

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Sorell Tudelep 206194IABB

**Monitooringu seadmete jälgimissüsteemi nõuete  
analüüs ning esmane realiseerimine  
Tarbijakaitse ja Tehnilise Järelevalve Ametile**

Bakalaureusetöö

Juhendaja: Viljam Puusep  
MSc

Tallinn 2024

## **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: Sorell Tudelep

20.05.2024

## **Annotatsioon**

Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakond tegeleb igapäevaselt Eesti sidevõrgu järelevalvega, et tagada vajaliku informatsiooni segamatu jagamine. Selle garanteerimiseks on Sideosakond paigaldanud võrgumonitorimise seadmed üle Eesti, mis kontrollivad võrku segavaid signaale. Käesoleva töö eesmärk on analüüsida süsteemi nõudeid ning koostada projekti esmane realiseerimine. Selle saavutamiseks analüüsib autor erinevaid lähenemisi, mille alusel otsustada projekti õige suund.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 24 leheküljel, 4 peatükki, 4 joonist.

## **Abstract**

### **Analysis of requirements and initial implementation of a Tracking System for Monitoring Surveillance Devices for the Consumer Protection and Technical Regulatory Authority**

The Consumer Protection and Technical Surveillance Authority's Communications Department deals daily with the supervision of Estonia's telecommunications network to ensure the uninterrupted sharing of necessary information. To guarantee this, the Communications Department has installed network monitoring devices across Estonia that check for signals that interfere with the network. The purpose of this work is to analyze the system requirements and to prepare an initial implementation of the project. To achieve this, the author analyzes various approaches to decide the correct direction for the project.

The thesis is in Estonian and contains 24 pages of text, 4 chapters, 4 figures.

## Lühendite ja mõistete sõnastik

Debug	Vigade otsimine
XAML	Extensible Application Markup Language/märgistuskeel
Ping	IP võrgu reageerimise mõõtmine
WPF	Windows Presentation Foundation
MWS	Web Map Service
MFS	Web Feature Service
MVP	Model-View-Presenter
MVVM	Model-View-ViewModel
MVC	Model-View-Controller
CLI	Command Line Interface
CLR	Common Language Runtime

# Sisukord

1 Sissejuhatus .....	9
1.1 Üldine .....	9
1.2 Probleem .....	9
1.3 Eesmärk .....	9
1.4 Töö struktuur .....	10
2 Metoodika.....	11
2.1 Objekt .....	11
2.2 Rakendusetüüp.....	11
2.3 Tehnoloogia .....	12
2.4 Serveripool ning kasutajaliides.....	13
2.5 Komponentide kirjeldus .....	14
2.6 Kaardi kihid .....	15
2.7 Töö protsess.....	16
3 Peamised tulemused .....	18
3.1 Nõuete määramine.....	18
3.1.1 Funktsionaalsed nõuded .....	18
3.1.2 Mittefunktsionaalsed nõuded.....	19
3.2 Abstraktsed nõuded .....	19
3.3 Arhitektuur.....	20
3.4 Kaart .....	21
3.5 Projekti esmane realiseerimine .....	22
4 Analüüs ja järeldused.....	25
4.1 Nõuete valimine.....	25
4.2 Tulemuste ülevaade .....	26
4.2.1 Analüüsi tulemuste ülevaade .....	27
4.2.2 Realiseerimise ülevaade .....	28

4.3 Alternatiivid.....	28
4.4 Kitsendused ja piirangud .....	29
4.5 Tulemuste valideerimine .....	30
4.6 Äriline põhjendamatus.....	31
4.1 Tuleviku tööd.....	31
5 Kokkuvõte .....	33
Kasutatud kirjandus .....	34
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	36
Lisa 2 – Laiendatud menüü paneel .....	37
Lisa 3 – Seadme informatsiooni muutmine .....	38
Lisa 4 – Seadmete nimekiri .....	39
Lisa 5 – Seadmete grupeerimine .....	40
Lisa 6 – Seadme otsing.....	41
Lisa 7 – Seadme kustutamine .....	42

## Jooniste loetelu

Joonis 1. Protsessi diagramm	18
Joonis 2. Model-View-ViewModel	22
Joonis 3. Interaktiivne kaart	24
Joonis 4. Seadme lisamine süsteemi	25



# **1 Sissejuhatus**

## **1.1 Üldine**

Raadiosageduste häireteta töö on kriitilise tähtsusega, et üle Eesti turvaliselt edastada informatsiooni. Tarbijakaitse ja Tehnilise Järelevalve Ameti (TTJA) Sideosakond tegeleb igapäevaselt sideteenuste järelvalvega, mille üheks osaks on raadiosageduste monitoorimine. Sideosakonna tööülesanded on raadioseadmete turujärelevalve, kaitsta võrku potentsiaalsete raadiohäirete eest ning vajadusel kõrvaldada segavad faktorid.

Käesolev bakalaureusetöö keskendub TTJA Sideosakonna töövahendi nõuete analüüsile ning rakenduse esmasele realiseerimisele, mille tulemusena on võimalik luua efektiivne jälgmissüsteem monitooringu seadmetele.

## **1.2 Probleem**

Tarbijakaitse ja Tehnilise Järelevalve Amet (TTJA) Sideosakond kasutas monitooringu seadmete jälgimiseks kahte eraldiseisvat lahendust. Esiteks, läbi PingInfoView ping-iti seadmeid, et tuvastada seadme võrgus olek. Kui ping-itav seade ei reageerinud 10ms jooksul, viitas see seadme mitte toimimisele. Teiseks, vigase seadme IP aadressi alusel otsiti kõvakettalt seadme nimetus. Seadme nimetus viitas seadme asukohale ning selle alusel otsiti arvuti tausta pildilt, mis kujutas Eesti kaarti, seadmete sisenemise kohta, mis oli määratud seadme asukoha järgi pildile. Antud lahendus oli väga ajakulukas ja tülikas igapäevane tegevus. Autor võttis eesmärgiks koostada uue süsteemi nõuete analüüsi ning luua esmane rakenduse realiseerimine, et tellijale oleks võimalik tulevikus arendada täisfunktsionaalne töövahend.

## **1.3 Eesmärk**

Projekti eesmärk on TTJA Sideosakonna nõuete analüüsimine ning esmase rakenduse arendamine, mis võimaldab efektiivsemalt monitoorida ja hallata monitooringu võrguseadmeid. Käesolev töö keskendub tellija poolt esitatud nõuete analüüsile, mille

abil on võimalik luua uus süsteem, mille kohaselt seadmete kontrollimine ning kaugjuurdepääs on kasutajasõbralikumad.

Analüüsi käigus selgitab autor välja parimad praktikad ja tehnilised lahendused koodi arendamisel kasutades võrdlevat metoodikat. See on vajalik, et tagada projekti nõuetele vastavuse ja optimeerida arendusprotsessi. Peale analüüsi realiseerib autor osa rakenduse koodist, et mõista, kas analüüsi käigus välja valitud tehnoloogiad on parimad, et projekti arendada. Selle tulemusena saavutab autor kindluse, et tellija jaoks on leitud kõige asjakohasemad ning võimekamad meetodid, millele toetudes on võimalik rakendus arendada.

## **1.4 Töö struktuur**

Lõputöö on jagatud neljaks peatükiks. Esimeses kirjeldatakse käesoleva töö eesmärke ning selgitatakse töö probleem. Teises peatükis kirjutatakse milliseid raamistikke ning meetode kasutada funktsiooni esmasel realiseerimisel. Seejärel toob autor kolmandas peatükis välja nõuete analüüsi ning esmase realiseerimise tulemused. Lõpptulemusena analüüsitakse saadud tulemust ning tuuakse välja edasised arenduse plaanid.

## **2 Metoodika**

Käesoleva lõputöö käigus analüüsitakse Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakonna poolt sisendina antud nõudeid ning tehakse tehnoloogia valikuid rakenduse arendamiseks. Analüüsil toetutakse võrdlevale metoodikale, mis kasutab regulaarset võrdlust väheste uurimisobjektide puhul [25], mis võimaldab objektiivset ja põhjalikku ülevaadet erinevatest valikutest, andes selge ettekujutuse, millised lahendused sobivad antud projekti jaoks kõige paremini.

### **2.1 Objekt**

Uurimisobjektiks on määratud Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakonna nõuded töövahendile, mille abil on võimalik jälgida monitooringu seadmete funktsioneerimist. Nõuded põhinevad Sideosakonna töötajate tagasisidel, mis on saadud läbi mitmete kohtumiste. Töö analüüsiv skoop sai kitsendatud vastavalt autori oskustele ja ressurssidele. Autori peamiseks suunaks oli valida arendusplatvorm ja tehnoloogiad, mis toetavad C# programmeerimiskeelt, arvestades autori varasemat kogemust ja kompetentsi selles valdkonnas. See lähenemine tagas, et tehnoloogia valik ja arendusprotsess oleksid tõhusad ja praktiliselt rakendatavad. Tehnoloogiate võrdlemise tulemusena valiti meetodid ja raamistikud, mida rakendati esialgse prototüübi loomisel. Rakenduse esmane realiseerimine andis aluse praktilisele analüüsitulemuste valideerimisele.

