

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Laura Herma  
164928IAPB

**ANDROIDI RAKENDUS  
ELEKTRITÕUKERATASTE LEIDMISEKS  
LINNAKESKKONNAS BLUETOOTH LE  
MAJAKATE ABIL**

Bakalaureusetöö

Juhendaja: Jaanus Kaugerand  
MSc

Tallinn 2019

## **Autorideklaratsioon**

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

Autor: Laura Herma

21.05.2019

## Annotatsioon

Käesoleva töö eesmärgiks on *Bluetooth* LE majakate abil leida üles elektrilised tõukerattad linnakeskkonnas, kus asukohamääramine satelliit navigatsiooni abil on linnakeskkonna tõttu piiratud täpsusega. Täpsemalt on töö eesmärgiks valmis teha Android rakendus, mis leiab tõukeratta poolt saadetud geograafilise asukoha abil üles tõukeratta üldise piirkonna ja majaka tehnoloogiat kasutades tõukeratta täpsema asukoha.

Töö käigus uuriti milliseid majakaid on turult kättesaadavad, mis formaate nad kasutavad ja mida peab silmas pidama majaka ostmisel. Rakenduse tegemiseks on konkreetselt kasutatud Kontakt.io majakat, millega rakendus koos töötab.

Töö tulemusena selgus, et majakaid ja satelliit navigatsiooni asukoha määramist saab kasutada koos elektritõukerataste otsimiseks, kuid pikematel distantidel kui 30 meetrit on seda raske anda täpselt edasi, sest RSSI kõigub väliskeskkonna tõttu palju.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 5 peatükki, 18 joonist, 3 tabelit.

## **Abstract**

### **Android Application for Finding Scooters in City Environment Using Bluetooth LE Beacons**

The purpose of this Thesis was to find how to search scooters with beacons in the city environment. More specifically to create an Android application that finds location of the scooters in the city environment using satellite navigation and beacons.

During the course of the Thesis, different options of beacons in the market were researched and studied what to keep in mind before purchasing beacons. The study involved which formats exist for beacons, how secure the devices are and which Bluetooth version to choose. In the end, Kontakt.io beacon was chosen for development.

For final solution an Android App and a small desktop webpage were created which emulated the behavior of scooters. Webpage sends location coordinates to server and app uses it to add markers in the correct place on Google Maps. If phone gets connection from beacon, the marker's color turns green. Otherwise markers are colored blue. After getting connection from beacon there is possibility to get more information when clicking on the marker – A progressbar and a label which indicate how many meters away the scooter approximately is.

As a result it was discovered that beacons make it easier to find scooters but nevertheless it is still quite hard to display the location precisely.

The thesis is in Estonian and contains 28 pages of text, 5 chapters, 18 figures, 3 tables.

## Lühendite ja mõistete sõnastik

<i>Bluetooth LE</i>	<i>Bluetooth Low Energy</i> , Bluetoothi tehnoloogia, mis käitub sarnaselt bluetoothi varasematele versioonidele, kuid kasutab vähem energiat 2.2.1
RSSI	Received Signal Strength Indicator ehk vastuvõetud signaalitugevuse indikaator
EID	<i>Ephemeral Identifier</i> ehk krüpteeritud ID Eddystone majakal
SDK	<i>Standard Development Kit</i> -ehk tarvaraarendus komplekt
URL	Uniform Resource Locator ehk internetiaadress
IP-68	Ingress Protection, kaitseaste, esimene number näitab kaitset tahketele objektidele ja teine number näitab kaitset vedelikele (6-täielikult kaitstud tolmu vastu; 8-veekindel, pikkadeks perioodideks vee all)
IP-40	<i>Ingress Protection- 4</i> -Protected against solid objects over 1.0mm e.g. Wires. 0-No protection  Ingress Protection, kaitseaste, esimene number näitab kaitset tahketele objektidele ja teine number näitab kaitset vedelikele (4-kaitse kõvadele asjadele, mis on suuremad kui 1 mm, 0-kaitse kondents vee vastu)
JVM	Java virtual machine ehk Java virtuaal masin
ID	Identifikaator

# Sisukord

1 Sissejuhatus.....	10
2 Analüüs.....	12
2.1 Beacon ehk majakas.....	12
2.1.1 Mis on beacon ehk majakas?.....	12
2.1.2 Kuidas majakas töötab.....	13
2.1.3 Populaarsemad majaka protokollid.....	14
2.1.4 Majakate turvalisus.....	15
2.1.5 Majakad sisetingimustest.....	16
2.1.6 Majakad välitingimustes.....	16
2.1.7 Elektriliste tõukerataste navigatsioon.....	16
2.2 Eeltöö programmi kirjutamise jaoks.....	17
2.2.1 Bluetooth 4 ja 5 võrdlus.....	17
2.2.2 Turuuuring.....	18
2.2.3 Kontakt.io majakas.....	20
3 Tehniline kirjeldus.....	21
3.1 Rakenduse metoodika.....	21
3.2 Rakenduse nõuded.....	21
3.2.1 Funktsionaalsed nõuded.....	21
3.2.2 Mittefunktsionaalsed nõuded.....	22
3.3 Arendus.....	23
3.3.1 Android ja Kotlin.....	23
3.3.2 Rakendus.....	23
3.3.3 Rakenduse suhtlemine majaka ja serveriga.....	27
3.4 Rakenduse manuaalne testimine.....	30
4 Tulemuste valideerimine.....	35
4.1 Edasiarenduse võimalused.....	35
5 Kokkuvõte.....	36
Kasutatud kirjandus.....	37

## Jooniste loetelu

Joonis 1. Näide tõukerataste asukohast. ....	11
Joonis 2. Majakate asetus järjestikku kolmnurkseks arvutuseks [25] .....	13
Joonis 3. Näide majakate töötamisest. [5] .....	14
Joonis 4. Näide Lime rakendusest. [13] .....	17
Joonis 5. Kontakt.io majakas.....	20
Joonis 6. Kontakt.io mobiilirakenduse vaade.....	21
Joonis 7. Valminud rakenduse Bluetoothi küsimise vaade.....	24
Joonis 8. Valminud rakenduse asukoha jälgimise loa küsimise vaade.....	24
Joonis 9. Valminud rakenduse hüplikakna vaade.....	25
Joonis 10. Valminud rakenduse majakaga ühendamata markerite vaade.....	26
Joonis 11. Valminud rakenduse majakaga ühenduses olev vaade.....	27
Joonis 12. Desktop rakenduse vaade.....	28
Joonis 13. Desktop rakenduse tegevusdiagramm.....	29
Joonis 14. Mobiilirakenduse tegevusdiagramm.....	30
Joonis 15. Katse 1.....	31
Joonis 16. Katse 2.....	32
Joonis 17. Katse 4.....	33
Joonis 18. Katse 5.....	34

## **Tabelite loetelu**

Tabel 1. iBeacon ja Eddystone majakaprotokolli võrdlus.....	14
Tabel 2. Bluetooth 4 ja 5 võrdlus.....	18
Tabel 3. Majakate võrdlus.....	19



# 1 Sissejuhatus

Linnastumise ja tehnoloogia arenguga on järjest populaarsemaks saanud elektrilised tõukerattad. Suurlinnades kõrghoonete vahel, aga tekkib tihti olukordi, kus vajalikud raadiosignaalid ei levi piisavalt hästi [1]. Lisaks jäetakse rattad kohtadesse, mis pole inimsilmale kergesti nähtavad ja hiljem on omanikel raske rattaid kokku koguda. Ühte näidet sellisest olukorrast demonstreerib Joonis 1. Selleks, et seda probleemi lahendada, on bakalaureusetöös uuritud kuidas kasutada *Bluetooth LE* ehk *Bluetooth Low Energy* majakate tehnoloogiat koos satelliitnavigatsiooniga asukoha määramiseks. Seda selleks, et omanikud või järgmised kliendid saaks neid kergemini üles leida. *Beacon* ehk majakas on *bluetoothi* abil töötav seade, mis kogub tänapäeva ühiskonnas aina enam populaarsust. Kuigi tegemist on pikalt turul olnud seadmega on see saavutanud suurema tuntuse tänu *Bluetoothi* arengule [2]. Seda kasutatakse nii reklaamimaks poodide allahindluseid, kui ka muuseumites ja tehastes täpsema asukoha määramiseks. Siiaamaani on kasutatud majakaid enamasti sisetingimustes, kuid näiteks 2019 aasta alguses asus tuntud firma Waze majakaid paigaldama New Yorgi tunnelitesse. Eesmärgiga, et saaks jätkata asukoha määramist kohas, kuhu satelliitnavigatsiooni raadiosignaalid levivad halvasti [3]. Bakalaureusetöö eesmärgiks on luua Android rakendus, mis kasutab Google Maps satelliitnavigatsiooniks rakendusliidest ja uurida töö analüüsi käigus kas majakaid ja satelliitnavigatsiooni põhise asukohamääramist on võimalik koos kasutada elektritõukerataste leidmiseks. Seda nii viisi, et kõigepealt leitakse üles üldine piirkond satelliitnavigatsiooni abil ning kui asukoha määramise täpsus pole enam piisav, kasutatakse majaka tehnoloogiat tõukeratta leidmiseks. Töö keerukus seisneb selles, et *beacon* tehnoloogiat ei ole siiani kasutatud nii palju välitingimustes, kui on seda tehtud sisetingimustes. Lisaks on hetkel rohkem näited sellest, kuidas on majakaid kasutatud, kui neid on palju üksteisele lähedal järjestikku asetatult. See töö keskendub sellele kui seadmeid ei ole palju üksteisele lähedal.

