

Janne Varjo

# GPS-PAIKANTIMEN INTEGROINTI PILVIPALVELUUN

Tietojenkäsittelyn koulutusohjelma

2019

## GPS-PAIKANTIMEN INTEGROINTI PILVIPALVELUUN

Varjo, Janne  
Satakunnan ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Kesäkuu 2019  
Sivumäärä: 30  
Liitteitä: 0

Asiasanat: satelliittipaikannus, seuraus, integraatio, esineiden internet, IoT

---

Tämän opinnäytteen aiheena oli integroida GPS-paikannin pilvipalveluun Bluugo Oy:n toimeksiantamana. GPS-paikantimien integrointi oli osa Bluugon Tracking Cloud™ -ohjelmistoalustan tuotekehitystä. Opinnäytetyötä varten valitut GPS-paikantimet otettiin testikäyttöön ja GPS-paikantimia varten kehitettiin ohjelmistorajapintoja Bluugon Tracking Cloud™ -ohjelmistoalustaan. Lisäksi GPS-paikantimia varten toteutetun ohjelmistokehityksen keskeisiä ratkaisuja ja osa-alueita tutkittiin ja kuvattiin yleisellä tasolla.

Opinnäytteen teoriaosuus keskittyi opinnäytteen toteutusosuuden käyttämien tekniikoiden ja kokonaisuuksien tutkintaan ja kuvaamiseen. Opinnäytteen teoriaosuuden luvuissa pyrittiin kertomaan kyseisen teoriaosuuden luvun käsittelemän asian lyhyt historia ja keskeiset ominaisuudet. Lisäksi teoriaosuuden luvuissa korostettiin tälle opinnäytetyölle keskeisiä ja tärkeitä asioita.

Opinnäytteen toteutusosuus oli ottaa käyttöön kaksi GPS-paikanninta ja kehittää Bluugon Tracking Cloud™ -ohjelmistoalustaan tuki kyseisiä GPS-paikantimia varten. Toteutusosuus sisältää myös varsinaisen ohjelmistokehityksen aikana syntyneitä toimintamalleja ja ratkaisuja. Lisäksi opinnäytteen toteutusosuudessa esiteltiin GPS-paikantimilla saatuja tuloksia.

Opinnäytteen loppuosuudessa käytiin läpi opittuja asioita ja ohjelmistokehityksen ja GPS-paikantimien testauksen aikana ilmenneitä ongelmia ja niiden ratkaisuja. Lisäksi loppuosuuteen sisällytettiin Bluugon näkökanta tämän opinnäytteen tuloksista ja niiden hyödyntämisestä tulevaisuudessa.

Opinnäytteen lopputuloksena ja tavoitteena oli kehittää Bluugon Tracking Cloud™ ohjelmistoalustaan alustava tuki GPS-paikantimia varten. Ratkaisukokonaisuus antaa hyvin suuntaa sille, miten esineiden Internet -tyyppisiä laitteita voidaan jatkossa integroida ja hyödyntää Bluugon asiakasprojekteissa. Tämän opinnäytteen ohella Bluugon Tracking Cloud™ -ohjelmistoalusta myös kehittyi huomattavasti kartta- ja geotietoalustojen osalta, jotka antavat hyvät perustyökälyt logistiikkakeskeisissä ohjelmistokehitysprojekteissa.

# INTEGRATION OF A GPS-LOCATOR TO A CLOUD SERVICE

Varjo, Janne

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in information technology

June 2019

Number of pages: 30

Appendices: 0

Keywords: satellite navigation, tracking, integration, internet of things, IoT

---

The subject of this thesis was to integrate a GPS locator to a cloud service as an assignment from Bluugo Oy. The integration of the GPS locators was a part of product development of Bluugo's Tracking Cloud™ software platform. The GPS locators picked for this thesis were initially deployed to test use. The testing was followed by implementing an application programming interface for the GPS locators on Bluugo's Tracking Cloud™ software platform. Additionally, key solutions and parts of the software development process were described and inspected on a general level.

The theory portion of this thesis focused on describing and investigating the technologies and concepts used by the implementation portion of this thesis. The sections in the theory portion of this thesis were aimed to contain a short history and essential properties of the corresponding subject. In addition, the theory portion of this thesis contains highlights of important topics in the perspective of this thesis.

The implementation portion of this thesis consisted of deploying two GPS locators and developing the Tracking Cloud™ software platform to support the said GPS locators. The implementation portion also includes the operating models and solutions developed during the software development. Lastly the end results created with the GPS locators were showcased.

The closing portion of this thesis was used to walk through all the things learned during the thesis and the issues that were discovered while testing the GPS locators. In addition, the closing portion of this thesis was extended with Bluugo's standpoint on this thesis' results and the utilization of the said results in the future.

The result and goal of this thesis was to develop an initial support for GPS locators on Bluugo's Tracking Cloud™ software platform. The solution gives good direction towards how "internet of things" devices can be integrated and utilized in Bluugo's customer projects. Alongside this thesis the map and geofencing features were further developed on Bluugo's Tracking Cloud™ software platform. The developed map and geofencing tools provide good basic tools for logistics-centric software development projects.

# SISÄLLYS

## SANASTO JA LYHENTEET

1	JOHDANTO .....	6
2	GEOAITAUKSEN TEKNINEN YMPÄRISTÖ .....	7
2.1	PostgreSQL.....	7
2.2	Django.....	9
3	KÄYTETYT TEKNIIKAT .....	13
3.1	GPS .....	13
3.2	Geoaitaus .....	14
3.3	M2M-dataliittymät.....	15
4	LAITTEET .....	16
4.1	GL505 .....	16
4.2	GL300A .....	17
4.3	Air Interface Protocol .....	18
5	TOTEUTUS .....	20
5.1	Laitteiden käyttöönotto .....	20
5.2	Tracking Cloudin kehitys .....	22
5.3	Testaus .....	24
6	POHDINTA JA TULEVAISUUS .....	26
6.1	Pohdinta .....	26
6.2	Tulevaisuus .....	28
	LÄHTEET.....	30
	LIITTEET	

## SANASTO JA LYHENTEET

3G, 4G, 5G	Lyhenne, joka kuvaa matkapuhelinteknologian sukupolvea.
GNSS	Global Navigation Satellite System, suomeksi maailmanlaajuinen satelliittipaikannusjärjestelmä.
GPS	Global Positioning System, suomeksi maailmanlaajuinen paikannusjärjestelmä.
GPRS	General Packet Radio Service, suomeksi yleinen radiopakettijärjestelmä.
GSM	Global System for Mobile Communications, suomeksi maailmanlaajuinen mobiilikommunikointijärjestelmä.
IoT	Internet of Things, suomeksi esineiden Internet.
IP-osoite	Internet Protocol, suomeksi Internet-protokolla.
IP-luokitus	International Protection Marking, suomeksi kansainvälinen tiiveysmerkintä.
JSON	JavaScript Object Notation, suomeksi JavaScriptin objektien merkin-tätapa.
M2M	Machine to Machine, suomeksi koneesta koneeseen.
mAh	Milliampere hour, suomeksi Milliampeeritunti.
MVC	Model-View-Controller, suomeksi malli-näkymä-ohjain.
NB-IoT	Narrowband IoT, suomeksi kapean kaistanleveyden IoT.
ORM	Object-relational Mapping, suomeksi objektien relaatiokartoitus.
SIM	Subscriber Identity Mobile, suomeksi asiakkaan tunistemuoduli.
SQL	Structured Query Language, suomeksi rakenteellinen kyselykieli.
TCP	Transmission Control Protocol, suomeksi (tiedon)siirron ohjauspro-tokolla.
URL	Uniform Resource Locator, suomeksi yhtenäinen resurssin paikan-nin.
UTC	Coordinated Universal Time, suomeksi koordinoitu maailmanlaajui-nen aika.
XML	Extensible Markup Language, suomeksi laajennettavissa oleva mer-kintäkieli.