### **2.2 Rakendusetüüp**

Projekti arendamisel tuleb arvestada erinevate rakendustüüpidega, mida võib kaaluda vastavalt projekti eesmärkidele ja nõuetele. Kuna tegemist on riikliku ettevõttega, toonitati kohtumistel rakenduse turvanõudeid ja vajadust tagada andmete kättesaamatus kõrvalistele isikutele. Seetõttu otsustati arendada töölaarakendus, mis paigaldatakse otse kasutaja arvutisse, mis tähendab, et selle toimimiseks ei ole vajalik andmete edastamine serverisse. See võimaldab rakendust kasutada igal ajal ning rakenduse andmed

salvestatakse lokaalselt töölauarvuti andmebaasi, mis muudab kolmandate osapoolte ligipääsu tundlikele andmetele raskendatuks. Lisaks ei nõua töölaarakendus pidevat andmete pilvevarundamist, pakkudes seeläbi paremat kontrolli andmete üle ja tugevdades andmeturvet [11].

## 2.3 Tehnoloogia

Toetudes lõputöö eesmärgile analüüsida projekti arenduse nõudeid võeti arvesse erinevaid tehnoloogiaid, et saavutada rakenduse kõige kasulikum kuju. Valikut kitsendasid autori ning tellija poolsed kitsendused, mis nägid ette rakenduse keelena C# programmeerimiskeelt ning töölaarakenduse olemasolu. Sellest tulenevalt kaaluti erinevaid raamistikke nagu WPF, .NET MAUI ning Avalonia, millel oli võimekus lisada kaarte rakendusele.

WPF on osa .NET raamistikust, mis kasutab XAML programmeerimiskeelt ning mille abil on võimalik paindlikult dünaamiliselt eraldada disain ja implementatsioon [13]. Tehnoloogia pakub võimalusi kontrollida kasutajaliidese paneele, nagu DockPanel, mis kinnitab sisu kindlale alale. Paneeli tüüp GridLayout reastab elemendid ridadesse ning veergudesse, pakkudes paindlikkust andmete liigutamisel. Andmete sidumine on WPFi funktsioon, mis hoiab objekti omadusi ja kolleksioone sünkroniseeritud kujul ühes või mitmes andmevaates. Andmesidumise eesmärk on vähendada vajadust käsitsi andmete uuendamise järele, kui andmed objektis muutuvad, uuendades andmeid, kui kasutaja muudab kontrole. WPF tagab rakenduse kasutamise ja ligipääsu ainult siseveebis ning rakenduse sõltumatuse serveritest. WPF installitud rakendus käitub samamoodi, kui tavaline installitud rakendus, kus ligipääs on ainult süsteemi andmetele. Raamistiku abil on võimalik luua Windows rakendusi, mis jooksevad eraldiseisvalt enda aknas mitte veebirakenduses [1].

Avalonia on avatud lähtekoodiga .NET platvormi ülene raamistik, mille abil on võimalik luua Windows, MacOS, Linux, iOS ning Androidi rakendusi [20]. Avalonia on edasiarendus WPF ning sellest tulenevalt on tal samad andmesidumise võimalused, XAML kasutamine, mallid ning erinevad disainimoodulid nagu DockPanel, mis on samuti WPF'il [21]. Avalonia kasutab compiletime sooseid, runtime asemel, mis teeb koodi debuggimise kergemaks. Avalonial on erinevused kuidas luuakse nimekirja, kui Avalonias on nimekiri ItemsRepeater siis WPF raamistikus kasutatakse ListView'd. Kuigi

nimetused on kahel platvormil sarnased, on Avalonia raamistiku on õppimine raskem kui WPFi [22].

.NET MAUI raamistik võimaldab luua mitmeplatvormilisi rakendusi. .NET MAUI võimaldab integreerida platvormi spetsiifilist koodi ja ressursse, pakkudes võimalust rakendada suurel määral rakenduse loogikat ja kasutajaliidese kujundust ühtsest koodibaasist. See raamistik on loodud toetama rakenduste töötamist erinevatel platvormidel ning pakub täiustatud navigatsiooni võimalusi, andmesidumist ja kasutajaliidese elementide kohandamist, mis muudab arendusprotsessi paindlikumaks ja tõhusamaks. .NET MAUI toob uuendusi ka kasutajaliidese komponentide esitamisel, pakkudes rikkalikke võimalusi kasutajakogemuse parandamiseks [12]. MAUI kasutab XAML kasutajaliidese ning C# loogika kirjeldamisel. Raamistikul puudub Grid elemendi maksimaalse laiuse ning pikkuse määramine. Samuti spetsiifiliste kontrollide tugi on raamistikul puudulik ning vahel rakendus reageerib valesti peale protseduuride loomist ning edastab valeandmed [22].

Võttes arvesse raamistike omadusi on WPF raamistik antud töös kõige parem valik. Kuigi Avalonia on edasiarendus WPF, ei ole antud projekti raames vajadust luua mitmele platvormile rakendust ning Avalonia pikem õppeperiood muudab raamistiku vähem atraktiivseks. .NET MAUI eesmärk on lihtsustada arendamist, kuid raamistikus on esinenud vigu, mis tulevad MAUI lähtekoodist ning sellest tulenevalt autor soovib kasutada WPF raamistiku arenduseks.

## **2.4 Serveripool ning kasutajaliides**

Projekti raamistik ning autori tehnilised piirangud nägid ette kasutajaliidese XAML programmeerimiskeelt ning code-behind-i osa C# keelega. Kuna XAML on sügavalt integreeritud valitud WPF raamistikuga, ei tekkinud küsimust, millist keelt kasutajaliidese arendamiseks kasutada. Vastavalt autori kitsendustele kirjutada serveri pool C# keeles, uuris töö omanik alternatiivseid võimalusi, et tagada tehtud valikute asjakohasus.

XAML, mida kasutatakse laialdaselt .NET raamistiku kasutajaliideste kirjeldamiseks, on laiendatav märgistuskeel, mis võimaldab atribuutide kaudu määrata, kuidas kasutajaliidese objektid välja näevad ja toimivad [2]. See hõlmab kontrollide

andmesidumist, mis võimaldab arendajatel kasutajaliidest dünaamiliselt muuta ilma alusprogrammi koodi muutmata, elimineerides vajaduse iga muudatuse järel programmi uuesti laadida. XAML-i kasutamine .NET rakendustes lihtsustab kasutajaliidese loomist ja muudab selle hoolduse tõhusamaks, kuna XAML on tõstutundlik ja nõuab atribuutide korrektset kirjutamist [9].

Autori valik jätkata code-behind-i arendamist C# keeles põhines tema eelneval kogemusel ja selle keele populaarsusel .NET platvormil. C# toetab objektorienteeritud ning funktsionaalseid programmeerimispõhimõtteid ja pakub automaatset mäluhaldust, mis on osa .NET Runtime'ist, lihtsustades sellega ressursside juhtimist [10]. Alternatiivina kaaluti F# keelt, mis samuti toetab objektorienteeritud ja funktsionaalset programmeerimist .NET raamistikus [19], kuid autor otsustas jätkata C# kasutamist, et ära kasutada oma varasemaid teadmisi ja vältida uue keele õppimisega kaasnevat ajakulu.

## **2.5 Komponentide kirjeldus**

Tellijal soovis, et rakenduses sisaldaks interaktiivset Eesti kaarti, millele lisada monitooringu seadmete asukohti ning kuvada seadmete kaetuse raadiust. Sellest tulenevalt uuris autor erinevate komponentide pakutavaid võimalusi.

Esri poolt arendatud ArcGis API on tuntud oma geoinfosüsteemide lahenduste poolest ning kasutades ArcGIS API kaardi konstruktorit, on võimalik luua aluskaarte. ArcGIS API teeb päringuid andmetele, sõltuvalt kaardi suuruselt ja detailide tasemest, ning ArcGIS server kuvab seejärel kaardi, tagades dünaamilise kasutajakogemuse. ArcGis API't, mis on loodud selleks, et võimaldada geograafiliste andmete põhjalikku hoiustamist säilitamise, analüüsimist ja haldamist. Süsteemi tarkvara on mõeldud kasutamiseks töölauaarvutitele, pakkudes asukoha põhiseid tööriistu ja funktsionaalsusi. ArcGis API integreerub ArcGis serveriga ning süsteemis kasutatakse C# programmeerimiskeelt lisafunktsioonide loomiseks [3]. Raamistiku pakub võimalust lisada kaardile graafika kiht, millele lisada punkte, jooni, teste ning sümbolikat [7]. ArcGis serverile juurdepääsu tagab URL, mis tagastab veebilehe. Kaardiandmed genereeritakse aerofotode põhjal ning andmed salvestatakse andmebaasi, nagu iga muud tüüpi info korral. Interaktiivsete kaartide peamine eesmärk on geograafilise

informatsiooni visuaalne esitamine, mis võimaldab kasutajatel läheneda või kaugeneda, tagades nii vaateulatuse paindliku kohandamise vastavalt vajadusele [3]. ArcGIS API pakub võimalust alla laadida veebis olev kaart ning talletada seda oma vahemälus. Võrguvälise kaardi abil on võimalik kasutada andmekihte, mida saab kasutada ka interneti ühenduseta [15].

Alternatiivina on saadaval QGIS Server, mis on avatud lähtekoodiga ja kirjutatud C++ keeles. Erinevalt paljudest teistest WMS tarkvaradest, kasutab QGIS Server kartograafilisi reegleid, et võimaldada nii serveri seadistamist kui ka kasutaja poolt määratud kartograafiliste reeglite rakendamist. See on üles ehitatud toimima veebiserveritega ning toimib FastCGI/CGI rakendusena. Lisaks on QGIS Server varustatud Pythoni pluginite toega, mis tagab uute funktsioonide kiire ja tõhusa arendamise ning juurutamise. QGIS Server võimaldab kaardi andmete salvestamine vahemällu. Raamistik toetab visuaalselt punktide lisamist kaardile kuid läbi QGIS Desktopi või kasutades kolmandate osapoolte tööriistu [23].

