete täitmiseks on kogutud, analüüsitud ja kliendiga kooskõlastatud. Analüütiku ülesandeks on ka osaleda kõrghariduse tehnilise lahenduse väljapakkumises (loogilises disainis). Analüütiku ülesanneteks on andmete analüüsimine, kasutuslugudele vastuvõtukriteeriumite määramine, lahendust kirjeldavate UML mudelite koostamine, kasutajaliidese prototüübi loomine ning vajadusel lisainformatsiooni küsimine kliendilt või teistest asjakohastest infoallikatest. Autori poolt läbi viidud analüüsi ja disaini koondtulemused moodustavad antud lõputöö põhiosa.

Projekti jooksul koostas autor mitu mudelite versiooni vastavalt toimivate nõuete muutmisele ja analüüsi käigus tehtud otsustele. Töös esitatakse kõige viimane mudelite versioon. Kõik esitatud mudelid on tõlgitud inglise keelest eesti keelde. Mudelite koostamisel jälgitakse UML keele spetsifikatsiooni [19].

### **3 Funktsionaalsed ja mittefunktsionaalsed nõuded süsteemile**

See peatükk kirjeldab kogutud funktsionaalseid ja mittefunktsionaalseid nõudeid süsteemile ning nende nõuete põhjal tehtud tähtsamaid analüüsi ning disaini puudutavaid otsuseid.

#### **3.1 Mittefunktsionaalsed nõuded**

Tarneteekondade pikkuste kalkulaator peab vastu võtma aadresse logistikaettevõttes kasutuses olevast tarnearuannete süsteemist ning pärast teekondade pikkuste arvutamist salvestama tulemused andmebaasi. Tulevikus plaanitakse ettevõttes ehitada uus süsteem, mis hakkab kasutama arvutatud teekonnapikkuseid tarnemarsruutide koostamiseks ja kulude arvutamiseks. Antud lahendus on mõeldud eelkõige logistikaettevõtte partnerettevõtte tarnete jaoks, mille kaupu tarnitakse maanteetranspordiga üle terve Euroopa.

Aadresse on kokku kolme tüüpi: kaubasaatjate, kaubasaajate ja logistikaettevõtte enda depoode aadressid, mida veokid kauba tarnimisel läbivad. Ärivajadustest lähtudes piisab, kui teekonnapikkused arvutatakse sihtnumbri täpsusega. Kuna tegemist on rahvusvaheliste tarnetega, ei mängi mõne kilomeetri suurune eksimus suurt rolli.

Tarneteekondade pikkuste arvutamine otsustati teostada PTV xServeri tarkvara poolt pakutud Route Planning API [20] kaudu. Otsus võeti vastu logistikaettevõtte esindajate poolt. Antud vahend valiti, sest teekonna pikkuse kalkuleerimisel arvestab see veoautode liikumise omapäradega, erinevalt näiteks Google Maps [21] tarkvarast, mis koostab teekonnad tõhusalt erasõidukite jaoks. Samuti on kasutatud PTV tarkvara logistikaettevõtte teistes projektides ning tulemusega rahule jäädud. Üheks oluliseks loodava süsteemi piiranguks on see, et ettevõtte tellitud PTV xServeri tarkvara kasutamise litsents seab piirangu päringute arvule aastas. Antud asjaolu on tähtis arvestada terve lahenduse planeerimisel ning püüda võimalikult palju vähendada PTV xServerisse saadetavate päringute arvu. Kui ettevõttes tekib kunagi tulevikus vajadus arvutada teekonnapikkused ka lennu-, mere- või raudteetranspordi jaoks, siis on võimalik PTV xServeri tarkvara asendada mõne alternatiivtarkvaraga, mis arvestaks teekonnapikkuste arvutamisel ka nende transpordiliikide omapäradega. PTV xServeri tarkvara arvutab aga teekonnapikkused ainult maanteetranspordi jaoks.



Teekonnapikkused tuleb arvutada kõikide süsteemi jõudnud aadresside vahel. Teekonnapikkused lähtekohast sihtkohani ja vastupidi sihtkohast lähtekohani peetakse üheks ja samaks pikkuseks, vaatamata sellele, et tegelikud teekonnapikkused edasi ja tagasi kahe asukoha vahel võivad liiklustingimuste tõttu natukene erineda. Võib öelda, et saadud arvutamistulemused peavad moodustama suunamata täisgraafi struktuuri [22], kus sõlmedeks on asukohad ning kaared näitavad kohtadevahelisi seoseid. Teekonnapikkused on kaarte kaalud.

Kuna süsteemi jõuavad aadressid sageli puudulike andmete või vigadega, mis takistavad korrektset teekondade pikkuste arvutamist, siis peab süsteem võimaldama aadresside parandamist. Kõigepealt oodatakse käsitsi parandamise võimalust.

Kõik aadresside versioonid, nii originaalid kui ka parandatud, peavad olema süsteemi salvestatud, et vältida korduvat teekonnapikkuste arvutamist ning tagada tulevikus tarneandmete kooskõlastus erinevate süsteemide vahel.

Andmebaasisüsteemina tuleb kasutada Oracle andmebaasisüsteemi, sest see on logistikaettevõttes juba kasutusel. Andmebaas tuleb realiseerida SQL-andmebaasina.

Tarkvara kasutajaliides peab olema ingliskeelne, sest sellel on rahvusvaheline kasutajaskond. Tarkvara peab olema veebipõhine ja töötama kõigi levinumate veebilehitsejatega.

### **3.2 Funktsionaalsete nõuete kirjeldus kasutuslugudena**

See alapeatükk kirjeldab põhilisi funktsionaalseid nõudeid süsteemile kasutuslugudena.

Kasutuslood on laialdaselt soovitatav abivahend agiilsete arendusmetoodikate puhul. Kasutuslugude kasutamise eeliseks on see, et need võimaldavad kiiresti kohaneda projekti jooksul toimuvate nõuete muutustega ning tänu oma lihtsale ja selgele struktuurile tõhustavad arutelu äri huvipoolte (ingl *Stakeholders*) ja arendustiimi vahel [4]. Hea kasutuslugu peab vastama *INVEST* akronüümile ehk olema iseseisev (ingl *Independent*), arutletav (ingl *Negotiable*), väärtuslik (ingl *Valuable*), hinnatav (ingl *Estimable*), väike (ingl *Small*) ja testitav (ingl *Testable*) [5].

Kasutuslugude kirjapanemiseks kasutatakse üldjuhul järgmist formaati [23], kuigi see võib varieeruda sõltuvalt konkreetse kasutusloole sisust:

“<Rollina> tahan <teha seda>, et <saavutada midagi>”

Täpsemaks nõuete paika panemiseks määratakse iga kasutusloole jaoks vastuvõtukriteeriumid [4]. Vastuvõtukriteeriumid lisavad kasutusloole detaile võimaldades arendustiimil täpselt hinnata funktsionaalsuse mahtu, mida tuleb kasutusloole raames arendada. Samuti annavad vastuvõtukriteeriumid tootjale võimaluse töötulemust valideerida. Kuid isegi vastuvõtukriteeriumitega ei kirjelda kasutuslugu üksi nõudeid väga detailselt – seda tehakse teiste vahendite abil (nt UML diagrammidena). Vastuvõtukriteeriumid koostatakse enne sprindi algust ning neid ei muudeta sprindi jooksul. Juhul, kui sprindi jooksul tekib uus nõue, luuakse selle jaoks eraldi kasutuslugu. Antud alapeatükis esitatakse kasutuslood, mis kirjeldavad süsteemi põhifunktsionaalsust.

**Kasutuslugu 1:** Administraatorina tahan, et süsteem oleks võimeline vastuvõtma sissetulevad aadressid, et töödelda neid ja arvutada tarneteekondade pikkused.

*Vastuvõtukriteeriumid:*

- Aadress jõuab süsteemi järgmiste andmetega: organisatsiooni nimi, riigikood, asula ja sihtnumber.
- Süsteem kontrollib, kas täpselt sama aadress juba eksisteerib andmebaasis. Selleks võrreldakse kolme aadressi põhikomponenti: riigikood, asula ja sihtnumber.
- Kui aadress eksisteerib, siis katkestab süsteem selle aadressi töötlemise.
- Kui aadress ei eksisteeri, siis salvestab süsteem aadressi edasiseks töötlemiseks andmebaasi.

**Kasutuslugu 2:** Administraatorina tahan, et süsteem võimalusel seostaks aadressi õige geograafilise asukohaga automaatselt, et vähendada käsitsi parandamise vajadust.

*Vastuvõtukriteeriumid:*

- Süsteem kontrollib, kas aadressi põhikomponendid on registreeritud. Kui vähemalt ühe põhikomponendi andmed puuduvad, siis märgistatakse aadress käsitsi parandamiseks.
- Süsteem seostab automaatselt aadressi PTV xServeri abil leitud geograafilise asukohaga, juhul kui vastuses on täpselt üks geograafiline asukoht, mille aadressi andmed on päringu andmetega identsed.
- Kõikidel muudel juhtudel märgistatakse aadress käsitsi parandamiseks.

**Kasutuslugu 3:** Administraatorina tahan saada teavituse, kui süsteemis tekivad aadressid, mis vajavad käsitsi parandamist.

*Vastuvõtukriteeriumid:*

- Kui leidub vähemalt üks aadress, mis vajab käsitsi parandamist, siis saab administraator üks kord ööpäevas e-kirja teavitusega, et süsteemis on mõned aadressid, mis vajavad käsitsi parandamist. Kui selliseid aadresse ei ole, siis teavitust ei tule.
- Peale teavituse teksti sisaldab iga e-kiri ka linki aadressi parandamise kasutajaliidesesse.

**Kasutuslugu 4:** Administraatorina tahan sisse logida süsteemi, et aadresse käsitsi parandada.

*Vastuvõtukriteeriumid:*

- Administraator peab sisestama oma kasutajanime.
- Administraator peab sisestama oma parooli.
- Kui üks väljadest on jäetud tühjaks, siis kuvatakse veateade.
- Kasutajanime või parooli mittesobivuse puhul kuvatakse veateade.

**Kasutuslugu 5:** Administraatorina tahan näha kõikide käsitsi parandamist vajavate aadresside nimekirja, et saada mugavalt ülevaade.

*Vastuvõtukriteeriumid:*

- Kasutaja näeb nimekirja kõikidest käsitsi parandamist vajavatest aadressidest.
- Nimekirjas on esitatud järgmised aadressi andmed: organisatsiooni nimi, riigikood, asula ja sihtnumber.
- Iga aadressi juures on olemas parandamise korralduse andmise võimalus.

