



TAMPEREEN  
AMMATTIKORKEAKOULU

# GEOMAGNEETTINEN SISÄTILAPAIKANNUS ANDROID ÄLYPUHELIMELLA

Eero Meriläinen

Opinnäytetyö  
Kesäkuu 2016  
Tietotekniikka  
Ohjelmistotekniikka



## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikka  
Ohjelmistotekniikka

MERILÄINEN EERO

GEOMAGNEETTINEN SISÄTILAPAIKANNUS ANDROID ÄLYPUHELIMELLA

Opinnäytetyö 27 sivua, joista liitteitä 1 sivu  
Kesäkuu 2016

---

Ihmiset käyttävät päivittäin GPS:ään perustuvia paikannus- ja navigointisovelluksia älypuhelimissaan ja navigaattoreissaan mm. löytääkseen suorimman reitin osoitteeseen. Niin mullistavaa kuin GPS on ollutkin ihmisten elämässä, on silläkin omat rajoitteensa. GPS toimii luotettavana paikannusteknologiana ulkotiloissa, mutta sisätiloihin GPS signaalit eivät ylety kattorakenteiden takia ja GPS paikannuksen virhe on noin 8 metrin luokkaa. Sisätilassa toteutettavaan paikannukseen tarvitaan siis vaihtoehtoisia ratkaisuja.

Sisätilapaikannuksessa ollaan pitkään käytetty teknologioita Wi-Fi verkkoa hyödyntäen. Uusimpana ja viime vuosien innovatiivisimpana ratkaisuna on kehitetty magneettinen sisätilapaikannus, joka perustuu geomagneettisen kentän muutoksiin rakennuksen sisällä. Maailmassa toistaiseksi ainut magneettisen sisätilapaikannuksen tarjoaja sovelluskehittäjille on IndoorAtlas, jonka tarjoama teknologia on kiistatta luotettavin, helppokäyttöisin ja rahaa säästävin vaihtoehto.

Sisätilapaikannus on ollut uusi käsitys kuluttajille mahdollisesti olemassa olleiden teknologioiden puutteen takia. Magneettiseen sisätilapaikannukseen perustuva paikannusteknologia saattaa kuitenkin olla läpimurto sisätilapaikannuksen kuluttajien tason käytön suhteen. Älypuhelimelle helposti ladattava sisätilapaikannussovellus navigointimahdollisuuksineen helpottaa käyttäjän liikkumista esimerkiksi ostoskeskuksessa. Teknologia mahdollistaa myös muiden samaa sovellusta käyttävien ihmisten paikallistamisen.

Tätä teknologiaa käyttäen oma sovellus saatiin toimimaan Android älypuhelimella, joka näyttää käyttäjän sijainnin kartalla. Paikannusalueena toimi Tampereen ammattikorkeakoulun pääkampuksen A-siiven 3. kerros. Jotta paikannuksen saa toimimaan omassa sovelluksessa, on paikannettava alue ensin kartoitettava. Magneettinen sisätilapaikannus osoittautui toimivaksi tekniikaksi sisätilapaikannuksen toteuttamisen suhteen.

---

Asiasanat: sisätilapaikannus, android sovelluskehitys

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Information Technology  
Software Engineering

MERILÄINEN EERO  
GEOMAGNETIC INDOOR POSITIONING WITH AN ANDROID SMARTPHONE

Bachelor's thesis 27 pages, appendices 1 page  
June 2016

---

People use daily positioning and navigation applications based on GPS in their smartphones and navigators for locating and finding the straightest route to their destination. As magnificent as GPS has been to people, it too has its limits. GPS is reliable in outdoor areas but indoor areas are not penetrated by GPS signals and the GPS has an accuracy of about 8 meters. Therefore alternative means for indoor positioning are required.

Indoor positioning has long been delivered by analyzing Wi-Fi signals in a building. The newest and arguably the most innovative solution for indoor positioning is magnetic indoor positioning, which is based on differences in the geomagnetic field inside a building. So far the only provider in the world for this kind of technology is IndoorAtlas which provides undoubtedly the most reliable, easy to use and money saving option for indoor positioning.

Indoor positioning has been a new concept for consumers possibly because of the lack of existing technologies. However, magnetic indoor positioning might turn out to become a breakthrough among consumer level users. An application that is easy to download for a smartphone with the ability to navigate for example inside a shopping mall helps the movement of the user. This technology also makes it possible to locate other people using the same application.

