

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Jevgeni Družkov 184976IADB

# **Lennuvälja teenindussõidukite liiklusinformatsiooni esitusmooduli arendamine ja testimine**

Bakalaureusetöö

Juhendaja: Meelis Antoi  
Magistrikraad  
Tarmo Aia

Tallinn 2021

## **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: Jevgeni Družkov

16.05.2021

## Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on arendada lennujaama maismaasõidukite maaseiresüsteemi kuuluvat moodulit, mille abil on võimalik jälgida maapealset liiklust lennujaamas. Nimetatud moodul on mõeldud peamiselt lennujuhtidele, kuid seda on võimalik kasutada teistes lennujaama üksustes, nagu rajameistrid, päästeteenistus ja lennuohutusüksus.

Lõputöö analüüsi käigus uuriti olemasolevaid süsteeme ja nende liiga suure maksumuse tõttu otsustati arendada majanduslikult soodsamat lahendust. Mooduli töö põhineb digitaalse raadiojaama poolt genereeritud andmete töötlemisel. Arhitektuuri poole pealt tegemist on juhtprogrammiga, mis on võimeline edastama digitaaljaamade andmed rakendusliidesesse, mille kaudu on võimalik serverida andmed erinevatele kasutajaliidestele. Moodul on võimeline väljastama aktiivsete sõidukite andmeid ning lisaks on võimalik pärida andmeid statistiliseks analüüsiks (nt kõige aktiivsemate lennujaama tsoonide väljaselgitamine ja lennuohutusosalaste uuringute tegemine).

Lisaks analüüsile ja mooduli arendusele kirjutati ühiktestid ning viidi läbi *user acceptance test* ajavahemikul, mis mahutab sisse vähemalt 1000 lennuoperatsiooni. Mooduli arendusalane põhifookus on suunatud võimalikult kiire lahenduse arendamisele, mida oleks võimalik integreerida maksimaalselt paljustesse lennujaama infosüsteemidesse.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 5 peatükki, 10 joonist, 7 tabelit.

## **Abstract**

### **Development and testing of the software module to display ground service vehicle's information**

The aim of this bachelor's thesis is to develop a module belonging to the ground monitoring system of airport land vehicles, which can be used to monitor ground traffic at the airport. This module is mainly intended for air traffic controllers, but can be used in other airport units such as airfield managers, rescue service and air safety unit.

During the analysis of the thesis, the existing systems were examined and due to their excessive cost, it was decided to develop a more economically advantageous solution. The operation of the module is based on the processing of data generated by a digital radio station. In terms of architecture, a driver is able to transmit data from digital stations to an application interface, through which it is possible to serve data to various user interfaces. The module is able to output active vehicle data and besides, it is possible to request data for statistical analysis (i.e. identification of the most active areas of the airport, so called hot spots, and conducting aviation safety surveys).

In addition to the analysis and development of the module, unit tests were written and a user acceptance test was performed over a period of at least 1000 flight operations. The main development focus of the module is aimed at developing an agile solution that could be integrated into as many airport information systems as possible.

The thesis is in Estonian and contains 30 pages of text, 5 chapters, 10 figures, 7 tables.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> - rakendusliides
ARG	<i>Advanced Routing Guidance</i> - Saab maismaasõidukite navigatsioonisüsteem
ARS	<i>Automatic Registration Service</i> - automaatse registreerimise teenus
AVT	<i>Airport Vehicle Tracking</i> - maismaasõidukite seiresüsteem
DB	<i>Database</i> - andmebaas
DMR	<i>Digital Mobile Radio</i> - digitaalne mobiilne raadiojaam
GLONASS	<i>Global satellite-based positioning systems</i> - globaalne satelliitnavigatsiooni süsteem
GNSS	<i>Global Navigation Satellite System</i> - ülemaailmne satelliitnavigatsiooni-süsteem
GPS	<i>Global Positioning System</i> - üleilmne asukoha määramise satelliitnavigatsiooni süsteem
HTML	<i>Hyper Text Markup Language</i> - hüpertekstdokumentide loomiseks kasutatav märgistuskeel
HTTP	<i>Hypertext Transfer Protocol</i> - hüperteksti edastusprotokoll
ID	<i>Identifier</i> - identifikaator
IP aadress	<i>Internet Protocol address</i> - internetiaadress
JSON	<i>Javascript Object Notation</i> - Javascript andmevahetusvorming
LRRP	<i>Location Request and Response Protocol</i> - asukoha päringu ja vastuse protokoll
PC	<i>Personal Computer</i> - personaalarvuti
QZSS	<i>Quasi-Zenith Satellite System</i> - Kvaasi-Zeniidi satelliitsüsteem
SQL	<i>Structured Query Language</i> - struktuurpäringukeel
TLS	<i>Transport Layer Security</i> - transpordikihi turbeprotokoll

UAT	<i>User Acceptance Testing</i> - kasutajapoolne testimine
UDP	<i>User Datagram Protocol</i> - kasutajadatagrammi protokoll
UI	<i>User Interface</i> - kasutajaliides
USB	<i>Universal Serial Bus</i> - universaalne jadasiin

## Sisukord

1 Sissejuhatus .....	11
1.1 Andmed ja meetodika.....	12
2 Ülevaade probleemist .....	13
2.1 Olemasolevad lahendused.....	14
2.1.1 SAAB AVT süsteem.....	14
2.1.2 ADB Safegate ACEMAX.....	14
2.1.3 Motorola DMR .....	15
2.2 Lahenduse valik.....	16
3 Lahenduse analüüs.....	18
3.1 Tehniliste nõuete kirjeldus.....	18
3.1.1 Funktsionaalsed nõuded.....	18
3.1.2 Mittefunktsionaalsed nõuded.....	19
3.2 Tehnoloogia valik .....	20
3.2.1 Programmeerimiskeele valik .....	20
3.2.2 Andmebaasi valik .....	22
3.2.3 Arenduskeskkonna valik.....	22
4 Arhitektuur.....	23
4.1 Juhtprogrammi ja rakendusliidese arendamine .....	25
4.1.1 Juhtprogramm .....	25
4.1.2 Rakendusliides .....	27
4.2 Kasutajaliidese arendamine .....	29
4.3 Administreerimine .....	31
5 Testimine .....	33
5.1 Ühiktestimine.....	33
5.2 UAT .....	34
6 Hinnang.....	36
6.1 Saavutatud kasutatavus .....	36
6.2 Uudsus .....	37

6.3 Projekti investeringu maksumuse analüüs .....	38
6.4 Edasiarendus .....	38
7 Kokkuvõte.....	40
Kasutatud kirjandus .....	41
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	44
Lisa 2 – Intervjuu baasküsimused .....	45
Lisa 3 – Intervjuu kodeerimisjuhend .....	46
Lisa 4 – Intervjuude <i>cross-case</i> maatriks .....	47
Lisa 5 – Lennuvälja kontrollpunktid .....	50
Lisa 6 – Lennuvälja kontrollpunktide kaart.....	51
Lisa 7 – Lennuvälja kontrollpunktide koordinaadid .....	52



## Jooniste loetelu

Joonis 1. Analüüsi ja arenduse käik. ....	12
Joonis 2. Ühendus DMR võrgu ja arvuti vahel. ....	23
Joonis 3. Veebiteenuse arhitektuur. ....	24
Joonis 4. ARS sõnumite ja LRRP protokollivi diagramm. ....	25
Joonis 5. Rakendusliidese töö. ....	27
Joonis 6. Testkeskkonna ekraanitõmmis. ....	30
Joonis 7. Vaade testimise ajal. ....	34
Joonis 8. Lennuvälja kontrollpunktid. ....	50
Joonis 9. Lennuvälja kontrollpunktide asukohad lennuvälja territooriumi vaates. ....	51
Joonis 10. Lennuvälja kontrollpunktide koordinaadid. ....	52

## Tabelite loetelu

Tabel 1. Programmeerimiskeelte võrdlus.....	21
Tabel 2. API päringud. ....	29
Tabel 3. DMR jaamade sätted. ....	32
Tabel 4. Juhtprogrammi ja rakendusliidese testitud funktsionaalsus. ....	33
Tabel 5. Projekti kulustruktuur. ....	38
Tabel 6. Intervjuu kodeerimisjuhend.....	46
Tabel 7. Intervjuude <i>cross-case</i> maatriks. ....	47

# 1 Sissejuhatus

Lennujuhtimine on iga lennuvälja kui lennuliikluse infrastruktuuri toimimiseks hädavajalik element, mille ülesandeks on piloodi juhendamine nähtamatute marsruutide kaudu õhus. Lisaks eelnevale lennujuhtimise teenust osutava isiku, lennujuhi, ülesandeks on maapealne lennujuhtimine, mis lisaks õhusõidukite juhendamisele hõlmab endas ka maapealsete teenindussõidukite, nn erimasinate, juhendamist lennuvälja aladel.

Lennujuhi otsustusprotsessi üks kriitilisematest elementidest on olukorratedlikkuse tagamine. Statistiliste andmete kohaselt on paljud lennujuhi poolt tingitud ohutusjuhtumitest seotud ebapiisava informatsiooni tagamisega [1]. Sellest tulenevalt vajab iga lennuvälja operatsioonidega tegelev organisatsioon tarkvara, mis on võimeline reaajas edastama lennuvälja liiklusalas asuvate sõidukite informatsiooni (nt asukohaandmed, kiirus, kutsung ja liikumissuund). Lennujuht peab alati teadma, kus ja mis koguses asuvad teenindussõidukid, et vältida õhusõiduki ja teenindussõiduki kokkupõrget. Nimetatud tarkvara poolt tekitatud andmestikku kasutatakse mitte ainult lennujuhtimiseks, kuid ka lennuohutusega seotud protsessides nagu siseaudit, vahejuhtumite menetlemine ja lennujuhtide koolitamine.

Käesolevas lõputöös analüüsitakse probleemi, mille kohaselt puudub lennuvälja käitajal võimekus lennuvälja maapealsete teenindussõidukite digitaalseks jälgimiseks. Paljudel lennuväljadel juhitakse lennuliiklust peamiselt raadioside abil ja visuaalnähtavuse piires kasutatakse optilisi seadmeid. Lennuvälja mahuka pindala tõttu on võimatu visuaalselt kontrollida liiklust täies ulatuses, mistõttu tuleb arvestada põhiliselt raadiosaatja teel edastatud informatsiooniga. Nimetatud olukord põhjustab lennujuhil suurt psühholoogilist ja intellektuaalset pinget, mille tulemuseks kasvab tõenäosus, et lennujuht teeb oma töös vea.

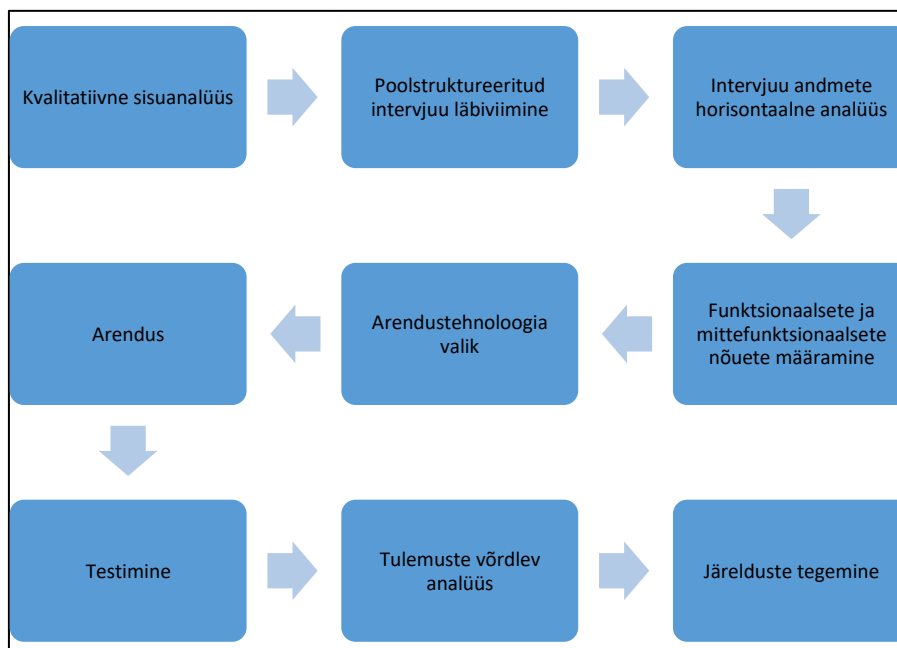
Lõputöö eesmärgiks on arendada ja testida lennuvälja teenindussõidukite jälgimise tarkvara eesmärgiga tagada parem olukorratedlikkus lennuliiklusalas (maa-ala, kus liiguvad õhusõidukid). See võimaldab lennujuhil paremaid tingimusi teenindussõidukite jälgimiseks ja vähendab lennujuhtimise vea tõenäosust.

## 1.1 Andmed ja metoodika

Käesoleva lõputöö uurimisstrateegia näol on tegemist juhtumuringuga (*case study*) [2]. Uurimistöös lähtutakse induktiivsest lähenemisest, kus kõigepealt kogutakse andmeid uurimisprobleemi lahendamiseks ja seejärel tehakse tehnoloogia valik pragmaatilise ja nõuetest lähtuva lahenduse arendamiseks [3].