**Kasutuslugu 6:** Administraatorina tahan parandada aadressi, et aidata süsteemil seostada aadressi õige geograafilise asukohaga.

*Vastuvõtukriteeriumid:*

- Parandamine toimub eraldi vaates, kuhu kasutaja suunatakse parandamise korralduse andmise järel.
- Parandamisvaatel on olemas neli sisestusvälja: organisatsiooni nimi, riigikood, asula ja sihtnumber.
- Organisatsiooni nime sisestusväljas andmeid muuta ei saa, ülejäänud väljades saab administraator andmeid muuta. Redigeerimisel valideeritakse, et kõik väljad on täidetud.
- Kinnitamise korralduse andmise järel:
  - kui süsteem suutis seostada aadressi ainulaadse geograafilise asukohaga, siis kuvatakse eduka seostamise kinnitussõnum;
  - kui süsteem ei suutnud leida ühtegi aadressile vastavat geograafilist asukohta, siis kuvatakse veateade, milles palutakse kasutajal uuesti aadressi parandada;
  - kui süsteem sai PTV xServerist mitu võimalikku geograafilist asukohta, siis kuvatakse nimekiri võimalikest variantidest. Administraator valib sobiva variandi, kinnitab oma valiku ning näeb eduka seostamise kinnitussõnumit.

Kommentaari: kuigi aadressi töötlemise jaoks on olulised ainult geograafilised koordinaadid, näidatakse administraatori valiku lihtsustamiseks nimekirjas kõiki andmeid, mis tulevad PTV xServerist.

Seetõttu on seal järgmised väljad: riigi kood (ingl *Country Code*), maakond (ingl *State*), linn (ingl *City*), linn2 (ingl *City2*), tänav (ingl *Street*), maja number (ingl *House Number*), geograafiline laius (ingl *Latitude*) ja geograafiline pikkus (ingl *Longitude*).

**Kasutuslugu 7:** Administraatorina tahan märkida aadressi kehtetuks, kui seda pole võimalik parandada, et vältida vajadust parandada seda aadressi uuesti.

*Vastuvõtukriteeriumid:*

- Parandamisvaatel on aadressi kehtetuks märkimise korralduse andmise võimalus.
- Korralduse saamisel küsitakse kasutajalt, kas ta on oma otsuses kindel.
- Kasutajalt kinnituse saamisega märgistab süsteem aadressi kehtetuks.

**Kasutuslugu 8:** Administraatorina tahan taastada eksikombel kehtetuks märgistatud aadressi, et parandada oma viga.

*Vastuvõtukriteeriumid:*

- Kasutajaliideses on olemas eraldi vaade, kus näidatakse kõik kehtetuks märgistatud aadressid.
- Iga aadressi juures on olemas taastamise korralduse andmise võimalus.
- Korralduse saamisel küsitakse kasutajalt, kas ta on oma otsuses kindel.
- Kasutajalt kinnituse saamisega märgistab süsteem aadressi käsitsi parandamiseks.

**Kasutuslugu 9:** Administraatorina tahan, et süsteem arvutaks teekonnapikkused kõikide salvestatud geograafiliste asukohtade vahel ning salvestaks neid andmebaasi.

*Vastuvõtukriteeriumid:*

- Kui süsteemis tekib uus geograafiline asukoht, siis arvutatakse kõik teekonnapikkused selle geograafilise asukoha ja kõikide varem salvestatud geograafiliste asukohtade vahel.
- Arvutatud teekonnapikkused salvestatakse kilomeetrites 0.5 kilomeetri täpsusega.

**Kasutuslugu 10:** Administraatorina tahan, et süsteem teostaks perioodilist tarneteekondade pikkuste ümberarvutamist, et tagada andmete aktuaalsus.

*Vastuvõtukriteeriumid:*

- Ümberarvutamine toimub üks kord aastas.
- Ümberarvutamine toimub automaatselt taustaprotsessina.

Antud peatükkis kirjeldatud süsteeminõuete ja disainiotsuste põhjal on koostatud järgmises peatükis esitatud süsteemi kirjeldavad UML mudelid.

### 3.3 Olulised otsused

Analüüsi käigus otsustati teekondade pikkuste kalkulaatori süsteemi keskseks olemitüübiks määrata *Aadress*, sest süsteemi tulevate aadresside töötlemisel põhinevad loodava tarkvara tähtsamad protsessid ning teekondade pikkuste arvutamine on aadressi töötlemise viimaseks etapiks. Kuna oma elutsükli jooksul läbib aadress mitu töötlemise etappi, siis otsustati registreerida iga aadressi puhul selle hetkeseisund [24] ning määrata seisundite põhjal toimingud, mida süsteem peab aadressiga igal töötlemise etapil tegema.

Ärивajaduse järgi peab süsteem arvutama teekonnapiikkused kõigi salvestatud aadresside vahel. See asjaolu seab süsteemile teatud piirangud, sest iga uus salvestatud aadress suurendab arvutamist vajavate teekonnapiikkuste arvu ning järelikult ka arvutamisele kuluvat aega. „Pudelikael“ loodava lahenduse puhul on PTV xServeri poolel – päringute arvu suurendamisega väheneb tunduvalt arvutamise kiirus. Selleks, et täpsemalt tuvastada kuidas salvestatud aadresside arv mõjutab teekonnapiikkuste arvutamist, testiti arendaja poolt PTV jõudlust. Testimise tulemused on esitatud Lisas 1 [3]. Testimine näitas, et mida rohkem aadresse on salvestatud süsteemi, seda vähem uusi aadresse suudab süsteem ööpäeva jooksul töödelda. Antud piirang püstitas keerulise analüüsiülesande – koostada selline aadressi töötlusprotsess, et xServerisse saadetavate teekonnapiikkuste arvutamispäringute arv oleks minimaalne. Selle eesmärgi saavutamiseks otsustati teostada mitmeastmeline aadressi kontroll, mis garanteerib, et samade asukohtade vahel ei toimuks korduvat teekonnapiikkuse arvutamist.

Eelmainitud piirangust tulenes ka järgmine oluline disainiotsus – saata teekonnapiikkuste arvutamise päringuga ainult aadresside geograafilised koordinaadid, mitte aadressid ise.

Antud juhul otsitakse koordinaate ehk geograafilist asukohta PTV xServeri abil enne teekonnapiikkuste arvutamist ning salvestatakse aadressist eraldi. Antud lähenemine on põhjendatud ka sissetulevate aadresside andmete kvaliteediga. Andmete analüüsi käigus selgus, et on väga levinud olukord, kui süsteemi saadetakse sama asukoha erinevad aadresside versioonid, mille erinevus seisneb näiteks selles, et aadressi registreerimisel kasutati linna nime kas kohalikus keeles või inglise keeles. Süsteemi jaoks paistavad need aadressid erinevad, kuigi tegelikult viitavad samale kohale. Korduva arvutamise vältimiseks seostatakse erinevad aadressi versioonid sama geograafilise asukohaga. Sel juhul toimub arvutamine ainult üks kord, kuigi salvestatud aadresside versioone on mitu. Samamoodi toimub protsess aadressi käsitsi parandamise korral – nii algne kui ka parandatud aadressi versioon seostatakse ühe ja sama geograafilise asukohaga.

Andmete analüüsi käigus tuvastati, et süsteemi võivad sattuda ka nii vigased aadressid, mida pole võimalik isegi käsitsi parandada. Selliste juhtumite käsitlemiseks otsustati realiseerida lisafunktsionaalsus, mis lubaks kasutajal märgistada aadressi kehtetuks. Kui sama kehtetu aadress jõuab uuesti süsteemi, siis süsteem välistab selle töötlemisprotsessist. Juhuks, kui kasutaja märgistab aadressi kehtetuks eksikombel, otsustati lisada ka aadressi taastamise funktsionaalsus.

Kuna ärivajadustest lähtudes piisab, kui arvutatakse teekonnapiikkused sihtnumbri täpsusega, otsustati, et tarneteekondade pikkuste kalkulaator käsitleb ainult kolme aadressi põhikomponenti: riigikood, asula ja sihtnumber. Aadressi unikaalsust süsteemis määratakse nende kolme põhiparameetri väärtuste järgi. Peale selle registreeritakse aadressiga seotud organisatsiooni nimi, et abistada kasutajat käsitsi aadressi parandamisel. Kuna organisatsiooni nimi jõuab süsteemi aadressi komponendina, mis ei osale aadressi töötlemise protsessis ning mängib süsteemis ainult informatiivse rolli, otsustati mitte käsitleda organisatsiooni eraldi olemitüübina, vaid registreerida organisatsiooni nimi olemitüübi *Aadress* atribuudi väärtusena. Selline lihtsustatud lähenemine aadresside haldamisel vähendab üldist aadresside arvu, mida süsteem peab töötleva. Antud asjaolu on suureks eeliseks, arvestades varem mainitud teekondade pikkuste arvutamise seotud lahenduse jõudluse piirangut ning ärivajadust vähendada PTV xServerisse saadetavate päringute arvu. Kui aadressi töötlemisel võetaks arvesse ka tänava nimi ja majanumber, siis suurendaks see neljakordselt töötlemist vajavate aadresside arvu, kuigi ei lisaks süsteemile suuremat ärilist väärtust. Lisaks on antud

lähenedamine tingitud teekondade pikkuste kalkulaatoriga integreerivate olemasolevate süsteemide omapäradest ja aadresside andmete kvaliteedist.

Andmete analüüsi käigus tuvastati, et kuigi tarneruannete süsteemis registreeritakse umbes kaks miljonit tarnet kuus, siis enamasti tarnetega seotud aadressid korduvad. Viimase pooleteise aasta jooksul (2018-2019) registreeriti süsteemis umbes pooltuhat põhikomponentide järgi unikaalset aadressi. See tähendab, et loodavasse tarneteekondade pikkuste kalkulaatori süsteemi jõuavad uued aadressid üsna harva. Kõige suuremat koormust süsteemile oodatakse kohe pärast süsteemi avalikustamist ja kasutuselevõtmist, sest kõik vastuvõetud aadressid on süsteemi jaoks uued ning vajavad töötlemist ja teekonnapiikkuste arvutamist. Aja jooksul hakkavad aadressid korduma ning koormus süsteemile väheneb tunduvalt, kuni kümne töötlemist vajava aadressini nädalas.



## 4 Detailanalüüs

See peatükk annab detailse ülevaate loodava süsteemi toimimisest ning selleks vajalikest andmetest. Peatükk sisaldab süsteemis toimuvate põhiprotsesside kirjeldavaid tegevusdiagramme, aadressi seisundidiagrammi ning kontseptuaalset andmemudelit.