## 1 JOHDANTO

Geoaitauksen käyttö kasvaa jatkuvasti. Varsinkin ympäristöissä, joissa GPS-signaalin saamiselle ei ole esteitä, on helppo alkaa seuraamaan esimerkiksi logistiikkaa. Vaikka geoaitaus yleensä perustuu GPS-paikantimen tuottaman sijaintitiedon hyödyntämiseen, ei seurattavan kohteen sijaintia kuitenkaan yleensä seurata suoraan. Pääpaino on kirjauksissa, joita luodaan seurattavan kohteen osuessa johonkin geoaitaan tai geoaidan ulkopuolelle. Geoaitauksella luotujen tapahtumien avulla voidaan esimerkiksi laskea keskimääräisiä aikoja, joita käytetään jonkin matkan kulkemiseen esimerkiksi ajoneuvolla. Geoaitauksen edut ilmenevät, kun perinteinen käsin kirjaaminen tai mikä tahansa ajoneuvon kuljettajan manuaalisesti tekemä toimenpide korvataan automaattisella geoaitauksella.

Geoaitoja voidaan käyttää myös hälytyksessä ja omaisuuden vartioinnissa. Esimerkiksi moottoripyörään kiinnitetty GPS-paikannin voisi hälyttää moottoripyörän omistajaa, kun moottoripyörä lähtee pois joltain ennalta määritellyltä alueelta. Tällöin myös perinteisempää tärinään tai liikkeeseen perustuvaa varashälytintä ei tarvitsisi kytkeä pois päältä esimerkiksi ilta-aikaan, jos kyseistä moottoripyörää pitää siirtää.

Geoaidoilla muodostettavat virtuaaliset portit ja alueet luovat rajattomat mahdollisuudet asioiden seurannalle ja mittaukselle. GPS:n tarkkuutta voidaan tulevaisuudessa hyödyntää vielä enemmän tehokkaammilla ja paremmilla vastaanottimilla. Kun GPS:n tarkkuus ei riitä, voidaan käyttää jotain paikallista paikannusjärjestelmää. Hyvä esimerkki tästä on Tom Scotin 29.4.2019 julkaisema YouTube-video ”The Hundred-Tonne Robots That Help Keep New Zealand Running” (suomeksi Sadan tonnin painoiset robotit, jotka pitävät Uuden Seelannin käynnissä), jossa kuvataan, miten eräässä Uudessa Seelannin Aucklandin sataman konttiterminalissa käytetään paikallista paikannusjärjestelmää automaattisten kontinsiirtimien paikantamiseen. Paikannuksen ansiosta konttien sijainti ei perustu enää asfalttiin maalattuihin merkintöihin vaan täysin virtuaaliseen karttaan. Perinteiset asfalttiin maalatut merkinnät myös aiheuttavat jatkuvaa kulumista lähes samaan paikkaan osuvien konttien kulmien takia. Virtuaalisen kartan ansiosta konttien paikkoja voidaan siirtää jatkuvasti pieniä määriä kerrallaan, joka johtaa asfaltin tasaisempaan kulumiseen. Samalla myös konttien

kuljetusajoneuvojen reitit muuttuvat ja niiden renkaiden aiheuttama kulutus asfaltin pintaan vähenee konttipaikkojen tapaan. (The Hundred-Tonne Robots That Help Keep New Zealand Running, 2019.)

Geoaitauksella ja paikantamisella on siis monta ilmenemismuotoa. Parhaimmillaan geoaitaus tarjoaa suuria parannuksia edeltävään ratkaisuun verrattuna. Geoaitauksen tuomat hyödyt ilmenevät varsinkin järjestelmän laajetessa. Esimerkiksi uutta konttipaikkaa lisätessä ei tarvitse maalata asfalttiin uusia merkintöjä.

Tämän opinnäytetyön toimeksiantaja on Bluugo Oy. Bluugo on kasvussa oleva pienyritys, ja Bluugolla on toimipisteet Vantaalla ja Porissa. Bluugo auttaa asiakkaitaan liiketoiminnan digitalisoinnissa, joka perustuu Bluugon räätälöityyn Tracking Cloud -alustaan. Bluugon asiakkaita ovat muun muassa Europress, Fortum ja Swissport. Tämän opinnäytetyön luvuissa 4.1 ja 4.2 kuvatut GPS-paikantimet ja niiden integrointi ovat osa Bluugon tulevan tuoteversion (”core 3.0”) ydintoiminnallisuutta.

## 2 GEOAITAUKSEN TEKNINEN YMPÄRISTÖ

Tässä opinnäytetyössä toteutettu tekninen ympäristö perustuu virtuaalipalvelimessa suoritettavaan PostgreSQL-tietokantaan sekä Python-pohjaiseen Django-ohjelmistokehitysalustaan. PostgreSQL-tietokantaa käytetään kaiken tiedon tallentamiseen, mukaan lukien geoaidat. Django-ohjelmistokehitysalustaa käytetään varsinaisen sovelluksen toteutukseen ja GPS-paikantimien lähettämien sanomien vastaanottamiseen ja käsittelyyn.

### 2.1 PostgreSQL

PostgreSQL on avoimen lähdekoodin olio-relaatiotietokantapalvelin, joka käyttää ja laajentaa SQL-kieltä. PostgreSQL:lle on saatavilla lisäosia kuten PostGIS-niminen geospaatialinen tietokantalaajennos. PostgreSQL:n kehityksen voidaan katsoa alkaneen jo vuonna 1986 osana POSTGRES-projektia Kalifornian Yliopistossa,

Berkeleyssä. PostgreSQL:n ydinosuutta on kehitetty yli kolmekymmentä vuotta. (PostgreSQL:n verkkosivut, 2019.)

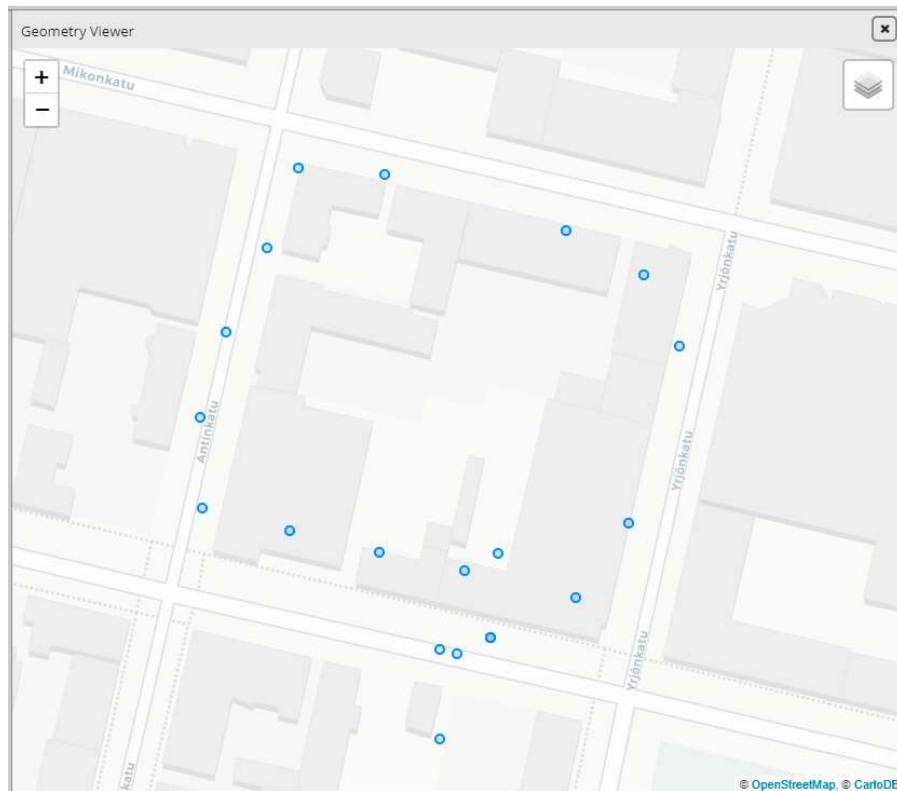
Ehkä selkein ero PostgreSQL:n ja perinteisempien SQL-palvelimien, kuten MySQL:n tai MariaDB:n välillä on PostgreSQL:n laajempi tuki erilaisille tietotyypeille, kuten JSON, XML ja geometriset muodot (esimerkiksi ympyrä, nelikulmio, viiva, piste ja monikulmio). Tämän opinnäytetyön kannalta PostgreSQL:n tärkein ominaisuus edellä mainittu PostGIS-tietokantalaajennos ja geometriset saraketyypit, joiden avulla geospaatialisia SQL-kyselyitä voidaan käyttää hyväksi geoaitamisessa. Kuvassa 1 on esitetty SQL-kysely, jossa haetaan rivit, joiden geoaita pitää sisällään tietyn pisteen. Geoaitaus tapahtuu funktiolla `ST_Contains`, jota käytetään kahdella parametrilla. Molemmat parametrit ovat tyypiltään geometrisiä muotoja. Kuvassa 1 esitettyssä esimerkki SQL-kyselyssä ensimmäinen `ST_Contains`-funktiolle annettu parametri (kuva 1 rivi 4) on viittaus taulun `geo_fence` sarakkeeseen `data`, joka sisältää geoaitojen tiedot. Toinen `ST_Contains`-funktiolle annettu parametri on tekstisyötteestä luotu geometrinen muoto (kuva 1 rivi 5) käyttäen koordinaatistojärjestelmää 4326 (muun muassa GPS käyttää koordinaattijärjestelmää 4326). Kuvassa 1 esitetty SQL-kysely palauttaa kaikki rivit taulusta ”`geo_fence`”, joiden sarakkeen ”`data`” geometrian sisäpuolella on piste, jonka latitudi on 61,4823598 ja longitudi 21,7944978.

```
1  SELECT *
2  FROM geo_fence
3  WHERE ST_Contains(
4  geo_fence.data,
5  ST_GeomFromText ('POINT(21.7944978 61.4823598 )', 4326)
6  )
```