Uurides kahe platvormi eripäraseid otsustas autor liikuda edasi ArcGIS API'ga. Kuigi QGIS Serveri dokumentatsioon on olemas jääb see siiski alla Esri poolt pakutavale detailsele informatsioonile. Mõlemad raamistikud suudavad luua interaktiivseid kaarte ning serveri maha kukkumisel hoian andmeid vahemälus kuid Esri poolt pakutav süsteem võimaldab punktide lisamist kaardile, kuid QGIS Server vajab selleks kolmandat osapoolt. Lisaks vaadeldes konkurentide kodulehtede andmeid annab Esri ArcGIS API tugevama ülevaate oma võimalustest nagu andmete dünaamiline analüüs ja visualiseerimine ning tuleviku laiendamisvõimalused.

## **2.6 Kaardi kihid**

Kaardikihtide analüüsimise vaatles autor WMS ning WFS erinevusi, et leida kõige sobilikum tehnoloogia projekti kaardile punktide ning visuaalide loomiseks. Keskendutakse tehnilistele omadustele ja piirangutele, mis potentsiaalselt mõjutaksid rakenduse tööd.

WMS võimaldab luua paindliku kaardirakenduse, kus kasutajad saavad kergelt lisada või eemaldada kaardikihte. Kihte on võimalik lisada juurde kasutades veebilehtede URL-e. Kihtide lisamise ja eemaldamise funktsioon muudab kaardi kasutajasõbralikumaks ja

kohandatavaks vastavalt kasutaja vajadustele. WMS-i abil on võimalik genereerida kohandatud georeferentseeritud pilt andmekogumist serverist koos seotud metaandmetega. Oluline osa on kihilisus, kus iga kiht on üks infoplokk ja iga kiht saab olla erinevate stiilidega [8]. Kaardid on mitmekihilised ning XML formaadis ning võimaldavad kasutajal korraga kuvada erinevaid kaardikihte. WMS kihtide kasutamine ja haldamine on muutunud lahutamatuks osaks kaasaegsetes geoinfosüsteemides, pakkudes kasutajatele ulatuslikke võimalusi andmete analüüsimiseks ja esitamiseks [18].

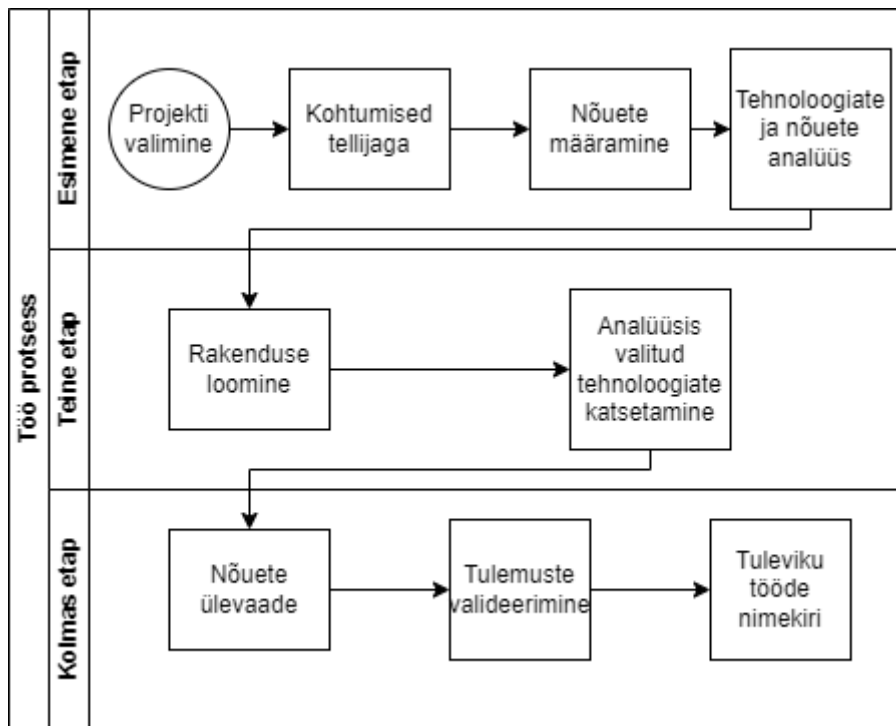
WFS kihid võimaldavad geograafiliste objektide jagamist ning muutmist koos nende geomeetria ja atribuutidega, pakkudes võimalust lisada kogu teenuse kaart või lisada kihte individuaalselt. WFS kihid toetavad telgede järjekorra kohandamist, mis on oluline koordinaatsüsteemide erisuste korral. WFSi on võimalik muuta URL-e, WFS versioone ja kohandatud päringu parameetrid. WFS funktsioneerimine võib sõltuda serveri võimekusest ning POST- päringutest ning samuti on kihtidel piiranguid nagu tehingute puudumine ja WFS kihtide jagamise puudumine veebikaardiga [16].

Analüüsi tulemusena jõuti järeldusele, et WMS on kahest valikust parim tehnoloogia, kuna pakub suuremat paindlikkust kaardi rakenduste loomisel ning WFS kihtide jagamine veebis oleva kaardiga on oluline puudus, mida projekt vajaks.

## **2.7 Töö protsess**

Antud bakalaureusetöö protsess on jaotatud kolmeks põhiliseks etapiks: nõuete analüüsi, rakenduse esmane realiseerimine ning tulemuste analüüsimine ja järelduste tegemine.





Joonis 1. Protsessi diagramm.

Esimeses etapis, nõuete analüüsis, valis autor projekti ning osales kohtumistel tellijaga, kus arutati projekti soovitud tulemuste ning funktsionaalsuse üle, mille alusel autor pani kirja esmased nõuded. Peale süvenemist projekti, osales autor detailsemal kohtumisel, et kinnitada projekti funktsionaalsete ja mittefunktsionaalsete nõuete kooskõla tellija soovidega, et saavutada tellija oodatav tulemus.

Autor analüüsis olemasolevaid tehnoloogiaid ja raamistikke, mis sobitusid projekti eesmärkidega. Analüüsi mitmeid raamistikke, programmeerimiskeeli ja komponente, vaadeldes nende sobivust nõuetega. Valikute tegemisel toetuti tootjate dokumentatsioonile, raamatutele ning veebis leitavale ametlikumale informatsioonile.

Rakenduse esimeses realiseerimises, koostati prototüüp, mille abil testiti analüüsifaasis valitud raamistike ja meetodite sobivus. Etapp võimaldas tuvastada, kas esialgsed tehnoloogiad, mis analüüsi käigus valiti olid asjakohased ning vastasid tellija nõuetele.

Kolmandal etapil analüüsit ja valideeriti tulemusi ning keskenduti saadud tulemuste suurimisele ja järelduste tegemisele. Selle faasi käigus vaadeldi, kuidas rakendatud lahendused vastasid projekti esialgsetele nõuetele.

## **3 Peamised tulemused**

Peamiste tulemuste peatükis toob autor välja tellija poolt tulnud nõuded, mis on jagatud funktsionaalseteks nõueteks ning mittefunktsionaalseteks nõueteks. Eraldi on välja toodud abstraktsed nõuded, mis rõhutavad projekti olulisust ning kõige tähtsamaid aspekte. Autor kirjeldab rakenduse potentsiaalselt arhitektuuri ning mustreid, mille abil on võimalik tulemus saavutada. Lisaks on välja toodud kaardi päritolu ning rakenduse esmase realiseerimise tulemus.

### **3.1 Nõuete määramine**

Nõuete määramisel toetuti Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakonna töötajate praegustele ning tulevikus vajaminevatele funktsionaalsusele, mis kaardistati läbi mitmete kohtumiste. Autor proovis mõista töötajate vajadusi ning kasutusel oleva süsteemi puudujääke, et selle alusel luua konkreetsed nõuded, millele projekti analüüsimisel ning arendamisel toetuda.

#### **3.1.1 Funktsionaalsed nõuded**

Sideosakonna töötajate nõuded tööriistale:

- Kasutajal peab olema võimalus lisada, muuta ja kustutada monitooringu seadmeid süsteemist.
- Rakenduses peab olema visuaalselt näha monitooringu seadmete asukohta.
- Visuaalselt peab süsteem kuvama seade võrgus olekut.
- Iga seadme kaetuse raadiust oleks võimalik Eesti kaardil kuvada.
- Seadmete nimekirjast on võimalik filtreerida võrgus olevaid seadmeid ja võrgust väljas olevaid seadmeid.
- Seadme nime, IP aadressi või füüsilise aadressi järgi otsida seadmeid süsteemist.
- Seadme juures olev lisainformatsiooni lisamise võimalus, mille kaudu osakonna siseselt informatsiooni vahetada.

- Kasutajal on võimalik vaadelda, kui kiiresti seadmed ping-i vastu võtavad.
- Süsteem võimaldab ping-ida seadmeid eraldi, et kontrollida nende tegevust.
- Rakenduse automaat ping-imise aega on võimalik muuta igal ajal.
- Kasutaja saab lisada uue seadme võrku vajutades Eesti kaardil kindlale asukohale.