### 4.1 Protsessid

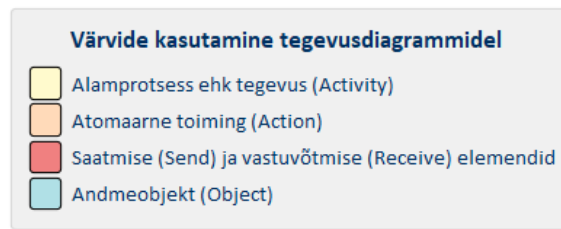
Süsteemis toimuvate põhiprotsesside töövood on kirjeldatud tegevusdiagrammide [7] abil. Tegevusdiagramm on mugav viis kuidas visuaalselt esitada rakenduselt oodatava toimimise loogikat. Projekti jooksul said tegevusdiagrammid oluliseks suhtlemisvahendiks tooteomaniku ja arendustiimi liikmete vahel, andes mõlemale osapoolle detailse ülevaate sellest, kuidas loodav lahendus töötama hakkab.

Teekonnapikkuste kalkulaatori loogika põhineb mitmeastmelisel aadressi töötlemise protsessil ehk üldprotsessil. Parema loetavuse eesmärgil on üldprotsess mitmekordselt tükeldatud alamprotsessideks. Alamprotsessid modelleeritakse põhidiagrammil tegevustena (ingl *Activity*) ning esitatakse eraldi diagrammidena. Ülejäänud tegusid modelleeritakse atomaarsete toimingutena (ingl *Action*).

Objekti tüüpi (ingl *Object*) elemendid esitavad tegevusdiagrammidel andmeobjekte, millele vastavaid andmeid tuleb andmebaasis talletada. Selle kaudu tekib mudelis seos kontseptuaalses andmemudelis väljendatud süsteemi andmevaatega (vt Joonis 9). Objekti seisundi muutused näidatakse märkmete (ingl *Note*) abil. Selle kaudu tekib mudelis seos põhiolemitüübi (*Aadress*) seisundidiagrammiga (vt Joonis 8).

Kõik tegevusdiagrammid läbisid *Enterprise Architect*-i sisse ehitatud valideerimise [25] ilma vigade ja hoiatusteta.

Joonis 1 esitatakse tegevusdiagrammidel kasutatud värvide tähendus:

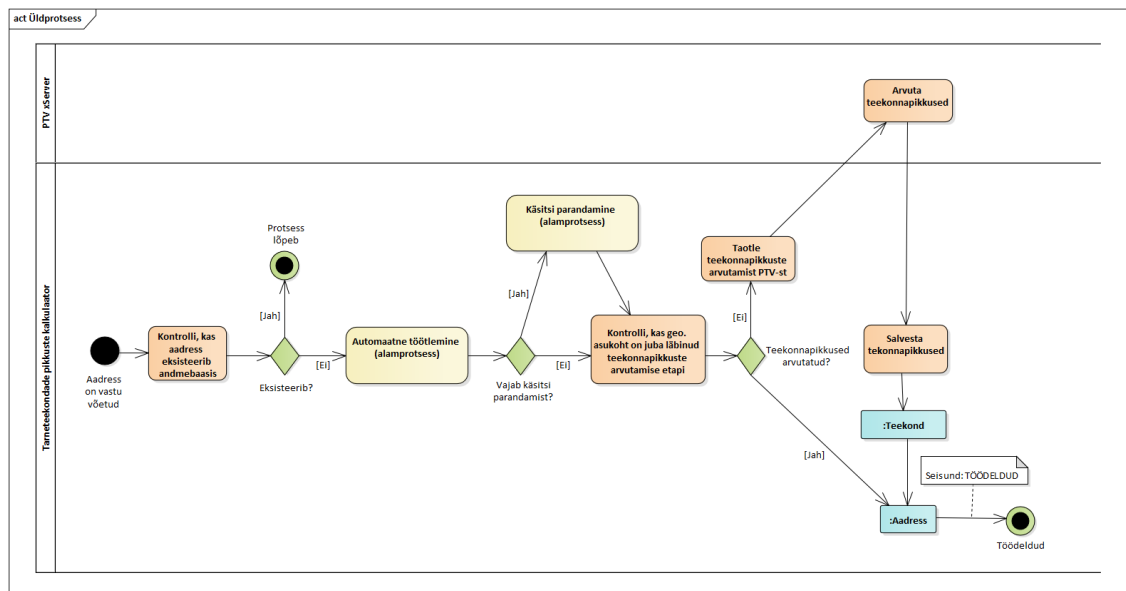


Joonis 1. Värvide kasutamine tegevusdiagrammidel.

Kokku on esitatud kuus tegevusdiagrammi.

#### 4.1.1 Üldprotsess

Järgnev tegevusdiagramm kirjeldab aadressi töötlemise üldprotsessi (Joonis 2).



Joonis 2. Üldprotsess.

Aadressi töötlemise protsessi alustatakse alati kontrollimisega, kas vastu võetud aadress oli varem süsteemis registreeritud ehk kas andmebaasis juba eksisteerib kirje täpselt samasuguste aadressi põhiandmetega: riigikood, asula ja sihtnumber. Juhul, kui selline kirje leitakse, siis katkestatakse aadressi töötlemine. See aitab vältida süsteemi ülekoormamist sama aadressi korduva töötlemisega. Antud kontroll on väga oluline, arvestades sellega, et ühe päeva jooksul võib sama aadress jõuda süsteemi sadu kordi.

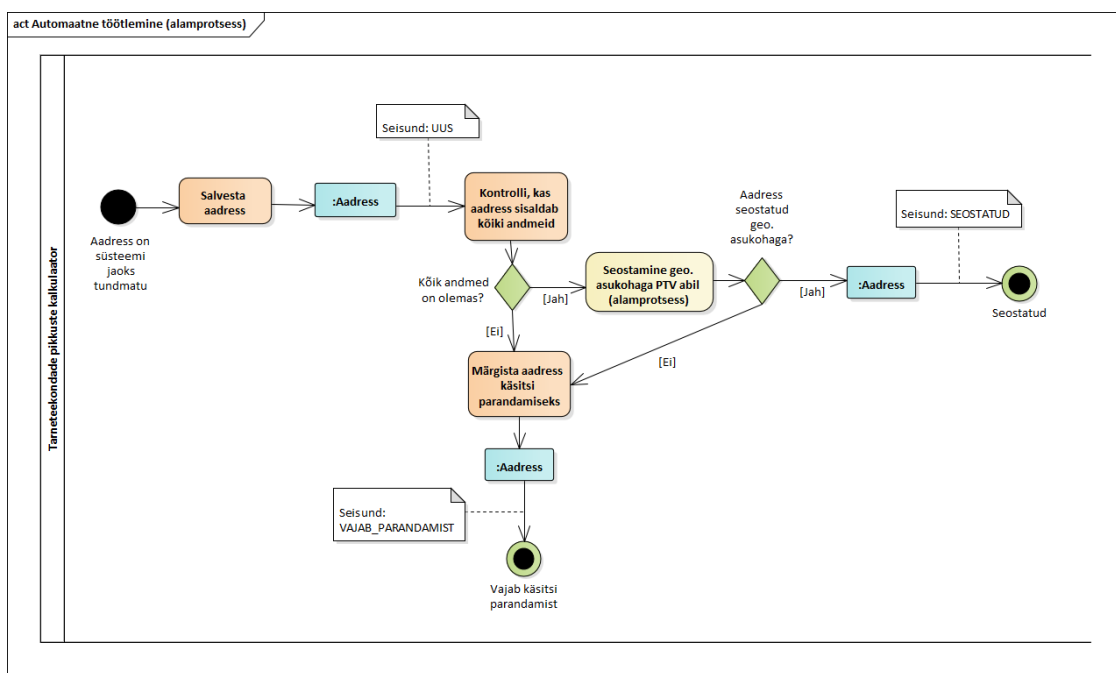
Aadress, mis on süsteemi jaoks tundmatu, läbib automaatse töötlemise etapi ning vajadusel ka käsitsi parandamise etapi. Need etapid moodustavad alamprotsessid, mida

kirjeldatakse eraldi diagrammidel. Mõlema alamprotsessi põhieesmärgiks on seostada aadress sellele vastava geograafilise asukohaga.

Üldprotsessi viimane etapp on seotud teekonnapiikkuste arvutamisega, mis teostatakse PTV xServeri API kaudu. Teekonnapiikkused arvutatakse aadressiga seostatud geograafilise asukoha ja kõikide eelnevalt süsteemis salvestatud geograafiliste asukohtade vahel. Juhul, kui aadress seostatakse geograafilise asukohaga, mis on juba teepikkuste arvutamise protsessis osalenud, siis ei toimu korduvat arvutamist.

#### 4.1.2 Automaatne töötlemine

Järgnev tegevusdiagramm kirjeldab aadressi automaatse töötlemise alamprotsessi (Joonis 3).



Joonis 3. Automaatse töötlemise alamprotsess.

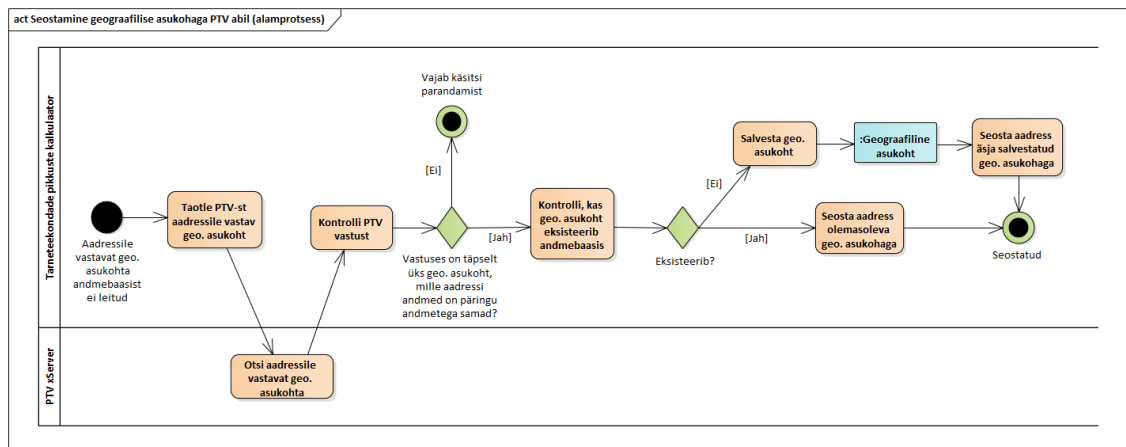
Automaatse töötlemise alamprotsessi peamiseks eesmärgiks on seostada süsteemi jaoks tundmatu aadress sellele vastava geograafilise asukohaga ilma administraatori kaasamiseta, vastasel juhul märgistada aadress vastava seisundiga käsitsi parandamiseks.