Kuva 1. SQL-kysely, jossa suoritetaan geoaitaus.



PostgreSQL:n mukana tulee myös PGAdmin-ylläpitosovellus, jonka avulla koko tietokantapalvelinta voi hallita. Geoaitauksen kannalta tärkeä ominaisuus PGAdminissa on mahdollisuus visualisoida geografisia sarakkeita karttanäkymän avulla, josta esimerkki kuvassa 2.



Kuva 2. PGAdminin karttanäkymä, jossa visualisoituna GPS-sijainteja.

## 2.2 Django

Django on Python-pohjainen web-ohjelmistokehitysalusta. Django on ensimmäinen versio kehitettiin vuosien 2003-2004 aikana sanomalehtien verkkosivuista vastaavan web-kehitystiimin toimesta. Kasvavan verkkosivumäärän takia lähdekoodin ja suunnittelumallien uudelleenkäyttö kasvoi. Yleisestä jaetusta lähdekoodista muodostui geneerinen web-ohjelmistokehitysalusta. Kyseinen geneerinen web-ohjelmistokehitysalusta julkaistiin Django-nimisenä avoimen lähdekoodin projektina kesäkuussa 2005. (Django introduction, MDN web docs:in verkkosivut, 2019.)

Django perustuu löyhästi MVC-arkkitehtuuriin. MVC:n malliosuudesta Django vastaa Django oma ORM tukien lukuisia eri tietokantapalvelimia kuten SQLite,

MySQL ja PostgreSQL. Djangoon malleja kuvataan luokilla, josta esimerkki kuvan 3 riveillä 7 – 11. Mallien tietokantasarakkeet kuvataan luokkainstansseilla, kuten esimerkiksi CharField (kuva 3 rivit 8 – 11). Malleja kysellään tietokannasta managerien avulla. Kaikilla malleilla on oletusmanageri ”objects” (kuva 3 rivi 14). Kuvassa 3 riveillä 15 – 18 on kuvattu suodatustoiminto, jossa käyttäjän liittymispäivämäärän on oltava päiväyksien 1.1.2019 ja 31.12.2019 välillä. Laajempia SQL-kieleen perustuvia toimintoja kuten tietokantafunktiot ja muut ominaisuudet, kuten BETWEEN-operaattori, kuvataan Djangoissa sarakekäsittelijöillä. Djangoon sarakekäsittelijöitä käytetään ”\_”-erottimen avulla sarakkeen nimen jälkeen. Sarakekäsittelijät ovat usein nimetty niiden käyttämän tietokantafunktion mukaisesti, kuten kuvassa 3 rivillä 15 esiintyvä date-sarakekäsittelijä. Sarakekäsittelijät kääntyvät SQL:ksi riippuen käytetystä tietokantapalvelimesta.

Esimerkiksi MySQL:ssä Djangoon date-sarakekäsittelijä kääntyy SQL:n DATE-tietokantafunktioksi. Vastaavasti kuvassa 3 rivillä 15 esiintyvä range-sarakekäsittelijä kääntyy MySQL:ssä BETWEEN-operaattoriksi. (Django ohjesivusto, 2019).

```

1  from datetime import date
2  from django.db import models
3  from django.utils.translation import gettext as _
4  from django.contrib.auth.models import AbstractUser
5
6
7  class User(AbstractUser):
8      customer_number = models.CharField(
9          max_length=128,
10         verbose_name=_('Customer number')
11     )
12
13
14     User.objects.filter(
15         date_joined__date__range=(
16             date(2019, 1, 1),
17             date(2019, 12, 31)
18         )
19     )

```

Kuva 3. Esimerkki mallin määrittämisessä ja mallien kyselystä tietokannasta.

MVC:n näkymäosuudesta Djangoissa vastaa sen funktio- ja luokkapohjaiset näkymät. Djangoon luokkapohjaiset näkymät käyttävät yleensä jotain asettelumallia HTML-sivun näyttämiseen. Kuvassa 4 on esimerkki luokkapohjaisesta näkymässä, jossa

määritetään yksittäisen käyttäjän profiilin näkymä. Käyttäjän profiilin näkymän määrittäminen koostuu mallista (kuva 4 rivi 7), vaaditusta käyttöoikeudesta (kuva 4 rivi 8) sekä asettelumallista (kuva 4 Rivi 9).

```

1  from django.contrib.auth.models import User
2  from django.views.generic import DetailView
3  from django.contrib.auth.mixins import PermissionRequiredMixin
4
5
6  class UserProfileView(PermissionRequiredMixin, DetailView):
7      model = User
8      permission_required = 'auth.view_user'
9      template_name = 'user/profile.html'

```

Kuva 4. Esimerkki Django luokkapohjaisesta näkymästä.

Djangon asettelumallien merkintäsyntaksi HTML:n ohella on oletuksena Jinja2, josta esimerkki kuvassa 5. Jinja2:ssa asettelumallifunktioita käytetään muun muassa toistuvien rakenteiden muodostamiseen esimerkiksi for-silmukalla (Kuva 5 rivi 2). Toinen käyttökohta Jinja2:n asettelumallifunktiolle on yhden arvon palautus, kuten esimerkiksi URL-polun rakennus (Kuva 5 rivi 3).

```

1  <ul>
2  {% for user in users %}
3      <li>
4          <a href="{% url 'user:user-profile' user.username %}" >
5              {{ user.username }}
6          </a>
7      </li>
8  {% endfor %}
9  </ul>

```

Kuva 5: Esimerkki Jinja-syntaksin käytöstä HTML-koodissa.

Djangossa ei ole varsinaista MVC-arkkitehtuurin mukaista käsittelijää tai ohjainta, vaan näkymät ovat joko luokkia tai funktioita, jotka vastaavat jotakin URL-polun osaa. MVC-arkkitehtuurin mukaisen käsittelijän roolista vastaa URL-polkujen reitittäjä, jossa jotakin polkua vastaa jokin näkymä. URL-polut sisältävät usein parametreja ja dynaamisia osuuksia. Kuvassa 6 on esitetty URL-polku, jossa dynaamista osaa kuvataan merkinnällä `<str:username>` (Kuva 6 rivi 7), jossa ”str” tarkoittaa, että parametri on tekstityyppinen ja ”username” parametrin nimeä.

```
1  from django.urls import path
2  from . import views
3
4
5  urlpatterns = [
6      path(
7          'user/<str:username>/profile',
8          views.UserProfileView.as_view(),
9          name='user-profile'
10     )
11 ]
```

Kuva 6. Esimerkki näkymän yhdistämisestä URL-polkuun.