### **3.1.2 Mittefunktsionaalsed nõuded**

Sideosakonna töötaja jaoks vajalikud mittefunktsionaalsed nõuded:

- Rakenduse ülesehitusel tagada kasutajasõbralik ning ühetimõistetavus.
- Rakendusse sisenemine ainult Tarbijakaitse ja Tehnilise Järelevalve Ameti sisevõrgu.
- Rakendu sei tohi sõltuda serverist.
- Eesti kaardi interaktiivsus, mis võimaldab kaardil läheneda ja kaugeneda.
- Kasutaja jaoks süsteemi lihtne orienteerumine.

### **3.2 Abstraktsed nõuded**

Süsteemi turvalisus ning kättesaadavus on samuti olulised aspektid, aidates valida sobivad tehnoloogilised raamistikud ja lahendused. Eesmärgiks seati TTJA Sideosakonna töötajate rutiinsete ülesannete kiirem ja sujuvam täitmine, vähendades manuaalsete toimingute hulka ning muutes protsessid süsteemi abil tõhusamaks ja lihtsamaks.

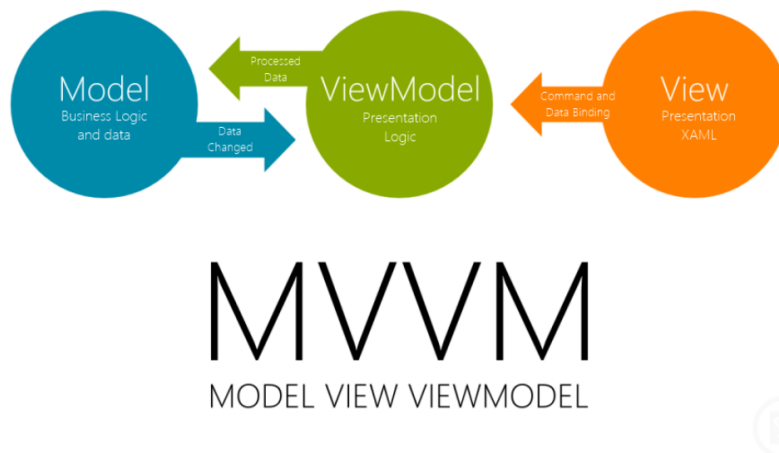
Süsteemi turvalisus ja pidev kättesaadavus on samuti projektil tähtsal kohal, arvestades süsteemi kasutust riikliku ettevõtte poolt. Tellija esitas nõude, et rakendus peab olema võimeline toimima ööpäevaringselt, kaasa arvatud võimalike veebiserverite tõrgete esinemisel. Rakenduse turvalisus oli väga oluline, ning juurdepääsetavus rakendusele on kontrollitud tingimustes, et vältida kolmandate osapoolte ligipääsu.

Rakendus peab olema hõlpsasti haldatav, tagamaks muretu ja arusaadava kasutuskogemuse kõigile kasutajatele, sõltumata nende tehnilisest taustast. Kasutajaliides tuleb kujundada kasutajasõbralikuks ja loogiliseks, et efektiivselt toetada igapäevast töövoogu.

Arvestades abstraktseid nõudeid, ei aita see mitte ainult suurendada süsteemi vastavust algsetele nõuetele, vaid loob ka kindla aluse võimalikeks tulevikulaiendusteks. See strateegiline lähenemine suunab tähelepanu süsteemi arenduse detailsematele tehnilistele aspektidele, tagades, et lõpptulemus vastab täielikult tellija ootustele ja vajadustele.

### **3.3 Arhitektuur**

Autori arhitektuuriline valik toetus WPF raamistiku omadustele eraldada rakenduse kasutajaliidest ning ärireegleid. Seda võimaldas teha MVVM muster, mis on WPF raamistiku puhul väga populaarne. Muster kasutab ära WPF-i võimalusi nagu andmesidumine, mis võimaldab tihedat koostööd vaate ja vaatemudeli vahel ilma, et vaade peaks tegelema andmete haldamise või ärireeglite rakendamisega. MVVM võimaldab hõlpsalt hallata mitmeid vaateid, mis jagavad ühte vaatemudelit, pakkudes paindlikkust kasutajaliidese disaini muutmiseks ilma ärioloogikat mõjutamata. Lisaks hoiab MVVM muster hästi struktureeritud koodi, kus on võimalik rakendada erinevaid disaini mustreid. WPF-i raamistiku osi nagu andmemallid ja käskude infrastruktuur, mis edendavad andmete ja käitumise selget eristamist vaatest [7]. Erinevalt MVP mustrist, kus vaade suhtleb mudeliga läbi esitleja, võimaldab MVVM muster andmesidumist vaate ja vaatemudeli vahel. Vaatemudel võtab andmeid mudelist ning käsitleb kasutaja interaktsiooni, näidates meetode ja käsklusi vaatest [6]. MVP mudel on mustri andmete kiht, mis vastutab andmete kättesaamise üle, vaade on kasutajaliidese kiht, mille abil andmeid näidatakse ning esitleja on loogikakiht, mis töötleb andmeid. MVP muster, kuigi kasulik teatud kontekstides, nõuab esitleja kasutamist andmete edastamiseks vaate ja mudeli vahel, mis võib tekitada lisakihistuse ja raskendada testide kirjutamist [5].



Joonis 2. Model-View-ViewModel.

MVVM muster lihtsustab rakenduse haldamist, võimaldab testida sõltumata kasutajaliidese elementidest ning kood on hästi organiseeritud, mis vähendab vigade teket. Mustri abil on võimalik rakenduse disaini kergelt muuta, mõjutamata rakenduse loogikat [14]. Vaatemudel võtab andmeid mudelist ning käsitleb kasutaja interaktsiooni, näidates meetode ja käsklusi vaatest [6].

### 3.4 Kaart

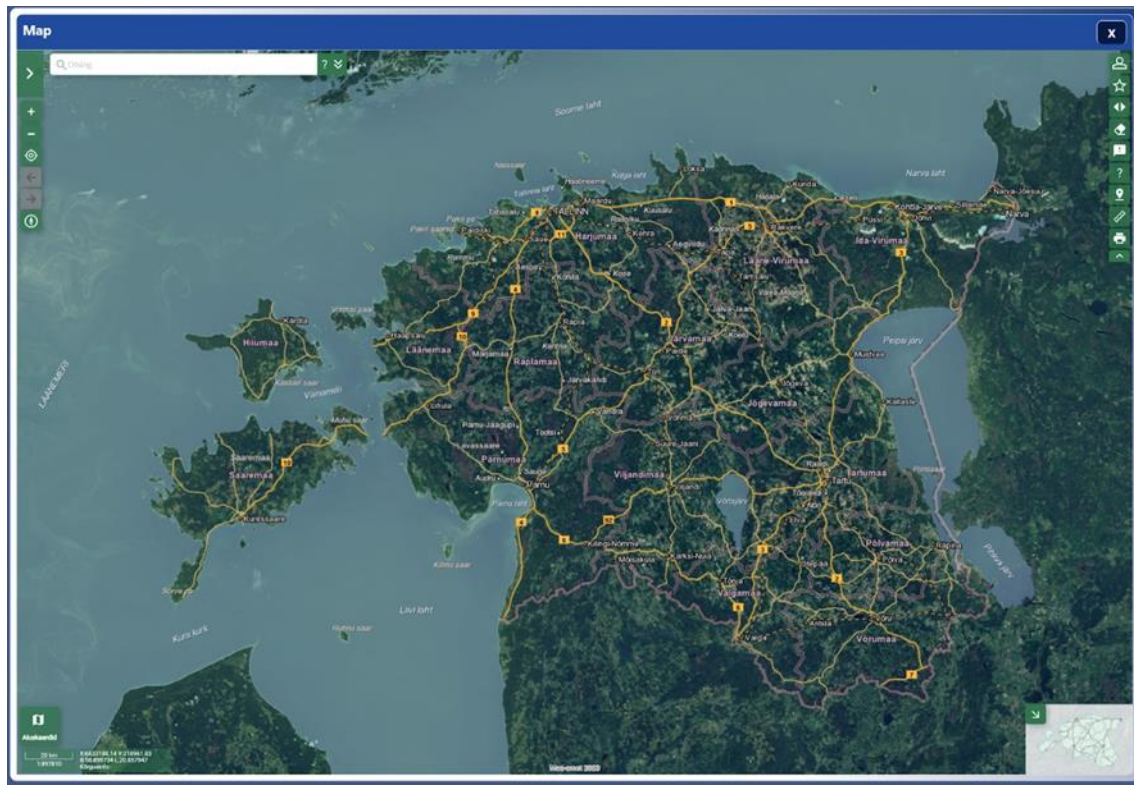
Geoloogilise kaardi valimisel kaaluti erinevaid lahendusi nagu Google Earth ning Maa-ameti Eesti kaart. Kuigi Google Earth on tasuta tarkvaraline kaart, kus kuvatakse erinevate piirkondade geoloogilise kaarte, otsustas autor Maa-ameti kaardi kasuks kuna Maa-amet pakub Eesti kaarti kihtidena ning sisaldab uuemaid andmeid. Samuti mõjutas kaardi valikut asjaolu, et Tarbijakaitse ja Tehnilise Järelevalve Amet on riiklik järelevalve ettevõtte, mis tegeleb Eesti ohutusjärelevalve, tururegulatsiooni ja kohustuste täitmise kontrollimisega [4]. Sellest tulenevalt otsustati minna edasi Eesti Maa-ameti kaardiga, kuna autoril oli soov luua Eesti andmete põhjal rakendus, mida saab kasutada meie sideteenuse monitoorimiseks.