Using this technology an own application was successfully run on an Android smartphone which shows the location of the user on a map. The area to be positioned was the third floor in the A-wing of TAMK main campus. In order to get the positioning to work in an own application, the area where positioning is expected to work has to be mapped first. Magnetic indoor positioning turned out to be a working solution for indoor positioning.

---

Key words: indoor positioning, android software development

## SISÄLLYS

1	JOHDANTO.....	6
2	MAGNEETTINEN SISÄTILAPAUKANNUS.....	7
2.1	IndoorAtlas ja sisätilapaikannuksen käyttösovellukset .....	8
2.2	Älypuhelimien laitevaatimukset .....	9
3	INDOORATLASIN KÄYTTÖÖNOTTO .....	10
3.1	IndoorAtlas Developer.....	10
3.2	IndoorAtlas MapCreator .....	11
4	OMAN INDOORATLAS SOVELLUKSEN TEKO .....	13
4.1	Android Studio ja työssä käytetty älypuhelin .....	13
4.2	Javasta yleisesti .....	13
4.3	TAMK Indoor Positioning System – ohjelman päivittäminen .....	14
4.4	Gradle ja Maven projektit .....	14
4.5	IndoorAtlas SDK .....	15
4.6	Projektin avaaminen ja asetusten kohdistaminen .....	16
4.6.1	Älypuhelimien asetukset .....	17
4.6.2	Projektin suorittaminen .....	18
4.7	Tehty sovellus .....	18
4.7.1	Java-tiedostot.....	18
4.7.2	XML-tiedostot.....	20
4.7.3	Gradle skriptit.....	21
4.7.4	Sovelluksen toiminnallisuus .....	22
4.8	Kehitysmahdollisuudet .....	24
5	YHTEENVETO .....	25
	LÄHTEET.....	26
	LIITTEET .....	27
	Liite 1. Projektin sisältämät tiedostot .....	27

**ERITYISSANASTO**

API	Application Programming Interface
GPS	Global Positioning System
SDK	Software Development Kit
TAMK	Tampereen ammattikorkeakoulu
Wi-Fi	WLAN, langaton verkko
WPS	Wi-Fi Positioning System

## 1 JOHDANTO

Tämä opinnäytetyö käsittelee magneettista sisätilapaikannusteknologiaa, sen nykyisiä käyttösovelluksia ja omaa sisätilapaikannusprojektia. Työssä perehdytään magneettiseen sisätilapaikannukseen käsitteenä, IndoorAtlas teknologiaan ja sen käyttöön, sekä oman sisätilapaikannussovelluksen tekemiseen Android Studiolla.

Ihmisille tutuksi tullut GPS on tärkeä työkalu tänäpäivänä paikantamisen suhteen. GPS:ää käytetään niin navigaattoreissa kuin älypuhelimissa. GPS ei kuitenkaan teknologiana ole riittävä sisätilapaikannusta varten ja tämän takia luotettavan sisätilapaikannusteknologian tuottaminen on saanut ihmisiä tekemään innovatiivisia ratkaisuja tarkalle sisätilapaikannukselle.

Yksi näistä ratkaisuista on suomalaislähtöisen IndoorAtlas Ltd:n kehittämä menetelmä jossa sisätilapaikannus perustuu maapallon magneettikentän muutoksiin rakennuksen sisällä. On havaittu, että maapallon magneettikenttä teräsrakenteiden sisällä pysyy samana ja luo rakennukselle yksilöllisen sormenjäljen, mikä tekee magneettisesta sisätilapaikannusteknologiasta ideaalista sisätilapaikannuksen näkökulmasta.

IndoorAtlas tarjoaa ilmaisen palvelun ohjelmistokehittäjille jotka haluavat käyttää magneettista sisätilapaikannusta omissa sovelluksissaan. Palvelussa älypuhelimessa oleva sovellus lähettää tietoa pilvipalveluun ja takaisin, jonka kautta saadaan älypuhelimessa toimivalle sovellukselle tieto käyttäjän sijainnista rakennuksen sisällä.