Djangon tuotekehitystä tukee Django Software Foundation (suomeksi Django ohjelmistosäätiö, myöhemmin **DSF**). DSF on voittoa tavoittelematon itsenäinen yhdistys ja DSF:n tavoitteina on muun muassa tukea Django kehitystä ja suojella Django:n immateriaalioikeuksia. DSF:ään kuuluu yritysjäseniä, jotka rahoittavat DSF:ää suoraan. DSF:n yritysjäsenet on jaettu eri tasoihin, ja esimerkiksi platinatasoon kuuluu JetBrains ja Instagram. (Djangon verkkosivut, 2019.)

JetBrains tuottaa ohjelmistokehitystyökaluja lukuisille ohjelmointikielille, mukaan lukien Python. JetBrainsin Pythonia varten luotu ohjelmistokehitystyökalu on PyCharm. PyCharm tukee Django-pohjaista webiohjelmistokehitystä suoraan PyCharmiin sisäänrakennetuilla aputoiminnoilla. PyCharm tukee monien Django perustoimintojen automaattitäydennystä, kuten esimerkiksi asettelumallifunktiot, tietokantamalliin perustuvien lomakkeiden kenttien nimet ja näkymän asettelumallin tiedostopolun. JetBrainsin PyCharmista on saatavilla ilmainen ja maksullinen versio. Maksullinen versio PyCharmista sisältää laajemman tuen Django:lle. Esimerkki maksullisen PyCharmin laajemmasta Django-tuesta on asettelumallien diagnostiikkatyökalut, jotka sallivat breakpointin (suomeksi pysäytyspiste) asettamisen asettelumalliin.

## 3 KÄYTETYT TEKNIIKAT

### 3.1 GPS

Geoaitaamisen mahdollistaa GPS-tekniikka. GPS:n avulla saadaan hyvissä olosuhteissa mitattua kohteen sijainti alle viiden metrin tarkkuudella, mikäli käytössä on tyyppinen älypuhelin (GPS Accuracy, 2019). Koska GPS perustuu maata kiertäviin satelliitteihin, on oleellista, että GPS-paikantimella on esteetön näköyhteys taivaalle. Tämän takia sisätiloissa, kuten esimerkiksi varastorakennuksissa tehtävä geoaitaus vaatii yleensä jonkun muun paikannusjärjestelmän.

GPS on alun perin Yhdysvaltain luoma satelliittipaikannusjärjestelmä. Yhdysvaltain laivasto teetätti 1960-luvulla kokeiluja, joissa ydinohjuksia kuljettavia sukellusveneidän sijaintia seurattiin kuudella maapallon napoja kiertävillä satelliiteilla. Sukellusveneidän paikannus perustui satelliittien lähettämän radiosignaalin ilmenevään Doppler-ilmiöön. Sukellusveneen sijainti pystyttiin määrittämään muutamassa minuutissa. (Mai, 2017.)

1970-luvulla Yhdysvaltain puolustusministeriö halusi varmistaa vankan satelliittinavigointijärjestelmän saatavuuden. Tulevan järjestelmän ensimmäinen satelliitti laukaisiin maata kiertävälle radalle vuonna 1978. 24:stä satelliitista koostuva järjestelmä valmistui vuonna 1993.

GPS koostuu kahdesta eritasoisesta palvelusta:

- Standard Positioning Service (SPS) - Normaali paikannusjärjestelmä
- Precise Positioning System (PPS) - Tarkka paikannusjärjestelmä

Normaali paikannusjärjestelmä on maailmanlaajuisesti kaikkien vapaassa käytössä. Tarkka paikannusjärjestelmä taas on rajoitettu Yhdysvaltain viranomaisten ja Yhdysvaltain sotilasliittolaisten käyttöön. (Mai, 2017.)

Nykyään GPS:ää on vaikea välttää – se löytyy lähes jokaisesta älypuhelimesta ja autosta.

## 3.2 Geoaitaus

Geoaitaus on yksinkertaisimmillaan jatkuvaa etäisyyden mittaamista jonkin seurattavan kohteen ja ennalta määritellyn sijainnin välillä. Seurattavan kohteen sijaintia seurataan yleensä GPS:n avulla. Geoaidan ja seurattavan kohteen muodostamaa kokonaisuutta kuvataan yleensä erinäisin karttanäkymin. Kun seurattava kohde on tarpeeksi lähellä ennalta määritettyä sijaintia kuten esimerkiksi 25 metriä, suoritetaan jokin haluttu toimenpide. Pelkkään etäisyyteen perustuva geoaitaus on yleensä kuvattu ympyränä, jossa ympyrän säde on geoaidan tunnistamisetäisyys. Tämä on geoaitauksen yksinkertaisin muoto ja se on yleensä sisäänrakennettu GPS-paikantimiin sen helpon tietoraketeen ansiosta. Esimerkiksi luvussa 4.1 esitelty Queclinkin laite GL505 tukee viittä sisäistä geoaitaa.

Ympyrää monimutkaisempi ja oikeaan maailmaan helpommin muotoutuva geoaidan muoto on monikulmio, joka luonnollisesti sisältää myös nelikulmiot. Kuvassa 2 on havainnollistettu monikulmiosta koostuvaa geoaitaa, joka ”aitaa” sisälleen osan korttelin sisällä olevasta pysäköintialueesta. Tummempi punainen viiva kuvaa geoaidan varsinaista ”aitaosuutta” ja vaalempi punainen alue kuvaa geoaidan sisälle sulkeutuvaa aluetta.

Kuvassa 7 esitetty geoaita siis luotaisiin jollain piirtotyökalulla, jossa geoaitoja piirretään tarkan kartta-aineiston avulla. Tällaiset piirtotyökalut, kuten esimerkiksi Mapboxin GL JS -kirjastolle luotu lisäosa Mapbox Draw, käyttävät jotain koordinaattijärjestelmää. Hyvä esimerkki koordinaattijärjestelmän käytöstä karttatyökalussa on Google Maps. Kun käyttäjä liikuttaa karttaa Google Maps -palvelussa, päivittyy näkyvillä olevan kartan keskipiste verkkosivun osoitteeseen käyttäen latitudi- ja longitudikoordinaatteja.

Kun haluttu geoaita on piirretty ja sen geometria on kuvattu jollain koordinaattijärjestelmällä, voidaan sitä käyttää geoaitaamiseen. Kuvan 7 mukaisen monikulmion geoaitaaminen ratkaistaan Point in Polygon (suomeksi piste monikulmiossa) -nimisen ongelman avulla. Luvussa 2.1 kuvattu PostgreSQL-tietokantapalvelimen PostGIS-tietokantalaajennos tukee tietokantafunktiota ST\_Contains, jossa tarkastetaan sisältääkö jokin geometrinen muoto toisen geometrisen muodon. Teknisesti tämä siis sallisi

tarkastuksen, jossa tutkitaan, onko jokin geoaita toisen geoaidan sisällä. Tässä opinäytetyössä ST\_Contains-tietokantafunktiota käytetään epäsuorasti GPS-sijaintien geoaitaamiseen.



Kuva 7. Geoaita visualisoituna karttapohjan päällä.

Geoaidat ovat siis erinomainen tapa lisätä virtuaalisia portteja, jotka seuraavat liikennöintiä. Niiden käyttö ei myöskään yleensä vaadi loppukäyttäjän kouluttamista, sillä karttapohjan päällä tapahtuva piirtäminen ja työskentely on varsin intuitiivista.