### 3.5 Projekti esmane realiseerimine

Projekti esmasel realiseerimisel keskendus autor analüüsi käigus välja selgitatud tehnoloogiate testimisele, et veenduda meetodika all kirjeldatud rakenduse osade sobilikkus. Funktsionaalsuse loomisel jälgiti tellija poolt tulnud nõudeid, et tagada Sideosakonna töövahendi õigsus.

Rakendus loodi kasutades Visual Studio 2022 IDE programmi, ning varundati GitHubi kaudu pilve, tagamaks andmete säilimise. Rakendus loodi kasutades WPF ning .NET 6 raamistikku, mille abil oli eraldati code-behind-i kood kasutajaliidesest. Kasutajaliides loodi XAML programmeerimiskeelega ning code-behind-i kood C#-ga, millega autor oli varasemalt kokku puutunud.

Suur rõhk oli ArcGIS API'l, mille abil oli võimalik rakendusse luua interaktiivne kaart. Kaardi kuvamine tuli Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakonna nõuetest ning oli projekti jaoks suure tähtsusega. ArcGIS API abil saavutati kaardil sisse ja välja zoomimine, mis oli vajalik visuaalne lahendus. Lisaks võimaldas ArcGIS API rakendada WMS kihte, millel on võimekus kuvada seadmete asukoha punkte. Projektis oli kavandatud punktide lisamise võimalus kaardile kuid on veel pooleli. Esialgne analüüs on tehtud, kuid lõpuni viimistlemine jäi ajapuuduse tõttu saavutamata. Punkti lisamise funktsioon on oluline järgmiste arenduse etappide jaoks koos raadiuse kuvamise, punktide erinevate disainide ning läbi kaardi seadmete lisamisega.



Joonis 3. Interaktiivne kaart.

Nõuetekohaselt soovis tellija lisada, kustutada ning muuta seadmeid rakenduse sees. Selle aitas saavutada C# abil loodud CRUD operatsioonid. XAML abil loodi ühtse disainiga nupud, mis võimaldavad funktsioonide kasutust.

**Lisa uus seade** X

Nimi  
\_\_\_\_\_

Asukoht:  
\_\_\_\_\_

IP:  
\_\_\_\_\_

Asukoha nimetus:  
\_\_\_\_\_

Info:  
\_\_\_\_\_

Raadius:  
\_\_\_\_\_

Töökorras:

Näita kaardil:

**Lisa**

Joonis 4. Seadme lisamine süsteemi.

Seadmete nimekiri sisaldab konkreetseteid tellija poolt edastatud monitooringu seadmeid, mis näitavad kõikide seadmete andmeid nagu nimi, asukoht, IP aadress, asukoha nimetus, kas seade on töökorras, seadme raadius, seadme lisainfo ning seadme raadiuse näitamine kaardil. Lisainfo väli oli samuti tellija nõue, et seadme kohta edastada asjakohast infot tagades kollegide vahel infovahetuse, kuid selle täisfunktsionaalsust ei loodud esmasel realiseerimisel. Täidetud sain tellija nõue nimekirja alusel otsida ning filtreerida seadmeid.

Ühe seadme pingimise funktsionaalsus jäi lõpetamata, kuid loodi eraldi popup aken IP aadresside pingimiseks. Serveripoolne kood jäi testimata ning autor kavatseb tulevikus keskenduda üksik pingimise funktsionaalsuse arendamisele, kui ka automaatse pingimise arendamisele. Täiendavate funktsioonide, nagu seadmete raadiuse kuvamine kaardil ja seadmete lisamine läbi kaardi, arendamine on samuti plaanitud.



## 4 Analüüs ja järeldused

Analüüsi ja järelduste peatükis toob autor välja miks spetsiifilised nõuded valiti, millised on analüüsi tulemused ning alternatiiv lahenduse projekti arendamisel. Kirjeldatakse kitsendusi, mis piirasid rakenduse analüüsi valikuid, tuuakse välja analüüsi ja projekti realiseerimise tulemused ning esitletakse tulevikus tehtavad tööd, et rakendus saavutaks tellija ootuste kohase funktsionaalsuse.

### 4.1 Nõuete valimine

Autor kohtus mitmel korral Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakonna töötajatega, et mõista nende vajadusi uue süsteemi järele. Tellija tegi ülevaate kasutusel olevast süsteemist, ning tõi välja funktsionaalsused, mis on uue süsteemi poolt oodatud.

Sideosakonna eelnev lahendus, koosnes kahest osast: süsteemi kaudu ping-imisest ning seadme otsimisest Eesti kaardilt, et sinna kaugelt sisse saada. PingInfoView rakenduse abil kontrolliti seadmete reageerimisaega läbi ping-imise, mis on internetiprogramm, mis võimaldab kasutajal testida ja kontrollida, kas seadme IP-aadress suudab vastu võtta päringuid arvutivõrkude haldamisel. Ping-imist kasutatakse diagnostiliselt, et veenuda, kas kasutaja sihtarvuti töötab [17]. Kui seade ei reageerinud 10ms jooksul, andis see märku, et tegu on ebakorrektselt funktsioneeriva seadmega. Rakendusest võeti aeglaselt reageeriva seadme IP aadress, mille alusel leiti arvuti kõvakettalt seadme nimi, mis sisaldas seadme asukohta. Seejärel otsiti arvuti tausta pildilt, mis kujutas Eesti kaarti, üles vigase seadme kaugelt sisse minemise rakendus. Hallatavad seadmete olid paigutatud seadme asukoha järgi Eesti kaardile, et kiirendada seadme rakenduse üles otsimise protsessi. Seadmesse logiti kaugjuurdepääsu abil sisse ning kontrollitakse, milles viga seisneb.

Olemasoleva süsteemi aja kulukuse ning ebamugavuse pärast soovis Sideosakond uue töövahendi loomist, mis lubab seadmeid kontrollida ning hallata ühe süsteemi kaudu. Tellija leidis, et peale uue rakenduse kasutuselevõttu väheneb ajaline kulu monitooringu seadmete kontrollimise puhul. PinginfoView rakenduses oli samuti võimalik lisada, kustutada ning muuta seadmete andmeid ja reguleerida ping-imiste vahel olevat aega. Sellest tulenevalt olid samad nõuded ka antud projektil.

Uute funktsioonidena soovis tellija lisada interaktiivse kaardi, mis kuvaks seadmete asukohta kui ka seadmete võrgus olekut ning lisaks seadmete võrgu katvust. Selle funktsionaalsuse abil on töötajal visuaalselt kiirelt näha, kas kõik seadmed üle Eesti on reageerinud või on kuskil piirkonnas tekkinud muresi. Seadmete võrgu kaetavuse lisamine tuleneb asjaolust, et Sideosakonnal on vaja teha ministriumile ülevaateid, kuidas võrk on üles seatud ning kui suure ala katavad monitooringu seadmed. Kui on olemas interaktiivne kaart, siis tellija soov oli lisada seadet vajutades kaardil seadme asukohale ning läbi selle lisamise protsess läbi teha.

Mugavuse pärast soovis tellija seadme otsingut ning grupeerimist, et seadmete nimekirjast oleksid aeglasemalt reageerivad seadmed kiiremini leitavad, kuna jälgimise all olevate seadmete arv on täna 30 ringis, kuid tulevikus võib suureneda. Tellija soovis võimalust seadmete ping-imise regulaarsust ise muuta, juhul kui tekivad süsteemis samaaegselt mitmeid probleeme, siis on võimalik muuta millise aja tagant seadmed pingivad. Samuti juurde tuua ühe seadme ping-imise võimalus, et kontrollida eraldi seadme reageerimisega, ilma, et peaks ootama millal uus automaatne ping-imise laine tuleb.

Kuna osakonnas liigub palju infot, soovis tellija, et iga seadme juures oleks märkmete tegemise võimalus. Leiti, et see on vajalik, kui seadme kohta on ilmunud uut vajalikku informatsiooni, mida edastada kolleegile, ilma info kadumiseta.

## **4.2 Tulemuste ülevaade**

Vastavalt tellija nõuetele ning tulevikku suunatud visioonile, on kavandatud rakendus töölaarakendus, mille andmed hoitakse lokaalselt arvuti vahemällu. See hõlmab ka kaarti, mis laaditakse serverist, kuid mille andmed on samuti varundatud lokaalselt, tagamaks rakenduse toimimist ka juhul, kui kaardiserver peaks katkema. Rakendus kasutab internetiühendust seadmete ping-imiseks ja kaardi kuvamiseks.

Kaardil kuvatakse seadmete asukohti värviliste punktidenä, mis näitavad, kas seade on võrgus või võrgust väljas. Võimalik on visuaalselt kuvada seadmete leviala raadiuseid, andes ülevaate nende kaetavuse ulatusest.

Rakenduse funktsionaalsused hõlmavad ka seadmete nimekirja haldust, võimaldades seadmete grupeerimist ja otsingut vastavalt kasutaja vajadustele. Seadmed on grupeeritavad erinevate kriteeriumide järgi. Kavandatud on automaat ping-imise

funktsioon, mis võimaldab regulaarselt kontrollida seadmete ühenduvust. Kasutajad saavad seadistada automaat ping-imise sagedust ning on võimalik teostada käsitsi ping-imist konkreetsetele IP-aadressidele.