## 2 MAGNEETTINEN SISÄTILAPAIKANNUS

Magneettinen sisätilapaikannus perustuu maapallon magneetikentän muutokseen ja sen kehitti alun perin Janne Haverinen ja Aleksi Kemppainen (Magnetic positioning, Wikipedia, 2016). Sisätilapaikannus perustuu maapallon magneetikentän muutokseen eikä pelkästään GPS:ään. GPS ei sovellu sisätilapaikannukseen sillä satelliitin mikroaallot heikkenevät ja hajoavat osuessaan kattoihin ja muihin koviin materiaaleihin. Myöskin GPS:n tarjoama tarkkuus on noin 8 metriä mikä on suuri virheen määrä sisätilapaikannusta varten (Community Health Maps, NIH, 2016). Sen sijaan magneetikentät teräsrakenteiden sisällä pysyvät samana ja luovat rakennukselle yksilöllisen ”sormenjäljen”.

Magneettisella sisätilapaikannuksella on suurta potentiaalia. Sisätilapaikantamiseen on aiemmin käytetty esimerkiksi WPS teknologiaa joka on Wi-Fi verkkoa käyttävä sisätilapaikannusmenetelmä. Yleisin tapa käyttää WPS teknologiaa sisätilapaikannukseen on mittaamalla päätelaitteen vastaanottaman Wi-Fi signaalin vahvuutta sisätilan eri kohdissa. WPS teknologiaan perustuvat sisätilapaikannusratkaisut ovat tarkkuudeltaan täysin sisätilassa olevan langattomien Wi-Fi lähettimien varassa (Indoor Positioning System, Wikipedia, 2016).

Koska magneettinen sisätilapaikannus perustuu rakennuksen geomagneettisten arvojen vaihteluihin, se on siten riippumaton Wi-Fi signaalin vahvuudesta itse paikantamisprosessissa. Täten magneettista sisätilapaikannusta voidaan käyttää ympäristöissä joista ei löydy Wi-Fi lähettimiä tai joiden signaali ei ole tarpeeksi vahva luotettavaan paikannukseen.

Magneettisen sisätilapaikannuksen ainut heikkous on se, että rakennuksen sisäiseen magneetikenttään vaikuttavat liikuteltavat metallimateriaalit kuten hissit ja metallihyllyt sekä elektroniset laitteet kuten tietokoneet (Magnetic Positioning, Wikipedia, 2016).

## 2.1 IndoorAtlas ja sisätilapaikannuksen käyttösovellukset

IndoorAtlas Ltd. on vuonna 2012 Janne Haverisen perustama yritys, joka kehittää sisätilapaikannusteknologiaa. IndoorAtlas aloitti pienenä startup yrityksenä ja sillä on tällä hetkellä toimistot viidessä eri maassa (About us, IndoorAtlas, 2016). IndoorAtlas on ainut laatuaan tarjoamaan ilmaisen kehitysalustan ohjelmistokehitykselle, joka käyttää geomagneettisia sensoreita sisätilapaikannustarkoituksessa.

IndoorAtlasin teknologiaa voidaan käyttää muun muassa kauppakeskuksissa, lentokentillä, juna-asemilla, urheilustadioneilla, museoissa, sairaaloissa ja muissa suurissa julkisissa rakennuksissa. Sisätilapaikannusteknologialla on mahdollista toteuttaa mm. navigointia ja käyttäjänetsimistöimintoja. Se soveltuu mainiosti esimerkiksi suuriin kauppaja- ja messukeskuksiin joissa on monenlaisia kohteita joiden välillä käyttäjä joutuu liikkumaan. Navigointiominaisuudella käyttäjä pystyy valitsemaan kohdepisteen ja ohjelma näyttää älypuhelimella ruudulla suorimman reitin pisteeseen.

Magneettinen sisätilapaikannusteknologia tuo esille myös uudenlaisia sosiaalisen kanavoinnin menetelmiä. Mobiilisovelluksen pystyessä keräämään tietoa ihmisten liikkeistä kartoitetussa sisätilassa on mahdollista pysyä yhteydessä ihmisiin rakennuksen sisällä joilla on sama sovellus omassa käytössä. Yksi tilanne jossa sisätilapaikannusta voitaten hyödyntää olisi esimerkiksi tilanne, jossa kaverukset saapuvat suureen kauppakeskukseen jossa jokaisella on tarve asioida eri liikkeissä. Sisätilapaikannussovelluksesta olisi sitten mahdollisuus seurata toisen henkilön liikkeitä kauppakeskuksessa ja siten pysyä kartalla missä hän liikkuu. Tämän kaltaisella sovelluksella olisi käyttösovelluksia myös sen suhteen että vanhemmat pystyvät seuraamaan lastensa liikkeitä vastaavissa ympäristöissä.