### 3.3 M2M-dataliittymät

GPS-paikantimien tietoja voidaan hyödyntää vasta, kun kerättyihin tietoihin päästään käsiksi. Yleensä tietoja siis lähetetään jonnekin sen sijaan, että ne noudettaisiin laitteesta fyysisesti. Tätä voidaan verrata hyvin nykyaikaisiin etäluettaviin sähkönkulutusmittareihin. Tietojen lähettämistä ja GPS-paikantimen etäohjausta varten tarvitaan siis jokin väylä, jota pitkin tietoa kuljetetaan. Tämän opinäytetyön kohdelaitteet (luvut 4.1 ja 4.2) käyttävät SIM-korttimoduulin avulla GSM- ja GPRS -tiedonsiirtopalveluita.

Koska GPS-laitteissa on SIM-korttimoduuli, voidaan tiedonsiirtoon käyttää normaalia matkapuhelinliittymää, jossa mobiilitiedonsiirto on sallittu. Tällöin tosin matkapuhelinliittymien perinteiset ominaisuudet kuten puhelut tai tekstiviestit olisivat turhaan päällä, tai niiden päällä olo nostaisi liittymän kuukausihintaa. Tämän takia IoT-laitteita varten on saatavilla erillisiä räätälöityjä mobiililiittymiä, joita kutsutaan M2M-liittymiksi.

IoT-laitteiden tiedonsiirron kanssa ilmenee usein myös tarve ohjata IoT-laitetta etänä. Perinteisessä kotitalouksien kiinteissä laajakaistaliittymässä on yleensä julkinen IP-osoite, joka siis mahdollistaa yhteyden muodostamisen laajakaistaliittymää käyttävään laitteeseen. 3G- ja 4G-liittymissä julkinen IP-osoite ei kuitenkaan ole yleensä oletuksena päällä ja pitää tilata operaattorilta erikseen. Varsin yleinen esimerkki julkisen IP:n vaativasta IoT-laitteesta on web-kamera. Tällöin web-kameran käyttöliittymään päästään käsiksi esimerkiksi verkkoselaimella, ja web-kameran lähettämää kuvaa voidaan katsella. Samaan tapaan IoT-laitteisiin voidaan luoda suora yhteys jollain Internetin protokollalla.

## 4 LAITTEET

### 4.1 GL505

GL505 on Queclinkin tarjoama vesitiivis GPS-paikannin. Se on tarkoitettu kiinteästi asennettavaksi neljän kiinnitysreiän avulla, jotka näkyvät kuvassa 8. GL505 käyttää sisäistä akkua, jonka käyttöikä yhdellä latauksella on 1800 päivää (melkein viisi vuotta), kun sijaintieto lähetetään kerran päivässä. GL505:n akun kapasiteetti on 1500 mAh.

GL505 on kahdesta kohdelaitteesta kevyempi malli, jonka tekniset ominaisuudet ovat kuvattu taulukossa 1. GL505 kestää IP-luokituksensa mukaisesti hetkellisen upotuksen veteen ja näinollen soveltuu ulkokäyttöön. Pienen kotelonsa ansiosta se ei myöskään herätä erityisemmin huomiota ja on vaikea havaita. GL505 soveltuu siis muun muassa kiinteän omaisuuden valvontaan. (Queclink GL505:n verkkosivut, 2019.)



GL505 voidaan käytön aikana vaihtaa tarpeen mukaan joko jatkuvaan raportointi- tai virransäästötilaan. Raportointitilassa laite lähettää tiedot GPS-sijainnistaan korkeimmillaan kerran minuutissa. Virransäästötilassa laite käynnistyy tietyn aikavälin ja ajoituksen mukaisesti.



Kuva 8. GL505 kuvattuna yläpuolelta. IMEI-koodi sensuroitu turvallisuussyistä.

IP-luokitus	IPX7 (vesitiivis, ks. liite 1: IP-luokitus)
Akun kapasiteetti	1500 mAh
Paino	140g (akun kanssa)
Koko (pituus x leveys x korkeus)	118mm x 69,5mm x 26,8mm
Toimintalämpötila	Alin: -20°C, ylin: +60°C

Taulukko 1. GL505:n ominaisuudet.

#### 4.2 GL300A

GL300A on Queclinkin tarjoama kannettava GPS-paikannin, joka on suunniteltu logistiikka- ja kuljetustoimintaa varten. Laite sisältää kaksi valoanturia sekä yhden sisäisen lämpötila- ja kosteusanturin. GL300A:ssa on kymmenen kertaa suurempi akku kuin GL505:ssa: 15000 mAh. (Queclink 300A:n verkkosivut, 2019.)

GL300A on kohdelaitteista tehokkaampi ja sen tekniset ominaisuudet on kuvattu taulukossa 2. GL300A on GL505:n tavoin vesitiivis. GL505:n tavoin GL300A tukee erilaisia toimintatiloja, kuten virransäästötilaa ja jatkuvan lähetyksen tilaa. Kuvassa 9 on kuvattu GL300A edestä ja sivulta.



Kuva 9. GL300A kuvattuna päältä ja sivulta.

IP-luokitus	IP65 (vesitiivis, ks. liite 1: IP-luokitus)
Akun kapasiteetti	15000 mAh
Paino	375g (akun kanssa)
Koko (pituus x leveys x korkeus)	74mm x 34,5mm x 151mm
Toimintalämpötila	Alin: -20°C, ylin: +60°C

Taulukko 2. GL300A:n ominaisuudet.

#### 4.3 Air Interface Protocol

Queclinkin laitteet käyttävät tekstipohjaista viestiprotokollaa, jonka nimi on Air Interface Protocol (suomeksi ilmarajapintaprotokolla, myöhemmin **AIP**). AIP:n sanomat kulkevat joko tekstiviesteinä tai Internetiä pitkin TCP:n kautta. AIP:n viestit alkavat

tunnisteella ja viestin varsinainen sisältö on pilkuin eroteltuja arvoja. Viesti päättyy lopetusmerkkiin, joka on \$ (dollarin merkki).

Kuvassa 10 on esitetty kokonaisuudessaan yksi GL300A:n lähettämästä sanomasta.

Sanoma koostuu seuraavista osista:

- **+RESP:** Raporttityyppinen sanoma
- **GTFRI:** Ajastettu sijainnin raportti, joka samalla määrittää sanoman parametrien merkityksen
- **490301:** Protokollan versio
- **860\*\*\*\*\*:** Laitteen IMEI
- **gl300a:** Laitteen nimi
- **0:** Raportin tunniste/lisäosan maski
- **0:** Raportin tyyppi (0, 1, tai 2)
- **1:** Sijaintietojen määrä
- **1:** GNSS-tarkkuus
- **0.0:** Nopeus
- **166:** Atsimuutti (suunnan horisontaalinen komponentti)
- **37.1:** Korkeus merenpinnasta
- **21.795203:** Longitudi
- **61.481928:** Latitudi
- **20190412141943:** GNSS UTC-aikaleima
- **0244:** MCC (Mobile Country Code) - Mobiiliyhteyden maakoodi. 244 on Suomen mobiilimaakoodi.
- **0005:** MNC (Mobile Network Code) – Mobiiliverkon koodi
- **0898:** LAC (Location Area Code) – Sijainnin aluekoodi
- **29FF:** CID (Cell ID) – Laitteen tunniste alueella (LAC)
- **<tyhjä>:** Kuljetun matkan etäisyys (tyhjä mikäli toiminto ei ole käytössä)
- **100:** Akun varausprosentti
- **20190412141945:** Lähetysaika
- **273A:** Viestien määrän laskuri

```
+RESP:GTFRI,490301,860*****,gl300a,0,0,1,1,0.0,166,  
37.1,21.795203,61.481928,20190412141943,0244,0005,0898,  
29FF,,100,20190412141945,273A$
```

Kuva 10. Esimerkki GL300A:n +RESP:GTFRI -sanomasta.