Samuti uuriti töö tegemiseks, milliseid majakaid on turul saadaval ja mida on võimalik linnakeskkonna tingimustes kasutada. Kuidas on võimalik kuvada majakatelt saadud

informatsiooni Google Maps kaardirakendusel nõnda, et see aitaks kasutajal efektiivselt töökeratas üles leida.

Loodud rakenduse kood on üleval GitLabi salves viide on jaluses <sup>1 2</sup>



Joonis 1. Näide töökeratase asukohast.

---

1 Mobiilirakenduse repositoorium <https://gitlab.com/lauraherma/iapb>

2 Desktoprakenduse repositoorium [https://gitlab.com/lauraherma/iapb\\_backend](https://gitlab.com/lauraherma/iapb_backend)

## 2 Analüüs

Bakalaureusetöö eesmärgiks on kirjutada Android rakendus, mis kasutab satelliitnavigatsiooni põhist asukoha määramist ja majakaid, et võimalikult täpselt tuvastada, kus asub elektritõukeratas. Selleks, et võimalikult täpselt teada, millist majakat ja kuidas neid kasutada on eelnevalt uuritud, mis on majakad, milliseid formaate nad kasutavad, kuidas tagada turvalisus ja mida täpsemalt jälgida majaka valimisel. Lisaks, mida on varem tehtud majakatega sise- ja välistingimustes. Samuti, mis on *Bluetooth*, mida majakas kasutab, mis vahe on erinevatel standardi versioonidel.

### 2.1 Beacon ehk majakas

#### 2.1.1 Mis on beacon ehk majakas?

*Beacon* ehk majakas on füüsiline seade, mis kasutab *Bluetoothi* standardit perioodilise raadiosignaali saatmiseks. Signaali saatmise intervall võib varieeruda sekunditest kuni minutiteni vastavalt sellele, mis on konkreetse majaka kasutamise eesmärgiks ja kuidas see on seadistatud. Täpsemalt suhtleb majakas nutiseadmega, mis toetab *Bluetooth* standardit ja majakas saadab nutiseadmele unikaalse ID, mis on igal majakal olemas [4]. Tegemist on seadmega, mis töötab üldiselt ühesuunaliselt ehk majakas saadab infot välja, kuid ei võta andmeid vastu. Mõningaid majakaid on, võimalik konfigureerida administratiivse rakenduse abil, mis tuleb kaasa majakat ostes või programmeerides, kui on kaasas SDK. Majakas ei näe ega kuule, mis toimub ümberringi vaid pigem annab pidevalt teada infot end kohta [5]. See tagab inimestele, teadmise, et seade ei saa neid koguaeg jälgida või nende kohta infot koguda. Majaka kasutamise jaoks luuakse iOS või Android rakendus, mis kuulab majakaid ning kasutab vastavat infot, kui telefon on majaka raadiuses. Majakas võimaldab määrata asukohta hinnates kui kaugel või lähedal inimene asub seadmest *Bluetooth* standardit kasutades, mis on olemas enamuses kasutusel olevatel nutiseadmetel [6]. Majaka kaugust inimesest saab määrata erinevat moodi. Üks viis on asetada mitu majakat üksteisele lähedale. Kui mitu majakat saavad kätte kasutaja

seadme *Bluetooth* signaali on võimalik kolmnurkselt välja arvutada, kus telefon asetseb. Selleks mõõdetakse kolme seadme RSSI väärtus välja ja võrreldakse neid omavahel. Sellisel juhul majakad ei liigu ja on ühekoha peal ning on teada kui kaugel need üksteisest asetatud on [7] (Joonis 2). Teine võimalus, mida on kasutatud käesoleva bakalaaurusetöö jaoks, on üksik majakas. Majaka jaoks on seadme tootja poolt välja mõõdetud RSSI väärtus 1m kauguselt. Kui kasutaja seade ühendub majakaga, siis saadab majakas tagasi hetke RSSI väärtuse [8]. Nende mõõtmiste järgi on hinnanguliselt võimalik välja arvutada kasutaja liikumise ajal umbkaudne kaugus majakast. Majaka signaali levimiskaugus sõltub konkreetsest seadmest, kuid tavaliselt varieerub see 50m-200m. Majakad kasutavad patareisid, mis kuluvad vastavalt selle, kui tihedaks on pandud signaali edastamise sagedus ja kui kõrgeks on määratud signaali edastus tugevus. Keskmise majaka vastupidavus on 12-24 kuud enne kui patarei tuleb välja vahetada. Selleks, et majakas saaks töötada peab inimese nutitelefoni olema *Bluetooth* sisse lülitatud ja ta peab olema nõus, et rakendus jälgib inimese asukohta. Ilma selleta ei ole võimalik rakendusel ühendust saada majakaga.

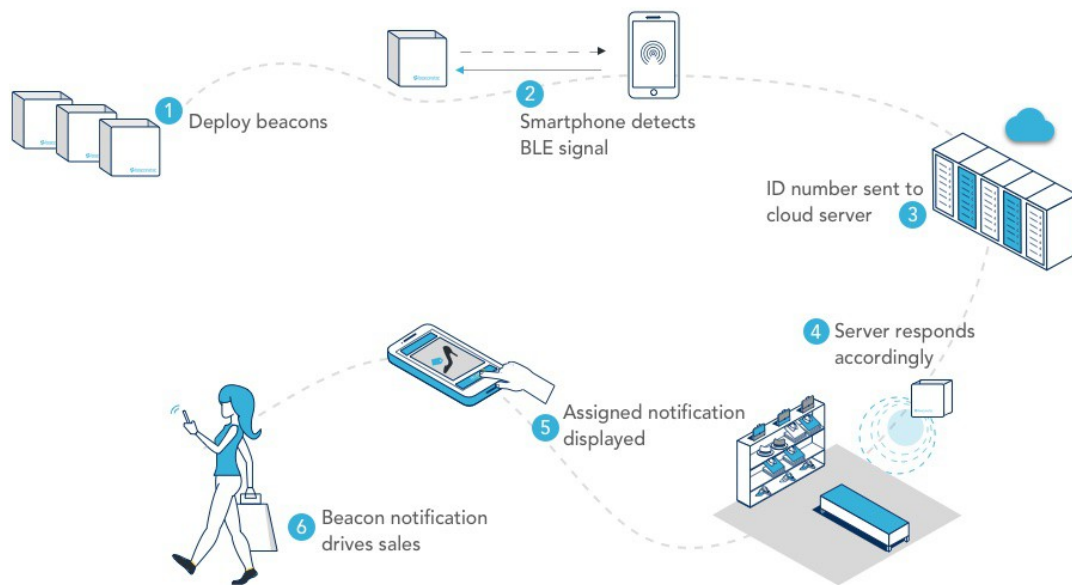


Joonis 2. Majakate asetuse järjestikku kolmnurkseks arvutuseks [7]

### 2.1.2 Kuidas majakas töötab

Kõige sagedasem näide majakate kasutamisest on kui neid kasutakse palju üksteise lähedal. Selleks kõigepealt paigaldatakse majakad üksteisele paarikümne meetri

kaugusele. Kui inimene kõnnib mööda majakast telefoniga ja tal on olemas vastav rakendus ning *Bluetooth* on sisselülitatud, siis selle peale saadetakse serverisse vajalik info. Täpsemalt saadetakse unikaalne ID ja vastavalt protokollidele majaka info. Server tagastab info vastavalt unikaalsele ID-le ja saadab kasutaja telefonile tõuketeavituse. Tõuketeavitus on teadeanne kasutajale telefoni poolt. Bakalaureusetöö käigus valminud rakendus töötab analoogselt, kuid täpsemaid erinevusi saab lugeda peatükis 3.3.3. Illustreeriv skeem on toodud majaka töötamisest Joonis 3'l



Joonis 3. Näide majakate töötamisest. [6]

### 2.1.3 Populaarsemad majaka protokollid

Majakad kasutavad erinevaid formaate. Kõige tuntumad majaka protokollid, mille abil *beaconid* töötavad on Eddystone ja iBeacon. Eddystone on välja töötatud Google poolt ja iBeacon Apple poolt, kuid mõlemad protokollid töötavad nii Androidi, kui ka iOS seadmete peal [9] [6]. Tabel 1 on toodud välja protokollide sarnasused ja erinevused.