Uurimuse juhtumiks on olukord, kus majanduslikel põhjustel puudub väiksematel lennuväljadel maaseire süsteem ning lennujuhtimine toimub ainult raadioside ja visuaalnähtavuse vahendite abil. Selleks, et nimetatud olukorda lahendada, kasutab autor kombineeritud uurimismeetodeid, mis jagunevad andmekogumis- ja andmeanalüüsimetoditeks.

Andmekogumismeetodite hulka kuuluvad kvalitatiivne sisuanalüüs (juhendite, käsiraamatute, tehnilise kirjanduse, dokumentide ja artiklite analüüs) ning poolstruktureeritud ekspertintervjuu [4]-[6]. Nimetatud meetodeid rakendati lahenduse funktsionaalsete ja mittefunktsionaalsete nõuete määramiseks. Andmeanalüüsi meetodite hulka kuuluvad *cross-case* analüüs, mida kasutatakse intervjuude analüüsimisel, ja kvantitatiivne võrdlev analüüs (programmi tulemuslikkuse hindamiseks ja järelduste tegemiseks) [3], [7]. Uurimuse käik on esitatud joonisel 1. Intervjuude analüüsiks vajalikud andmed on esitatud lisas 2 ja lisades 3-4 esitatud tabelites 6-7.



Joonis 1. Analüüsi ja arenduse käik.

## 2 Ülevaade probleemist

Maapealne lennujuhtimine, kui osa tervest lennujuhtimise funktsioonist, on olulise riski faktoriga tegevus. Nimetatud riski kujundavad lennujaama liiklusalal liikuvad objektid, nagu õhusõidukid, tehnilised teenindussõidukid ja üksikud liiklejad. Kõige suuremaks väljakutseks on tagada ohutu ja sujuv nimetatud objektide liikumine maismaal, sünkroniseerides liiklust õhus.

Eeltoodud riski maandumise maatriks sisaldab gradatsioonis olevaid riske koos nende mõju vähendavate faktoritega. Enamus lennujaamadest peab suuremaks riskiks õhusõiduki kokkupõrget mistahes objektiga õhus ja maismaal. Selle vältimiseks integreerib lennujaam erinevaid lahendusi ja kõige tähtsam osa lahendusest on just õhusõiduki jälgimise võimekuse digitaliseerimine.

Olenevalt lennujaama kui infrastruktuuri läbilaskevõimest ja taristu suuruselt valitakse seire seadmed. Nimetatud seiresüsteemide funktsionaalsus peab olema maksimaalselt kuluefektiivne, kuna seiresüsteemid on piisavalt kallid, et iga lennujaam lubaks endale mistahes lahendusi. Siinkohal on oluline mainida, et paljud väiksemat tüüpi lennujaamad, nagu näiteks väikeste piirkondade või hobilennujaamad, ei suuda majanduslikel põhjustel soetada kompleksseid süsteeme piirdudes minimaalse funktsionaalsusega. Sellest tulenevalt jäävad integreerimata maismaaseire süsteemid, mis aitaksid hoida kontrolli all olulise riskifaktoriga tegevusi nagu lennuvälja hooldustegevus, lennuvahendi tankimis- või päästeoperatsioonid, õhusõiduki jäätõrje ja pinnakatte remont. Süsteem ei pea olema sertifitseeritud ja seda ei pea tingimata kasutama õhusõidukite hajutamisel.

Eeltoodud probleemi lahendamiseks uurib autor erinevaid võimalusi, mis aitaks lennujaamal digitaliseerida maismaa teenindussõidukite liikumist eesmärgiga tagada lennujuhile paremat olukorratadlikkust. Autori uurimistöö analüüsi fookusesse jäävad erineva funktsionaalsuse astmestikuga süsteemid, mis on mõeldud erineva suurusega lennujaamale. Vaatluse alla ei kuulu suuremahulised maaradaritega varustatud süsteemid kontseptuaalse ülesehituse ja kõrge hinnaklassi tõttu [8].

## 2.1 Olemasolevad lahendused

### 2.1.1 SAAB AVT süsteem

Üheks maailmatuntuks lennunduses kasutatava tehnoloogia tootjaks on Rootsi ettevõtte Saab. Nimetatud firma pakub erinevaid komplekslahendusi paljudele lennunduse valdkondadele. Saab ettevõtte kõige populaarsemaks lennunduse tooteks on kaasaegne Saab Gripen hävitaja, mis on disainitud sõjaliste õhutorbe missioonide tarbeks.

Lisaks eelnevale toodab Saab infosüsteeme lennujuhtimisteenistusele. Nende hulka kuulub Saab AVT (inglise keeles *Airport Vehicle Tracking*) süsteem, mis lahendab maapealsete teenindussõidukite jälgimise probleemi. Nimetatud tüüpsüsteemi andmeedastuse komponendiks on VL-1090 ja VL-4G mobiilne seade, mis kinnitatakse sõiduki kere külge ning ühendatakse sõiduki vooluvõrku. Käesoleva süsteemi kindlaks eeliseks on ilmastikukindlus, mida arendati põhjamaade suhteliselt külma ilmastiku tingimustes ja mida on tootja tõestanud mitmetes põhjamaade lennujaamades. Veel üks põhiline eelis seisneb selles, et seade vastab mitmele valdkonna standardile ja omab ühilduvust paljude 1090 MHz sagedusel või mobiilsidevõrgul põhinevate seiresüsteemidega. [9]

Eeltoodud süsteemi suuremaks nõrkuseks on suhteliselt suured soetus- ja eksploatatsioonikulud. Esiteks, kui lennujaamas puuduvad eeltoodud parameetritega seiresüsteemid, siis tuleb hankida Saabi komplekslahendus, mis koosneb mobiilsetest andmeedastusseadmetest, seiresüsteemist ja tarkvarast, mis on kaitstud organisatsiooni autoriõigustega ja seetõttu suure tõenäosusega kohalik arendusmeeskond ei saa seda muuta enda spetsiifikale vastavaks. Teiseks, mistahes teiste süsteemidega ühildumine ja seejärel süsteemi versiooni uuendamine tähendab täiendavat lisakulu meeskonna koolituste, tehnilise personali sisseostmise või muu lisateenuse näol.

### 2.1.2 ADB Safegate ACEMAX

Lisaks eraldiseisvatele süsteemidele, nagu Saab AVT, eksisteerivad komplekslahendused. Üheks komplekslahenduseks on ADB Safegate ACEMAX, mis hõlmab endas erinevaid teenuseid nagu nt *Advanced Surface Movement Guidance and Control System* ja *ARG Advanced Routing and Guidance*. [10]

Tegemist on moodularhitektuurist koosneva süsteemiga, mis hõlmab endas lisaks teenindussõidukitele ka lennuvahendite seirevõimalusi. Nimetatud tootja lahendused on kasutuses paljudes lennujaamades üle maailma ja üheks eestimaiseks näiteks on AS Tallinna Lennujaam. [11]-[13]

Tehniline realisatsioon on sarnane Saab lahendusega. Süsteem koosneb andmesaatjast, vastuvõtjast ja tarkvarast, mis töötleb andmepakette ja kuvab sõidukite andmeid lennujuhi arvutiekraanile. Mainimist väärt on aga lisafunktsioonide olemasolu:

- Andmete arhiveerimine ja situatsiooni taasesitamine kaardil, mille abil saab näha sõidukite liikumist ajaloos ja teha visuaalset analüüsi.
- Statistiliste andmete kogumine ja analüütiliste tööriistade olemasolu, mille abil on võimalik hinnata lennujuhi protseduuride efektiivsust ja teha ettepanekuid protsesside optimeerimiseks.
- Kaardikihi redaktor, kus on võimalik paigaldada kasutajale sobiv kaardikiht.
- Icoonide redaktor, mille abil on võimalik kujundada sobiva ja kaardil esitatava sõiduki ikooni (nt lisada lisaks kutsungile sõidukiirust ja suunda).

Analüüsi tulemusel selgus, et antud näide sobib keskmisele alustavale või uuele tasemele suubuva aktiivse piirkonna lennujaamale. Põhjuseks on suhteliselt suured soetus- ja eksploatatsioonikulud ning süsteemi keerukus. Madala läbilaskevõime või suhteliselt väikse maismaa operatsiooni pindala tõttu on paljudele väikestele lennujaamadele enamused ACEMAX funktsioonid pigem soovituslikud kui kriitilised.

### **2.1.3 Motorola DMR**

Maapealsete sõidukite seireks kasutatakse laialt DMR (inglise keeles *Digital Mobile Radio*) ehk digitaalse mobiilse raadiojaama standardit. Üheks nimetatud standardi esindajaks on Motorola Mototrbo põlvkonna raadiojaamad, mis on kasutuses erinevates tsiviil operatiivüksustes üle maailma, nagu näiteks kiirabi või politsei teenused. Põhiliseks erinevuseks analoograadioside jaamade ja DMR vahel on andmeedastusvõimalus võrgu teel. [14]

DMR tehniliste omaduste poolest konventsionaalselt erineb käesoleva uurimistöo eelmistes osades kirjeldatud lahendustest. Kõige olulisemad erinevused on:

- **Mobiilsus.** DMR jaama on võimalik kasutada jalgsi liiklejate jälgimiseks, kuna jaam on varustatud akuga ja ei vaja toidet sõiduki vooluvõrgust. Eriti oluline funktsioon, mis tuleb kasuks erinevate remont- või hooldustööde teostamisel, kuhu kaasatakse rohkem inimesi, suurendades riskifaktorit. Lisaks, funktsioon aitab hoida kokku kulud, kuna ühte seadet saab kasutada erinevates sõidukites, sest raadiojaam liigub koos sõiduki juhiga.
- **Häälside.** DMR on analoograadiojaama edasiarendus, mille algne põhifunktsioon on häälsidevõimaluse realiseerimine raadioside teel. DMR abil on võimalik kasutada kitsamaid ribalaiuseid, suurendades selleks kanalite hulka.
- **Andmeside krüpteerimine.** Digitaalse andmeedastuse juures, eriti aeronavigatsiooni jaoks kriitilise info edastamisel, on oluline andmete adekvaatsuse ja terviklikkuse kaitse, mida saab realiseerida kasutades DMR erinevaid võimalusi. Selleks ei ole vaja tarkvaraarendaja abi, kuna krüpteerimise funktsiooni sätteid saab seadistada DMR administreerimise tarkvara abil vastavalt organisatsiooni poolt esitatud nõuetele.
- **DMR kaks ühes – saatja ja vastuvõtja funktsioon.** Eelmise lahenduse ülesehitus nõudis seiresüsteemi, mis võtab vastu seadmest välja saadetud andmepaketid. DMR jaam suudab saata ja vastu võtta andmepakette, st igat jaama on võimalik seadistada dispetšeri jaamaks, mis saadab andmed edasi töötlevasse programmi (kasutajaliides või mõni muu süsteemi osa).

Nagu iga analüüsitud süsteem omab ka DMR puudusi. Kõige suuremaks puuduseks autori arvates on akulaadimise vajadus. Motorola DMR kestab orienteeruvalt kaheksa tundi andmeside režiimil, st on oluline tagada aku vahetamise või perioodilise aku laadimise võimalus. Aku iga on limiteeritud ja sõltuvalt laadimisetsüklitest on vaja aeg-ajalt akud välja vahetada.

## **2.2 Lahenduse valik**

Antud uurimistöo raames käsitletud probleemi kirjelduse kohaselt on oluline välja tuua, et maismaasõidukite liikumise andmeid kuvav programm peab sobima ühe lennurajaga lennujaamale, mille lennuraja pikkus on üle 1800 m (ICAO kood nr 4). Samuti peab



süsteem sobima kahe lennurajaga lennujaamale, mille pikkus on kuni 1800 m. [15] Võttes arvesse olemasolevaid ja eelpool toodud tüüplahendusi otsustas autor maaseire süsteemi välja valida järgmiste põhimõtete alusel:

- Süsteem peab sisaldama minimaalset arvu komponente. See aitab hoida kulusid kokku süsteemi administreerimisel, koolitamisel ja integreerimisel.
- Lahendus peab sisaldama võimalikult palju kriitilise tähtsusega funktsioone ja võimalikult vähe lisafunktsioone, mis ei ole kriitilised, aga mille eest on tarvis täiendavalt maksta.
- Süsteem peab sobima kasutamiseks igas maismaa sõidukis.
- Lahendus peab olema ümber konfigureeritav. Kaardikiht, ikoonid ja kuvatavate telemetria andmete hulk peab olema muudetav vastavalt kasutaja soovile. Muudatused viib sisse kas administraator või kasutaja.
- Süsteemi peab olema võimalik mastabeerida.