Süsteemi võivad jõuda aadressid puudulike andmetega, mistõttu kontrollitakse enne aadressi edasist töötlemist, kas aadress sisaldab kõike põhikomponente: riigikoodi, asulat ja sihtnumbrit. Kui vähemalt üks nendest komponentidest puudub, siis märgistatakse

aadress käsitsi parandamiseks, muul juhul teostatakse aadressi seostamine geograafilise asukohaga PTV xServeri abil. See alamprotsess on kirjeldatud eraldi diagrammina.

#### 4.1.3 Seostamine geograafilise asukohaga PTV abil

Järgnev tegevusdiagramm kirjeldab PTV abil geograafilise asukohaga seostamise alamprotsessi (Joonis 4).



Joonis 4. PTV abil geograafilise asukohaga seostamise alamprotsess.

Antud alamprotsessi põhieesmärgiks on leida aadressile vastav geograafiline asukoht PTV xServeri API kaudu. PTV xServerisse saadetakse päring, mis sisaldab aadressi põhiandmeid.

Näidispäring, mis saadetakse PTV xServerisse:

```
country=ES; state=null; postCode=28806; city= Alcalá de Henares; city2=null;
street=null; houseNumber=null
```

Näidisvastus PTV xServerist:

```
country=ES; state= Comunidad de Madrid; postCode=28806; city= Alcalá de Henares;
city2=null; street=null; houseNumber=null; x=-3.37545; y=40.4961; z=null
```

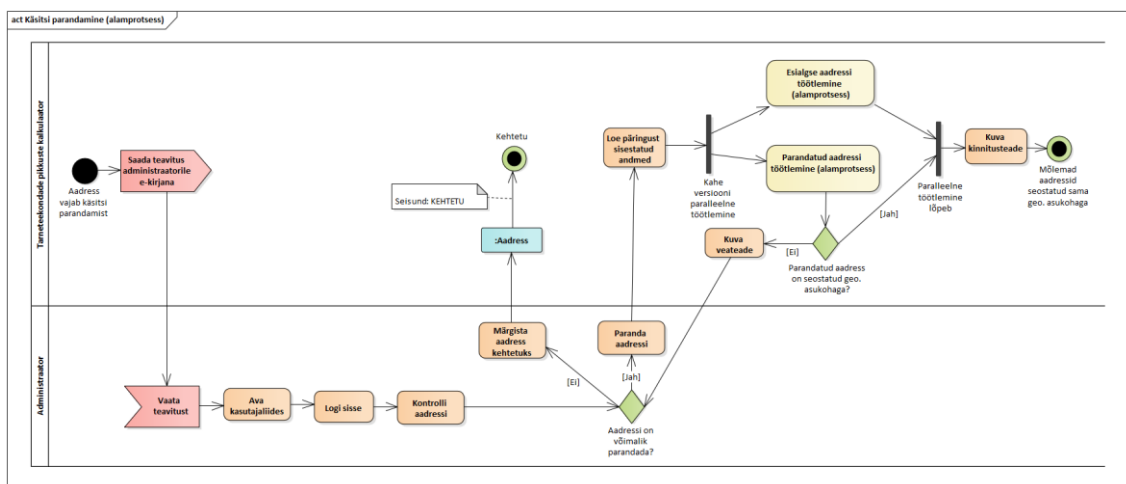
PTV xServerist saadud vastuste puhul tähendab „x“ geograafilist pikkust ning „y“ geograafilist laiust.

Päringu vastuses on null või rohkem geograafilist asukohta. Tulemuses võib olla null asukohta, kui sisendandmed olid puudulikud või neis on viga. Aadressi on võimalik seostada leitud geograafilise asukohaga ainult siis, kui vastuseks saadi täpselt üks geograafiline asukoht, millel riigikood, asula ja sihtnumber on täpselt samad päringu

omadega. Enne uue geograafilise asukoha andmebaasi lisamist kontrollitakse, et sama asukoht ei oleks varem süsteemi juba salvestatud. Kui andmebaasist leitakse selline asukoht, seostatakse aadress sellega. Kui andmebaasist asukohta ei leita, siis salvestatakse uus asukoht ning seostatakse aadress sellega. Antud lähenemine aitab vältida andmebaasis andmete dubleerimist ning korduvat tekkonnapikkuste kalkuleerimist.

#### 4.1.4 Käsitsi parandamine

Järgnev tegevusdiagramm kirjeldab aadressi käsitsi parandamise alamprotsessi (Joonis 5).



Joonis 5. Käsitsi parandamise alamprotsess.

Käsitsi parandamise alamprotsessi põhieesmärgiks on parandada ja seostada geograafilise asukohaga need aadressid, millele süsteem ei suutnud ise automaatse töötlemise käigus geograafilise asukohaga vastendada.

Kui süsteemis tekib käsitsi parandamist vajav aadress, siis saadetakse administraatorile selle kohta e-kirja teel teavitust, mis sisaldab linki aadressi parandamise kasutajaliidesesse.

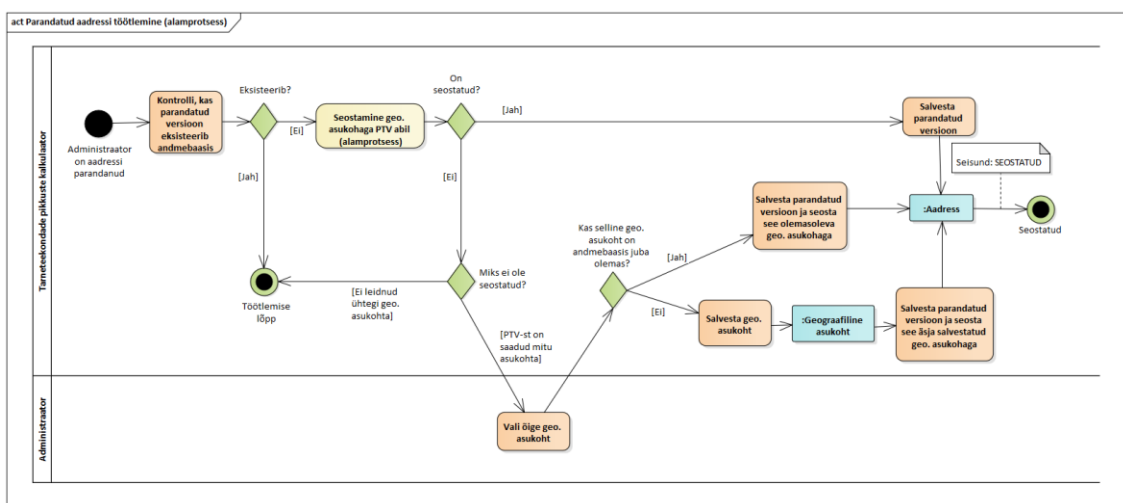
Kasutajaliideses saab administraator parandada ja kinnitada aadressi, lastes süsteemil aadressi seostamist geograafilise asukohaga uuesti proovida. Parandamise käigus tekib peale esialgse aadressi ka parandatud versioon, mida töödeldakse samaaegselt esialgsena. Parandatud versiooni töötlemise käigust sõltub, kuidas esialgset aadressi töödeldakse. Mõlemate versioonide töötlemise alamprotsessid on esitatud eraldi diagrammidena. Juhul, kui isegi pärast aadressi parandamist ei suutnud süsteem leida ühtegi sellele

vastavat geograafilist asukohta, palutakse administraatoril aadressi uuesti parandada, sest tõenäoliselt oli aadressi andmete sisestamisel tehtud tõsine sisuline viga.

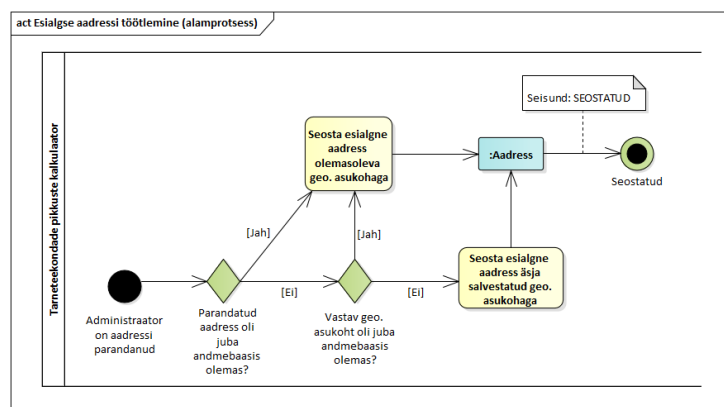
Juhul kui saadud aadressi pole üldse võimalik parandada (nt aadressi andmetest on teada ainult riigikood), siis saab administraator märgistada aadressi kehtetuks. Järgmine kord, kui selline puudulik aadress saabub, siis välistab süsteem selle töötlemise protsessist.

#### 4.1.5 Esialgse ja parandatud aadressi versioonide töötlemine

Järgnevad tegevusdiagrammid kirjeldavad parandatud ja esialgse aadressi versioonide töötlemise alamprotsesse (Joonis 6 ja Joonis 7).



Joonis 6. Parandatud aadressi töötlemise alamprotsess.



Joonis 7. Esialgse aadressi töötlemise alamprotsess.