Tabel 1. iBeacon ja Eddystone majakaprotokolli võrdlus

Eddystone	iBeacon
-----------	---------

Unikaalne ID 16 byte (Namespace ID + Instance ID)	Unikaalne ID 16 byte (Proximity UUID + Major + Minor)
Lühendatud URL	Puudub URL
Majaka staatuse info	Puuduvad staatused
EID	ID krüpteerimine puudub
Vastuvõetud signaalitugevuse indikaator (0m)	Vastuvõetud signaalitugevuse indikaator (1m)
Avatud lähte kood	Ei ole avatud lähtekoodiga
Kolme tüüpi andmeid	Ühte tüüpi andmed

Eddystone ja iBeacon'i ID formaat on sarnane, kui Eddystone kasutab väljades *namespace* ja *instance*, siis iBeacon kasutab UUID, *Major*'it ja *Minor*'it. Eddystone kasutab EID, mis krüpteerib unikaalse ID ja muudab seda perioodiliselt, selleks, et teised rakendused ei saaks majakaid kasutada. Põhiliselt on see selleks, et majakaid muuta turvalisemaks [9]. Eddystone majaka staatuse info annab näiteks teada, kui täis on patarei ja mis temperatuur on seadmel. iBeacon Major ütleb, mis asukohta majakas kuulub [6]. Näiteks kui neid on palju ühes kohas, siis on võimalik öelda, et need asuvad kõik samas hoones või ruumis. Dokumentatsioonis märgitud vastuvõetud signaalitugevuse indikaator on tavaliselt mõõdetud RSSI 1 või 0 meetri kauguselt. iBeacon'il ja Eddystone'l on see selleks, et määrata kui kaugel majakas asub, iBeacon'il määratakse seda 1m ja Eddystone'il 0m kauguselt. Eddystone on avatud lähtekoodiga, kuid iBeacon seda ei ole. Eddystone saab unikaalse ID, URL aadressi ja majaka staatuse info.

#### 2.1.4 Majakate turvalisus

Praegusel hetkel enamik majakaid ei ole väga turvalised. Majakaid on võimalik rünnata neljal erineval viisil, ründaja võib varastada ja kuulata unikaalset ID, mis on tõenäolisem, kui kasutatakse iBeacon formaati, sest Eddystone'il on unikaalne ID krüpteeritud ja krüpteerimist muudetakse pidevalt [9]. Kui turvalisusega ei ole tegeletud on võimalik lihtsalt teha *man-in-the-middle* rünnakut seadmele [10]. Lisaks on võimalik kloonida majaka konfiguratsioone. Kolmandaks enamasti ei krüpteerita andmeid, mida saadetakse majakatele. Näiteks, kui majaka kasutamiseks kasutatakse

parooli võib ründaja seda näha. Samuti on võimalus, et majakaid rünnatakse füüsiliselt. Kuna tegemist on laialt kasutava seadmega, siis on oluline, et majakaid valides oleks firma selliste probleemidega tegelema [11]. Sellepärast on käes olevas bakalaureusetöös kasutatud Kontakt.io majakaid, kes on need riskid üritanud seadmel viia miinimumini. Peatükis 2.2.3 on täpsemalt kirjeldatud, milliseid meetmeid on kasutatud.

### **2.1.5 Majakad sisetingimustest**

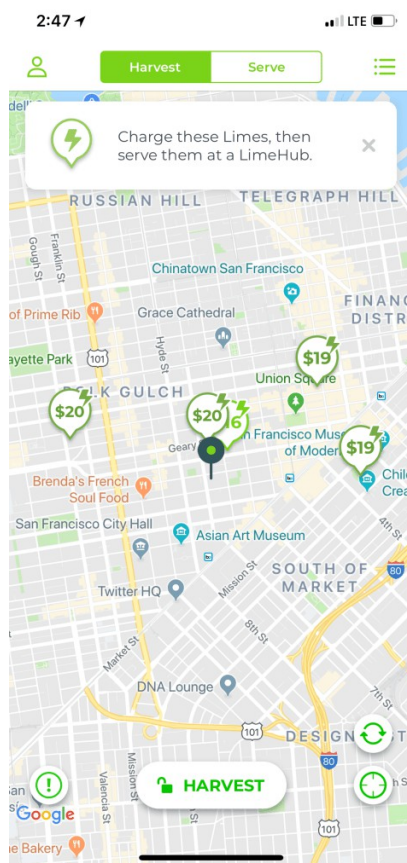
Majakad on saanud populaarseks eelkõige siseruumides, sest algselt hakati neid kasutama eesmärgina kaardistada hoone sisemust kuna satelliitnavigatsiooni asukoha määramist ei saa sellistes tingimustes kasutada. Neid on kasutusele võetud näiteks turunduses, et saata inimestele tõuketeavitusi sooduskupongide kohta. Inimene saab teavituse poest möödudes või poes sees olles, et suurendada võimalust, et klient läheduses viibides sooritaks ostu [12]. Samuti on majakad kasutusel muuseumites ja tehastes teekonna kaardistamiseks. Muuseumites kaardistatakse parim teekond ekskursiooni läbimiseks ning teoste kohta saab lugeda lisa informatsiooni näitusel ringi käies [13]. Tehastes leitakse majakate abil kõige optimaalsem tee kuidas vajaliku esemeni jõuda [14]. Samuti on kasutatud majakaid lennujaamades kohvri ülesleidmiseks. Inimesele saadeti teavitus, kui ta oli selle läheduses [15].

### **2.1.6 Majakad välitingimustes**

2019 aasta jaanuaris asus tuntud firma Waze, kes töötab navigatsiooni rakendusega paigaldama majakaid New Yorgi tunnelitesse. Majakad on paigaldatud üksteisest 40 m kaugusele moodustades võrktopoloogia ja kasutavad Eddystone protokollid. Waze pakub majakate abil navigatsiooni tunnelites, kus vajalikud raadiosignaalid ei levi hästi [3]. Soomes tegi McDonalds kampaania, kui inimene oli allalaadinud nende appi ja kui ta läks mööda nende restoranist saadeti selle peale tõuketeavitus pakkumise kohta [16]. Välitingimuste näiteid majakate kohta on võimalik leida palju vähem, sest eelkõige kasutatakse hetkel neid siseturunduse eesmärgil.

## 2.1.7 Elektriliste tõukerataste navigatsioon

Elektriliste tõukerataste leidmine võib mõningatel juhtudel keeruline olla. Mõned inimesed võtavad elektritõukerataid endale tuppa, samuti jäetakse tõukerataid varjatud kohtadesse või viiakse aiaga piiratud alale. Selleks, et seda probleemi lahendada on kõige tuntumad elektritõukerataste firmad nagu Bird ja Lime eraldi palgale võtnud inimesed, kes otsivad rattaid ja laevad neid enda kodus [17]. Mõlemad firmad maksavad rataste eest vastavalt sellele, kui lähedal on teised rattad ja kui raske neid kätte saada on [18]. Näited Lime rakendusest saab näha Joonis 5'l. Kuigi elektritõukerataste otsimine võib tunduda lihtne, kurdavad töötajad, et tõukerataste asukoht kaardil ei vasta alati täpsele asukohale [19].



Joonis 5. Näide Lime rakendusest. [18]



## 2.2 Eeltöö programmi kirjutamise jaoks

Hetkel on turul olemas palju erinevate omaduste ja hinnaga majakaid. Parima valiku tegemise jaoks on enne ostu sooritamist tehtud turuanalüüs. Kuna analüüsi käigus tuli välja, et majakad kasutavad nii *Bluetooth* 4 kui 5 on selle kohta tehtud võrdlus, et aru saada, kui oluline see valiku juures on.

### 2.2.1 *Bluetooth* 4 ja 5 võrdlus

*Bluetooth* on lühikese raadiusega juhtmevaba tehnoloogia, mis võimaldab erinevatel seadmetel omavahel suhelda. *Bluetooth* loodi peamiselt selleks, et seadmed ei vajaks enam kaableid [20]. Algselt võimaldas *Bluetooth* teha 10m raadiuse võrgu enda ümber, mis ühendas 2-7 seadet [21]. Kuna turu uuringut tehes tuli välja, et erinevad majakad kasutavad erinevat *Bluetoothi* versiooni tuli uurida, milles seisneb erinevus, selleks, et osata teha korrektne otsus majaka valimisel. Järgnevalt on Tabel 2 toodud *Bluetooth* 4 ja 5 omadused [22] [23].

Tabel 2. *Bluetooth* 4 ja 5 võrdlus

Võrdlus	<i>Bluetooth</i> 4	<i>Bluetooth</i> 5
Sagedus (GHz)	2.4	2.4
Kaugus (m)	100	200
Sõnumi suurus (byte)	31byte'i	255byte'i
Andme edastamise kiirus (Mb/s)	1	2
Seadmete arv võrgus	Piiramatu	Piiramatu

*Bluetooth* 4 ja 5 kuuluvad *Low Energy* klassi ning seetõttu ei kuluta nii palju akut ja võimaldavad palju rohkem teha kui varasemad *Bluetooth* versioonid. See on aidanud asjade interneti arengu seadmetele palju kaasa. *Bluetooth* 4 ja 5 omadused ei erine palju, kuid põhiliseks suureks vaheks on levi kaugus seadmest, andmete edastus kiirus ja sõnumi suurus. Kõik need kolm kriteeriumit on *Bluetooth* 5 paremad. Kuigi enamik seadmetel on hetkel kasutusel *Bluetooth* 4, muutub asjade interneti areng ja *beaconite* kasutus veel suuremaks, kui seadmed hakkavad kasutama *Bluetooth* 5.