## 5 TOTEUTUS

### 5.1 Laitteiden käyttöönotto

Ensimmäinen toimenpide laitteiden kanssa oli asentaa niihin M2M-mobiililiittymien SIM-kortit. Molemmat laitteet käyttävät normaalikokoista SIM-korttia, jonka ansiosta isosta luottokorttiaihioista irrotettua SIM-korttia ei tarvitse leikata enää pienemmäksi.

Laitteiden varsinaisen fyysisen käyttöönoton jälkeen laitteisiin muodostettiin yhteys USB-väylän avulla (kuva 11). Jokaiselle Queclinkin laitteelle on saatavilla Windows-ohjelmisto, jolla laitteen tietoja voidaan tarkastella ja hallita USB-väylän kautta (kuva 11). Ensimmäisissä kokeiluissa GL505 asetettiin tilaan, jossa GPS-sijaintitietoja raportoidaan tekstiviestein. Pian kuitenkin ilmeni, että GL505 saattoi mennä virransäästötilaan kesken USB-yhteyden. Queclinkin toimittaman ohjeen mukaisesti GL505 saatiin herätettyä virransäästötilasta, jonka jälkeen virransäästötila myös kytkettiin pois päältä.



Kuva 11. USB-tiedonsiirtokaapeli kytkettynä GL505:een.

GL505:n lähettämät tekstiviestit toimivat odotetusti. Tämän jälkeen vuorossa oli TCP-yhteydellä toimivan tiedonsiirron testaus. Koska aluksi saatavilla ei ollut julkisella IP-osoitteella varustettua Internet-yhteyttä, piti TCP-yhteyttä testata ngrok.com -palvelulla. Tällä palvelulla voi luoda julkisen URL:n, johon kohdistuva liikennöinti ohjataan paikalliseen tietokoneeseen ngrok-sovelluksen avulla. Ngrokin avulla luodulla TCP-tunnelilla muodostettiin yhteys GL505:stä kannettavaan tietokoneeseen. Tässä vaiheessa varsin yksinkertainen JavaScript-pohjainen TCP-palvelin oli jo käynnissä, ja useamman yrityksen jälkeen ensimmäiset TCP-sanomat saatiin tulostettua komentokehotteeseen. Kuvassa 12 on TCP-yhteyden kokeilussa käytetyn TCP-palvelimen lähdekoodi. Lopullisessa tuotteessa käytetään Python-pohjaista TCP-palvelinta, jonka käynnistymistä ohjataan Djangoilla.

```

1  const net = require('net');
2
3  net.createServer(socket => {
4    socket.on('data', function(data) {
5      console.log('Echoing: %s', data.toString());
6      socket.write(data.toString())
7    });
8  }).listen(8001);

```

Kuva 12. Esimerkki NodeJS-pohjaisesta TCP-palvelimesta.

## 5.2 Tracking Cloudin kehitys

GPS-paikantimien TCP-yhteyden toiminnan varmistuttua alettiin kehittää Tracking Cloudiin uutta antureista vastaavaa osakokonaisuutta. Tracking Cloudissa antureilla kuvataan yksittäisiä laitteita, jotka lähettävät sanomia Tracking Cloudiin. Antureiden lisäksi Tracking Cloudia laajennettiin geoaita- ja karttaominaisuuksin. Esimerkiksi listanäkymään lisättiin tuki perinteisen taulukkonäkymän ohelle karttanäkymä. Tracking Cloudin karttanäkymää käytetään havainnollistamaan geopisteitä ja geoaitoja.

Tracking Cloudissa kuvataan liiketoimintaa lähtökohtaisesti kolmen perustietomallin mukaan, jotka ovat sijainti, seurattava kohde ja yritys/yhtiö. Kaikista perustietomalleista voidaan luoda rajattomasti eri tyyppisiä malleja kuvaaman asiakasyrityksen liiketoiminnan eri osia, kuten esimerkiksi varastoja, asiakasrekistereitä ja huoltotilauksia. Tämän opinnäytetyön kannalta olennaiset perustietomallit ovat sijainti ja seurattava kohde.

Sijainnilla kuvataan nimensä mukaisesti sijainteja ja alueita. Sijainnit ovat hierarkkisia ja yleensä kuvaavat asiakasyrityksen toimialueita. Esimerkiksi kiinteistövälitysyrittäjällä sijaintiin perustuva tietomalli voisi olla alue, jossa kiinteistövälitystoimintaa harjoitetaan. Sijainneilla on myös geoaita - sijainnin geoaita määrittää, mitkä toimenpiteet suoritetaan seurattavan kohteen siirtyessä geoaidan sisäpuolelle ja lähtiessä geoaidan alueelta pois.

Seurattava kohde on yleensä asiakkaan liiketoiminnassa esiintyvä asia tai kokonaisuus. Seurattavia kohteita on siis monia, ja esimerkiksi kiinteistövälitysyrittäjällä seurattava kohde voisi olla yksinkertaisimmillaan kiinteistö. Seurattavat kohteet ovat yleensä riippuvia muista tietotyypeistä ja seurattavien kohteiden viittauksia muihin tietorakenteisiin, kuten esimerkiksi sijaintiin, kuvataan suhteilla. Sijainnin ja seurattavan kohteen suhde on sidoksissa, johon GPS-paikantimena ilmenevä anturi kiinnittyy. Mikäli seurattavan kohteen ja sijainnin sidoksissa anturi on tyypiltään geoaitaava, muuttuu sidoksessa oleva sijainti anturin luomien geoaitaustapahtumien mukaan. Anturi myös ylläpitää seurattavan kohteen ja sijainnin sidoksessa geoaidan poistumistietoa. Tällöin seurattavan kohteen viimeisin tunnettu sijainti on yhä tiedossa ja seurattava kohde on merkittävä poistuneeksi sijainnistaan.

Seuraavaksi toteutettiin Tracking Cloudiin logiikka, jonka avulla geoaitaus suoritetaan. Geoaitaus alkaa, kun jokin GPS-paikannin rekisteröi uuden sijainnin Tracking Cloudissa. GPS-paikantimen ilmoittaman GPS-sijainnin on erottava kyseisen GPS-paikantimen edeltävästä GPS-sijainnista koordinaattien ja GPS:n UTC-aikaleiman osalta. GPS-sijainnin koordinaattien ja GPS:n UTC-aikaleiman muuttumisen vaatiminen on varsin yksinkertainen tapa varmistaa, että esimerkiksi GPS-paikantimen puskurista tulevat vanhemmat sanomat eivät aiheuta sekaannusta.

GPS-paikantimen antamaa sijaintia vastaavat geoaidat haetaan tietokannasta. GPS-paikantimen sijaintia vastaavia geoaitoja ei kuitenkaan välttämättä ole ollenkaan, joten GPS-paikantimen sijaintia vastaavien geoaitojen määrä on välillä 0-n. Kuvassa 13 on esitetty, miten Djangoissa voidaan GeoDjangon avulla suorittaa geoaitaus. Kuvan 13 rivillä 2 käytetään GeoDjangon tarjoamaa sarakkeen käsittelijää ”contains”. ”contains”-sarakekäsittelijä kääntyy PostgreSQL:ssä PostGIS:in tietokantafunktioksi ”ST\_Contains”, joka on kuvattu luvussa 2.1. Django sarakekäsittelijät ovat lisäksi kuvattu hieman tarkemmin luvussa 2.2.

```

1     matching_geo_area_locations = Location.objects.filter(
2         geo_area__isnull=False, geo_area__contains=geo_location
3     )

```

Kuva 13. Esimerkki geoaitaamisesta Django-ohjelmistokehitysalustassa GeoDjangon avulla.