#### **4.2.1 Analüüsi tulemuste ülevaade**

Bakalaureusetöö võrdleva analüüsi põhjal jõuti järeldusele, et Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakonna töövahendiks on kõige sobilikum kasutada järgmisi tehnoloogilisi lahendusi.

Turvalisuse kaalutlustel on kõige mõistlikum kasutada töölaarakendust, mille kokkuleppisid autor ning tellija projekti algfaasis. See valik aitab tagada, et kolmandatel osapooltel on keerukam ligipääs kriitilistele andmetele, suurendades seeläbi süsteemi turvalisust.

Tehnoloogial valikul toetuti autori ning tellija poolsetele kitsendustele ning WPF raamistik sai valitud oma kindlusele ja lihtsusele omadustele. Raamistik kuulub .NET alla ning võimaldab tõhusalt eraldada code-behind koodi kasutajaliidese koodist, muutes rakenduse arendusprotsessi paindlikumaks ja lihtsamini hallatavaks.

Kasutajaliidese koodi arendamisel kasutati XAML-i, mis on standard WPF rakendustes. Code-behind koodi juures jäi autor kindlaks C# keele, millega tal oli eelnev kogemus ülikooliõpingute käigus. See valik välistas vajaduse õppida uut programmeerimiskeelt, vähendades arendus aega ja tõstes koodi kvaliteeti. C# abil oli võimalik luua rakendusele funktsionaalsused, mida nõuded ette nägid nagu seadmete haldamine, otsing, filtreerimine. Keele abil on võimalik lisada süsteemi sisene ping-imine.

ArcGIS API valiti oma rikkaliku dokumentatsiooni ja geograafiliste andmete analüüsivõime tõttu. See tehnoloogia pakub paindlikke funktsioone ja kindlat ligipääsu kaardi andmetele URLi kaudu, mis on oluline dünaamilise kaardi põhise info esitamiseks. ArcGIS API võimaldab serveri väliselt interaktiivse kaardi kasutust, mis oli oluline nõue tellija poolt tagamaks rakenduse stabiilse funktsioneerimise.

WMS kihtide kasutamine võimaldas projektil lisada ja hallata kaardikihte hõlpsalt. Kuna projekt nõudis punktide lisamist kaardile ilma geoloogiliste andmete muutmata, osutus WMS kihtide kasutamine optimaalseks lahenduseks. Eraldi kaardikihi ehitamine võimaldab lisada aluskaardile punkte ning sümboleid, mis on tellija nõuete kohaselt väga

olulised, et kuvada võrgus olevaid seadmeid ning nende kaetavuse raadiusi. Samuti luues kihi on võimalik tekitada kaardi lisamise funktsionaalsus läbi kaardikihi, mille tulemusena on võimalik kaardil vajutades punkti (seadme asukohta) lisada.

#### **4.2.2 Realiseerimise ülevaade**

Projekti esmase realiseerimise tulemusena sai kinnitust, et analüüsi käigus valitud tehnoloogiad on sobilikud rakenduse arendamiseks. Kuigi kõikide nõuete funktsionaalsuste kontrolli ei jõutud teha, sai töö omanik siiski kindluse, et rakendus on loodud kasutades autori arvates parimaid tehnoloogilisi lahendusi.

- Kasutades ArcGIS API'd, loodi rakendusse interaktiivne kaart, mis võimaldab sisse ja välja suumimist. Antud funktsionaalsus oli tellija nõue, et kasutajad saaksid vaadelda kaarti erinevatel detaili tasemetel. See on osutunud võtmetähtsusega funktsionaalsuseks, mis aitab visualiseerida seadmete paigutust ja võrgu katvust.
- Rakendusele lisati võimekus lisada, kustutada ja muuta seadmeid, kasutades C# keeles loodud CRUD operatsioone. See funktsionaalsus on hädavajalik süsteemi administreerimiseks ja päevast päeva toimimiseks, võimaldades hõlpsat seadmete haldust.
- XAML-i abil kujundati kasutajaliides, mis pakub ühtset ja intuitiivset kogemust, muutes seadmete otsimise ja filtreerimise lihtsaks ning arusaadavaks.

Analüüsi tulemuste realiseerimisel loodi esialgne rakendus inglise keelsena, ning alustati rakenduse eesti keelsele üleminekut. Kuigi mitmed rakenduse eesmärgid jäid saavutamata nagu punkti lisamine, punktide värvide kuvamine, seadme kaetavuse raadiuse kuvamine ning seadmete pingimine, sai autor kindluse, et valitud raamistikud ning tehnoloogiad on kõige etemad antud projekti arenduseks.

### **4.3 Alternatiivid**

Alternatiivina on analüüsi peatükis 2 välja toodud erinevad alternatiiv lahendused mis analüüsi tulemusena välja jäid.

Alternatiiv tehnoloogiad WPF raamistikule olid Avalonia ning .NET MAUI. Avalonia jäi valikust välja kuna mitut platvormi toetav lahendus oli keerulisem ning antud projekt ei vajanud lisaks töölaarakendusele veebi- või mobiili tuge. .NET MAUI oli kõige uuem raamistik kuid raamistikus oli ilmnenud vigu, mis oleksid muutnud projekti realiseerimise keerulisemaks ning vähem usaldusväärsemaks.

Code-behind koodil tõi välja autor alternatiiv keelena F#, mis on .NET raamistikule sobilik ning toetab objektorienteeritud programmeerimist kuid C# keelega oli autoril suurem kokkupuude ning seetõttu jäi F# valikust välja.

Interaktiivse kaardi loomisel kaaluti ArcGIS API ning QGIS Serveri vahel ning loobuti QGIS Serveri kasutamisest kuna platvormi kohta oli vähem infot ning QGIS Server ei toetanud punkti lisamist rakendusele.

Kaardikihtide uurimisel vaadeldi WMS ja WFS erinevusi ning leiti, et WFS on kehvem valik kuna tal puudub kihtide jagamise võimekus veebikaardiga ning kihid rohkem keskenuvad geoloogiliste andmete muutmisele, kui kaardikihtide visuaalsele kuvamisele.

Autor ei suutnud leida alternatiiv rakendust, mis koosneks seadmete reageerimisaja kontrollimisest ning seadmete kuvamisest interaktiivsel kaardil, mis olid Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakonna kindlad nõuded. Sellise spetsiifilise kombinatsiooni puudumine viitab asjaolule, et on vaja arendada sobiv kohandatud rakendus.

#### **4.4 Kitsendused ja piirangud**

Projekti algusfaasis seadis autor piirangud, valides C# keele ning tellija soovil töölaua rakenduse platvormi, et lihtsustada esmase prototüübi valideerimisprotsessi. Valik tagas, et rakenduse esmane arendus on autori jaoks hallatav ning autori tugevamad teadmised C# keelega võimaldasid arendusprotsessi kiirendada. Antud piirangud olid strateegiliselt valitud, et optimeerida arendusprotsessi ja tagada projekti nõuetekohane täitmine.

Tellijal esitas mitmeid piiranguid, mille alusel tuli rakendust arendada, et tagada lõpptulemus, mis vastaks Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakonna vajadustele. Üheks olulisemaks kitsenduseks oli rakenduse töökindlus serveri võimaliku tõrke korral. Nimelt nõudis tellija, et isegi serveri maha kukkumise korral peab rakendus

jätkama toimimist. Selle saavutamiseks kasutati ArcGIS API-d, mis võimaldab veebikaardi andmeid vahemällu laadida, et tagada katkematu juurdepääs kaardiinfole. Lisaks kaardi interaktiivsusele, nagu suumimisvõimalus, tuli rakendusse integreerida funktsionaalsused nagu punktide lisamine kaardile ja nende staatus muutuste kuvamine, samuti iga punkti raadiuse visualiseerimine, et anda kasutajatele selge ülevaade seadmete kaetavusest.

Rakenduse arendamise käigus peeti silmas ka laiendatavust ja tulevikukindlust. Autor soovis tagada süsteemi arhitektuuri piisava paindlikkuse, et toetada tulevasi uuendusi ja integratsioone, näiteks täiendavate andmeallikate lisamist või uute funktsionaalsuste integreerimist vastavalt Sideosakonna arenevatele vajadustele.

Seega, rakenduse arendamisel tuli tasakaalustada tellija nõuded töökindluse, funktsionaalsuse ja kasutajasõbralikkuse vahel, tagades samas süsteemi laiendatavuse ja tuleviku valmiduse.

#### **4.5 Tulemuste valideerimine**

Analüüsi koostamisel toetus autor erinevatele ametlikele allikatele, sealhulgas raamatutele, veebipõhistele materjalidele ning raamistike või meetodite ametlikele kodulehtedele, mis käsitlesid geoinfosüsteeme, tarkvara arendust ning aitasid leida parimaid lähenemisi projekti nõuetele. Autor kitsendas rakenduse skoopi, et praktilise osa loomine oleks võimetekohane, sellest tulenesid raamistiku ning meetodite valikud. Allikad aitasid kindlaks määrata meetodite teoreetilist sobivust ning andsid aluse rakenduse esmasele arenduskäigule.

Projekti realiseerimise tulemusena on näha, kuidas potentsiaalselt rakendus tööle hakkab, mis kinnitab valitud meetodite ja raamistike sobivust. Esialgne rakenduse prototüüp demonstreeris WPF ning ArcGIS API kasutamise võimekust interaktiivse kaardirakenduse loomisel. Näiteks Eesti kaardi kuvamine ning sisse ja välja suumimine kinnitavad, et valitud lähenemine oli asjakohane.