Võttes arvesse eeltoodud valikukriteeriumeid otsustas autor valida välja DMR tehnoloogial põhineva tüüplahenduse. Uurimuse raames kasutatakse DMR süsteemi järgmistel põhjustel:

- Süsteem pakub kahte kriitilist võimekust – digitaalne/analoo häälside ja maismaaseireks vajaliku andmeside.
- Lahendust saab mastabeerida, DMR on mobiilne, st seda saab kasutada liikleja ja sõiduki tasemel. Iga uue DMR jaama lisamisel ei ole vaja teha installeerimistöid, vaid piisab ainult jaama seadistamisest.
- Andmesidepakette vastuvõetav programmi osa võib olla arendatav kohaliku arendusmeeskonna poolt, vastavalt organisatsiooni vajadustele. See annab liidestusvõimalust mistahes olemasoleva süsteemiga.
- Süsteem on testitav, kuna autoril on kasutada üle 10 Motorola DMR jaama.

## **3 Lahenduse analüüs**

### **3.1 Tehniliste nõuete kirjeldus**

Käesoleva rakenduse nõuete paika panemisel arvestatakse kahe põhilise põhimõttega. Esiteks, tehniline lahendus peab toetama Motorola Mototrbo digitaalsest raadiojaamast tulenevate andmepakettide töötlemist. Teiseks, arendusprotsess peab olema teostatud maksimaalse kuluefektiivsusega, st arendamise, testimise ja integreerimisega seotud kogukulu peab olema minimaalne.

Lisaks eelnevale on tähtis mainida, et antud lahendus on mõeldud erinevate organisatsioonide poolt kasutatavatele süsteemidele, mis kasutavad oma kasutajaliidest ja kasutajahalduskeskkonda, seega võetakse antud uurimistöös kasutusele vabavaraline kaardipõhi, mille peal demonstreeritakse rakenduse tööd.

Tehniliste nõuete määramisel lähtuti lõppkasutaja arvamusest, mida analüüsiti ekspertintervjuude abil. Selleks, et intervjuu käigus kogutud andmed oleksid rakendatavad erinevatele lennujaama üksustes, viidi läbi poolstruktureeritud intervjuud rajameistri ja lennujuhiga.

#### **3.1.1 Funktsionaalsed nõuded**

Funktsionaalsed nõuded tavakasutaja tasemest lähtuvalt:

1. Rakenduses on valitud kaardipõhi, kus on ära toodud vähemalt lennuvälja ruleerimisteed, õhusõidukite perroonid ja lennurada.
2. Lahenduses kasutatav kaardipõhi peab olema sisse ja välja suunitav.
3. Iga raadiosaatja, mida lülitakse sisse rakenduses kasutatavale raadiosidekanalile, peab olema kasutajale nähtav. Antud funktsioon tagab iga teenindussõiduki fikseerimise käimas oleva lennuvälja operatsiooni kontekstis.
4. Iga raadiosaatja, mis hetkel on rakenduse poolt kasutatava kanali peal ja mida lülitakse välja või millel muudetakse raadioside kanal, peab olema kaardipõhjast eemaldatud. Sellest tulenevalt näeb kasutaja, millised teenindussõidukid on alalt väljunud.

5. Iga raadiosaatja, mis siseneb rakenduse poolt kasutatavale sidekanalile, peab edastama liikumiskiiruse, suuna ja geograafilised asukohakoordinaadid *lat/long* formaadis. Selle abil genereerib rakendus kaardipõhjal liikuva ikooni, kus on esitatud eeltoodud andmed.
6. Rakenduses peab olema võimalus teha väljavõtte liiklusalal liikunud sõidukite andmete kohta.

Funktsionaalsed nõuded administraatori tasemest lähtuvalt:

1. Digitaalsest raadiosaatjast vastuvõetud andmesidepakettid töödeldakse organisatsiooni poolt hallatavas kohalikus arvutis või serveris ning saadetakse edasi organisatsioonisisese võrgu kaudu. Töödeldud andmete turvalisuse tagamiseks ei kasutata ühendust välisvõrguga andmeside pakettide säilitamiseks, seega peab olema tagatud sisevõrgus andmeedastuse ja säilimise funktsioon.
2. Töödeldud raadiosaatja andmepakett peab olema välja saadetud kasutades JSON standardit, et tagada elemendi liidestuse funktsioon erinevate transpordi ja seire infosüsteemidega.
3. Olenemata organisatsioonipoolsest infosüsteemi ärioloogikast peab olema liidesel ühendus andmebaasiga, et tagada ligipääs andmebaasi funktsioonidele realiseerimaks kasutajapoolseid funktsioone.

### **3.1.2 Mittefunktsionaalsed nõuded**

Mittefunktsionaalsed nõuded tavakasutaja tasemest lähtuvalt:

1. Kuvatav tekst ja andmed peavad olema inglise keeles (lähtuvalt rahvusvahelisest lennujuhtimise keelest) [16].
2. Kuvatavad andmed peavad kasutama rahvusvahelist SI mõõteühikute süsteemi.
3. Lahendus peab töötama kasutades vähemalt kolme Windows operatsiooni süsteemi peal töötavat veebilehitsejat: Google Chrome, Internet Explorer, Microsoft Edge.
4. Lahendust kasutatakse ainult statsionaarse kontori arvutitehnikal. Esialgu ei pea lahendus toetama mobiilseid seadmeid.

Mittefunktsionaalsed nõuded administraatori tasemest lähtuvalt:

1. Koodibaasis sisaldavad elemendid (nagu funktsioonid ja klassid) peavad olema kommenteeritud.
2. Funktsionaalsus peab olema testitud ja testid peavad olema omakorda kommenteeritud.
3. Koodibaasi kommenteerimise keeleks on inglise keel.

## 3.2 Tehnoloogia valik

### 3.2.1 Programmeerimiskeele valik

Käesolevas lõputöös keskendutakse süsteemi programmeerimisele (inglise keeles *systems programming*), mille põhieesmärgiks on luua teenuseid standardiseeritud programmidele, võimaldades telemeetria andmete jõudmine DMR allikast organisatsiooni poolt rakendatud kasutajaliidesesse. Tegemist on juhtprogrammi (draiveri) arendusega, mis võtab vastu, töötleb ja saadab andmepaketid teisele süsteemi osale. Lisaks eelnevale luuakse kasutajaliides demonstreerimaks nimetatud teenuste funktsioneerimist. Lahendus tervikuna võib olla kasutatud rakendustarkvara näol (inglise keeles *application program*), kuid minimaalse funktsionaalsusega. [17]

Juhtprogrammi arenduseks tihti kasutatakse madal- või kesktaseme programmeerimiskeeli. Üks tuntumatest keeltest on C imperatiivne, protseduuriline ja suhteliselt kiire keel. Nimetatud keelel on mitmeid tugevusi, nagu nt madal- ja kõrgtaseme programmeerimise võimekuse elemendid (inglise keeles *low and high level programming*), mis käesoleva uurimistöö jaoks on eriti suure tähtsusega, kuna juhtprogrammi töö seostatakse DMR madaltaseme liidese ja kasutajaliidese tööga. C keelel on ka mitu puudust, nagu kõrge õppimiskeerukus ja raske loetavus, objektorienteeritud programmeerimise kontseptsiooni puudumine (nt abstraktsioon ja polümorfism, inglise keeles *abstraction and polymorphism*), keeruline veatuvastus. [18]

Eeltoodud C keele edasiarenduseks peetakse C++ programmeerimiskeelt. Mitmed C keele puudused on lahendatud C++ keeles. Nimetatud keel hõlmab endas objektorienteerituse ja polümorfismi elemente. Lisaks eelnevale, C++ omab C keele funktsionaalsust juhtida mälu madaltasemel (*low-level memory manipulation*), mis tagab programmi suhteliselt kõrge kiiruse ja efektiivsema mälukasutuse. C++ mõnedeks

puudusteks võib välja tuua keskmisest kõrgema õppimiskeerukuse ja dünaamilise mälu eraldamise võimekuse puudumist (inglise keeles *dynamic memory allocation*). [19]

Süsteemi programmeerimise näidetes võib kohata Python programmeerimiskeelt. Näiteks võib tuua USA Riikliku Aeronautika- ja Kosmosevalitsuse NASA projekti nimega X-Plane Communication Toolbox, mille abil edastatakse erinevate sõidukite andmeid lennusimulaatori keskkonnas ja tagatakse logimise, lennuprofiilide visualiseerimise ja erinevaid autopiloodi funktsioone. [20] Python järgib objektorienteerituse printsiipe ja seda loetakse kõrgetaseme programmeerimiskeeleks. Lisaks on Python keelt suhteliselt kerge õppida ja koodist aru saada [21].

Uurimistöo analüüsi skooopi kuulub ka Rust mitmeparadigmiline keel. Rust süsteemiprogrammeerimise keelt kasutatakse turvaliste lahenduste arendamisel, kuna viimane on ülesehitatud mälu turvalisuse aspektist lähtuvalt. Rusti tugevuseks peetakse keskmisest kõrgemat sooritustaset ja kiirust. Vaatamata Rust keele süntaksi sarnasusele C ja C++, loetakse teda keskmise õppimiskeerukusega keeleks. Rust järgib objektorienteerituse ja polümorfismi mustreid. [22]

Eeltoodud tehnoloogiate kirjeldust arvesse võttes otsustas autor kasutada süsteemi programmeerimiseks kahte keelt – Python ja Rust (vt tabel 1). Kuna autoril on eelnevalt olnud kokkupuude Rust keelega süsteemi programmeerimise kontekstis, siis autor soovib arendada antud oskust ning seetõttu soovib lisaks probleemi lahendamisele saada paremat programmeerimise oskust Rust keeles. Python keel valiti põhjusel, mille kohaselt autoril on varasem kogemus Python keelega ja nimetatud keele abil on võimalik Rust koodi testida ja luua erinevaid kliendi rakendusi, mis simuleerivad DMR liikumist kaardikihil.

Tabel 1. Programmeerimiskeelte võrdlus.

<b>Tehnoloogia</b>	<b>Õppimiskeerukus</b>	<b>Kogemus</b>
C	Kõrge	Puudub
C++	Kõrge	Puudub
Python	Madal	Hea
Rust	Keskmine	Halb

### 3.2.2 Andmebaasi valik

Lõputöö raames arendatakse moodulit, mis algselt hakkab asuma ühes arvutis lennujuhtimistornis. Võttes arvesse fakti, et enne mooduli kasutusvõttu tuleb see korralikult ära testida ja UAT on reeglina suhteliselt ajamahukas, siis mooduli arenduse algusfaasis kasutatakse SQLite failipõhist andmebaasi. Nimetatud andmebaas on piisavalt aktuaalne ja suhteliselt levinud tehnoloogia, mis toetab tarkvara kiiremat ja mugavamat arendamist [23].

Nimetatud andmebaasi eelis seisneb selle suhtelises lihtsuses. SQLite ei nõua täiendavaid ressursse, mis teeb ta majanduslikult efektiivseks. Lisaks eelnevale, SQLite tagab arendajale sõltumatus ja ei nõua serveri olemasolu [23].

Olenevalt kasutatavatest raamistikest ja programmeerimistehnoloogiatest on võimalik tulevikus asendada SQLite andmebaasi muu, organisatsiooni jaoks meelepärasema, tehnoloogiaga. See annab võimaluse kasutada andmebaasi ka teistele kasutajatele.

### 3.2.3 Arenduskeskkonna valik

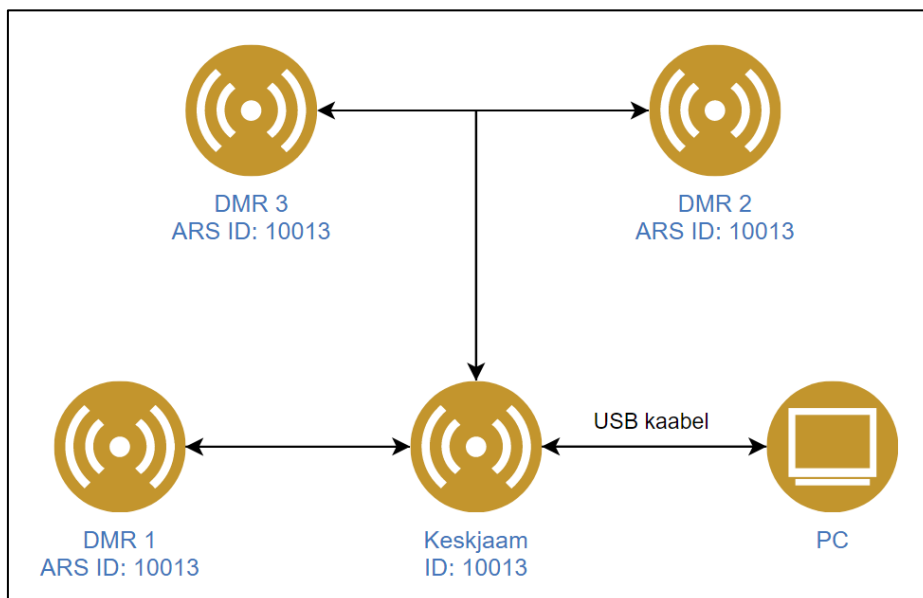
Käesoleva uurimuse raames kasutatakse mitmeid keskkondi. Programmeerimiseks Rust koodis kasutatakse MS Visual Studio Code keskkonda [24]. Nimetatud tööriist omab piisavalt teke ja laiendeid (inglise keeles *plugin*), et arendus Rust keeles oleks mugav ja kiire.