## 2.2.2 Turuuuring

Antud bakalaureusetöö jaoks on tehtud turuuuring, sest müügil on palju erineva hinnaklassi ja omadustega majakaid. Algselt on võrreldud omavahel majakaid leheküljelt alibaba.com, kus on võimalik saada majakaid odava hinnaga, kuid neid tuleb osta suurtes kogustes. Sellest uuringust on valitud 2 tükki välja, et võrrelda majakatega, mis on rohkem levinud meedias. Kontakt.io, Estimote, BlueUp'i majakate kohta saab rohkem infot internetist otsides ja need tundusid töö autorile usaldusväärsemad. Tabel 3 on toodud täpsemad erinevused välja.

Tabel 3. Majakate võrdlus

Firma	Protokoll	Tarkvara arendus teek	Bluetooth verisoon	Raadiuse ulatavus	Turvalisus	Ilmastiku kindlus
Kontakt.io tough beacon [24]	iBeacon, Eddystone	Olemas	Bluetooth LE 4	70m	Turvaline	IP-68
Estimote LTE Beacon (Sissehitatud GPS) [25]	Estimote proximit, iBeacon, Eddystone	Olemas	Bluetooth LE 4	200m	Pole teada	Tolmu ja pritsme kindel, kui mitte 100% vee kindel.
BlueUp BlueBeacon Forte [26]	iBeacon, Eddystone, Quuppa	Olemas	Bluetooth LE 4	Pole teada	Krüpteeritud parool.	Väli-tingimus-tele mõeldud.
FEASYCOM [27]	EddyStone, iBeacon, AltBeacon	Olemas	Bluetooth 5.0	80m	Parool	IP40
IbeTag [28]	iBeacon, Eddystone	Olemas	Bluetooth LE 4	100m	E i o l e teada	Vee-kindel

Analüüsist on näha, et kõikidel majakatel on olemas tuntud protokollid Eddystone ja iBeacon ning tarkvara arendus teek, mida saab kasutada rakenduse ühendamise jaoks. Tuntumate majakate juures on hea, et tarkvara arendus teegiga on võimalik eelnevalt

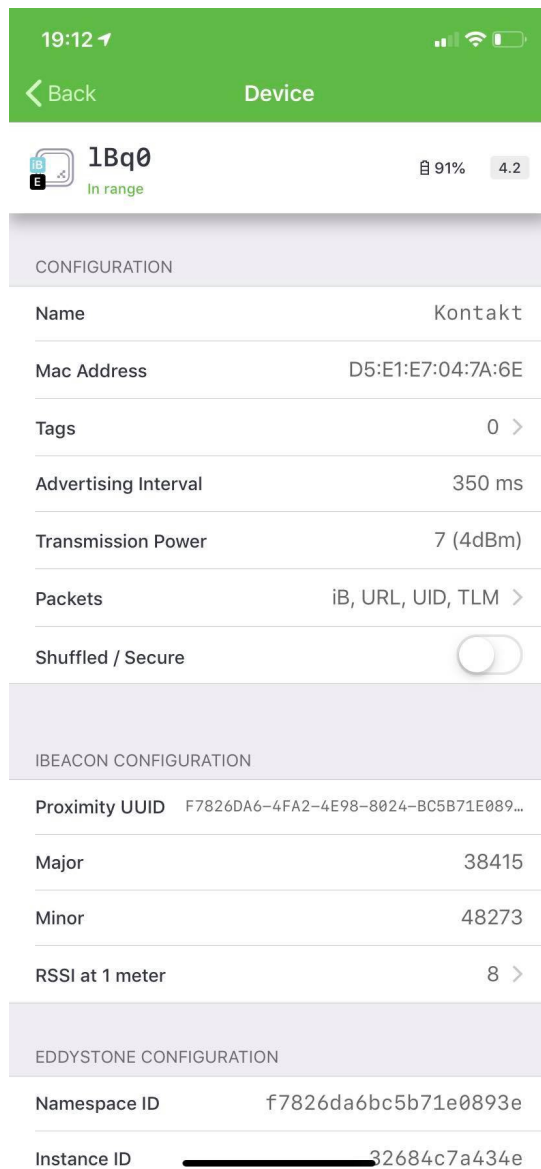
tutvuda ja on olemas veebilehe tugi, kus vajadusel infot juurde küsida. Enamik majakaid kasutavad *Bluetooth 4*, kuigi *Bluetooth 5* on parem kaalusid ülejäänud tingimused valiku tegemisel üle. Põhiliseks probleemiks on, et *Bluetooth 5* ei ole olemas veel paljudel nutitelefonidel. Kuigi *Bluetooth 5* kaugus peaks olema suurem, siis turuuringu seadmete infot vaadates oli näha, et osadel *Bluetooth 4* seadmetel reklaamiti kaugus suuremaks, kui *Bluetooth 5* omal. Samuti oli näha, et paljud firmad on tegelenud turvalisusega vähe või ei ole üldse sellele rõhku pannud. Enamik majakatest olid ilmastiku kindlad, välja arvatud FEASYCOM majakas, kuid see ei saanud valiku tegemisel määravaks. Kuna Kontakt.io majaka tehniline kirjeldus oli väga sarnane teistele majakatele, kuid tundus töö autorile oluliselt turvalisem ja veebilehekülg oli hea tugisüsteemiga otsustati selle kasuks [24] .

### **2.2.3 Kontakt.io majakas**

Töö käigus valminud Android rakenduse tegemisel kasutati Kontakt.io Tough Beacon TB15-1(Joonis 6). See majakas kasutab nii Eddystone kui ka iBeacon protokolle, lisaks on see spetsiaalselt mõeldud välitingimustele [24] . Majakatega tuleb kaasa administratiivne *desktop* ja mobiili rakendus, kust saab muuta majakate konfiguratsiooni. Kontakt.io *desktop*'i administratiivses halduskeskkonnast saab serveri rakendusliidese võtme, mida saab kasutada rakenduse programmeerimiseks. Mobiilirakendusest saab näha majaka andmete väärtusi ja muuta neid vajadusel. Lisaks tuntud protokollidele on rakendusel olemas ka enda konfiguratsioon: Mac aadress, *tagid*, signaali edastamise tihedus, edastusvõimsus (Joonis 7). Kõik info on krüpteeritud ja parooli ei saadeta läbi õhu. Samuti on olemas segamis funktsioon, mis on oluline, kui on soovitud kasutada iBeacon formaati. See tähendab, et Majorid ja Minorid on segatud iga päevaselt ning unikaalset ID ei saa ära varastada [29] . Füüsiliselt on majakat kaitstud tarkvara lukuga. Kontakt.io reklaamib ennast, kui kõige turvalisemat majakat, mis on hetkel turul [30] .



Joonis 6. Kontakt.io majakas.



Joonis 7. Kontakt.io mobiilirakenduse vaade.

## 3 Tehniline kirjeldus

### 3.1 Rakenduse metoodika

Rakenduse tegemise jaoks kasutati erinevaid tehnoloogilisi vahendeid nagu Kotlin, Android, BLE, Google Maps, PostgreSQL, Node.js ja Kontakt.io SDK. Tulemusi kontrollitakse manuaalse testimise käigus.

## 3.2 Rakenduse nõuded

Selleks, et rakendust ehitada on välja tood funktsionaalsed ja mittefunktsionaalsed nõuded. Funktsionaalsed nõuded kirjeldavad, mida süsteem peab tegema-süsteemi peamised eesmärgid ja toimingud. Mittefunktsionaalsed nõuded kirjeldavad kuidas süsteem töötab [31] .

### 3.2.1 Funktsionaalsed nõuded

Käesolevas peatükis on kirjeldatud funktsionaalsed nõuded:

- 1) Rakendus saab serverist elekritõukerataste koordinaadid.
- 2) Rakendus saab serverist elektitõukeratastega seotud majakate ID.
- 3) Rakendus annab teada, kui *Bluetooth* ei ole sisselülitatud.
- 4) Rakendus küsib luba kasutajalt asukoha jälgimise kohta.
- 5) Rakendus saab majakaga ühenduse majaka raadiusesse jõudmisel.
- 6) RSSI kaudu arvutakse välja kui kaugel on elekritõukeratas telefonist.
- 7) Rakendus näitab, kui on üles leitud satelliitnavigatsiooni raadiosignaali koordinaadid.
- 8) Rakendus näitab, kui on saadud ühendus majakaga.
- 9) Rakenduse jaoks on tehtud serveri poolne kasutajaliides.
- 10) Serveripoolse kasutajaliidesega on võimalik emuleerida elektrilise tõukeratta satelliitnavigatsiooni raadiosignaali käitumist.