Tulevikus on plaanis läbi viia põhjalikud kasutajatestid ja tehnilised testid, et valideerida rakenduse korrektset funktsioneerimist, mille abil on võimalik saada kinnitus, et rakendus mitte ainult ei vasta tehnilistele nõuetele, vaid on ka kasutajate jaoks efektiivne.

## 4.6 Äriline põhjendamatus

Autori ei leidnud ärilist põhjendust luua kaardile seadmete raadiuse kuvamise funktsionaalsus. Tellija nõue nägi ette iga seadme juures võimalus kuvada kaardil tema katvuse piirkond läbikumava ringina, mis aitaks visuaalselt näidata seadme poolt kaetud ala. Antud funktsiooni arendamise soov tulenes vajadusest demonstreerida ministritele, kuidas on Eesti võrgud geograafiliselt kaetud. Kuigi kaetavuse ülevaade on peamiselt vajalik uute ministrite ametisse astumisel, et anda neile kiire ja arusaadav ülevaade võrgu ulatusest, leidis autor, et projekti piiratud ressursside tõttu ei õigusta selle funktsionaalsuse väljatöötamine end äriliselt ära. Vaatamata sellele, tunnistab ta, et funktsionaalsus on tellija jaoks strateegiliselt tähtis. Seega, arvestades tellija vajadust ja projekti pikaajalisi eesmärke, on edasistes arendus etappides plaanis raadiuse kuvamine kaardil lisada.

## 4.1 Tuleviku tööd

Bakalaureusetöös keskenduti nõuete analüüsile ning esmase rakenduse realiseerimisele, mis pani aluse põhjalikule ja funktsionaalsele rakendusele. Kuigi esmane arendus vastas mitmetele olulistele tellija nõuetele, on selge, et süsteemi potentsiaali saavutamine nõuab edasist arendamist.

Läbi ArcGIS API võimaldab rakendus interaktiivselt kuvada Eesti kaarti, pakkudes kasutajatele kaardi sisse ja välja zoomimise funktsionaalsust. Nõuetele toetudes on projekti kaardile tulevikus vaja lisada seadmete asukohtade dünaamiline kuvamine, mis võimaldab visuaalselt jälgida võrguseadmete seisundit. Oluline on punktide värvide vahetus, mis näitab seadmete võrgus olemise või võrgust väljas oleku staatust. Esmane analüüs punktide lisamise kohta on autoril lõpetatud, kuid punkti arendus on planeeritud järgmisesse etappi. Süsteemile on lisamata seadmete raadiuse kuvamise võimalus kaardil, mis on samuti oluline arendust vajav funktsionaalsus. Seda suudab luua kasutades ArcGIS APIt ning antud lisaväärtus aitab tutvustada Sideosakonnal Eesti võrgu monitooringu ülevaadet ministriumile.

Esmase arenduse käigus integreeriti seadmete nimekiri, mis toetab seadmete otsimist ja grupeerimist vastavalt nende võrgusoleku staatusele. Andmebaasi on lisatud seadmete

nimed, IP aadressid, asukohad, asukoha nimetused. Töö käigus arendati XAML-il põhinev kasutajaliides, mis kuvab lisainfo, raadiuse ning raadiuse kuvamise välja, kuid funktsionaalsuse code-behind lisati tuleviku tööde hulka.

Praeguseks on välja töötatud alus ühekaupa seadme ping-imine kuid see vajab täiendavat arendamist, et antud funktsioon vigadeta tööle saada. Tuleviku tööde nimekirja jääb üksiku ping-imise arenduse lõpetamine ning automaatping-imise funktsiooni loomine koos ping-imise perioodi seadistamise võimalusega.

Antud analüüs määrab selgeid samme tulevikuks, tagades projekti jätkusuutliku arengu ja kasutajate vajaduste täitmise.



## 5 Kokkuvõte

Tarbijakaitse ja Tehnilise Järelevalve Ameti Sideosakond tegeleb sideteenuste järelevalvega, mille tõttu ettevõtte kontrollib, kas võrku on tekkinud raadiosagedusi, mis võivad rikkuda Eesti raadiosageduste tööd. Algne võrguseadmete monitoorimis lahendus, mida Sideosakond igapäevaseks tööks kasutas oli ebapraktiline ning tülikas.

Projekti eesmärk oli tellija nõuete analüüsimine ning esmase rakenduse arendamine, mille põhjal on võimalik tulevikus luua funktsioneeriva töövahendi. Autor analüüsis raamistikke ning meetodeid, et leida sobivad vahendid realiseerimiseks. Välja valiti WPF raamistikul põhinev töölaarakendus, mis kasutab XAML ning C# programmeerimiskeelena. Töös tuuakse välja autori loodud nõuded, mis põhinevad Sideosakonna tagasisidel ning uue süsteemi ootustel. Autor toob eraldi välja abstraktsed nõuded, et rõhutada projekti olulisust tellijale. Peale analüüsi koostab autor projekti esmase analüüsimise, mille käigus testitakse eelnevalt välja valitud raamistike ning meetodite sobivust ning võimekust.

Nõuete analüüsi tegemine andis kindla sihi ning tööriistad, mida kasutada rakenduse arenduseks. Esmase realiseerimisega leiti, et valitud vahendid olid sobilikud ning toetasid rakenduse loomist. Kuigi projekti raames loodi prototüüp, mille ülesanne oli veenduda analüüsi tulemuste õigsuses, on rakendusel potentsiaali muutuda tellija töövahendiks, mis aitab kiirendada igapäeva toiminguid.

## Kasutatud kirjandus

- [1] [1] L.Moroney, „Foundations of WPF: An Introduction to Windows Presentation Foundation” [Võrgumaterjal], [https://books.google.ee/books?hl=en&lr=&id=Se8OSNB95rcC&oi=fnd&pg=PP1&dq=what+is+wpf&ots=whrcTYDnN\\_&sig=IzwZjdaO\\_iPPDQSAjPAD-Qcg7Bs&redir\\_esc=y#v=onepage&q=what%20is%20wpf&f=false](https://books.google.ee/books?hl=en&lr=&id=Se8OSNB95rcC&oi=fnd&pg=PP1&dq=what+is+wpf&ots=whrcTYDnN_&sig=IzwZjdaO_iPPDQSAjPAD-Qcg7Bs&redir_esc=y#v=onepage&q=what%20is%20wpf&f=false) [Kasutatud 21.04.2024]
- [2] [2] M.Dalal, A.Ghoda, „XAML Developer Reference“ [Võrgumaterjal], [https://www.google.ee/books/edition/XAML\\_Developer\\_Reference/I6pCAwAAQBAJ?hl=en&gbpv=1](https://www.google.ee/books/edition/XAML_Developer_Reference/I6pCAwAAQBAJ?hl=en&gbpv=1) [Kasutatud 21.04.2024]
- [3] [3] R.Rubalcava, “ArcGis Web Development”, [Võrgumaterjal], [https://books.google.ee/books?hl=en&lr=&id=ZTgzEAAAQBAJ&oi=fnd&pg=PT12&dq=arcGis+api&ots=fCcBZbbxmu&sig=Wi6AfeUBvfmmpCyWvYoLSNEs8k&redir\\_esc=y#v=onepage&q=arcGis%20api&f=true](https://books.google.ee/books?hl=en&lr=&id=ZTgzEAAAQBAJ&oi=fnd&pg=PT12&dq=arcGis+api&ots=fCcBZbbxmu&sig=Wi6AfeUBvfmmpCyWvYoLSNEs8k&redir_esc=y#v=onepage&q=arcGis%20api&f=true) [Kasutatud 21.04.2024]
- [4] [4] Tarbijakaitse ja Tehnilise Järelevale Amet, „Tutvustus ja Struktuur“, [Võrgumaterjal], <https://www.ttja.ee/ariklient/ametist/ametist/tutvustus-ja-struktuur> [Kasutatud 21.04.2024]
- [5] [5] D.D.Li, X.Y.Liu, „Research on MVP Design Pattern Modeling Based on MDA“, [Võrgumaterjal], <https://www.sciencedirect.com/science/article/pii/S1877050920301344> [Kasutatud 21.04.2024]
- [6] [6] R.Vincent, „Learning ArcGis Runtime SDK for .NET“, [Võrgumaterjal], [https://www.google.ee/books/edition/Learning\\_ArcGIS\\_Runtime\\_SDK\\_for\\_NET/wf9vDQAAQBAJ?hl=en&gbpv=1&dq=WMS+layer+wpf&printsec=frontcover](https://www.google.ee/books/edition/Learning_ArcGIS_Runtime_SDK_for_NET/wf9vDQAAQBAJ?hl=en&gbpv=1&dq=WMS+layer+wpf&printsec=frontcover) [Kasutatud 21.04.2024]
- [7] [7] J. Smith, “Patterns – WPF Apps With The Model-View-ViewModel Design Pattern”, [Võrgumaterjal], <https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern> [Kasutatud 17.05.2024]
- [8] [8] J.D. Blower, A.L. Gemmell, G.H. Griffiths, K. Haines, A. Santokhee, X. Yang, „A Web Map Service implementation for the visualization of multidimensional gridded environmental data“, [Võrgumaterjal], <https://www.sciencedirect.com/journal/environmental-modelling-and-software/vol/47/suppl/C> [Kasutatud 12.05.2024]
- [9] [9] „XAML overview (WPF .NET)“, [Võrgumaterjal], <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-8.0> [Kasutatud 13.05.2024]
- [10] [10] „A tour of the C# language“, [Võrgumaterjal], <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview> [Kasutatud 13.05.2024]
- [11] [11] P. Paul, “8 Benefits of Desktop Over Web-Based Applications”, [Võrgumaterjal], <https://www.trionds.com/benefits-of-desktop-over-web-based-applications/> [Kasutatud 13.05.2024]
- [12] [12] “What is .NET MAUI?”, [Võrgumaterjal], <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-8.0> [Kasutatud 13.05.2024]