Parandatud ja esialgse aadressi paralleelse töötlemise alamprotsesside põhieesmärgiks on seostada käsitsi parandamise käigus tekkinud uus aadress ja esmane aadress nendele

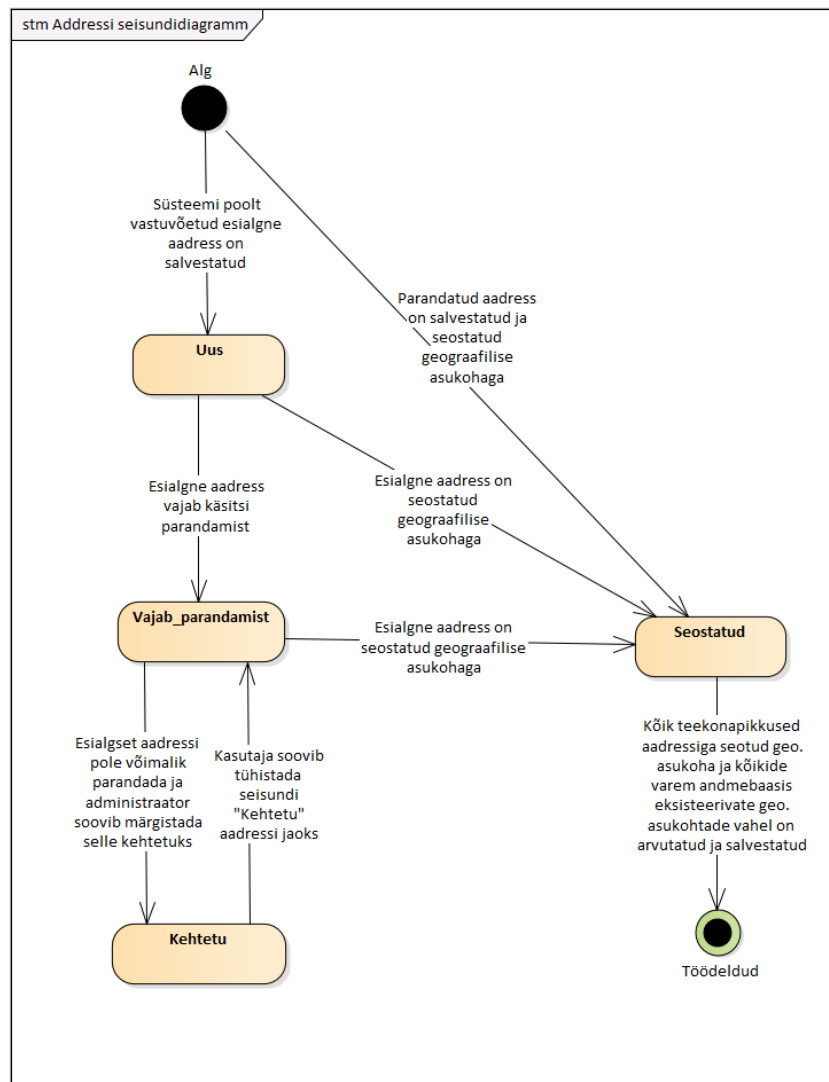
vastava ühe ja sama geograafilise asukohaga. Parandatud aadressi töötlemine süsteemi poolt on sarnane varem kirjeldatud automaatse töötlemisega.

Esiialgu kontrollitakse, et parandatud aadressi ei ole andmebaasis varem registreeritud. Kui on, siis sama aadressi uuesti ei salvestata, vaid seostatakse ainult esialgne aadress sama geograafilise asukohaga, millega on seostatud andmebaasist leitud aadress. Kui aga parandatud aadress on süsteemi jaoks uus, toimub seostamine geograafilise asukohaga PTV xServeri abil. Antud alamprotsess toimub täpselt samamoodi, nagu oli varasemalt kirjeldatud (vt Joonis 4). Juhul kui süsteem suudab selles etapis leida vastava geograafilise asukoha, salvestatakse nii parandatud aadress kui ka leitud asukoht ning seostatakse mõlemad aadressi versioonid selle asukohaga.

Juhul, kui PTV-st tuli vastusena mitu geograafilist asukohta, siis peab administraator ise käsitsi valima kõige paremini sobiva geograafilise asukoha. Selle jaoks kuvab süsteem kõik saadud asukohad tabelina, kus peale geograafiliste koordinaatide näidetakse ka teisi PTV-st saadud aadresside andmeid, et administraatorile valiku tegemist lihtsamaks muuta. Pärast seda toimub nii parandatud kui ka esialgse aadressi salvestamine ja seostamine geograafilise asukohaga samamoodi, nagu see toimuks, kui vastav asukoht oleks kindlaks määratud süsteemi poolt kohe, ilma administraatorikasutaja sekkumiseta.

## 4.2 Aadressi seisundidiagramm

See alapeatükk kirjeldab aadressi seisundidiagrammi [7], mis näitab antud süsteemi kontekstis aadresside kõikvõimalikke elutsükleid ja annab ülevaate aadressi võimalikest seisunditest ja sündmustest, mille tõttu aadressi seisund võib muutuda (Joonis 8).



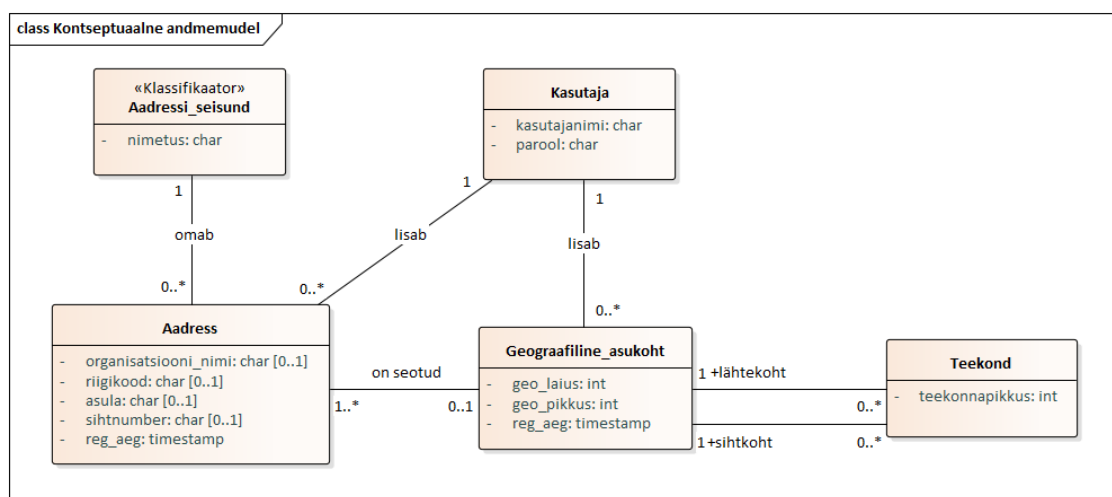
Joonis 8. Aadressi seisundidiagramm.

Diagrammil on nooltega näidatud üleminekud seisundite vahel ning noolte peale on kirjutatud sündmused, mis tingivad aadressi üleviimise uude seisundisse.



### 4.3 Kontseptuaalne andmemudel

Kontseptuaalne andmemudel esitab nõuded andmebaasi andmekoosseisule [8]. Kontseptuaalne andmemudel koosneb olemi-suhte diagrammist (Joonis 9) ning sellel olevate olemitüüpide ja atribuutide sõnalistest kirjeldustest. Olemi-suhte diagrammi koostamiseks kasutatakse antud juhul UML klassidiagrammi [7].



Joonis 9. Kontseptuaalne andmemudel.

Tabel 1 esitab olemi-suhte diagrammidel esitatud olemitüüpide sõnalised kirjeldused.

Tabel 1. Olemitüüpide sõnalised kirjeldused.

Olemitüübi nimi	Definitsioon
Aadress	Tarnega seotud kaupade saatja, kaupade saaja või depoo aadress.
Aadressi seisund	Seisundiklassifikaator, mis võimaldab iga aadressi puhul registreerida selle hetkeseisundi vastavalt aadressi objekti elutsüklile. Võimalikud seisundid kirjeldatakse aadressi seisundidiagrammis.
Geograafiline asukoht	Aadressile vastav asukoht maakeral ekvaatori ja algmeridiaani suhtes, määratud geograafiliste koordinaatide abil.
Teekond	Tarnimismarsruuti lõik mõnest lähtepunktist sihtpunkti.
Kasutaja	Tarneteekondade pikkuste arvutamise süsteemi kasutava isiku või teise süsteemi mandaat. Kuna esialgses süsteemi versioonis on kasutajaid väga vähe, siis lisatakse neid otse andmebaasi. Eraldi kasutajate haldamisfunktsionaalsus realiseeretakse vajadusel hiljem. Selline lähenemine on võimalik ainult

<b>Olemitüübi nimi</b>	<b>Definitsioon</b>
	sellepärast, et süsteem on ettevõtte sisemiseks kasutamiseks.

Tabel 2 esitab atribuutide sõnalised kirjeldused.

Tabel 2. Atribuutide sõnalised kirjeldused.

<b>Olemitüübi nimi</b>	<b>Atribuudi nimi</b>	<b>Atribuudi definitsioon</b>	<b>Näiteväärtus</b>
Aadress	organisatsiooni_nimi	Depoo, kaupade saatja või kaupade saaja organisatsiooni nimi, millele kuulub aadress.	Logistics Ltd
Aadress	riigikood	Üks aadressi põhikomponentidest, mis näitab, millises riigis asub aadressiga määratud asukoht. Õigete andmete puhul vastab ISO 3166-1 alpha-2 või ISO 3166-1 alpha-3 standarditele [26].	EE (või EST)
Aadress	asula	Üks aadressi põhikomponentidest, mis näitab, millises asulas asub aadressiga määratud asukoht. Kuna aadresside allikasüsteemis ega PTV xServeris ei tehta vahet millise asula tüübiga tegemist on, käsitleb tarneteekonnapiikkuste kalkulaator erinevaid asulate tüüpe ühe aadressi komponendina. Teadaolevate aadresside andmete uurimisel selgus, et kõikide asukohtade puhul on võimalik määrata asulad, kus nad asuvad.	Madrid
Aadress	sihtnumber	Üks aadressi põhikomponentidest, postikood, mis tähistab postipiirkonda.	28071
Aadress	reg_aeg	Aadressi süsteemi lisamise täpne aeg kuupäeva ja kellaajaga, kuid ilma ajavööndi ja sekundi mürdosadeta.	22.12.2019 16:59:05

Olemitüübi nimi	Atribuudi nimi	Atribuudi definitsioon	Näiteväärtus
		Süsteem määrab selle automaatselt.	
Aadressi seisund	nimetus	Aadressi elutsükli seisundi nimetus.	Uus (ingl <i>New</i> )
Geograafiline asukoht	geo_laius	Geograafiline laius, maapealse punkti nurkkoordinaat kümnenndmurdarvu kujul. Koos geograafilise pikkusega võimaldab täpselt määrata punkti asukohta maapinnal.	59.428671
Geograafiline asukoht	geo_pikkus	Geograafiline pikkus, maapealse punkti nurkkoordinaat kümnenndmurdarvu kujul. Koos geograafilise laieusega võimaldab täpselt määrata punkti asukohta maakera pinnal.	24.773727
Geograafiline asukoht	reg_aeg	Asukoha süsteemi lisamise täpne aeg kuupäeva ja kellaajaga, kuid ilma ajavööndi ja sekundi murdosadeta. Süsteem määrab selle automaatselt.	29.12.2019 16:59:05
Teekond	teekonnapiikkus	Tarnimismarsruudi löigu pikkus kilomeetrites, ümardatud 0.5 kilomeetri täpsusega.	177.5
Kasutaja	kasutajanimi	Kasutaja unikaalne tunnus süsteemis	mart.mets
Kasutaja	parool	Kasutaja identsust tõendav tunnus. Andmebaasis salvestatakse parooli ja pipra põhjal leitud räsiväärtus. Kasutatakse rakenduse lähtekoodi salvestatud pipart [27].	F31MQUUbN8s1 /fP8WPMs8XjV Yj16Omtg5GtL6 3Hv3xzmzJctPX5 KY10/vkxHKpH pDNL5KJiYloDu CS1wo/qCqIQ==