### 3.2.2 Mittefunktsionaalsed nõuded

Käesolevas peatükis on kirjeldatud mittefunktsionaalsed nõuded:

- 1) Rakendus on tehtud kasutades Kotlini.
- 2) Rakendus kasutab Node.js serveri jaoks.

3) Rakendus kasutab PostgreSQL andmebaasi jaoks.

4) Rakenduse kasutajaliides on lihtsasti mõistetav.

### 3.3 Arendus

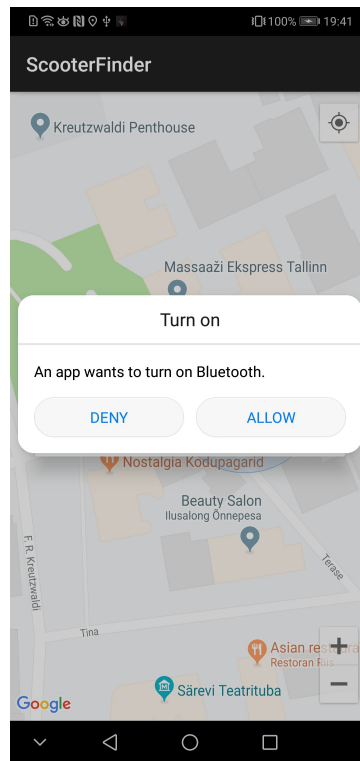
#### 3.3.1 Android ja Kotlin

Töö rakendus on kirjutatud Androidi jaoks kasutades programmeerimiskeelt Kotlin. Android on tarkvaraarendus pakett nutiseadmetele, mis hõlmab operatsioonisüsteemi, vahetarkvara ja põhilisi rakendusi. See on arendatud algselt Google ja hiljem OHA poolt [32]. Enamik Androidi rakendusi on kirjutatud programmeerimiskeeles Java, kuid 2017 aasta nimetas Android Kotlini kolmandaks ametlikuks programmeerimiskeeleks. Kotlin on ehitatud JetBrainsi loojate poolt, kes on samuti programmeerinud ka arenduskeskkonna Android Studio, kus on kirjutatud bakalaureusetöö rakendus. Kotlinis on omavahel kombineeritud funktsionaalne ja objektorienteeritud programmeerimine ja see töötab JVM peal [33]. Hetkel soovib Google kasutada arendamise jaoks Kotlini ning paljud rakendused, mis eelnevalt on kasutanud Javat on ümber kirjutatud Kotlini peale [34]. Javat ja Kotlinit on võimalik omavahel vaheldumisi kirjutada, mis võib tekitada segadust koodi kirjutamisel, kuid on samas mugav, kui soovida Java koodist minna ümber Kotlini peale. Kotlin aitab näiteks vähendada *null pointer exceptioneid*, mis tähendab, et tõenäosus sooritada tegevus mingi väärtusega mida ei ole olemas väheneb. Samuti on olemas *data* klassid, mis automaatselt genereerivad *boilerplate* ehk koodi, mis on mõeldud, et rakendus töötaks, aga see ei täida rakenduse tegevuse eesmärki. Kotlini programmeerimiskoodi on võimalik kirjutada palju lühemalt ja selgemalt kui Java oma [35].

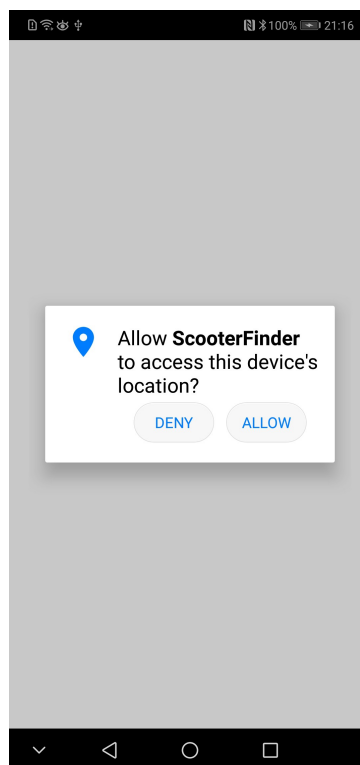
#### 3.3.2 Rakendus

Rakendus on arendatud Androidis kasutades Kotlinit, mis kasutab Google Mapsi ja Kontakt.io SDK. Lisaks tehakse päringuid kasutades Node.js. Google Mapsi kasutamise jaoks oli vaja eraldi luua rakendusliidese võti, selle abil saab ligi Google Mapsi serveritele, ning see kuvab visuaalselt kaardi. Google Mapsi kasutamise jaoks on kasutajalt asukoha määramise jaoks eelnevalt vaja luba küsida (Joonis 8), ilma selleta ei

ole rakendusel luba kasutada nutitelefoni asukohamääramise võimekust, ehk kasutaja ei saaks jätkata rakenduse kasutamist. Lisaks peab kasutajal olema *Bluetooth* sisselülitatud, et majakaga ühendust saada (Joonis 9).



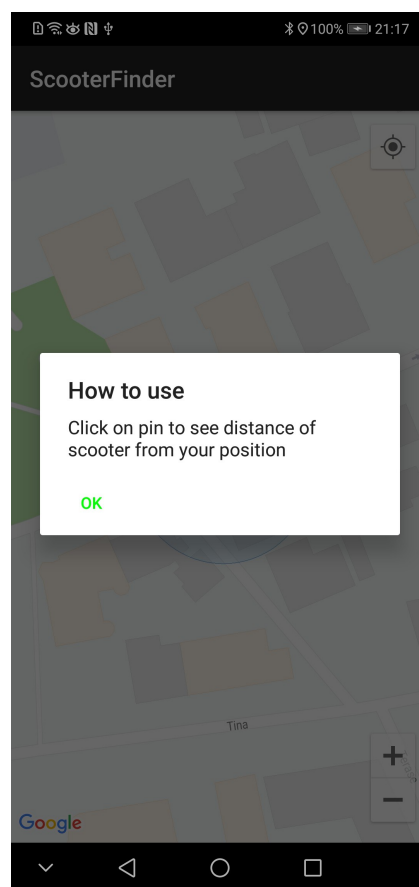
Joonis 8. Valminud rakenduse Bluetoothi küsimise vaade.



Joonis 9. Valminud rakenduse asukoha jälgimise loa küsimise vaade.



Rakenduse esmakordsel käivitamisel kuvatakse kasutajale kõigepealt Androidi *dialog*, mis on väike hüplikaken (Joonis 10), mille abil on kasutajale esitatud lisainfo, kuidas rakendust kasutada, sest markerile klikkimine ei pruugi rakendust kasutades olla intuiitiivne. Enamasti loodavad inimesed, et satelliitnavigatsioon näitab neile täpse asukoha ja ei arva, et peaks eraldi kuskile vajutama. Lisaks kaalus autor oma töös Android *toast* teavituse kasutamist, mis on kasutajale info edastamine teavitusena, kuid katsetuste käigus selgus, et see ei äratanud piisavalt tähelepanu kasutajas, et teada, mida teha edasi.



Joonis 10. Valminud rakenduse hüplikakna vaade.

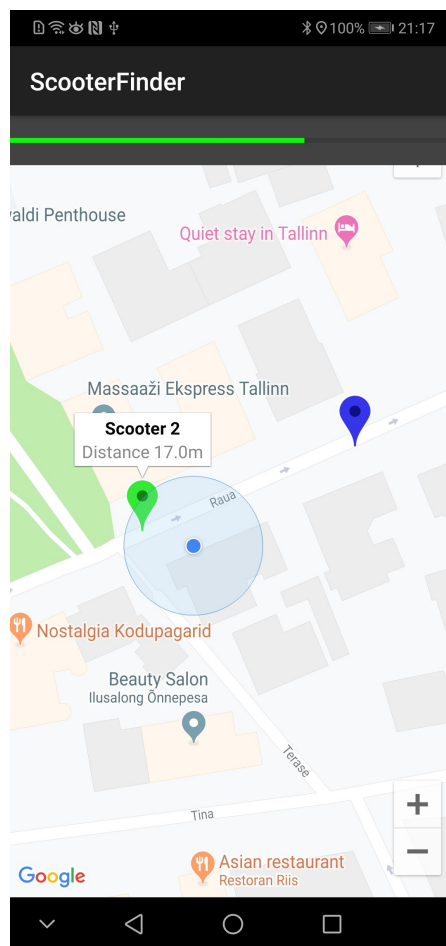
Google Mapi kaardile on arendajal võimalik lisada graafilisi elemente vastavalt oma soovile. Markerid on tavaline viis, kuidas näidata objektide asukohta, värvid on muudetud, et anda parimal viisil edasi tegevusi. Kui rakendus ei ole elektritõukerataste majakatega ühendust saanud on marker sinine ja peale ühenduse saamist muutub

rohelisteks. Kui vajutada, mitte ühendust saanud ratta markerile on kuvatud ainult elektritõukeratta nimi. (Joonis 11)



Joonis 11. Valminud rakenduse majakaga ühendamata markerite vaade.