- [13] [13] GeeksForGeeks, “What is WPF?”, [Võrgumaterjal], <https://www.geeksforgeeks.org/what-is-wpf/> [Kasutatud 13.05.2024]
- [14] [14] J. Smith, “Patterns - WPF Apps With The Model-View-ViewModel Design Pattern”, [Võrgumaterjal], <https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern> [Kasutatud 13.05.2024]
- [15] [15] ArcGIS Developers, “Offline maps”, [Võrgumaterjal], <https://developers.arcgis.com/documentation/mapping-apis-and-services/offline/offline-maps/working-with-offline-maps/> [Kasutatud 17.05.2024]
- [16] [16] ArcGIS Pro, “Use WFS services layers”, [Võrgumaterjal], <https://pro.arcgis.com/en/pro-app/latest/help/data/services/use-wfs-services.htm> [Kasutatud 17.05.2024]
- [17] [17] A. Zola, “What is a ping?”, [Võrgumaterjal], <https://www.techtarget.com/searchnetworking/definition/ping> [Kasutatud 13.05.2024]
- [18] [18] ArcGIS Pro, “Use WMS services layers”, [Võrgumaterjal], <https://pro.arcgis.com/en/pro-app/latest/help/data/services/use-wms-services.htm> [Kasutatud 17.05.2024]
- [19] [19] C. Smith, “Programming F#”, [Võrgumaterjal], [https://books.google.ee/books?hl=en&lr=&id=gzVdyw2WoXMC&oi=fnd&pg=PR5&dq=f%23&ots=I\\_UiInSZQC&sig=o4lckqHevExITm24xNLCVbqiflU&redir\\_esc=y#v=onepage&q=f%23&f=false](https://books.google.ee/books?hl=en&lr=&id=gzVdyw2WoXMC&oi=fnd&pg=PR5&dq=f%23&ots=I_UiInSZQC&sig=o4lckqHevExITm24xNLCVbqiflU&redir_esc=y#v=onepage&q=f%23&f=false) [Kasutatud 17.05.2024]
- [20] [20] Avalonia UI, “Create Multi-Platform Apps with .NET”, [Võrgumaterjal], <https://avaloniaui.net/> [Kasutatud 17.05.2024]
- [21] [21] Avalonia, “WPF and UWP Comparison”, [Võrgumaterjal], <https://docs.avaloniaui.net/docs/get-started/wpf/comparison-of-avalonia-with-wpf-and-uwp> [Kasutatud 17.05.2024]
- [22] [22] BC. D. Valko, “C#/.NET GUI toolkit comparison”, [Võrgumaterjal], [file:///C:/Users/stude/Downloads/C\\_\\_NET\\_GUI\\_toolkit\\_comparison\\_Archive.pdf](file:///C:/Users/stude/Downloads/C__NET_GUI_toolkit_comparison_Archive.pdf) [Kasutatud 17.05.2024]
- [23] [23] QGIS Documentation, “QGIS Server Guide/Manual”, [Võrgumaterjal], [https://docs.qgis.org/3.34/en/docs/server\\_manual/index.html](https://docs.qgis.org/3.34/en/docs/server_manual/index.html) [Kasutatud 17.05.2024]
- [24] [24] M. R. Saeed, “Model-View-ViewModel (MVVM-Part 1 - Overview”, [Võrgumaterjal], <https://rehansaeed.com/model-view-viewmodel-mvvm-part1-overview/> [Kasutatud 19.05.2024]
- [25] [25] D. Collier, “The Comparative Method”, [Võrgumaterjal], [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=1540884](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1540884) [Kasutatud 19.05.2024]

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

Mina, Sorell Tudelep

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Monitooringu seadmete jälgimissüsteemi nõuete analüüs ning esmane realiseerimine Tarbijakaitse ja Tehnilise Järelevalve Ametile“, mille juhendaja on Viljam Puusep
  - 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.

20.05.2024

---

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

## Lisa 2 – Laiendatud menüü paneel



## Lisa 3 – Seadme informatsiooni muutmine

### Muuda seadet X

Nimi

Asukoht:

IP:

Asukoha nimi:

Info:

Raadius:

Näita kaardil:

Töökorras:

**Uuneda**

# Lisa 4 – Seadmete nimekiri

Name	Location	IP	Location Name	In Working	Radius	Info	Show radius on map		
ES206	10.30.40.11	KOHTLA-JÄRVE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
ES206	110.30.40.61	HIASTE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
EM206	10.30.40.70	LAONAMÄE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
LOGOP	10.30.40.72	LAONAMÄE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
DESKTOP	10.30.40.80	PÄRNU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
EM200	10.30.40.81	PÄRNU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
EM200	10.30.40.91	KIBESSAARE	False	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
AG0	190.80.123.73	KOHTLA-JÄRVE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
MIKROTIK	190.80.123.74	KOHTLA-JÄRVE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
TELIA	190.80.126.09	TARTU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
MIKROTIK	190.80.126.70	TARTU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
EM200	10.30.40.21	TARTU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
MIKROTIK	212.197.47.171	MÕISU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
TELIA	194.126.126.9	SIURPALU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
MIKROTIK	194.126.126.10	SIURPALU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
MIKROTIK	88.190.299.130	HIASTE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
TELIA	88.190.299.129	HIASTE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
AG0	213.33.228.249	LAONAMÄE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
MIKROTIK	213.33.228.250	LAONAMÄE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
TELIA	80.215.113.81	PÄRNU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
MIKROTIK	80.215.113.82	PÄRNU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
MIKROTIK	80.89.32.228	KIBESSAARE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
DESKTOP	10.30.40.30	KIVIMÄE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
TELIA	80.89.32.225	KIBESSAARE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
EM206	10.30.40.31	KIVIMÄE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
DESKTOP	10.30.40.40	MÕISU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
ES200	10.30.40.41	MÕISU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
DESKTOP	10.30.40.50	SIURPALU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
EM200	10.30.40.51	SIURPALU	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	
DESKTOP	10.30.40.60	HIASTE	True	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	TTIA_Models.DeviceMod	0.00	Online	

# Lisa 5 – Seadmete grupeeriing

Devices List									
Name	Location	IP	Location Name	Is Working	Radius	Info	Show radius on map		
ES200	10.30.40.11	KOHTLA-JÄRVE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
ES200	110.30.40.61	HAASTE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
EM300	10.30.40.70	LASNAMÄE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
LOGO2	10.30.40.72	LASNAMÄE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
EM300	10.30.40.81	PÄRNÜ	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
ASO	195.80.123.73	KOHTLA-JÄRVE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
MIKROTIK	195.80.123.74	KOHTLA-JÄRVE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
TELIA	195.80.126.49	TARTU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
MIKROTIK	195.80.126.70	TARTU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
EM300	10.30.40.21	TARTU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
MIKROTIK	212.107.47.771	MÕIGU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
TELIA	194.126.126.9	SUURPALU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
MIKROTIK	194.126.126.10	SUURPALU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
MIKROTIK	88.196.209.138	HAASTE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
TELIA	88.196.209.129	HAASTE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
ASO	213.35.228.249	LASNAMÄE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
MIKROTIK	213.35.228.250	LASNAMÄE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
TELIA	80.235.13.81	PÄRNÜ	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
MIKROTIK	80.235.13.82	PÄRNÜ	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
MIKROTIK	85.89.32.226	KURESSAARE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
DESKTOP	10.30.40.30	KIVIMÄE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
TELIA	85.89.32.225	KURESSAARE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
EM300	10.30.40.31	KIVIMÄE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
DESKTOP	10.30.40.40	MÕIGU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
ES200	10.30.40.41	MÕIGU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
DESKTOP	10.30.40.50	SUURPALU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
EM300	10.30.40.51	SUURPALU	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
DESKTOP	10.30.40.60	HAASTE	True	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
DESKTOP	10.30.40.80	PÄRNÜ	False	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	
EM300	10.30.40.51	KURESSAARE	False	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	TTIA.Models.DeviceMod	500	Delete	



# Lisa 6 – Seadme otsing

### Devices List

Search:  Matches: 6  Group Online and Offline Devices

Name	Location	IP	Location Name	Is Working	Radius	Info	Show radius on map	edit	Delete
EM200	10.30.40.70	LASHAMAE	True	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	edit	Delete	
EM200	10.30.40.81	PÄRNI	True	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	edit	Delete	
EM200	10.30.40.51	KURESSAARE	False	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	edit	Delete	
EM200	10.30.40.21	TARTU	True	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	edit	Delete	
EM200	10.30.40.31	KIVIIMAE	True	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	edit	Delete	
EM200	10.30.40.51	SUURPALU	True	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	TTJA_Models.DeviceMod	edit	Delete	

 TARELIKAITSE JA  
TEHNILISE JÄRELEVALVE  
AMET

## Lisa 7 – Seadme kustutamine