IndoorAtlas mahdollistaa mm. Google Maps palvelun integroimisen omiin sisätilapaikannussovelluksiin. Täten on mahdollista tuottaa sovelluksia jotka sisältävät vaikkapa useita eri kartoitettuja rakennuksia ja käyttäjä siirtyy ulkoreittien kautta rakennuksesta toiseen. Oman sisätilasijainnin pystyy myös näyttämään Google Maps palvelussa ottamalla omaan projektiin käyttöön Google Maps Android API:n (Developer Guide, IndoorAtlas, 2016).



## 2.2 Älypuhelimien laitevaatimukset

IndoorAtlas teknologiaan perustuva sovellus kerää tietoa älypuhelimien nopeudesta, asennosta ja geomagneettisista arvoista, lähettää ne pilvipalveluun ja kokoaa tiedoista rakennukselle ominaisen ”geomagneettisen pohjapiirroksen”. Älypuhelimelta vaaditaan siten kiihtyvyyssanturi, geomagneettinen anturi ja gyrometri sekä vähintään Android 4.3 tai iOS 6 käyttöjärjestelmä (IndoorAtlas, FAQ, 2016). Nämä ominaisuudet löytyvät käytännössä kaikista uusista Android ja iPhone älypuhelimista. Mikäli haluaa käyttää IndoorAtlas teknologiaa vanhempimallisissa puhelimissa, tulee tarkistaa löytyykö älypuhelimesta vähimmäisvaatimukset. Älypuhelimella on myös oltava yhteys joko 3G, 4G tai langattomaan verkkoon jotta kommunikointi IndoorAtlasin pilvipalvelun ja älypuhelimien välillä onnistuisi.

### 3 INDOORATLASIN KÄYTTÖNOTTO

IndoorAtlasin käyttöönottoa varten tarvitsee käyttäjän aluksi luoda oma tili joko tietokoneen kautta selaimella tai älypuhelimella IndoorAtlas MapCreator -sovelluksen kautta. Kirjaututtua sisään tulee käyttäjän mennä IndoorAtlasin Developer osioon ja seurata ohjeita annetussa järjestyksessä.

#### 3.1 IndoorAtlas Developer

Aluksi luodaan selaimella IndoorAtlas Developer osiossa oma kohde (venue) ja annetaan sille nimi. Tämän jälkeen luodaan uusi pohjapiirros, määritetään sille nimi ja kerros sekä valitaan kuvatiedosto jonka tulee olla mahdollisimman selkeä ja tarkka pohjapiirros kartoitettavasta rakennuksesta. Tämän jälkeen kuva asetellaan karttaan sille kuuluvaan kohtaan. Asettelen voi myös tehdä älypuhelimella IndoorAtlas MapCreator sovelluksella, mutta selaimella asetelu saadaan toteutettua paljon tarkemmin. Kun pohjapiirroksen asetelu ja editointi on valmis, klikataan näppäintä submit joka vie tämän pohjapiirroksen tietokantaan ja luo arvot FloorplanId, FloorId ja VenueId -muuttujille. Näitä muuttujia voi käyttää hyödyksi omassa sovelluskehityksessä. Kuvassa 1 näytetään pohjapiirroksen asetelua selaimella.

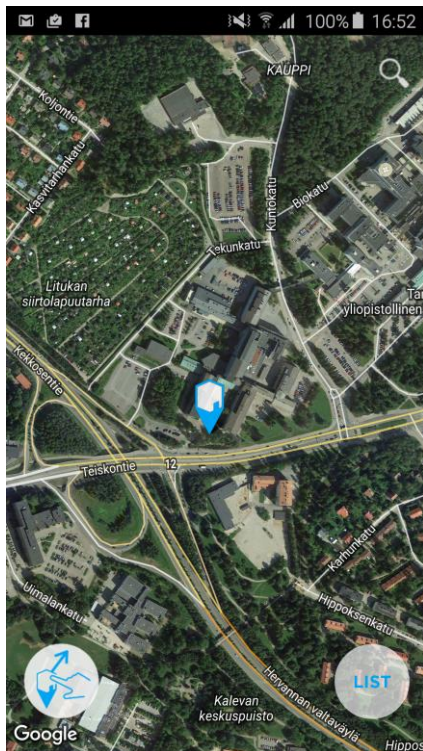


Kuva 1: Pohjapiirroksen asetelu tietokoneselaimella. Kuvassa asetellaan TAMK:n A-siiven 3. kerroksen pohjapiirrosta.