Kui vajutada majakaga ühendust saanud elektri tõukerattale saab näha, kui kaugel ollakse rattast meetrites. Samuti tekib seejärel rakenduse ülaossa visuaalne indikaator, mis näitab kui kaugel rattast ollakse. Indikaatori jaoks on kasutatud *progress bar*'i, mis on tavaline viis, kuidas anda kasutajale tagasisidet, kui palju on midagi vaadatud või palju on midagi tehtud. Indikaatori maksimum on 70 meetrit, mis on lubatud suurus Kontakt.io poolt Tough Beacon TB15-1 majakale [24] . (Joonis 12)

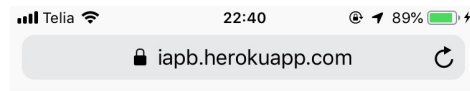


Joonis 12. Valminud rakenduse majakaga ühenduses olev vaade

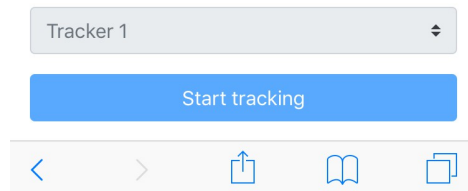
### 3.3.3 Rakenduse suhtlemine majaka ja serveriga

Android rakenduse jaoks serveriga suhtlemiseks on tehtud lihtne veebilehekülg, mis on üleväl leheküljel <https://iapb.herokuapp.com/track>. Joonis 13 Lehekülg emuleerib ühe elektrilisetõukeratta käitumist, kui selle asukohta jälgitakse rakenduse abil. Veebilehele minnes saab valida ratta, mille asukohta soovitakse jälgida. Lihtsustuse eesmärgina on ratas ja majakas käsitsi eelnevalt ühendatud, veebileheküljel seda eraldi teha ei saa. Pärast jälgimise alustamist salvestatakse iga 3 sekundi tagant PostgreSQL andmebaasi koordinaadid. Andmebaasist saab kätte ratta viimase registreeritud asukoha. Samamoodi on ka elektritõukeratastel, mingi teatud aja tagant salvestakse asukoht ja kui kasutaja lõpetab sõitmise jäetakse viimane asukoht meelde. PostgreSQL on valitud sellepärast, et tegemist avatud lähtekoodiga tarkvaraga, hea dokumentatsioon ja jõudlusega [36]. Jõudlus on oluline, sest ratta koordinaatide saatmise intervall on tihe ja

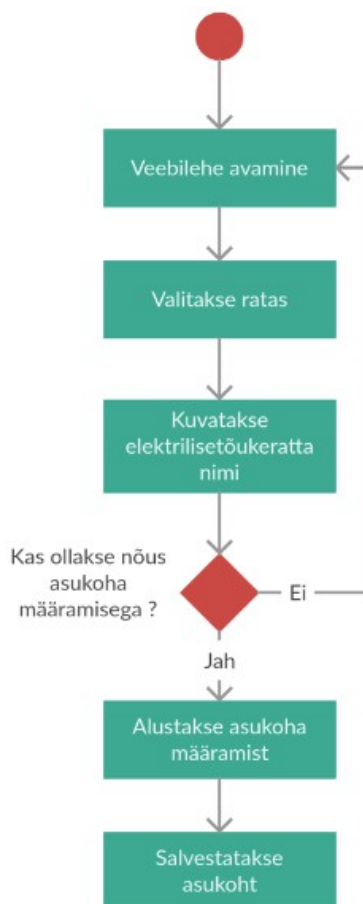
kui rattaid oleks palju peab andmebaas olemas suuteline koormusega hakkama saada. Hetkel on rakenduses ainult kolm elektritõukeratast, kuid rataste hulga suurendamisel mängiks see olulist rolli. Andmebaasist võetakse välja andmed ja tagastatakse JSON formaadis, mis on saadaval aadressil <https://iapb.herokuapp.com/trackers>. XML formaati ei tagastata, sest selleks ei näinud autor põhjust. (Joonis 14)



59.43672257671569  
24.770154487203538



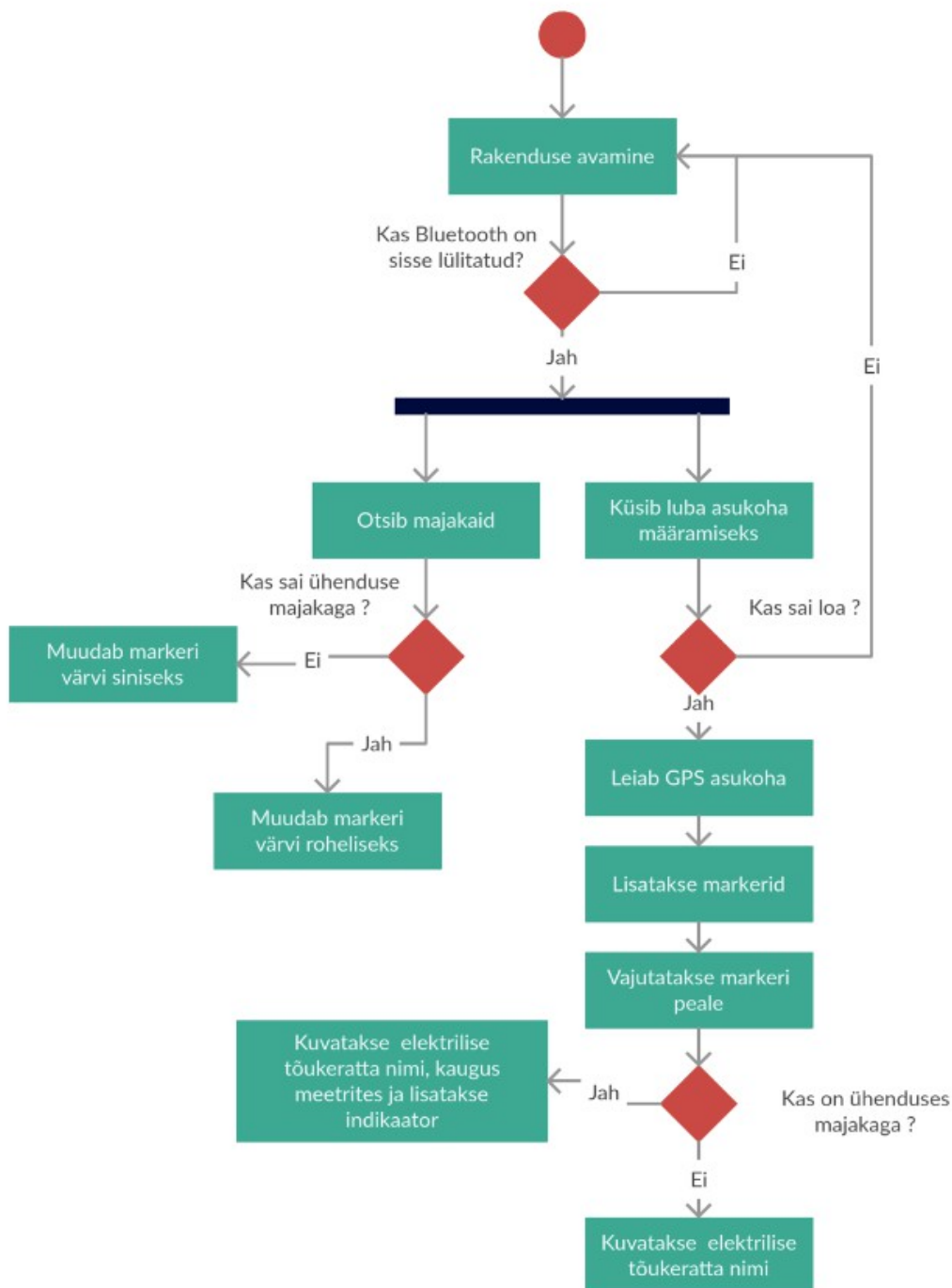
Joonis 13. *Desktop* rakenduse vaade



Joonis 14. Desktop rakenduse tegevusdiagramm

Rakenduse majakaga suhtlemise jaoks jagab majakas oma infot iga 350ms tagant, seda saab muuta vaid nutitelefoniga administratiivse rakendusega. Mis on algne määratud väärtus Kontakt.io poolt. Seadme värskendus intervall oli algselt pandud 3000ms, kuid testimise käigus otsustas autor, et 1000ms tagant saab täpsema info ja RSSI hüppamisel tekkiv meetrite ebatäpsus ei esine nii tihti. Markereid uuendatakse iga 2000ms tagant. Telefonile saadetakse infot on võimalik välja lugeda hetke RSSI väärtus, mille abil on võimalik arvutada majaka kaugus. Arvutuste tegemisel kasutasin AltBeaconi algoritmi, kes on välja töötanud, kuidas seda meetrites arvutada. Samuti on kasutatud Kontakt.io mõõtmisi meetrite kaupa. Kahe firma arvutused on omavahel kombineeritud, et saada täpsem tulemus. Arenduses käigus katsetati algoritmi nii iBeaconi kui ka Eddystone protokolliga mõõtetud RSSI väärtusega. Täpsemaks osutus iBeacon protokolliga määratud mõõtetud RSSI tulemus. Meetrite näitamine rakenduses

ei pruugi alati täpne olla, sest RSSI on mõjutatud välisest keskkonnast. Erinevatelt pindadelt peegeldus võib sama kohapeal anda erineva RSSI väärtuse [37] . (Joonis 15)