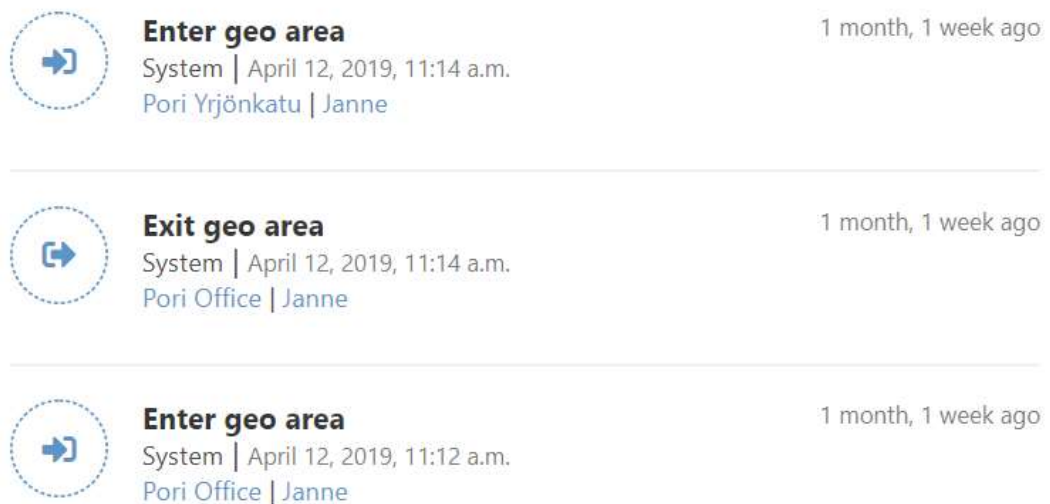
Geoaitojen hakemisen jälkeen käydään läpi kaikki seurattavat kohteet, jotka ovat kytkettynä kyseiseen GPS-paikantimeen. GPS-paikantimeen kytketyille seurattaville kohteille kerrotaan mihin geoaitoihin (0-n kappaletta) on siirretty sisään, joka taas laukaisee varsinaiset geoaidan omistaman sijainnin määrittämät geoaitaamistapahtumat.

Varsinainen geoaitaustapahtumien laukaisu seurattavassa kohteessa perustuvat peräkkäin suoritettaviin ”sisään” ja ”ulos”-tapahtumiin. Mikäli GPS-paikantimen GPS-sijaintia vastaavien geoaitojen määrä on nolla, päättyy geoaitaus ”ulos”-tapahtumaan, sillä seurattava kohde ei ole enää minkään geoaidan sisäpuolella.

Kun GPS-paikantimen GPS-sijaintia vastaavia geoaitoja on vähintään yksi, suoritetaan ”ulos”-tapahtuma, mikäli seurattavan kohteen nykyinen sijainti ei esiinny GPS-paikantimen GPS-sijaintia vastaavissa geoaidoissa. Toisin sanoen seurattava kohde ei siis voi siirtyä pois ja siirtyä sisään samassa alueessa peräkkäisinä tapahtumina.

Lopuksi suoritetaan ”sisään”-tapahtuma niiden geoaitojen osuudelta, jotka eivät vastaa seurattavan kohteen nykyistä sijaintia.

”Sisään”- ja ”ulos”-tapahtumat laukaisevat varsinaiset sijainnille asetetut geoaitaus-tapahtumat ja päivittävät seurattavan kohteen sijainnin. Sijainnille määritellyt geoaitaus-tapahtumat näkyvät käyttöliittymässä (kuva 14) tapahtumina.



Kuva 14. Esimerkki käyttöliittymässä näkyvistä geoaitaus-tapahtumista.

### 5.3 Testaus

Geoaitaamista testattiin määrittämällä Porin keskustaan neljä geoaitaa (kuva 15). Testausreitit muodon takia kuvassa 15 pohjoinen on poikkeuksellisesti kuvan oikeassa reunassa ja etelä vastaavasti vasemmassa reunassa. Testausta varten määritellyillä geoaidoilla oli tarkoitus seurata liikkumista Bluugo Oy:n Porin toimipisteen ja testaus-kerralle valitun lounasravintolan välillä. Testauslaitteeksi valittiin GL300A ja GL300A:n lähetyksintervalliksi asetettiin 2 sekuntia.

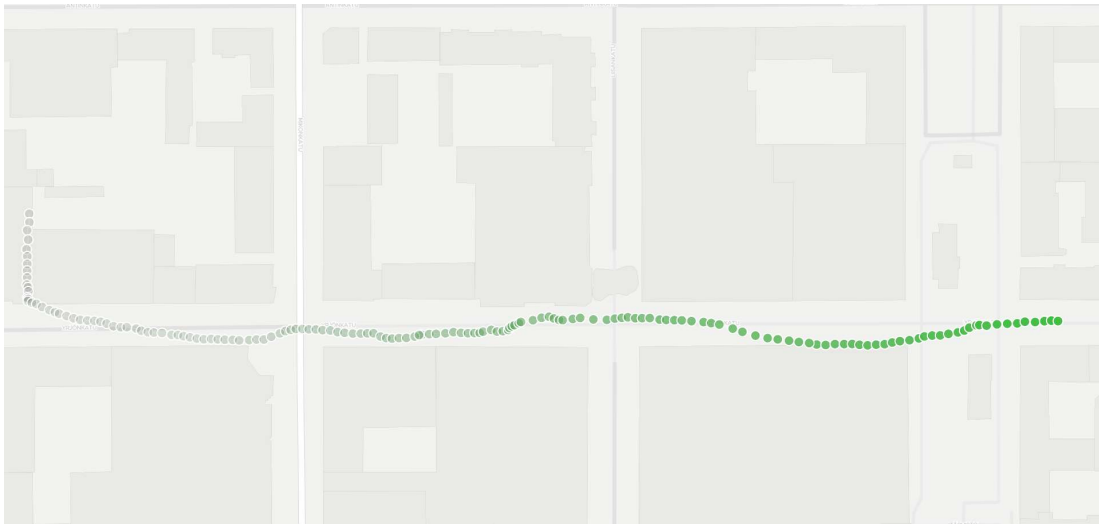


Kuvassa 15 keltainen nelikulmio ja numero 1 kuvaavat Bluugo Oy:n Porin toimipisteen sisältävää rakennusta ja kyseisen rakennuksen välitöntä läheisyyttä. Vastaavasti kuvassa 15 sininen nelikulmio ja numero 2 kuvaavat testausreitit osuutta Porin Yrjönkadulla. Vihreällä nelikulmiolla ja numerolla 3 merkitty alue kuvassa 15 kuvaa testausreitit osuutta Porin kävelykadulla. Testausreitit määränpää on merkitty kuvaan 15 punaisella nelikulmiolla ja numerolla 4.



Kuva 15. Testauksessa käytetyt geoaidat.

GL300A:n nauhoittama reitti on esillä kuvassa 16. Koska lähetysintervalli oli asetettu GL300A:ssa kahteen sekuntiin, saatiin GPS-pisteitä varsin tiheästi. Kuvassa 16 näkyvissä GPS-pisteissä ilmenee hieman virheitä, jotka todennäköisesti johtuvat ympäröivästä korkeista rakennuksista.



Kuva 16. GL300A:lla nauhoitettu testireitti.

Kokonaisuutena testaus onnistui hyvin. GL300A:n GPS-sijainnin raportointi toimi erinomaisesti, kun GL300A:lla oli esteetön näköyhteys taivaalle. Varsinaisia ongelmia ilmeni vasta, kun GL300A:n yhteys GPS-satelliitteihin katkesi. GL300A:n käyttäytyminen GPS-signaalin yhteyden katketessa on kuvattu luvussa 6.1.

Yhteenvetona GL505:n ja GL300A:n testikäytöstä voidaan todeta, että molemmat GPS-paikantimet soveltuvat hyvin kyseisille GPS-paikantimille määritetyille käyttötarkoituksille. GL505 on harvemmalla intervallilla toimiva kiinteän omaisuuden valvontaan soveltuva GPS-paikannin. GL300A taas soveltuu tiheämpään ja vaihtelevan kohteen seurantaan suuren akkukapasiteettinsa ja kotelointinsa ansiosta.