Arendamiseks Python keeles kasutatakse samuti MS Visual Studio Code programmi, mis erinevate laienditega toetab Python keele kasutust. Antud lahenduse kasutuselevõttu toetab autori endine akadeemiline kogemus Python arendusprojektide juures.

Antud projekti raames kasutatakse Linux Ubuntu 16.04 operatsioonisüsteemi. Nimetatud operatsioonisüsteem valiti erinevatel põhjustel: süsteem on vabavaraline, ei nõua suurt arvutijõudlust, süsteemil töötab eelnevalt toodud arenduskeskkond ja autoril on kogemused Ubuntu süsteemil töötamisega.

## 4 Arhitektuur

Mooduli ülesehituse määramisel lähtuti DMR standardi eripärast. DMR on võimeline ise tekitama võrgu, kuhu saavad ühineda teised, vastavalt programmeeritud, DMR jaamad. Nimetatud võrgu keskpunktiks on keskjaam (*control station*) ehk kontrolljaam, mis võtab vastu teiste jaamade andmepaketid ja edastab neid kas otse serverisse või arvutisse USB liidese kaudu (vt joonis 2). Antud juhul kasutatakse USB lahendust, tagades andmetele parema ühenduse ja turvalisuse. Selleks ühendatakse keskjaam arvutiga Motorola USB kaabli abil.



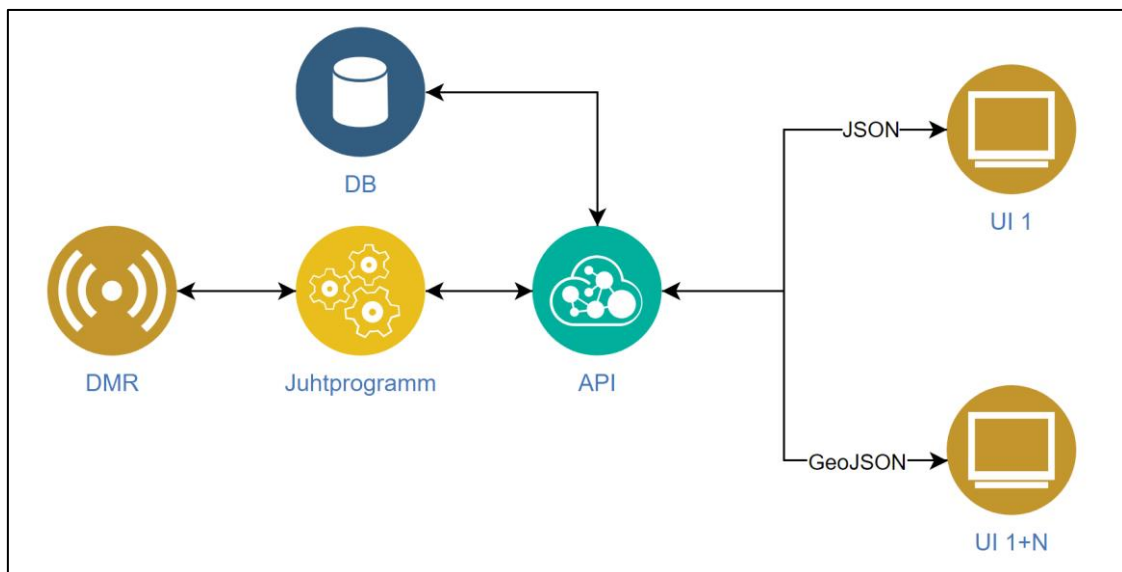
Joonis 2. Ühendus DMR võrgu ja arvuti vahel.

Keskjaama peamine ülesanne on Motorola võrgu monitooring. Selle tulemusel jõuavad arvutisse kontrollitud ja identifitseeritud andmed igalt võrguühenduses olevalt jaamalt. DMR võrk on seadistatav, st on võimalik reguleerida pakettide sisu ja pakettide edastamise ajalisi intervalli.

Eelpool nimetatud arvutis on installeeritud juhtprogramm, mille ülesanne on vastavalt eelsätetud konfiguratsioonifailile käivitada kontrolljaama monitooringu funktsioon, mille abil luuakse andmesideühendus keskjaama ja juhtprogrammi vahel.

Peale seda, kui juhtprogramm saab kätte keskjaamast edastatud andmepaketid, toimub andmepaketi kontroll ja töötlemine. Seejärel saadab juhtprogramm kontrollitud andmed API teenusele. API elemendi poolt vastuvõetud andmed kirjutatakse andmebaasi, mis tagab andmete säilivuse ja võimaluse edaspidiseks kasutamiseks.

Lõputöö raames testitakse juhtprogrammi ja API funktsioneerimist ainult ühe kasutajaliidese abil, kuid tulevikus toimuva mastabeerimise eesmärgil on tarvis pakkuda andmeid mitmele kasutajaliidesele (vt joonis 3). Nimetatud olukord põhjendab API liidese arendamise valikut.



Joonis 3. Veebiteenuse arhitektuur.

Nimetatud API teenust kasutab kliendirakendus, kes vastavalt oma üleschitusele küsib andmeid jaamade kohta. Kuna antud moodul on loodud kaardirakendustele, siis on arukas tagada ühendus erinevate formaatide alusel, mida tavaliselt kasutatakse kaardirakenduste poolt. Enimlevinud standardite hulka kuuluvad nt JSON ja GeoJSON standardid.

Eeltoodud standardite abil on võimalik edastada struktureeritud andmed objekti kohta, nagu nt maimaasõiduki kiirus ja asukoht. Standardite valik annab rohkem võimalusi teenuse kasutamiseks võimalikult paljudes kaardirakendustes ja tagab süsteemide ühilduvuse.

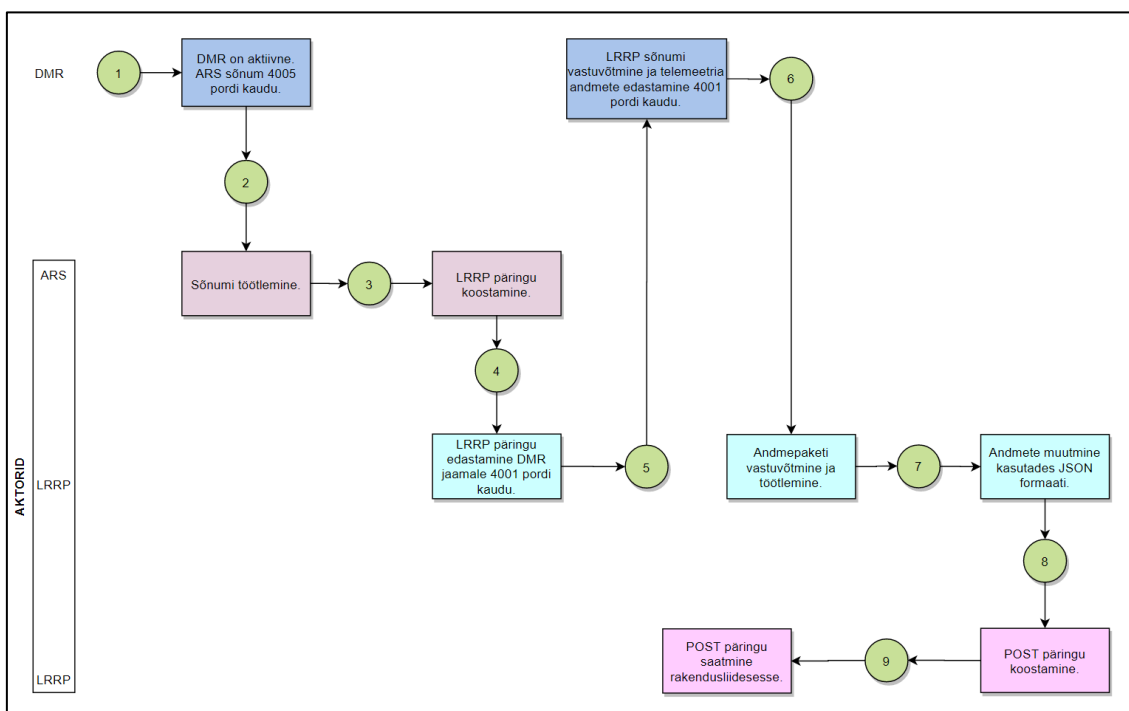


## 4.1 Juhtprogrammi ja rakendusliidese arendamine

### 4.1.1 Juhtprogramm

Uurimistöö raames analüüsitud ja arendatud juhtprogramm põhineb Motorola DMR tööpõhimõttel. Osa juhtprogrammi tööst põhineb varem kaitstud lõputöö tulemustel, mille raames tõestati tehnilist võimalust ühendada Motorola põlvkonna jaamad riigiasutuses arendatud tarkvaraga [25]. Selleks, et kirjeldada arendatud juhtprogrammi tööd, on tähtis mõista digitaaljaama võrguprotokolli aspekte. Esiteks on vaja tagada olukord, kus kontrolljaam oleks nõuetekohaselt seadistatud ja ühendatud arvutiga, kus töötab juhtprogramm. Teiseks peavad jaamad olema vastavalt programmeeritud.

Kui eeltoodud tingimused on täidetud, siis iga kord, kui kanalisse ühendatakse jaam, edastab viimane ARS sõnumi kontrolljaamale. Antud olukorras toimub jaama registreerimine võrgus. ARS protokolliga edastatud sõnum annab kontrolljaamale teada, et nüüd on võimalik pidada andmevahetust LRRP protokolliga abil ja saata detailsemaid andmeid. ARS ja LRRP protokollide töö on kirjeldatud joonisel 4.



Joonis 4. ARS sõnumite ja LRRP protokolliga voodiagramm.

Sellest tulenevalt peab juhtprogrammi arendamisel tagama ARS ja LRRP andmesidepakettide vastuvõtmist, genereerimist, saatmist ja kontrollimist. Lisaks eelnevale arvuti, milles töötab juhtprogramm, peab olema seadistatud andmepakettide

väljasaatmise marsruut vastavalt jaama seadistustele (st kindla lüüsi määramine jaama ja juhtprogrammi vahel).

Eeltoodud voodiagramm toetab juhtprogrammi struktuuri lahti seletamist. Ärioloogikast tulenevalt jaotati juhtprogramm neljaks struktuurseks osaks – ARS ja LRRP pakettide töötlemise, üldine Motorola pakettide töötlemise ja juhtprogrammi konfiguratsiooni osad. Igal osal on kindel funktsionaalsus ja tulenevalt Rust keele eripärast ei eksisteeri koodis ühtegi parameetrit või funktsiooni väljundit, mida juhtprogramm ei kasuta. Allpool on toodud juhtprogrammi struktuuri üldine kirjeldus.

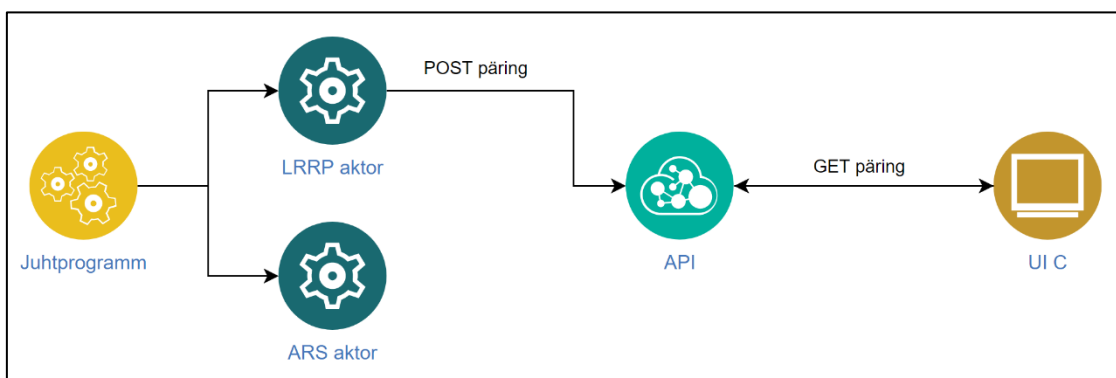
- ARS osa järgib Actor (edaspidi aktor) mudelit (*actor model*). See võimaldab arendada juhtprogrammi kasutades mitmeid iseseisvalt töötavaid aktoreid, mis omavahel koordineeritult töötavad. Aktori mudeli kasutamist võimaldab Actix teek, mis on võimeline genereerima terve aktorite süsteemi (*actor system*) ning kapseldama aktorite käitumist ja olekut. Rust aktori mudelis on võimalik kasutada kõike Rust andmetüüpe, mis teeb arenduse kiireks ja mugavaks. [26] Nimetatud funktsionaalsus võimaldab realiseerida ARS elemendi käitumist selliselt, et ta ei sõltuks teistest, nt LRRP aktorist. Kuna juhtprogramm vahetab sõnumeid DMR jaamaga, siis on vaja tagada erinevatele aktoritele erinevaid porte (nt ARS aktor kasutab porti 4005 ja LRRP on seadistatud portile 4001). ARS aktoris realiseeriti funktsioonid, mille abil on võimalik saata, vastu võtta ja töödelda Motorola ARS andmepakette. Juhtprogrammi ja Motorola DMR vaheliseks andmesideks kasutatakse UDP ühendust, selle realiseerimine on võimalik Tokio teegi abil [27]. UDP valiti põhjusel, et ta on oma alternatiivist kiirem. Lisaks eelnevale, ARS aktor kasutab dekodeerimise funktsioone, kus on realiseeritud andmete valideerimine ja vigade kontroll.
- LRRP elemendi tehnilise realisatsiooni muster on sarnane ARS osaga – LRRP arendati kasutades aktor mudelit. Nimetatud aktor omab samasugust funktsionaalsust nagu ARS, vahe on aga pakettides ja selle sisus (DMR võrk kasutab eraldi iseseisvat LRRP liidest, mis on erinev ARS omast). LRRP aktoris on lisatud funktsioon, mis on võimeline välja saatma POST päringu API aadressile.
- Motorola osa on mõeldud LRRP aktori funktsionaalsuse laiendamiseks. Motorola element dekodeerib ja valideerib LRRP sõnumeid. Selle tulemusel valmistatakse

struktureeritud objekt, mis sisaldab (DMR jaama) identifitseerimistunnust, objekti genereerimise aega, GNSS ülemaailmse satelliitnavigatsioonisüsteemi koordinaate laius- ja pikkuskraadide näol, kõrguse andmeid, kiirust ja liikumissuunda.