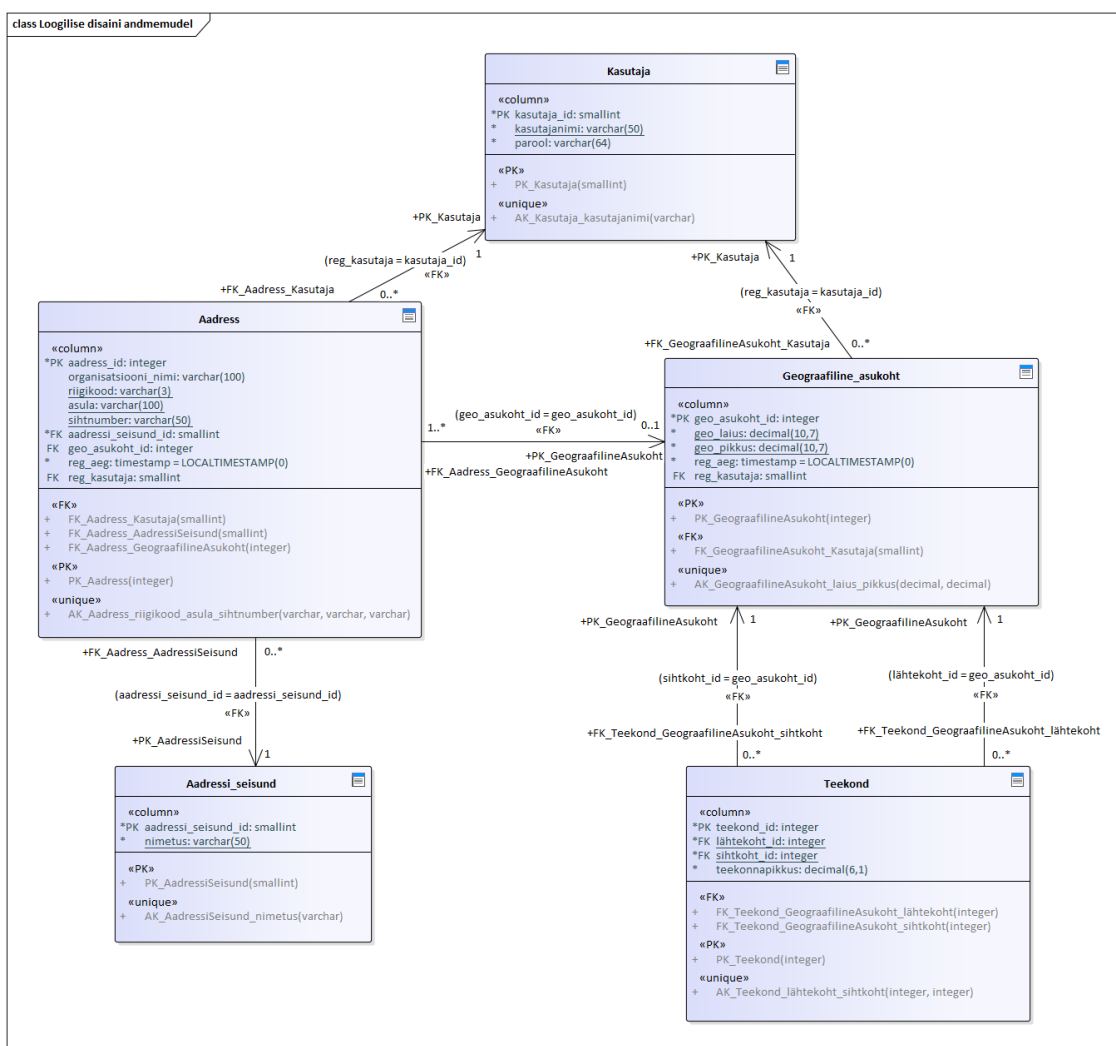
Kontseptuaalne andmemudel on sisendiks loogilise disaini andmemudeli koostamisele.

## 5 Disain

See peatükk kirjeldab tarkvara tehnilise lahenduse neid aspekte, mille kirjelduse koostamine oli autori ülesanne. Peatükk sisaldab SQL-andmebaasi loogilise disaini mudelit ja kasutajaliidese prototüüpi.

### 5.1 Loogilise disaini andmemudel

Alapeatükk sisaldab kontseptuaalse andmemudeli põhjal koostatud loogilise disaini andmemudelit (Joonis 10), mis kirjeldab andmebaasi struktuuri pidades silmas andmebaasi aluseks olevat andmemudelit (antud juhul SQLi aluseks olev andmemudel), kuid mitte arvestades konkreetse andmebaasisüsteemi omapäradega [8]. Andmebaasi loogilise disaini diagrammi koostamiseks kasutatakse UML klassidiagrammi [7].

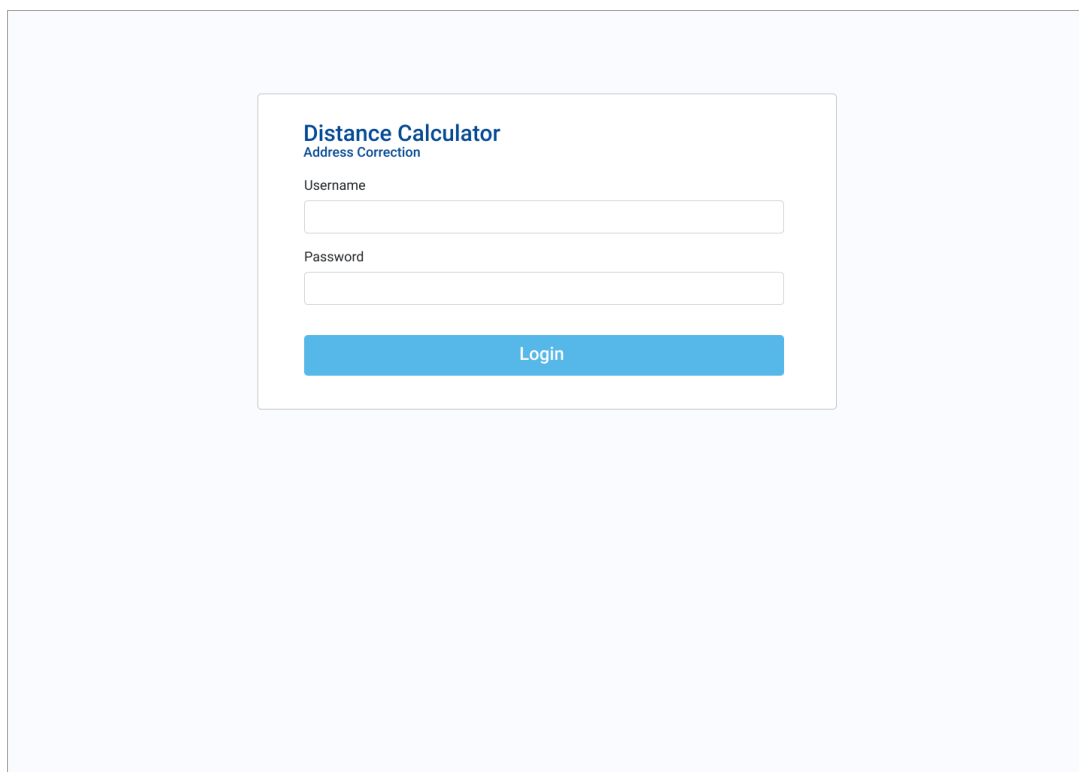


Joonis 10. Loogilise disaini andmemudel.

Loogilise disaini andmemudel loodi kasutades Enterprise Architect tarkvara poolt pakutavat kontseptuaalsest andmemudelist disaini mudeli genereerimise vahendit ning genereerimise tulemust parandati ja täiendati. Andmemudelis kasutatakse andmetüüpe ning funktsioonide nimesid, mida kirjeldab SQL standard [28]. Andmemudelis ei kasutata nimesid inglise keelt, ei kasutata Oracle andmetüpe ning funktsioone, ei kirjeldata indekseid – see kõik oleks juba füüsiline disain.

## 5.2 Kasutajaliidese prototüüp

See peatükk kirjeldab kasutajaliidese prototüüpi. Prototüüp on tehtud *Figma* [11] tarkvara abil. Prototüübist sai tõhus abivahend läbirääkimistel arendustiimi, tooteomaniku ja teiste tellija ettevõtte esindajate vahel, sest aitas anda kõikidele osapooltele ühtemoodi ettekujutuse loodava süsteemi kasutajaliidese. Prototüübi koostamisel püüdis autor läbi mõelda kasutajaliidest niimoodi, et see oleks maksimaalselt lihte ja kasutajale mugav. Kuna tegemist on rahvusvahelise logistikaettevõttega, on kasutajaliides ja selle prototüüp ingliskeelsed. Prototüüp on esitatud ekraanipiltidena (Joonis 11-Joonis 23). Konfidentsiaalsuse säilitamiseks on ettevõtte logo piltidelt eemaldatud ning näidisandmed asendatud.



The image shows a login form for a 'Distance Calculator' application. The form is centered on a light blue background. It has a title 'Distance Calculator' in blue, followed by a subtitle 'Address Correction' in a smaller font. Below the subtitle are two input fields: 'Username' and 'Password'. The 'Username' field is a simple white box with a light blue border. The 'Password' field is a white box with a light blue border and a small eye icon on the right side. Below the password field is a solid blue button with the text 'Login' in white.

Joonis 11. Sisselogimisvaade.

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

**Addresses for correction**

Company Name	Country Code	City	Postal Code	Edit
Logistics Ltd.	BE	Gent	-	<a href="#">Edit</a>
New Group	ES	Madrid	28100	<a href="#">Edit</a>
Fast Delivery Co.	DE	-	15-231	<a href="#">Edit</a>
Medical Group	IT	Veronna	01002	<a href="#">Edit</a>
Good Time Ltd.	GR	Athens	71772	<a href="#">Edit</a>
-	GR	Athen	71772	<a href="#">Edit</a>
General Co.	PL	Warszawa	001-232	<a href="#">Edit</a>
-	PT	-	-	<a href="#">Edit</a>
X Company	FR	Paris	FR 12309	<a href="#">Edit</a>

Joonis 12. Parandamist vajavate aadresside nimekirja vaade.

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

[Back to list](#)

Company Name

Country code

City

Postal code  
  
Field is required

Joonis 13. Redigeerimisvaade.

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

[Back to list](#)

Company Name  
Logistics Ltd.