Joonis 15. Mobiilirakenduse tegevusdiagramm

### 3.4 Rakenduse manuaalne testimine

Rakenduse manuaalsel testimisel osales kaks inimest, üks oli kasutaja ja teine elektrilise tõukeratta rollis hoides ühes käes majakat ja teises käes nudiseadet iPhone 6s operatsioonisüsteemi versiooniga iOS 12.2. iOS telefonis käivitati veebilehekül, mis emuleeris elektrilise tõukeratta tegevust. Android rakendust testiti HUAWEI P20 lite telefoni peale Android versiooniga 8.0.0 ehk Android Oreo, mis on 2018 aastal üks populaarsemaid Androidi versioone [38]. Kokku tehti 5 katset. Katse visualiseerimiseks on hiljem käsitsi graafiliselt juurde lisatud punane X, mis tähistab ratta tegelikku asukohta. Samuti on lisatud tumesinise tausta ja musta äärejoonega ring, mis tähistab, kus kasutaja seisis. See on lisatud sellepärast, et telefon ei tuvastanud vahepeal kasutaja tegelikku asukohta õigesti. Google Mapi poolt lisatud, kasutaja asukoht on näha helesinise valge joonega ringina. Sammu pikkuseks on võetud umbes 0.8 meetrit.

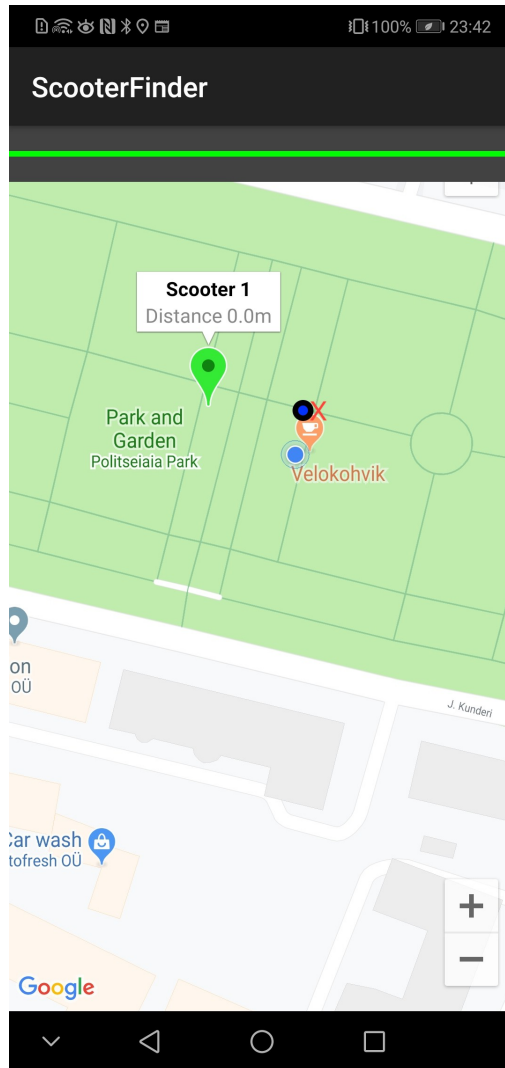
1. Katse eesmärgiks oli tuvastada, kui elektriline tõukeratas viia maja taha, kui täpselt seda oleks võimalik tuvastada. Joonis 16 on näha, et asukohamääramine satelliitnavigatsiooni abil on üsna täpne, kuid kaardilt võib jääda mulje nagu ratas oleks tuppa viidud. Samas on rakendusest näha, et majaka kauguse kalkulatsioon üsna täpne. Kalkulatsiooni võrdluseks mõõtis autor sammudes kui kaugel on teine inimene saades tulemuseks 31 sammu.



Joonis 16. Katse 1

2. Katse eesmärgiks oli näidata, et asukohamääramise satelliitnavigatsiooni signaal levib pargis halvasti. Tulemusest on näha, et majakas sai signaali kätte ja kuvab õigesti, et asub otse tõukeratta kõrval. Samal ajal kui marker asub täiesti teises kohas. Lisaks on jooniselt näha, et lisaks markerile näitab ka Google Maps kasutaja asupaika vales kohas. (Joonis 17)





Joonis 17. Katse 2

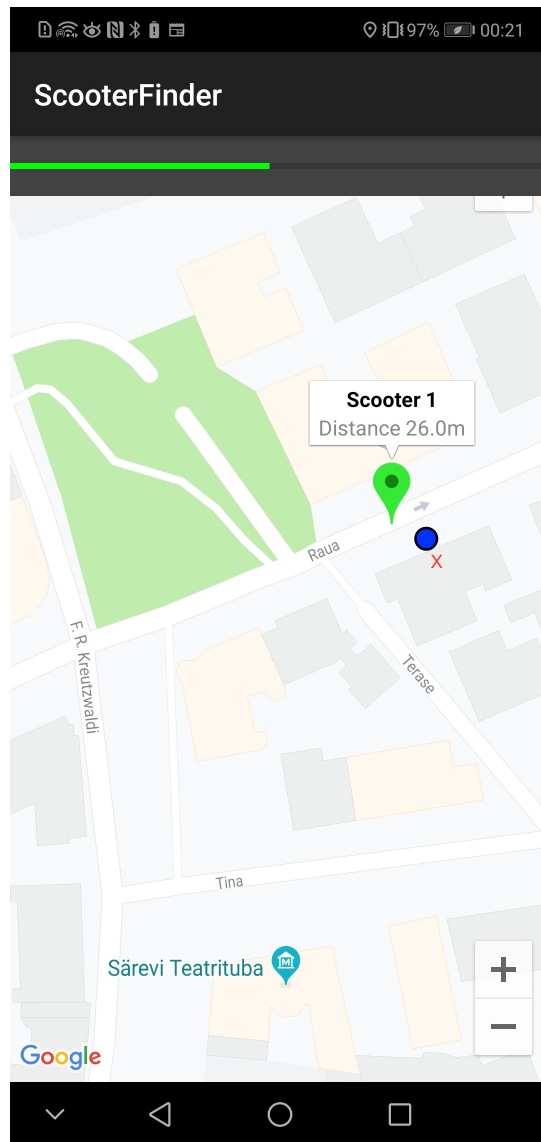
3. Katse eesmärgiks oli näidata, kui pikalt on võimalik kätte saada maksimum signaal, kui kasutaja ja elektrilise tõukeratta vahel ei ole takistusi. Kahjuks osutus pika distantsi katse probleemiks majaka RSSI ebatäpsus. Kuigi majakas hoidis kontakti pikalt, sammudes 76 sammu ehk umbes 60m, siis maksimum tulemus, mis saadi mõõtmisel oli 42m. Pikema distantsi peal hakkas signaal juba palju hüplema ja andis meetrites eba täpseid tulemusi. Sellel katsel oli ka satelliitnavigatsiooni signaal üsna täpne, sest pikal vahemaal ei olnud puid nii palju ümber.
4. Katse eesmärgiks oli testida rakendust tallinnas kõrgemate hoonete vahel Tornimäel. Teatavasti levib Tallinnas asukohamääramise satelliitnavigatsiooni

signaal üsna hästi. Soov oli näha, kas signaal kõigub samamoodi nagu pargis või on täpne. Katse käigus selgus, et tallinnas kõrghoonete vahel on asukohamääramine satelliitnavigatsiooni signaal üsna täpne. Väike satelliitnavigatsiooni viga on näidatud Joonis 18'1.



Joonis 18. Katse 4

5. Katse eesmärgiks on tuvastada, kas majakas töötab ka siis kui inimene on väljas ja elektri tõukeratas asuks maja sees. Kaardilt on näha, et asukohamääramine satelliitnavigatsiooni abil on üsna täpne. Tulemuse käigus selgus, et rakendus saab ühendust majakaga ja näitab kauguseks 26m (Joonis 19), mis on toas, kui kasutaja on samal ajal õues. Katse kaugus meetrites, aga ei olnud väga täpne. Majakas asus teisel korrusel ja eeldatav kõrgus oleks võinud olla umbes 15m.



Joonis 19. Katse 5

## **4 Rakenduse testimise tulemuste järeldused**

Bakalaureusetöö eesmärgiks oli teha Android rakendus, mis töötaks nii kasutades satelliitnavigatsiooni kui ka majakat elektrilise tõukeratta leidmiseks ja kuvaks võimalikult täpselt kaugel on inimene rattast. Rakenduse valideerimiseks tehti viis katset, millest saab järeldada, et tulemus õnnestus osaliselt. Kaardile lisatakse marker vastavalt satelliitnavigatsiooni signaalile ja majakaga saadakse välistingimustes ühendust. Majaka kauguse tuvastus on üsna täpne kui ollakse elektrilisele tõukerattale lähedal, kuid kui minna kaugemale kui 30m, siis RSSI erinevus suureneb ja meetreid ei kuvata enam nii täpselt.