- Konfiguratsiooni fail on mõeldud kiireks juhtprogrammi seadistamiseks. Sellest tulenevalt on võimalik ühes keskses kohas ära määrata ARS ja LRRP pordid ja IP aadressid. Konfiguratsiooni faili lugemiseks realiseeriti eraldi abistav funktsionaalsus, mida kasutatakse juhtprogrammi käivitamisel. Juhtprogrammis ei ole seadistatud vaikimisi eelpool nimetatud seaded, st ilma konfiguratsioonifailita juhtprogrammi töö ei ole võimalik.

#### 4.1.2 Rakendusliides

Käesolevas lõputöös arendati rakendusliides eesmärgiga tagada DMR andmete säilitamine. Lisaks eelnevale on API tehnoloogia abil võimalik tagada ligipääs andmete kliendirakenduse poolt. Joonisel 5 esitatakse rakendusliidese tööpõhimõtte koos võimalikke ühendustega.



Joonis 5. Rakendusliidese töö.

Lähtuvalt mooduli arhitektuurist ühendatakse API külge LRRP aktor ja kasutajaliides. LRRP ühendus on tagatud DMR paketi andmete kirjutamiseks andmebaasi. Vastavalt moodulile esitatud tehnilistele nõuetele on tarvis säilitada kõik DMR andmed, vastasel juhul ei ole võimalik tagada andmete adekvaatsuse statistilise analüüsi läbiviimiseks.

LRRP tasemel kasutatakse Reqwest teeki [28]. Nimetatud teek võimaldab JSON objektide saatmist üle võrgu. Kuna LRRP tasemel dekodeeritakse DMR andmed, siis on arukas kasutada antud tasemel POST päringu tüüpi selleks, et edastada andmed konverteeritud kujul otse rakendusliidesesse.

Kasutajaliidese ühendamine API rakendusliideselega on võimalik mitmel moel. Kuna antud töö raames on tegemist prototüübi arendamisega, siis osa kasutajaliidese funktsioonidest ei realiseerita (nt statistiliseks analüüsiks realiseeritakse ainult päringud, mille tulemusel edastatakse andmed JSON kujul). Arenduse põhirõhk on aktiivsete jaamade info edastamisel, st need andmed on mõeldud kasutamiseks kaardirakenduses ja need on kriitilised mooduli testimise jaoks. Küll aga on võimalik vajadusel realiseerida andmete filtreerimist kasutajaliidese poole peal, kuid tulevikus on plaanis arendada täisfunktsionaalsus.

API realiseeriti Rust keeles kirjutatud Rocket veebiraamistikku kasutades. Nimetatud raamistiku ülesanne on tagada saabuvate päringute suunamine deklareeritud marsruutide kaudu. Rocket töötleb päringuid kolme etapi järgi. Kõigepealt teostatakse päringu valideerimine, mille abil kontrollitakse andmetüüpide korrektsust. Seejärel toimub päringu töötlemine, kus töödeldakse andmeid arendatud äriloojika alusel. Kolmandaks etapiks on vastuse saatmine. Selleks kasutatakse HTTP protokollit, kuhu lisatakse juurde vastusesse minevad ja eelnevalt töödeldud andmed. Nimetatud raamistik on valitud põhjusel, et see on piisavalt dokumenteeritud ja raamistikku on võimalik kasutada erinevate SQL andmebaasi tüüpide jaoks (nt PostgreSQL ja MySQL). [29] Päringute kirjeldust on võimalik näha tabelis 2 esitatud andmete põhjal.

Tabel 2. API päringud.

Päringu kirjeldus	Päringu tüüp	Ressurss
<i>DMR andmete kirjutamine andmebaasi</i>		
DMR andmete salvestamine	POST	/packets
<i>DMR andmete pärimine kaardirakenduse tarbeks</i>		
DMR andmete pärimine JSON kujul	GET	/packets
Mitme aktiivse DMR jaama andmete pärimine GeoJSON kujul	GET	/geo_json_packets
Mitme aktiivse DMR jaama andmete pärimine JSON kujul	GET	/json_packets
<i>DMR andmete pärimine statistiliseks analüüsiks (JSON)</i>		
Ühe DMR jaama andmete pärimine ID järgi	GET	/packets/<id>
Ühe DMR jaama andmete pärimine ID ja kuupäeva alusel	GET	/packets/date/<date>/<id>
Mitme DMR jaama andmete pärimine kuupäeva alusel	GET	/packets/date/<date>
<i>Haldus ja mooduli testimine (deaktiveeritud rakenduse töötamise ajal)</i>		
Kõikide andmete kustutamine andmebaasist	DELETE	/packets
Ühe jaama andmete kustutamine andmebaasist	DELETE	/packets/<id>

## 4.2 Kasutajaliidese arendamine

Uurimistöo skooopi ei kuulu täisväärtusliku kasutajaliidese arendamine. Sellest olenemata on tarvis arendada testkeskkond, kus oleks võimalik demonstreerida mooduli tööd. See aga ei tähenda, et lõputöö testkeskkonda ei oleks võimalik integreerida mõnda eksisteerivasse lahendusse.

Lõputöö autor valis testkeskkonnaks Google Maps API teenuse. Nimetatud teenuse kasutamiseks on mitu põhjust. Esiteks, teenus pakub vabavaralist (kuni teatud kasutamiskiiranguteni) aluskaarti mitmel erineval kujul. [30] Kuna antud lahendus peab sobima lennujuhtimisteenistusele, siis on oluline, et kaardikiht sisaldaks kriitilisi objekte, mida toodi välja ekspertintervjuudes. Analüüsides erinevaid kaardikihte selgus, et kõige parem on satelliidi põhi, kuna viimast uuendatakse piisavalt tihti ja võrreldes teiste kaartidega esineb isegi lennuvälja markeering, mis on äärmiselt oluline maapealses

lennujuhtimises, kuna annab võimaluse detailselt aru saada maapealse sõiduki asukohast ja tema liikumissuunast. Google Maps API teenuse vaade on esitatud joonisel 6.



Joonis 6. Testkeskkonna ekraanitõmmis.

Teine põhjus, miks valiti just Google Maps API teenus testkeskkonna arendamisel, on suhteliselt suur funktsionaalsus. Nimetatud rakendusliides annab valikuid kaardi sätete seadistamiseks, suurt ikoonide valikut ja erinevaid võimalusi objekti lisainformatsiooni esitamiseks.

Kolmas põhjus seisneb selles, et Google Maps API paljud tööriistad on arendatud JavaScript keeles. Sellest tulenevalt on võimalik kasutada lisaks API teenuse funktsionaalsusele ka JavaScript võimeid. Üks kaasnevatest eelistest on JavaScripti integreerimise paindlikkus, mis suurendab testkeskkonna kasutamist tuleviku perspektiivis. [31]

Kaardipõhja seadistamisel on oluline määrata vaikimisi parameetrid, mille abil genereeritakse kaardivaade. Antud ülesandepüstitusest tulenevalt on oluline, et kaart oleks staatiliselt ühes konkreetses vaates, kuid vajadusel saab seda muuta. Selleks on vaja paika panna järgmised sätted.

- Kaardipõhja keskkoha koordinaadid, mis võetakse aluseks uue kaardi genereerimisel ning millega tagatakse konkreetne kaardi staatiline vaade.
- Sisse suumimise aste, mis tagab maksimaalselt suure kaardivaate.

- Kaardipõhja pööramise nurk, mille abil keeratakse kaart meelepärasemaks vaateks. See on eriti oluline ning seda on välja toodud ekspertintervjuudes. Kui kaart ei ole loomulikul kujul, siis see võib takistada olukorrast arusaamist.
- Kaardipõhja tüüp ja kaardi resolutsioon.

Lisaks kaardipõhjale on tarvis arendada vajalik funktsionaalsus ikoonide kuvamisel. Selleks arendati funktsioon, mis pöördub mooduli API poole ning saadab päringu objekti andmete saamiseks. Siinkohal on tähtis mainida, et ikoonid ehk markerid peavad muutma oma asukohta vastavalt maismaa sõiduki asukoha muutumisele. Selleks on arendatud funktsioon, mis teatud ajalise intervalliga uuendab markerite arvu ja positsiooni. Oluline on tagada, et iga marker iseloomustaks konkreetset ja kindlat sõidukit.

Ikooni informatsiooni kuvamiseks arendati funktsioon, mis võtab vastu andmebaasis salvestatud objekti, valideerib selle ja muudab markeri objektiks. See annab võimaluse kasutada objekti erinevaid välju, nagu kiirus ja identifitseerimise number, ning sisestada neid andmeid disainitud HTML vormi. Lisaks eelnevale, on võimalik kasutada infoakna funktsiooni, kuhu sisestatakse vajalik objekti lisainformatsioon.

### **4.3 Administreerimine**

Lõputöö raames arendatav moodul paigaldatakse kõigepealt ühte arvutisse. Sellest tulenevalt kõik mooduli osad, juhtprogramm ja rakendusliides, töötavad algselt ühes arvutis enne, kui funktsionaalsus on täielikult läbi testitud. Nimetatud arvutis peavad olema installeeritud vajalikud tööriistad, mida tuuakse välja tehnoloogia valiku kirjelduses.

Mooduli haldamisel ja uute jaamade programmeerimisel tuleb jälgida marsruutide sätteid. Peab olema tehtud jaamade ja mooduli portide seadistamine, et tagada ühendus aktorite ja jaamade vahel. Lisaks on tarvis seadistada lüüs, mille kaudu programm edastab DMR võrgu IP aadressidega sõnumeid. DMR võrgu konfiguratsiooni loeb välja Motorola jaamade programmeerimise tarkvara abil. Täpsemad sätted on esitatud tabelis 3.

Tabel 3. DMR jaamade sätted.

<b>Parameetri nimetus</b>	<b>Väärtus</b>
<i>Liikuv jaam</i>	
GNSS	GPS/QZSS/GLONASS, On
ARS radio ID	10073
ARS UDP port	4005
Location server UDP port	4001
<i>Kontrolljaam</i>	
Radio ID	10073
GNSS	GPS/QZSS/GLONASS, On
Data modem system type	Digital
Forward to PC	Via USB
ARS UDP port	4005
Location server UDP port	4001

Administreerimisel on oluline teha andmetest varukoopia. Mooduli prototüübi faasis ei ole see otseselt nõutud, kuid tuleviku perspektiivis on varukoopiate tegemine administraatori ülesanne. API rakendusliideses realiseeriti andmete kustutamise funktsioone, seda kahes versioonis – ühe konkreetse jaama andmete kustutamine või kõikide jaamade andmete kustutamine. Neid funktsioone tuleb käsitsi sisse lülitada ja vajadusel kasutada.



## 5 Testimine

### 5.1 Ühiktestimine

Lõputöö raames arendatud moodulit testitakse ühiktestide abil. Selleks jaotati testid kahte loogilisse gruppi – juhtprogrammi ja rakendusliidese testid (vt tabel 4). Kuna autoril olid kätte saadavad osa varem Python keeles kirjutatud testidest, mida oli võimalik muuta ja seejärel sobitada käesoleva juhtprogrammi testimiseks, siis otsustati aja kokkuhoiu mõttes arendada osa teste Python keeles.

Lisaks eelnevale, Python keeles on arendatud mitmeid mugavaid ja antud projekti jaoks aktuaalseid teke, mille abil on võimalik testida madaltasemelisi võrguliidese protsesse. Nendest näiteks võib tuua Socket ja Scapy, mida saab kasutada võrgupakettide saatmiseks ja vastuvõtmiseks ning mis toetab mitmeid protokolle [32]-[33]. LRRP pakettide genereerimiseks kasutati projektsioonide ja koordinaatide teisendamise teeki Pyproj [34].

Tabel 4. Juhtprogrammi ja rakendusliidese testitud funktsionaalsus.