Country code  
BE

City  
Gent

Postal code  
8712

[Edit address](#)

Multiple response was received from PTV. Please select an option with the most accurate coordinates.

	Country Code	State	City	City 2	Postal Code	Street	House Number	Longitude	Latitude
<input checked="" type="radio"/>	BE	Vlaams Gewest	Gent	Desteldonk	8712	-	-	57.11579	8.85595
<input type="radio"/>	BE	Vlaams Gewest	Gent	Mendonk	8712	-	-	57.15493	8.83407
<input type="radio"/>	BE	Vlaams Gewest	Gent	Sint-Kruis-Winkel	8712	-	-	57.19493	8.71819

[Approve](#) [Address is invalid](#)

Joonis 14. Geograafilise asukoha valimise vaade.

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

[Back to list](#)

Company Name  
GSK Belgium

Country code  
BE

City  
Gent

Postal code  
8712

[Edit address](#)

Multiple response was received from PTV. Please select an option with the most accurate coordinates.

Approve this address with the selected geographic location?  
This action can't be undone.

[Yes](#) [Cancel](#)

	Country Code	State	City	City 2	Postal Code	Street	House Number	Longitude	Latitude
<input checked="" type="radio"/>	BE	Vlaams Gewest	Gent	Desteldonk	8712	-	-	57.11579	8.85595
<input type="radio"/>	BE	Vlaams Gewest	Gent	Mendonk	8712	-	-	57.15493	8.83407
<input type="radio"/>	BE	Vlaams Gewest	Gent	Sint-Kruis-Winkel	8712	-	-	57.19493	8.71819

[Approve](#) [Address is invalid](#)

Joonis 15. Geograafilise asukoha valiku kinnitamise vaade.

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

[Back to list](#)

Company Name  
Logistics Ltd.

Country code  
BE

City  
Gent

Postal code  
8712

Address has been successfully approved and resolved!

[OK](#)

[Edit address](#)

Multiple response was received from PTV. Please select an option with the most accurate coordinates.

	Country Code	State	City	City 2	Postal Code	Street	House Number	Longitude	Latitude
<input checked="" type="radio"/>	BE	Vlaams Gewest	Gent	Desteldonk	8712	-	-	57.11579	8.85595
<input type="radio"/>	BE	Vlaams Gewest	Gent	Mendonk	8712	-	-	57.15493	8.83407
<input type="radio"/>	BE	Vlaams Gewest	Gent	Sint-Kruis-Winkel	8712	-	-	57.19493	8.71819

[Approve](#) [Address is invalid](#)

Joonis 16. Eduka aadressi geograafilise asukohaga seostamise vaade (variant 1).

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

[Back to list](#)

Company Name  
Logistics Ltd.

Country code  
BE

City  
Gent

Postal code  
8712

Address has been successfully approved and resolved!

[OK](#)

[Approve](#) [Address is invalid](#)

Joonis 17. Eduka aadressi geograafilise asukohaga seostamise vaade (variant 2).



Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

[Back to list](#)

Company Name

Country code

City

Postal code

Sorry, provided address cannot be resolved. Please edit address and try again.

Joonis 18. Ebaõnnestunud geograafilise asukohaga seostamise vaade.

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

[Back to list](#)

Company Name

Country code

City

Postal code

Field is required

Mark this address as invalid?  
It will be excluded from distance calculation.

Joonis 19. Kehtetuks aadressi märgistamise kinnitamise vaade.

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

[Back to list](#)

Company Name

Country code

City

Postal code  
  
Field is required

Address has been marked as invalid.

[OK](#)

[Approve](#) [Address is invalid](#)

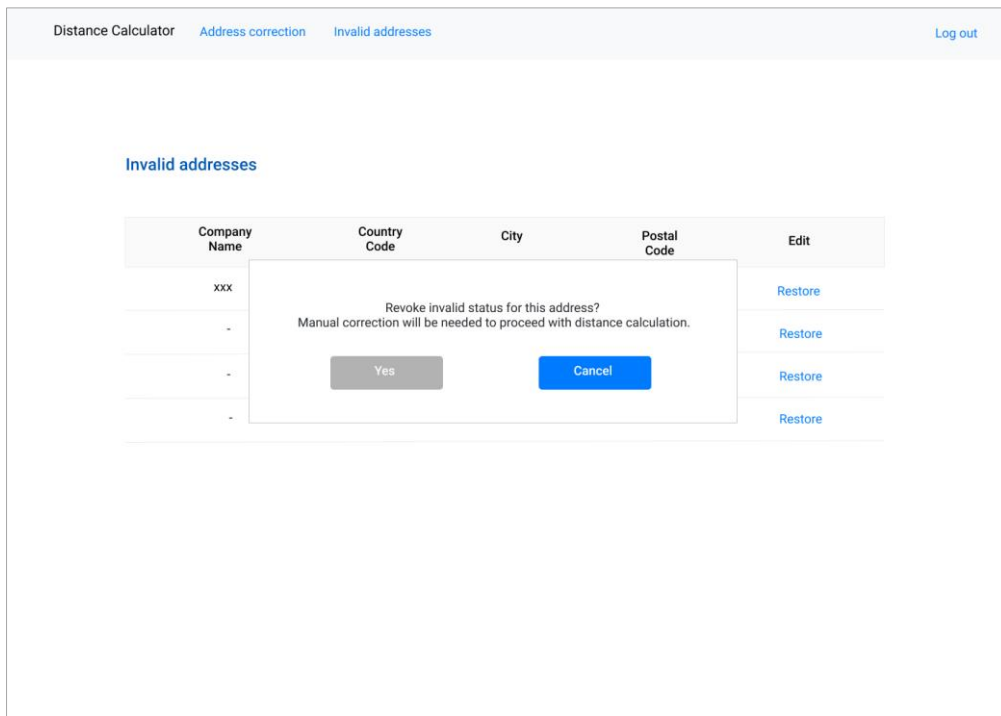
Joonis 20. Eduka kehtetuks aadressi märgistamise vaade.

Distance Calculator [Address correction](#) [Invalid addresses](#) [Log out](#)

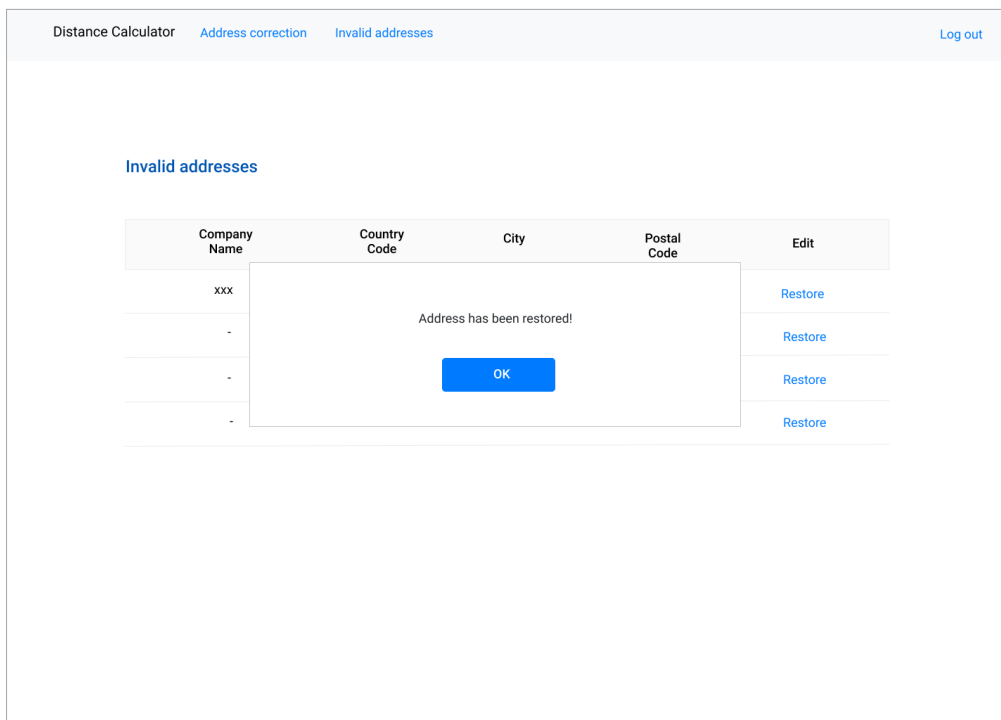
[Invalid addresses](#)

Company Name	Country Code	City	Postal Code	Edit
xxx	-	xxx	-	<a href="#">Restore</a>
-	TO	-	-	<a href="#">Restore</a>
-	DE	-	1	<a href="#">Restore</a>
-	MA	-	-	<a href="#">Restore</a>

Joonis 21. Kehtetu aadresside nimekirja vaade.



Joonis 22. Aadressi taastamise kinnitamise vaade.



Joonis 23. Eduka aadressi taastamise vaade.

Käesolev prototüüp on aluseks arendatava süsteemi kasutajaliidese loomiseks.

## 6 Olulisemad järeldused

Selles peatükis tuuakse välja olulisemad järeldused, milleni autor projekti tehes jõudis.

Kõigepealt on projekti edukaks realiseerimiseks ülioluline pidev suhtlemine arendustiimi ja kliendi ettevõtte huve esindava tooteomaniku vahel. Isegi kui tundub, et kõik osapooled mõistavad vajaminevat funktsionaalsust ühtemoodi, on vaja vähemalt iga nädal suhelda ja tööd kooskõlastada, olemaks kindel selles, et arendamine liigub õiges suunas. Tähtsate küsimuste läbirääkimisel peavad osalema mitte ainult tooteomanik ja analüütik, vaid terve arendustiim, et arendajad saaksid kohe anda tagasisidet tehnilise realisatsiooni küsimustes. Oluline roll on ka rüsinameistril, kes veendub selles, et kõik osapooled saavad süsteemi nõuetest ühtemoodi aru ning toetab ja motiveerib projekti arendustiimi.

Koostöö läbipaistvuse ja kooskõllalisuse saavutamisele aitavad kaasa ühine tööjärg ja projekti dokumentatsioon, millele on vaba ligipääs nii arendustiimil kui ka tooteomanikul. Samas peavad arendajad lisama tehtud arendusülesannete juurde asjakohaseid kommentaare, mis aitaksid teistel osapooltel arendusprotsessi paremini mõista. Analüütik peab omakorda õigeaegselt uuendama projekti dokumentatsiooni vastavalt toimuvate muudatustele.

Vähem tähtis pole ka pidev suhtlemine arendustiimi sees. Tõhus koostöö on võimalik siis, kui analüütik oskab tuua kliendi nõuded arusaadavalt arendajateni ning arendajad teavitavad analüütikut arenduse kulgemisest. Selles osas aitavad väga Scrumi raamistikus ettenähtud igapäevased püstijalakoosolekud, kus arendustiimi liikmed jagavad üksteisega infot oma töö edenemise ja takistuste kohta.