## 6 POHDINTA JA TULEVAISUUS

### 6.1 Pohdinta

Tämän opinnäytetyön aikana opin paljon IoT-laitteille keskeisistä tekniikoista. M2M-mobiililiittymät olivat entuudestaan tuttuja vain lyhenteensä puolesta. M2M-liittymien konkreettiset käyttökohteet ilmenivät varsin nopeasti juuri itsenäisten IoT-laitteiden muodossa - varsinkin IoT-laitteille, jotka liikkuvat tai joita liikutellaan. En yllättyisi,

jos löytäisin luvussa 4.1. kuvatun tapaisen GPS-paikantimen esimerkiksi jostain vuokrattavasta kalustosta, kuten vaikka peräkärystä.

M2M-mobiililiittymien ohella opin lisää mobiilidataliittymien tuotteistuksesta ja miten esimerkiksi julkinen IP-osoite ei ole oletuksena päällä 3G- tai 4G-liittymissä. Työkentely M2M-mobiililiittymien parissa antoi arvokasta kokemusta, jota on helppo hyödyntää myynissä, markkinoinnissa ja asiakaspalvelussa.

Kohdelaitteita testatessa ilmeni, että GL505:n ja GL300A:n GPS-paikannin ei välttämättä ole yhtä pitkälle kehitetty, kuin nykyisten älypuhelinien sisäinen GPS-paikannin. GPS-sijainnin haku sisätiloissa on lähes mahdotonta ja sisätiloissa testaaminen onnistui parhaiten ikkunan vieressä. Eräässä kokeilussa otin GL300A:n mukaan lounaalle. Kävelin Porissa Yrjönkatua pohjoiseen noin puolen kilometrin matkan. Itse lounaspaikka sijaitsi rakennuksen kellarikerroksessa, joka katkaisi GPS-paikantimen näköyhteyden satelliitteihin. Lounaan aikana GL300A:n ilmoittama sijainti oli alkanut itsestään siirtyä pohjoiseen useamman korttelin verran. Paluumatkalla ehdin kävellä useamman minuutin Yrjönkatua etelään ennen, kuin GL300A sai haettua GPS-sijainnin uudelleen ja korjattua sijaintinsa.

Yksinkertainen kokeilu, jossa GPS-paikannin menettää näköyhteyden GPS-satelliitteihin antoi arvokasta kokemusta siitä, miten GPS-paikantimen antamiin tietoihin ei voi aina luottaa sokeasti. Tämän takia Queclinkin laitteet ilmoittavat GPS-sijaintiedon ohella myös tarkkuuden, jolla GPS-sijaintitieto on saatu. Saapuvia GPS-sijaintitietoja pitää siis aina jalostaa ja käsitellä, jotta mahdollisesti virheellinen sijaintitieto ei aiheuta ongelmia.

Arvokasta kokemusta kertyi myös PostgreSQL:stä ja GeoDjangosta. Näiden tekniikoiden yhdistelmällä on helppo toteuttaa verkkosovellus, jossa käsitellään geoaitoja. GeoDjangon oletuskarttakomponentit ovat varsin karuja, jonka takia varsinaisessa tuotteessa käytettiin Mapbox GL JS ja Mapbox Draw -kirjastoja. Mapboxin kirjastojen avulla karttapohjaa voidaan mukauttaa tuotteen värien ja teeman mukaisiksi ja luoda sovelluksen ulkoasun mukainen piirto- ja muokkausnäky geoidoille.

Karttakomponenttien (Mapbox GL JS ja Mapbox Draw) sovellus geoaitojen muokkaustyökaluna ei kuitenkaan ollut täysin vaivatonta. Mapbox käyttää monen muun selainpohjaisen karttatyökalun tapaan latitudiin ja longitudiin perustuvaa koordinaattijärjestelmää. GeoDjangossa oletuskoordinaattijärjestelmä on 3857, vaikka dokumentaation mukaan se on 4326. Tämä aiheutti hämmennystä, koska latitudi- ja longitudi-koordinaateista koostuva piirretty geoalue ei tallentunut tietokantaan ollenkaan. Vasta GeoDjangon lähdekoodia selatessa selvisi, että koordinaattijärjestelmä oli väärä. Kun koordinaattijärjestelmäksi oli asetettu 4326, tallentui geoaita viimeinkin oikein.

Teknisen toteutuksen aikana opitut asiat auttoivat ymmärtämään IoT-laitteista koostuvien kokonaisuuksien toteuttamisen mahdollisia haasteita, varsinkin teollisessa käytössä. Tämä opinnäytetyö antoi myös harvinaisen tilaisuuden keskittyä yksittäiseen aihealueeseen ja sen keskeiseen sisältöön ilman suurempia keskeytyksiä. Työrauhan ansiosta ohjelmistokehitys ja oppiminen olivat jatkuvaa ja sisältö pysyi hyvin hallussa. On todennäköistä, seuraava minun työpöydälleni ilmestytävä GPS-paikannin perustuu tiedonsiirron osalta NB-IoT -verkkoon. Haasteita ja tilaisuuksia oppia uusia tekniikoita on siis tarjolla jatkossakin.

## 6.2 Tulevaisuus

Bluugo Oy (tämän opinnäytetyön toimeksiantaja) auttaa asiakasyrityksiään tuotteiden ja palveluiden digitalisoinnissa. Bluugo käyttää liiketoiminnassaan omaa Tracking Cloud -palvelualustaa, joka kytkee uusimman IoT-tekniikan, laitteet ja ohjelmistot SaaS-pohjaiseksi palvelukokonaisuudeksi.

Paikannustietoa hyödyntävillä ratkaisuilla on keskeinen rooli Tracking Cloud -palvelussa tiedonkeruuseen liittyen; varsinkin logistiikan parissa toimivissa yrityksissä tarve geoaitauksen hyödyntämiseen on koko ajan kasvussa.

Tässä opinnäytetyössä rakennetut työkalut edesauttavat ja helpottavat merkittävästi GPS-tietoa hyödyntävien ratkaisuiden käyttöönottoa ja edesauttavat näin Tracking Cloud -palvelun kehittämistä. Geoaitaukseen liittyvät sovellukset antavat asiakkaille

modernit työkalut esim. geoaitauksen ylläpitoa ja hallintaa varten. Vaikka työssä on keskitytty yhden valmistajan laitemalleihin, kehitetyt työkalut mahdollistavat jatkossa eri valmistajien ja eri tyyppisten paikannuslaitteiden käytön ja helpon kytkettävyyden. Tällä on erittäin suuri merkitys esimerkiksi logistiikassa, jossa monissa seurantakoh-teissa ja ajoneuvoissa tarvittavat laitteet ovat jo asennettuna. Bluugo Oy voi näin suo-raan hyödyntää jo olemassa olevaa teknologiaa ilman lisäkustannuksia. Periaatetasolla tässä opinnäytetyössä kehitetyt mallit mahdollistavat älypuhelimien, dedikoitujen GPS-paikantimien ja tulevaisuudessa myös NB-IoT / 5G -pohjaisten sensorien käyttä-misen.

## LÄHTEET

Django introduction, MDN web docs:in verkkosivut, 2019. Viitattu 18.5.2019.  
<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>

Djangon ohjesivusto, 2019. Viitattu 20.5.2019.  
<https://docs.djangoproject.com/en/2.2/>

Djangon verkkosivut, 2019. Viitattu 18.5.2019.  
<https://www.djangoproject.com>

GPS Accuracy, 2019. Official U.S. government information about the Global Positioning System (GPS) and related topics. Viitattu 13.5.2019.  
<https://www.gps.gov/systems/gps/performance/accuracy/>

Mai, T. 2017. Global Positioning System History. Viitattu 13.5.2019.  
[https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS\\_History.html](https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html)

PostgreSQL:n verkkosivut, 2019. Viitattu 15.5.2019.  
<https://www.postgresql.org/about/>

Queclink GL505:n verkkosivut, 2019. Viitattu 13.5.2019.  
<http://www.queclink.com/GL505>

Queclink GL300A:n verkkosivut, 2019. Viitattu 13.5.2019.  
<http://www.queclink.com/GL300A>

The Hundred-Tonne Robots That Help Keep New Zealand Running, 2019. Viitattu 21.5.2019.  
<https://www.youtube.com/watch?v=kQ8WI3nc1I0>