Mooduli osa	Testitud funktsionaalsus
Juhtprogramm	ARS pakettide saatmine
	ARS pakettide vastuvõtmine
	LRRP pakettide saatmine
	LRRP pakettide vastuvõtmine
Rakendusliides	Mitme aktiivse jaama andmete pärimine JSON kujul
	Ühe DMR jaama andmete pärimine ID järgi (JSON kujul)
	Ühe DMR jaama andmete pärimine ID ja Timestamp alusel (JSON kujul)
	Mitme DMR jaama andmete pärimine Timestamp alusel (JSON kujul)
	DMR andmete salvestamine
	Kõikide andmete kustutamine andmebaasist
	Ühe jaama andmete kustutamine andmebaasist

## 5.2 UAT

Uurimistöö lähteülesannetest tulenevalt arendatakse moodul lennujuhtimisteenistusele. Kuna tegemist on konkreetse sihtgrupiga, siis äärmiselt oluline on saada tagasisidet just lõppkasutajalt. Selleks viidi läbi UAT reaalse lennuvälja objekti peal.

UAT testimine on jaotatud kahte faasi: mooduli testimine kohapeal koostöös lennujuhiga ja testimine ilma arendaja (autori) kohalolekuta. Põhimõtteline erinevus seisneb selles, et esimese testimise ajal pannakse rakendus tööle, viiakse läbi kasutajale instruktaaž ja tehakse esimene test reaalsete maismaasõidukitega. Teise testimise ajal kasutab lennujuht moodulit iseseisvalt ja paneb kirja kõik probleemid ja kitsaskohad.

Selleks, et saada tulemusi mooduli töö adekvaatsusest, tehti test ühe, viie ja kümne maismaasõidukiga alal suurusega 3 ruutkilomeetrit (vt joonis 7, lisades 5-6 esitatud joonised 8-9). Antud testi eesmärk on võrrelda mooduli poolt esitatavate objektide koordinaadid reaalsete objektide asukohakoordinaatidega. Kui esimene test ühe DMR jaamaga näitab pigem objektide asukoha täpsust, siis testid viie ja kümne sõidukiga aitavad aru saada, kui suurt mõju avaldab DMR võrgule sõidukite arvu suurendamine.



Joonis 7. Vaade testimise ajal.

Testimise tarbeks tehti kontrollpunktide kaart, mida hiljem pidi maismaasõiduk läbima. Sõiduki juhil oli kaasas väljaprintitud kontrollpunktide kaart ja suurendatud aerofoto igast objektist, et aidata autojuhil paremini orienteeruda. Kuna tegemist on päris

lennujaamaga, siis olid abiks lennuvälja markeeringud, mille koordinaadid on võimalik välja võtta ArcPRO programmi abil (vt lisas 7 esitatud joonis 10) [35]. Sellest tulenevalt lepidi kokku, et kontrollpunktile saabudes asetatakse DMR kontrollpunktile, et oleks võimalik fikseerida reaalne DMR asukoht ja hiljem võrrelda seda lisas 7 esitatud koordinaatidega. Lisaks maismaasõidukile seadistati kontrolljaam ühes lennujaama administratiivhoones, kus toimus ka kogu testimise protsessi juhtimine.

Kontrolltulemuste analüüsist tuli välja, et moodul esitab andmed 3,7 meetri täpsusega, mis rahuldab lähteülesande tulemust. Kaardipõhi, ikoonid ja infoaknad olid selgelt loetavad ja täitsid oma eesmärgi. Kogu testimise ajal ei esinenud takistusi ega tõrkeid testimisprotsessis.

Üheks oluliseks tulemuseks, millega on arukas arvestada, on andmeside kiirus. Kui oletada, et DMR saadab iga sekundi tagant asukoha andmed, siis kasutajaliidesesse nad jõuavad 3-9-sekundilise viitega. Selle põhjuseks on leviala suurus ja erinevad signaali häired, mille tõttu pole võimalik saavutada suuremat kiirust. Kui oletada, et maismaasõiduki keskmine kiirus lennuväljal on 35 km/h, siis kolme sekundiga jõuab sõiduk läbida 29 meetrit. Kuna lõputöös arendatava mooduli eesmärk on olukorradeadlikkuse parandamine, mitte tööriista arendamine lennuvahendite hajutamisprotseduuride jaoks, siis nimetatud olukord on aktsepteeritav, arvestades ala suurust (3 ruutkilomeetrit).

## 6 Hinnang

### 6.1 Saavutatud kasutatavus

Lõputöös arendatud mooduli kasutatavust saab hinnata realiseeritud funktsionaalsuse abil. Uurimistöö kirjutamisel jõuti arendada kõik lõppkasutaja poolt määratud funktsioonid. Töö üheks oluliseks osaks oli lõppkasutaja mõju kasutatavusele. Mooduli funktsionaalsus on testitud nii arenduskeskkonnas kui ka lõppkliendi juures, mistõttu oli autoril võimalik tutvuda lõppkasutaja hinnanguga ja viia sisse täiendused.

Moodul on mõeldud kasutamiseks organisatsiooni sisevõrgus ja mooduli töö nõuab eelnevalt programmeeritud Motorola Mototrbo DMR komplekti (vähemalt üks kontrolljaam, üks võrku ühendatav jaam ja Motorola USB kaabel). Juhtprogrammi töö on testitud Linux operatsioonisüsteemidel, MS Windows operatsioonisüsteeme kasutati ainult andmete kuvamiseks kliendile, st viimasel pole juhtprogrammi töö testitud. DMR andmete vahendamine rakendusliidese ja kliendirakenduse vahel toimub JSON andmevahetusvormingu abil, mis võimaldab mitmekülgseid integreerimisvõimalusi.

Arendatud kood on struktureeritud loogilisteks osadeks, iga funktsioon on kommenteeritud inglise keeles. Nimetatud asjaolu muudab koodi hallatavaks ja tagab võimaluse edasiarenduseks. Lisaks eelnevale arendati testid, mille abil on võimalik paremini jälgida programmi tööd. Rust tehnoloogia on suhteliselt hästi dokumenteeritud, seega on võimalik detailsemalt tutvuda koodi erinevate funktsioonidega.

Uurimuse raames arendatud kliendirakendus kannab ainult testimise funktsiooni, st kasutajate loomist ja haldamist see ei võimalda. Kliendirakendus demonstreerib mooduli tööd ja hõlmab funktsionaalsust, mida lõppklient pidas vajalikuks. Kliendirakendust toetavad nõuetes esitatud veebilehitsejad ja selle kood võib olla kasutatud mistahes teise kliendirakenduse osana.

## 6.2 Uudsus

Uurimistöös arendatud mooduli uudsust saab hinnata erinevatest tasemetest lähtuvalt. Maaseiresüsteemi kui terviku näol on tegemist nii digitaalse monitooringusüsteemi kui ka häälside lahendusega. Mototrbo ettevõtte küll pakub omapoolset kasutajaliidest, kuid viimane on tasu eest. Lõputöös arendati moodulit akadeemilistel eesmärkidel ja selle kasutamine ei nõua kasutajapoolseid lisakulusid.

Lahenduse integreerimine ja kasutamine on võrreldes teiste maaseiresüsteemidega oluliselt lihtsam. Esiteks, integreerimine ei vaja sisseostetud tehnilist personali – DMR seadistamise ja kasutamise juhend on piisavalt detailne ja arusaadav. Teiseks, ei ole vaja installeerida maismaasõidukitesse erinevaid jälgimisseadmeid. Kolmandaks, haldamine ja edasiarendamine ei nõua lisakulusid ega sisseostetud tööjõudu, kuna süsteemi haldamine on õpetatav organisatsioonisiselt. Neljandaks, uute DMR jaamade hankimine ja integreerimine käimasolevasse võrku ei vaja sisseostetud teenuse kaasamist.

Mooduli arendamisel kasutati Rust programmeerimiskeelt, mille süntaksi inimloetavus on autori arvamusel palju kõrgem võrreldes teiste süsteemi programmeerimiskeeltega, nagu nt C või C++. See annab võimaluse arendada arusaadavamalt arhitektuuri, mis tagab parema hallatavuse.

Oluliseks osaks on lahenduse osade iseseisvus, st on võimalik kasutada ja modifitseerida süsteemi osi vastavalt oma infosüsteemi vajadustele. Nt on võimalik kasutada eraldiseisvalt juhtprogrammi osa, modifitseerides selleks DMR objektide konverteerimise ja edastamise osa. Tähtis on mainida, et viimane osa võib olla arendatud nii klient-server arhitektuuri kontekstis kui ka *desktop* rakenduse jaoks.

Motrbo perekonda kuuluvad DMR mudelid kasutavad Motorola standardiseeritud protokollid, mis annab võimaluse kasutada juhtprogrammi Mototrbo erinevate mudelite jaoks. Enne juhtprogrammi kasutamist tuleb veenduda, et kasutatava mudeli sätteid on vastavalt seadistatud ning mudel toetab ARS ja LRRP protokollide edastamist DP4600 mudeliga võrdses konfiguratsioonis.

### 6.3 Projekti investeringu maksumuse analüüs

Käesoleva lõputöö raames valminud lahenduse kui projekti realiseerimise tarbeks on vajalik arvutada selle eeldatav maksumus. Selleks kasutab autor erakogus olevaid andmeid projekti erinevate osade maksumuse hindamiseks. Projekti kulude struktuuri esitatakse tabelis 5.

Tabel 5. Projekti kulustruktuur.

Kuluartikkel	Kogus (tk)	Kulu tk kohta (EUR)	Kulu kokku (EUR)
Motorola DP4600	11	560	6160
DP4600 kaabel	1	45	45
Arvuti	1	300	300
6 pesaga laadija	2	450	900
Tööjõukulu (8h, tehnik)	1	65	65
Kogukulu (EUR)			7470

### 6.4 Edasiarendus

Autori hinnangul on antud moodulil mitmeid komponente, mida oleks mõistlik arendada enne, kui süsteem läheb päriselt kasutamiseks. Ühest küljest on moodul arendatud selliselt, et seda on sisuliselt kohe võimalik integreerida, kuid lähtuvalt erinevatest organisatsiooninõuetest võivad olla kasulikud ja vajalikud allpool loetletud võimalused edasiseks arendamiseks.

- Üheks turvariskiks võib osutada krüpteerimata protokollide kasutamine rakendusliidese ja süsteemi osade vahel (juhul, kui kasutatakse ühendust organisatsioonivõrgust väljapoole). Sellest tulenevalt on arukas integreerida nt transpordikihi turbeprotokoll (inglise keeles *Transport Layer Security, TLS*). Nimetatud funktsionaalsus annab võimalust krüpteerida edastatavaid andmeid ja tagada autentsus. Sellega aidatakse kaasa andmete käideldavusele, terviklikkusele ja konfidentsiaalsusele.
- Hetkel kasutab moodul Rocket teeki, mis loob failipõhise andmebaasi. Erinevates olukordades ei pruugi see vastata organisatsiooni nõuetele, seega tuleb arendada ühendust teistsuguse andmebaasiga. Kuna autoril oli võimatu prognoosida

andmebaasi eelistusi antud mooduli kasutamise osas, siis sai valitud Rocket teek põhjusel, et viimane omab ühilduvust paljude teiste andmebaasidega, nagu nt PostgreSQL ja MySQL.

- Erinevates olukordades võib osutada vajalikuks mooduli kasutamine mobiilses vaates. Nt rajameister, kelle töö põhineb suures osas liikumisel lennuvälja taristul, vajab mobiilset aktiivsusmonitori, mille abil on võimalik hinnata erinevate maismaasõidukite paiknemist lennuväljal. Lisaks eelnevale on võimalik aru saada, millises tööfaasis asuvad parasjagu lennuvälja hooldussõidukid. See muudab kiiremaks ja tõhusamaks tööde planeerimist ja juhtimist.
- Andmebaasis salvestatavatest andmetest on alati mõistlik teha varukoopia. Kui on tarvis tulevikus moodulit sertifitseerida riigi pädeva asutuse poolt, siis üheks tõenäoliseks kriteeriumiks, mida tuleb täita, on andmete varundamine.

## 7 Kokkuvõte

Käesoleva lõputöö eesmärk oli arendada ja testida lennuvälja teenindussõidukite jälgimise moodul eesmärgiga tagada parem olukorrateadlikkus lennuliiklusalas. Eesmärgi saavutamiseks arendati moodul, mis kasutab teenindussõidukis asuvast DMR käsiraadiojaamast saadud andmepakette, töötleb neid ja kuvab käsijaama aktiivset asukohta kaardirakenduses.

Töö analüüsi tulemustest järeldati, et lennujaamades saab kasutada Motorola Mototrbo perekonna raadiojaamu, mis kasutavad Motorola võrguprotokollid ning mille abil on võimalik saada pakettide sisu (nt asukoha lat/long koordinaadid, sõiduki kiirus ja liikumissuund). Eeltoodud sisu on võimalik objektina lahti parsida ning edastada rakendusliidesesse. Sellest tulenevalt arendati rakendusliidese funktsionaalsus selliselt, et API teenus oleks võimeline väljastama DMR jaamade andmeid võrgupäringute teel. DMR andmete päringuid kasutab kasutajaliides, et kuvada jaamade asukohad koos täiendava informatsiooniga kaardil. Nimetatud kaardirakendust simuleeriti Google Maps API vabavara tehnoloogia abil ning testiti ühiktestide ja UAT-ga. Analüüsi tulemusi kokku võttes valmis moodul, mis sisaldab endas DMR võrgutööd reguleerivat juhtprogrammi ja rakendusliidest, mis salvestab kõik DMR andmed ja saadab neid kliendirakendusele vastava päringu alusel.