Agiilse projekti puhul on vaja olla paindlik ning osata kiiresti kohaneda toimuvate muudatustega. Analüütiku seisukohalt tähendab see mitte proovida kohe projekti alguses kõike ette näha ning välja töötada ideaalset lõplikku lahendust, vaid liikuda analüüsis edasi samm sammult, täiendades projekti analüüsi detailidega käigu pealt.

Reaalsete ettevõtetega seotud projekti puhul on tähtis arvestada raha ja aja piirangutega. Teades, et projektil on ettevõtete vahel kokkulepitud eelarve ja ajapiirang, on vaja töömahtu mõistlikult planeerida, et jõuda tarnida kliendile õigeaegselt ootustele vastavat ja kvaliteetset tarkvara. Lisafunktsionaalsust saab lisada juhul, kui selle realiseerimine ei ületa kokkulepitud ajapiirangut ja on tõesti vajalik.

Disainiotsuste vastuvõtmisel on oluline arvestada loodava lahenduse kontekstiga, mis sõltub kliendi ettevõtte ja lahendusega seotud süsteemide omapärast. Tarneteekondade pikkuste arvutamise kalkulaator ei ole eraldiseisev süsteem, vaid abivahend teiste süsteemide jaoks. Selleks, et loodud lahendus vastaks reaalsele vajadusele, on oluline analüüsi käigus läbi viia korralik taustauuring ning analüüsida lahendusega seotud andmed.

Ülaltoodud järeldused moodustavad kogumi soovitudest, mida jälgides on autori arvates võimalik projektiga jõuda parema tulemuseni.

## 7 Kokkuvõte

Bakalaureusetöös koguti, analüüsiti ja esitati tarneteekonnapikkuste arvutamise süsteemi nõudeid. Samuti kirjeldati töös projekti arendusmetoodikat ning metoodikat, mille alusel viis autor läbi süsteemianalüüsi. Töö tulemusena valmis funktsionaalsete nõuete kirjeldus kasutuslugudena ning mittefunktsionaalsete nõuete kirjeldus. Autor koostas süsteemi tööd kirjeldavad tegevusdiagrammid ja põhiolemitüübi seisundidiagrammi, kontseptuaalse ja loogilise disaini andmemudelid ning kasutajaliidese prototüübi. Samuti põhjendati analüüsi käigus tehtud otsuseid ning mõnede otsuste puhul selgitati, miks alternatiivide hulgast tehti just mainitud valik.

Tarneteekondade pikkuste arvutamise kalkulaatori projekti tulemusena luuakse tarkvara, mis abistab maanteetranspordi abil kaupade tarnimist. Tarkvara arvutab kogu kauba tarnimise marsruudi ning kulude kalkuleerimiseks vajalike geograafiliste asukohtade vahelised teekondade pikkuseid, kontrollides eelnevalt neile asukohtadele vastavate aadresside korrektsust ning vajadusel võimaldades neid käsitsi parandada. Loodav süsteem vastab kliendi ootustele. See on logistikaettevõtte jaoks esimene samm, mis alustab üleminekut efektiivsemale tarnimiskulude arvutamise meetodile, kus arvestatakse mitte ainult kauba parameetreid, vaid ka tarneteekondade pikkuseid.

Süsteem peab lõputöös kirjeldatud kujul valmima 2020. aasta jaanuari lõpuks. Peale projekti lõppu on vajalik tarkvara edasine hooldamine. Kui tarkvara kasutamisel selgub, et mõned protsessid vajavad korrigeerimist, viiakse vastavad parandused ellu. Üheks võimalikuks süsteemi täienduseks on automaatse aadressi parandamise loogika lisamine lihtsate tüüpvide parandamiseks. Samuti plaanitakse tulevikus integreerida tarneteekondade pikkuste kalkulaator logistikaettevõtte andmehaldussüsteemiga, mille abil saaks osa aadresse automaatselt valideerida ja parandada.

## Kasutatud kirjandus

- [1] Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R. (2011). Action design research. MIS Quart. 37-56 [WWW]  
<https://pdfs.semanticscholar.org/309d/493c4d9979b05b5bde925e0d96d30e32020c.pdf> (12.11.2019)
- [2] Basili V., Briand L., Bianculli D., Nejati S., Pastore F., Sabetzadeh M. (2018). Software engineering research and industry: a symbiotic relationship to Foster impact. IEEE Software, 35(5), 44-49 [WWW]  
[https://orbilu.uni.lu/bitstream/10993/35832/1/SO\\_SWSI-2018-01-0009.R1\\_Briand.pdf](https://orbilu.uni.lu/bitstream/10993/35832/1/SO_SWSI-2018-01-0009.R1_Briand.pdf) (12.11.2019)
- [3] Kaasatud ettevõtete sisemine dokumentatsioon.
- [4] Cohn, M. User Stories. [WWW]. [mountaingoatsoftware.com/agile/user-stories](http://mountaingoatsoftware.com/agile/user-stories) (20.12.2019)
- [5] Huang, S. (2018). Writing Technical User Stories. [WWW]  
<https://medium.com/tribalscale/writing-technical-user-stories-434bf96f1dd5> (21.12.2019)
- [6] Adžić, G. (2011). Specification by example: How successful teams deliver the right software. Manning Publications Co.
- [7] Miles, R., Hamilton, K. (2008). Learning UML 2.0. O'Reilly Media. Kindle Edition.
- [8] Andmebaasid I õppematerjalid [WWW] <https://maurus.ttu.ee/372> (23.12.2019)
- [9] Enterprise Architect 15.0 [WWW] <https://sparxsystems.com/products/ea/> (26.12.2019)
- [10] Oracle SQL Developer [WWW]  
<https://www.oracle.com/database/technologies/appdev/sql-developer.html> (26.12.2019)
- [11] Figma Prototyping Tool [WWW] <https://www.figma.com/> (26.12.2019)
- [12] Schwaber, K., Sutherland J. (2017). The Scrum Guide, [WWW]  
<https://www.scrum.org/resources/scrum-guide> (08.12.2019)
- [13] What is Kanban? [WWW] <https://www.digite.com/kanban/what-is-kanban/> (08.12.2019)
- [14] C. A. Standardipõhine tarkvaratehnika sõnastik [WWW] <https://stats.cyber.ee/> (05.01.2020)
- [15] What Is Epic In Agile Methodology (Definition & Template of Epic) [WWW]  
<https://www.yodiz.com/blog/what-is-epic-in-agile-methodology-definition-and-template-of-epic/> (08.12.2019)
- [16] Cohn, M. Planning Poker. [WWW]  
<https://www.mountaingoatsoftware.com/agile/planning-poker> (08.12.2019)
- [17] Waldock, B. Retrospective Sailboat Guide & Video [WWW]  
<https://www.tastycupcakes.org/2019/01/retrospective-sailing-being-agile/> (08.12.2019)
- [18] Scrum: Iteration 0 [WWW] <https://portal.netobjectives.com/reading-path/scrum/scrum-iteration-0/> (15.12.2019)
- [19] About the unified modeling language specification version 2.5 [WWW]  
<https://www.omg.org/spec/UML/2.5/About-UML/> (11.12.2019)

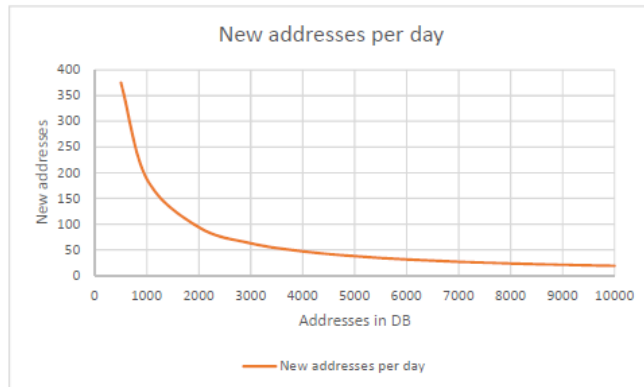
- [20] Route Planning API by PTV xServer [WWW]  
<https://www.ptvgroup.com/en/solutions/products/ptv-xserver/developer-zone/route-planning-api/> (09.12.2019)
- [21] Google Maps [WWW] <https://maps.google.com> (19.11.2019)
- [22] Complete Graph [WWW] <http://mathworld.wolfram.com/CompleteGraph.html>  
(9.12.2019)
- [23] Kasutajalood, mittefunktsionaalsed nõuded ja ära kasutajalood. [WWW]  
<https://mitmekylgsus.wordpress.com/2014/07/19/kasutajalood-mittefunktsionaalsed-nouuded-ja-arakasutajalood/> (9.12.2019)
- [24] Peek, S. Infosüsteemi põhiollemite seisundite esitamine SQL-andmebaasides. [WWW]  
<https://digikogu.taltech.ee/et/Item/101f40e2-d9fe-4bef-ab25-54508f18c078> (22.12.2019)
- [25] Model Validation [WWW]  
[https://sparxsystems.com/enterprise\\_architect\\_user\\_guide/14.0/model\\_domains/model\\_validation.html](https://sparxsystems.com/enterprise_architect_user_guide/14.0/model_domains/model_validation.html) (11.12.2019)
- [26] ISO 3166 Country Codes. Glossary for ISO 3166. [WWW] <https://www.iso.org/glossary-for-iso-3166.html> (19.12.2019)
- [27] Let's add some pepper [WWW] [https://www.martinstoeckli.ch/hash/en/hash\\_pepper.php](https://www.martinstoeckli.ch/hash/en/hash_pepper.php)  
(22.12.2019)
- [28] Melton, J. (2011). Information technology - Database languages - SQL. Part 2: Foundation (SQL/Foundation). ISO/IEC JTC 1/SC 32. IWD 9075-2:201?(E). [WWW]  
<https://www.wiscorp.com/sql20nn.zip> (29.12.2019)



## Lisa 1 – Jõudluse testimise tulemused

Single calculation (distances are calculated one by one)

Addresses in DB	Time (s)	New addresses per day
500	230	376
1000	460	188
2000	920	94
3000	1380	63
4000	1840	47
5000	2300	38
6000	2760	31
7000	3220	27
8000	3680	23
9000	4140	21
10000	4600	19



Parallel calculation (distances are calculated in batches)

Addresses in DB	Time (s)	New addresses per day
500	180	480
1000	360	240
2000	720	120
3000	1080	80
4000	1440	60
5000	1800	48
6000	2160	40
7000	2520	34
8000	2880	30
9000	3240	27
10000	3600	24