### **4.1 Edasiarenduse võimalused**

Bakalaureusetöö jaoks valmis esialgne rakenduse prototüüp, kuidas majakad ja satelliitnavigatsiooni signaali määramine võiks koos töötada, kui on kasutatud üksikut majakat. Edasiarenduseks on autoril mõned mõtted, mida võiks tulevikus kasutada. Esiteks võiks proovida, kui lisada juurde kolmnurkne meetod kui rattaid on palju, kas oleks võimalik üksiku majaka kalkuleerimist ja kolmnurkset arvutust koos kasutades võimalik saada täpsemad tulemused. Selle all mõtleb autor, et kui katsete käigus kaugel oleva ratta mõõtmine ei tulnud täpne, kas niiviisi oleks meetrite arvutus võimalik saada täpsemaks. Seda muidugi juhul, kui kolmnurkne arvutus on võimalik, kui majakad on üksteisest erineval kaugusel ja ei seisa ühel kohal. Samuti oleks hea teha katseid erinevate firmade majakatega, sest antud töös kasutatud majaka RSSI hüppab skaalal palju kuna on mõjutatud väliskeskkonnast. Hea oleks näha kas pikema distantisi peal on erinevatel majakatel teistsugused tulemused. Lisaks hetkel on majaka ID ratastega pandud käsitsi koodi, tulevikus võiks olla administratiivne süsteem, kuidas rattaid ja majakaid omavahel siduda. Samuti võiks proovida välja arvutada viga, mis tekib kauguse näitamise käigus.

## 5 Kokkuvõte

Käesoleva töö eesmärgiks oli luua Android rakendus, mis leiaks üles elektritõukerattad kasutades satelliitnavigatsiooni signaali ja majakaid. Töö saavutamiseks analüüsi, milliseid majakaid on turul olemas, mis formaate ja *Bluetooth* versioone seadmed kasutavad. Samuti uuriti, mida peab silmas pidama turvalisuse osas majakaid valides.

Töö tulemusena valmis Android rakenduse prototüüp, mille abil on võimalik tuvastada elektri tõukerattaste asukohta. Lisaks tehti lihtne desktop rakendus, mis emuleerib elektri tõukeratta tegevust. Desktop rakendus saadab serverile koordinaate ratta asukohast ja mobiilirakendus asetab selle põhjal markeri kaardile. Kui mobiilirakendus ei ole ühendust saanud majakaga on marker sinine ja kui on saanud ühendus, siis muutub marker roheliseks. Lisaks kuvatakse visuaalselt kaugel majakast ollakse.

Rakenduse valideerimiseks tehti viis katset, mis näitavad, et kui rakendus ei ole väga kaugel majakast annab see üsna täpseid tulemusi. Probleemid tekivad, siis kui minnakse rattast kaugemale kui 30m ja RSSI väärtust ei ole enam nii täpselt võimalik kätte saada.

Kokkuvõtteks majakad teevad elektritõukerattaste tuvastamise lihtsamaks, kuid selleks, et saada väga täpset tuvastust meetritest oleks vaja välja mõelda, kuidas saaks meetrite tuvastust täpsemaks viia.

## Kasutatud kirjandus

- [1] Miura, S., Hsu, L.T., Chen, F. and Kamijo, S., 2015. GPS error correction with pseudorange evaluation using three-dimensional maps. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), pp.3104-3115.
- [2] Jeon, K.E., She, J., Soonsawad, P. and Ng, P.C., 2018. Ble beacons for internet of things applications: Survey, challenges, and opportunities. *IEEE Internet of Things Journal*, 5(2), pp.811-828.
- [3] <https://gamer4k.com/waze-adds-beacons-for-drivers-not-to-lose-gps-signal-in-tunnels/> [vaadatud 22.03.2019]
- [4] <https://kontakt.io/beacon-basics/what-is-a-beacon/> [vaadatud 19.04.2019]
- [5] <https://academy.pulsatehq.com/bluetooth-beacons/> [vaadatud 16.04.2019]
- [6] <https://www.beaconstac.com/ibeacon-and-eddystone> [vaadatud 16.04.2019]
- [7] <https://proximi.io/accurate-indoor-positioning-bluetooth-beacons/> [vaadatud 16.04.2019]
- [8] <https://iotandelectronics.wordpress.com/2016/10/07/how-to-calculate-distance-from-the-rssi-value-of-the-ble-beacon/> [vaadatud 16.04.2019]
- [9] <https://developers.google.com/beacons/eddystone> [vaadatud 18.05.2019]
- [10] <https://www.beaconzone.co.uk/blog/man-in-the-middle-attacks-on-beacons/> [vaadatud 18.05.2019]
- [11] <https://kontakt.io/blog/why-you-should-use-the-kontakt-io-proximity-sdk-for-ios-or-android-with-kontakt-io-secure/> [vaadatud 12.04.2019]
- [12] [https://medium.com/@the\\_manifest/a-beginners-guide-to-beacon-marketing-in-2018-15ac361d4226](https://medium.com/@the_manifest/a-beginners-guide-to-beacon-marketing-in-2018-15ac361d4226) [vaadatud 12.04.2019]
- [13] <https://blog.beaconstac.com/2014/06/how-museums-can-use-beacons-to-enhance-visitor/> [vaadatud 12.04.2019]
- [14] <https://www.beaconzone.co.uk/blog/bluetooth-beacons-in-factories-iot-and-industry-4-0/> [vaadatud 13.05.2019]
- [15] <https://medium.com/@urish/traveling-with-beacons-checked-luggage-made-easy-bbd664765ea3> [vaadatud 13.05.2019]
- [16] <https://blog.beaconstac.com/2016/08/7-ways-beacons-are-disrupting-the-ooh-advertising-space/> [vaadatud 18.05.2019]
- [17] <https://www.ridester.com/lime-charger/> [vaadatud 13.05.2019]
- [18] <https://cityformillennials.com/electric-scooter-share-lime-s/> [vaadatud 13.05.2019]

- [19] <https://www.theatlantic.com/technology/archive/2018/05/charging-electric-scooters-is-a-cutthroat-business/560747/> [vaadatud 17.04.2019]
- [20] <https://www.lifewire.com/what-is-bluetooth-2377412> [vaadatud 17.04.2019]
- [21] <https://medium.com/iotforall/6-things-you-didnt-know-about-bluetooth-3ebb50ba370e> [vaadatud 17.04.2019]
- [22] <https://ieeexplore.ieee.org/abstract/document/8419192> [vaadatud 05.05.2019]
- [23] <https://ieeexplore.ieee.org/abstract/document/7224764> [vaadatud 05.05.2019]
- [24] <https://store.kontakt.io/our-products/28-tough-beacon-tb15-1.html> [vaadatud 25.04.2019]
- [25] <https://community.estimote.com/hc/en-us/articles/204092986-Technical-specification-of-Estimote-Beacons-and-Stickers> [vaadatud 04.05.2019]
- [26] <https://www.blueupbeacons.com/index.php?page=forte> [vaadatud 04.05.2019]
- [27] [https://www.alibaba.com/product-detail/Factory-OEM-ultra-thin-bluetooth-ibeacon\\_60785909351.html?spm=a2700.7724857.normalList.80.28fd2636ZuSArI](https://www.alibaba.com/product-detail/Factory-OEM-ultra-thin-bluetooth-ibeacon_60785909351.html?spm=a2700.7724857.normalList.80.28fd2636ZuSArI) [vaadatud 04.05.2019]
- [28] [https://www.alibaba.com/product-detail/Bluetooth-Accelerometer-Ble-Beacon-With-Temperature\\_60484143894.html?spm=a2700.details.pronpeci14.9.4ff244c0yX6jkx](https://www.alibaba.com/product-detail/Bluetooth-Accelerometer-Ble-Beacon-With-Temperature_60484143894.html?spm=a2700.details.pronpeci14.9.4ff244c0yX6jkx) [vaadatud 16.05.2019]
- [29] <https://kontakt.io/blog/beacon-security/> [vaadatud 16.05.2019]
- [30] <https://kontakt.io/blog/why-you-should-use-the-kontakt-io-proximity-sdk-for-ios-or-android-with-kontakt-io-secure/> [vaadatud 16.05.2019]
- [31] <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/> [vaadatud 14.05.2019]
- [32] <https://www.lifewire.com/what-is-google-android-1616887> [vaadatud 14.05.2019]
- [33] <https://www.jetbrains.com/research/kotlin-census-2018/> [vaadatud 14.05.2019]
- [34] <https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/> [vaadatud 14.05.2019]
- [35] <https://dzone.com/articles/why-kotlin-is-the-future-of-android-app-developmen> [vaadatud 11.05.2019]
- [36] <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems> [vaadatud 11.05.2019]
- [37] <https://www.bluetooth.com/blog/proximity-and-rssi/> [vaadatud 11.05.2019]
- [38] <https://fossbytes.com/most-popular-android-versions-always-updated/> [vaadatud 10.05.2019]