Projekti võib lugeda õnnestunuks, kuna tulemusel saadud mooduli abil on võimalik kuvada teenindussõidukite positsiooni liiklusalal, vähendades sellega lennujuhi intellektuaalset ja psühholoogilist koormust ning tagades statistilist andmestikku erinevateks organisatsiooni protsessideks.



## Kasutatud kirjandus

- [1] A. Pape, D. Wiegmann ja S. Shappell, „Air traffic control (ATC) related accidents and incidents: a human factors analysis.“ [Võrgumaterjal]. Available: [https://www.faa.gov/about/initiatives/maintenance\\_hf/library/documents/media/human\\_factors\\_maintenance/air\\_traffic\\_control\\_\(atc\)\\_related\\_accidents\\_and\\_incidents.a\\_human\\_factors\\_analysis.pdf](https://www.faa.gov/about/initiatives/maintenance_hf/library/documents/media/human_factors_maintenance/air_traffic_control_(atc)_related_accidents_and_incidents.a_human_factors_analysis.pdf). [Kasutatud 24 03 2021].
- [2] S. Hirsjärvi, P. Remes ja P. Sajavaara, Uuri ja kirjuta, Tallinn: Kirjastus Medicina, 2005.
- [3] L. Õunapuu, Kvalitatiivne ja kvantitatiivne uurimisviis sotsiaalteaduses, Tartu: Tartu Ülikooli Kirjastus, 2014.
- [4] C. Roberts, „A conceptual framework for quantitative text analysis“, *Kluwer Academic Publishers*, kd. 1, pp. 259-274, 2000.
- [5] C. Briggs, Learning how to ask. A sociolinguistic appraisal of the role of the interview in social science research., Cambridge: Cambridge University Press, 1986.
- [6] J. Spradley, The ethnographic interview, New York: Holt, Rinehart & Winston, 1979.
- [7] U. Kelle, Computer-aided qualitative data analysis: Theory, methods and practice., London: Sage, 1995.
- [8] SKYbrary, „Surface Movement Radar“, [Võrgumaterjal]. Available: [https://www.skybrary.aero/index.php/Surface\\_Movement\\_Radar](https://www.skybrary.aero/index.php/Surface_Movement_Radar). [Kasutatud 24 03 2021].
- [9] Saab, „Airport Vehicle Tracking“, [Võrgumaterjal]. Available: <https://www.saab.com/products/airport-vehicle-tracking>. [Kasutatud 24 03 2021].
- [10] ADB Safegate, „ACEMAX Advanced Surface Movement Guidance and Control System“, [Võrgumaterjal]. Available: <https://adbsafegate.com/product-center/tower/?prod=acemax-advanced-surface-movement-guidance-and-control-system>. [Kasutatud 24 03 2021].
- [11] T. L. I. j. Hannus Vard, Interviewee, *Lennuvälja maaseire süsteem*. [Intervjuu]. 08 02 2021.
- [12] ATCnetwork, „AviBit to provide its integrated Electronic Flight Strips and A-SMGCS to EANS for Tallinn Airport“, [Võrgumaterjal]. Available: <https://www.atc-network.com/atc-news/avibit-to-provide-its-integrated-electronic-flight-strips-and-a-smgcs-to-eans-for-tallinn-airport>. [Kasutatud 25 03 2021].
- [13] EANS, „Tallinna lennujuhtimistornis võeti kasutusele uus lennujuhtimissüsteem“, [Võrgumaterjal]. Available: <http://www.eans.ee/tallinna-lennujuhtimistornis-voeti-kasutusele-uus-lennujuhtimissusteem/>. [Kasutatud 25 03 2021].
- [14] Motorola, „Professional Digital two-way radio MOTOTRBO. User Guide.“, 2014. [Võrgumaterjal]. Available: [https://www.dnd.hu/uploads/termek\\_doc/Motorola\\_DP4800\\_DP4801\\_manual\\_en.pdf](https://www.dnd.hu/uploads/termek_doc/Motorola_DP4800_DP4801_manual_en.pdf). [Kasutatud 24 03 2021].

- [15] SKYbrary, „ICAO Aerodrome Reference Code,“ [Võrgumaterjal]. Available: [https://www.skybrary.aero/index.php/ICAO\\_Aerodrome\\_Reference\\_Code](https://www.skybrary.aero/index.php/ICAO_Aerodrome_Reference_Code). [Kasutatud 24 03 2021].
- [16] Aero Language, „The language of aviation,“ [Võrgumaterjal]. Available: <http://language.aero/en/about/aviation-language>. [Kasutatud 24 03 2021].
- [17] Channel 9, „Panel: Systems Programming in 2014 and Beyond,“ [Võrgumaterjal]. Available: <https://channel9.msdn.com/Events/Lang-NEXT/Lang-NEXT-2014/Panel-Systems-Programming-Languages-in-2014-and-Beyond>. [Kasutatud 10 02 2021].
- [18] B. M. Kernighan and D. M. Ritchie, The C Programming Language, Engelwood Cliffs: Prentice Hall, 1978.
- [19] T. J. Bergin ja R. G. Gibson, History of programming languages II, New York: Association for Computing Machinery, 1996.
- [20] J. Duley ja B. Reist, „X-Plane Communication Toolbox (XPC),“ [Võrgumaterjal]. Available: <https://software.nasa.gov/software/ARC-17185-1>. [Kasutatud 10 02 2021].
- [21] S. Holden, „The Python Wiki,“ [Võrgumaterjal]. Available: <https://wiki.python.org/moin/>. [Kasutatud 10 02 2021].
- [22] Rust Core Team, „Rust Programming Language,“ [Võrgumaterjal]. Available: <https://www.rust-lang.org/>. [Kasutatud 10 02 2021].
- [23] SQLite Core Team, „SQLite,“ [Võrgumaterjal]. Available: <https://www.sqlite.org/whentouse.html>. [Kasutatud 15 03 2021].
- [24] Microsoft, „Visual Studio Code,“ [Võrgumaterjal]. Available: <https://code.visualstudio.com/>. [Kasutatud 10 02 2021].
- [25] S. Kukk, Motorola Mototrbo raadiosüsteemi GPS andmete liidestamine Kaitseväe olukorra ja lahinguteadlikkuse süsteemiga, Tartu: Eesti Lennuakadeemia, 2020.
- [26] Actix Core Team, [Võrgumaterjal]. Available: <https://actix.rs/book/actix/sec-2-actor.html>. [Kasutatud 16 03 2021].
- [27] Tokio Core Team, „Crate tokio,“ [Võrgumaterjal]. Available: <https://docs.rs/tokio/1.3.0/tokio/>. [Kasutatud 16 03 2021].
- [28] Rust Core Team, „Crate reqwest,“ [Võrgumaterjal]. Available: <https://docs.rs/reqwest/0.11.2/reqwest/>. [Kasutatud 17 03 2021].
- [29] Rocket Core Team, „Rocket overview,“ [Võrgumaterjal]. Available: <https://rocket.rs/v0.4/overview/>. [Kasutatud 17 03 2021].
- [30] Google, „Build awesome apps with Google's knowledge of the real world.,“ [Võrgumaterjal]. Available: <https://developers.google.com/maps>. [Kasutatud 17 03 2021].
- [31] Google, „Maps JavaScript API,“ [Võrgumaterjal]. Available: <https://developers.google.com/maps/documentation/javascript/overview>. [Kasutatud 17 03 2021].
- [32] Python team, „Socket - low-level networking interface,“ [Võrgumaterjal]. Available: <https://docs.python.org/3/library/socket.html>. [Kasutatud 18 03 2021].
- [33] Scapy team, „Introduction,“ [Võrgumaterjal]. Available: <https://scapy.readthedocs.io/en/latest/introduction.html#what-makes-scapy-so-special>. [Kasutatud 18 03 2021].
- [34] Pyproj team, „Pyproj Documentation,“ [Võrgumaterjal]. Available: <https://pyproj4.github.io/pyproj/stable/index.html>. [Kasutatud 17 03 2021].

- [35] Esri, „ArcGIS Pro Next-generation Desktop GIS,“ [Võrgumaterjal]. Available: <https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview>. [Kasutatud 17 03 2021].

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Jevgeni Družkov

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Lennuvälja teenindussõidukite liiklusinformatsiooni esitatava mooduli arendamine ja testimine“, mille juhendajad on Meelis Antoi ja Tarmo Aia
  - 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.

16.05.2021

---

<sup>1</sup> 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 – Intervjuu baasküsimused

Nimi:

Ametikoht:

Haridus:

1. Millistel protseduuridel või olukordades oleks lennuvälja maaseire süsteem kasulik?
2. Millistel juhtudel oleks kasulik väljavõtte maaseire süsteemi statistiliste andmete kohta?
3. Millised objektid peavad olema kaardipõhjal nähtavad?
4. Millised objektid ei pea olema kaardil esitatud, kuid nende olemasolu mõjutaks positiivselt lennujuhi otsustusprotsessi?
5. Kas kaardipõhi peab olema suunitav? Miks?
6. Kas kaardipõhi peab olema pööratav? Miks?
7. Millise kujuga võiks olla maismaasõiduki ikoon?
8. Millist värvitooni oleks mõistlik kasutada ikooni disainimisel?
9. Milline info peab olema nähtav ikoonil?
10. Milline info ei pea olema ikoonil esitatud, kuid selle olemasolu mõjutaks lennujuhi otsustusprotsessi positiivselt?
11. Millist infot on mõistlik kasutada infoaknas, mis ilmub ikoonile vajutades?
12. Milline info ei pea olema infoaknal esitatud, kuid selle olemasolu mõjutaks lennujuhi otsustusprotsessi positiivselt?
13. Millise ajalise intervalliga peavad objekti liikumisandmed jõudma kaardipõhjale?

## Lisa 3 – Intervjuu kodeerimisjuhend

Tabel 6. Intervjuu kodeerimisjuhend.

<b>Põhikategooria</b>	<b>Alamkategooria</b>	<b>Kood</b>	<b>Intervjuu küsimuse nr</b>
1. Kasutusvaldkond	1.1 Tööprotsessid	1.1.1 Protseduurid	1
	1.2 Statistika	1.2.1 Andmete variatiivsus	2
2. Kaardipõhi	2.1 Objektid	2.1.1 Kohustuslikud	3
		2.1.2 Valikulised	4
	2.2 Kaart	2.2.1 Pööramine ja suum	5, 6
3. Sõiduki ikoon	3.1 Disain	3.1.1 Kuju	7
		3.1.2 Värvitoon	8
	3.2 Ikooni info	3.2.1 Kohustuslik	9
		3.2.2 Valikuline	10
	3.3 Infoaken	3.3.1 Kohustuslik info	11
		3.3.2 Valikuline info	12
3.4 Intervall	3.4.1 Aeg	13	

## Lisa 4 – Intervjuude *cross-case* maatriks

Tabel 7. Intervjuude *cross-case* maatriks.

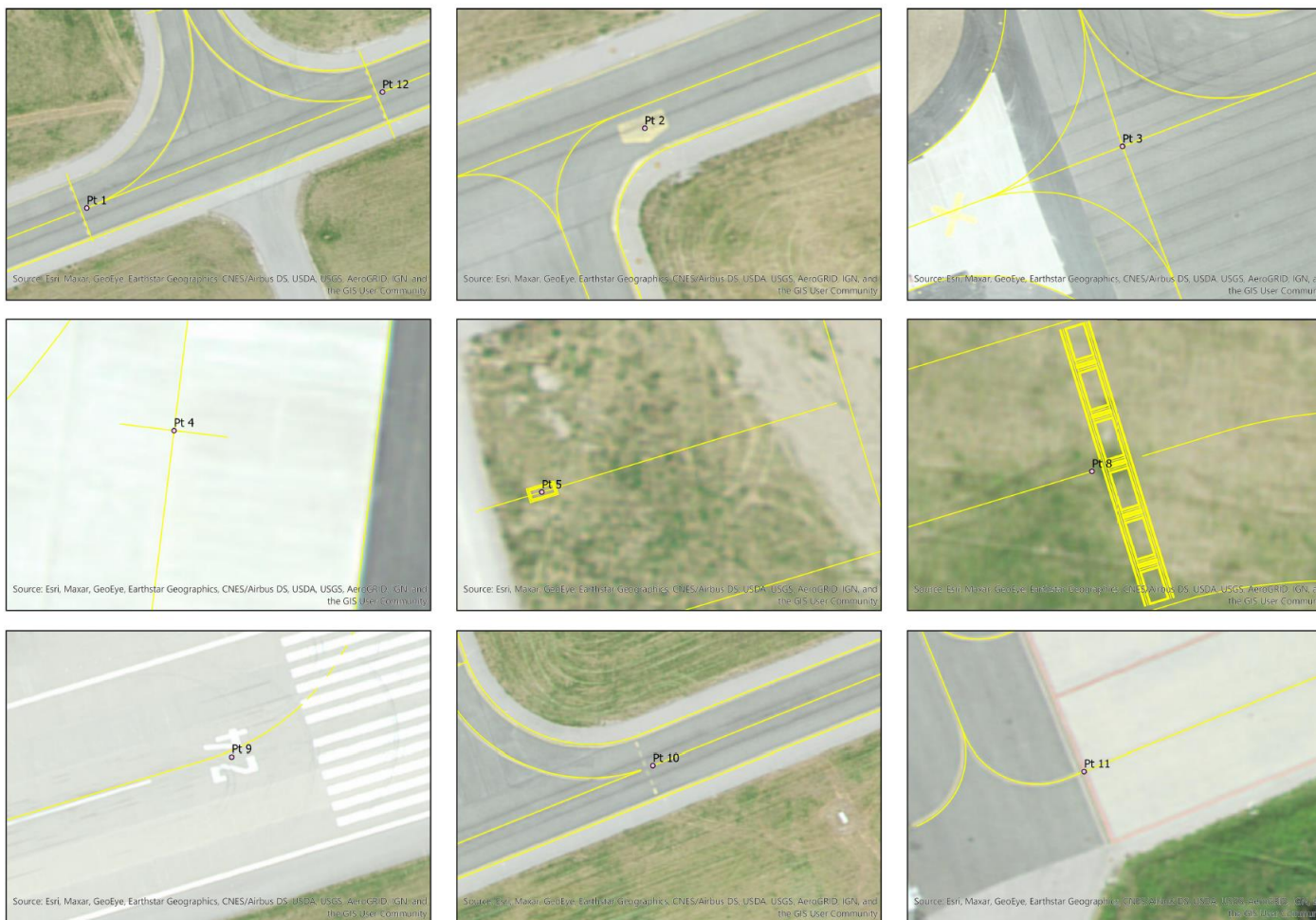
Kood	Intervjuu 1 (Siim Rumjantsev, lennujuht ja rajameister)	Intervjuu 2 (David-Andreas Mellov, lennujuht)
1.1.1 Protseduurid	<p>a. Hooldustööde jaotamine ja kontrollimine. Asukoha digitaalne visualiseerimine annab infot, kus parasjagu hooldusmasin asub. Kuna rajameistri töökäsk sisaldab hooldustööde jaotamist, siis asukoha järgi on võimalik aru saada, kui millises töö faasis meeskond asub. See annab parema ülevaate hooldusoperatsiooni kestusest ja annab parema võimaluse formuleerida järgmist töökäsku.</p> <p>b. Piiratud nähtavuse korral, nt udu või tiheda lumesaju tõttu, puudub visuaalne kontakt. See tähendab, et ainus asukoha määramise võimalus on raadioside pidamine. Võimalik on tutvuda olukorraga koha peal, kuid rajameistri tööhulka kuuluvad ka muud ülesanded ning suure vastutusala tõttu on tihti võimatu koha peale minna.</p>	<p>a. Situatsiooni teadlikkus. Märgatavalt kiiremini saab positsiooni teada.</p> <p>b. Piiratud nähtavuse protseduur – kui kaamerad ei anna tulemust.</p> <p>c. Häälside korral ei anta alati adekvaatset positsiooni.</p> <p>d. Üle andmisel kiire ülevaade sõidukite asukohtadest lennuväljal.</p> <p>e. Kui kanalis on sides mitu jaama, siis saab üle kontrollida.</p>
1.2.1 Andmete variatiivsus	<p>a. Väljavõtteid saab grupeerida ala ja ilmatingimuste alusel, mis annab võimaluse kvantitatiivselt hinnata, palju teatud lennuvälja ala võtab hooldamiseks aega. See on suureks abiks hooldustööde optimeerimisel ja hooldusaja prognoosimisel. Hetkel on see igal rajameistril subjektiivne ja põhineb ainult töökogemusel. See aitab analüüsida erinevate rajameistrite töövõtteid ja vajadusel suunata või aidata paremini hooldustöid planeerida.</p> <p>b. Statistika aitab teha paremaid ja kvaliteetsemaid ettekandeid juhkonnale.</p>	<p>a. <i>Hotspot</i> – kriitilised punktid, kus toimub masinate kogunemine.</p> <p>b. Liiklusala läbilaskevõime arvutamine.</p>
2.1.1 Kohustuslikud	<p>a. Lennurada, kõik ruleerimisteed ja perroonid.</p>	<p>a. Kogu liiklusala.</p>

<b>Kood</b>	<b>Intervjuu 1 (Siim Rumjantsev, lennujuht ja rajameister)</b>	<b>Intervjuu 2 (David-Andreas Mellov, lennujuht)</b>
2.1.2 Valikulised	a. Kogu lennujaama administratiivala. Kuna hooldusmeeskonnaliikmed täidavad ka muid protseduure lisaks lennuraja hooldamisele (nt sõiduki tankimine või rajakeemia varude täiendamine), siis on tihti oluline teada, kus parasjagu töötajad asuvad.	a. Ei ole muud vaja kui liiklusalala.
2.2.1 Pööramine ja suum	a. Suumimine on oluline, mõnikord on vaja detailsemalt vaadata – eriti väiksed, aga tähtsad objektid, nt õhusõiduki angaari ümbrus või kindel piiritletud ala. Digitaalse pildi abil saab aru, kas inimene on töökäsust õigesti aru saanud ja kas ta liigub õigele positsioonile või mitte. Vajadusel saab töötajale täiendavaid juhiseid anda. b. Pööramist pole tarvis. Kaart asub diagonaalis, mistõttu mahub maksimaalselt palju objekte peale. Peamine, et see oleks õiges mõõdus/mastaabis. Oleks isegi parem, kui lennuvälja kaart oleks loomulikul kujul. Harjumuspärane kujud aitab kiiremini olukorrast aru saada. Pigem asendada kaardi pööramine nt vahemaa mõõtmise funktsiooniga, mille abil saab mõõta kahe punkti vahel distantsti. Mõnikord annab ala mõõtmine paremat sisendit planeerimisprotsessis.	a. Suumimine on oluline - piiratud nähtavuse korral saab suumida ja vaadata teatud joonte ületamist. b. Pööramine pole oluline.
3.1.1 Kuju	a. Peamine, et oleks selgelt arusaadav, millisesse maastikupunkti ikoon näitab. Kujud iseenesest ei olegi tähtis. Ikoon peab olema võimalikult kompaktne.	a. Oluline on eristatavus, hea näide on Google Maps standardikoon.
3.1.2 Värvitoon	a. Kaardi taustast erinev – ergas värv, nt punane.	a. Peamine, et ei oleks sama tooni nagu teised maismaaobjektid.
3.2.1 Kohustuslik	a. Sõiduvahendi või isiku identifitseerimise number. Võib olla kokkuleppeline number, mis oleks kinnitatud konkreetsele isikule või sõidukile.	a. Oluline on sõiduki identifikaator, mis oleks selge ja ainulaadne.
3.2.2 Valikuline	a. Polegi muud infot tarvis, tähtis on näha sõiduki identifitseerimise sümbolit ja asukohta.	a. Ei ole midagi lisada.



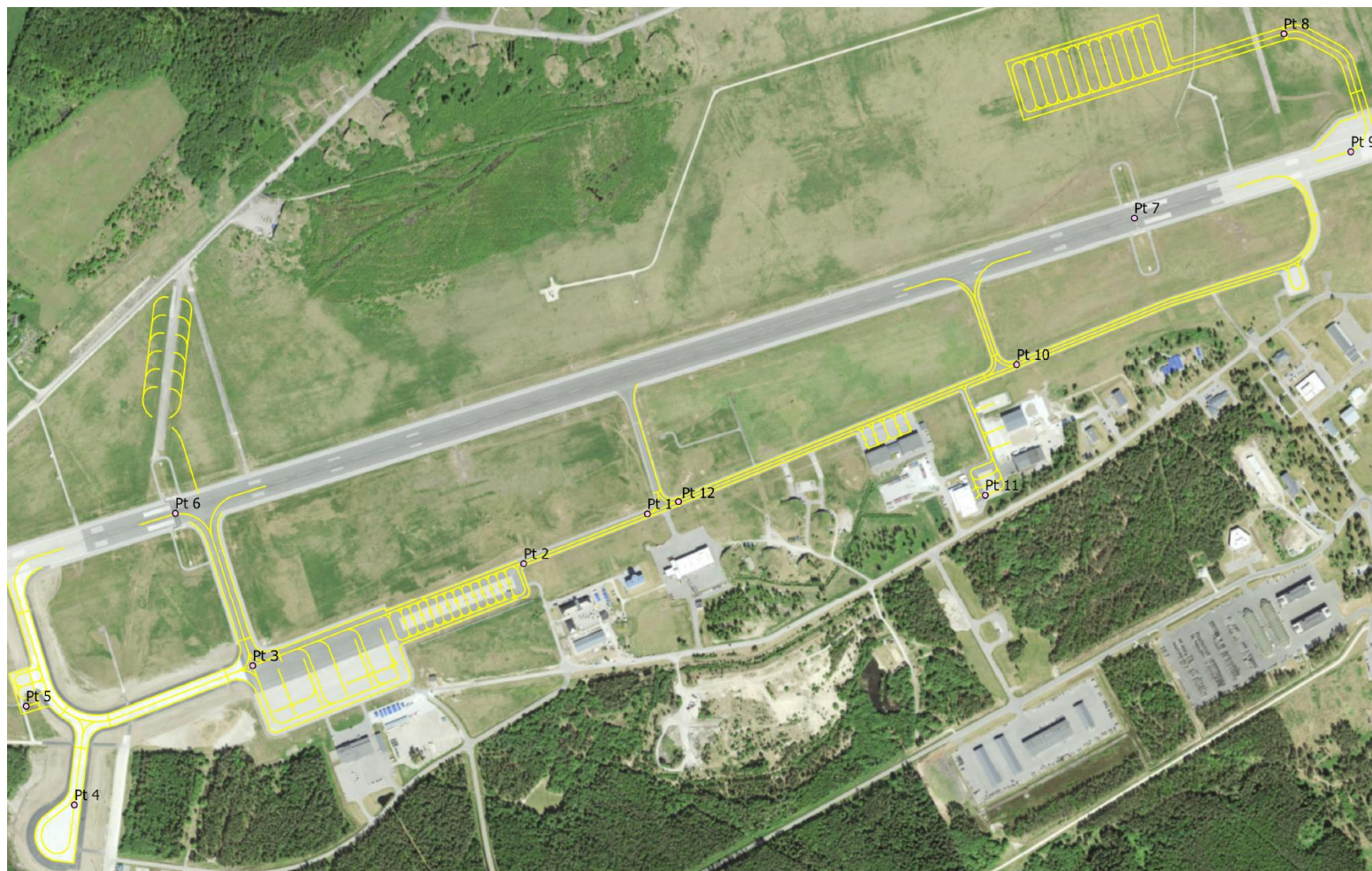
<b>Kood</b>	<b>Intervjuu 1 (Siim Rumjantsev, lennujuht ja rajameister)</b>	<b>Intervjuu 2 (David-Andreas Mellov, lennujuht)</b>
3.3.1 Kohustuslik info	a. Kiirus ja suund. Kuna sõiduki ikoonid liiguvad teatud sammuga, siis mõnikord on tähtis aru saada, mis suunas ta liigub.	a. Kutsung, kiirus ja suund.
3.3.2 Valikuline info	a. Kui infoaken ei lähe ebamugavalt suureks, siis võiks abiks olla sõiduki või isiku kutsung.	a. Ei ole midagi lisada.
3.4.1 Aeg	a. 10-15 sekundit. Kuna ala on reeglina suur ja keskmine liikumiskiirus hooldustöödel on 35-45 km/h, siis nimetatud ajaga ei jõua sõiduk läbida liiga suurt vahemaad, põhjustades teistele ohtu.	a. Kui lahendust kasutatakse mitte hajutamisel, siis kuni 10 sekundit on aktsepteeritav.

## Lisa 5 – Lennuvälja kontrollpunktid



Joonis 8. Lennuvälja kontrollpunktid.

## Lisa 6 – Lennuvälja kontrollpunktide kaart



Joonis 9. Lennuvälja kontrollpunktide asukohad lennuvälja territooriumi vaates.

## Lisa 7 – Lennuvälja kontrollpunktide koordinaadid

OBJECTID	Name	MGRS	Notes
1	Pt 1	35VLF 40629 72033	
2	Pt 2	35VLF 40403 71943	
3	Pt 3	35VLF 39910 71757	
4	Pt 4	35VLF 39584 71504	
5	Pt 5	35VLF 39496 71684	
6	Pt 6	35VLF 39768 72034	
7	Pt 7	35VLF 41517 72567	
8	Pt 8	35VLF 41790 72901	
9	Pt 9	35VLF 41912 72688	
10	Pt 10	35VLF 41302 72304	
11	Pt 11	35VLF 41246 72067	
12	Pt 12	35VLF 40686 72056	

Joonis 10. Lennuvälja kontrollpunktide koordinaadid.