Tässä vaiheessa kaikki on valmista kartoittamisen aloittamiseksi. Mikäli tarkoitus on käyttää IndoorAtlas teknologiaa omassa sovelluksessa, pitää vielä luoda selaimen kautta oma sovellus Developer osiossa ”Apps” välilehden kautta. Uuden sovelluksen luonti luo sovellukselle ID ja Secret – muuttujat jotka sijoitetaan omaan sovellukseen.

### 3.2 IndoorAtlas MapCreator

IndoorAtlas MapCreator on Android-sovellus, jonka avulla voidaan luoda uusia kohteita ja kartoittaa niitä. Näitä kartoitettuja kohteita voidaan sitten käyttää omassa Android sovelluksessa. Kun MapCreator sovellukseen on kirjaututtu, navigoidaan maailmankartalla kartoitettavan kohteen luokse. Uuden kohteen (venue) pystyy siis luomaan sekä selaimella että Android-sovelluksella. Jos kohdetta ei ole aiemmin luotu, luodaan se vasemmassa alareunassa näkyvästä nappulasta. Kuvassa 2 näkyy yleiskatsaus IndoorAtlas MapCreator sovelluksesta.

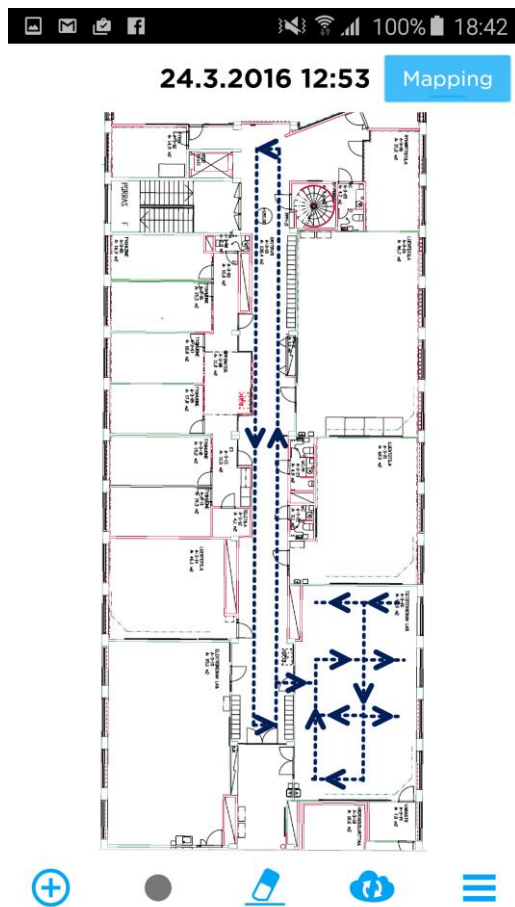


Kuva 2: Luotu venue IndoorAtlas MapCreator sovelluksessa.

Jokaiselle kohteelle voidaan luoda useita kerroksia joihin tuodaan kerrallaan omat pohjapiirroksensa. Pohjapiirroksen tulee olla mahdollisimman tarkka, mieluiten

esimerkiksi AutoCAD:llä tehty, mutta myös yksinkertaisella kuvaeditorilla tehty pohjapiirros toimii.

Kun kohteelle on tuotu pohjapiirros, se voidaan kartoittaa ”Mapping” välilehdeltä. Kuvassa 3 on näkymä IndoorAtlas MapCreatorin kartoitusvaiheesta. Ennen kartoitusta älypuhelin pitää kuitenkin kalibroida ”Calibration” välilehden kautta. Kalibroinnissa älypuhelimta pyöritetään jokaisen akselinsa ympäri samanaikaisesti. Tämä prosessi kestää toimivalla puhelimella ja oikein suoritettuna noin 5 sekuntia.



Kuva 3: TAMK:n A-siiven 3. kerroksen kartoitusta IndoorAtlas MapCreator sovelluksessa. Pohjapiirrosta kartoittaessa määritetään ensin nuolella kartoitettava kohta ja sitten tämä reitti kävellen pitäen älypuhelin vaakasuorassa asennossa kartoituksen ajan. Kartoituksen jälkeen MapCreator sovellusta voidaan käyttää kartoitettujen reittien testaamiseen.

## 4 OMAN INDOORATLAS SOVELLUKSEN TEKO

### 4.1 Android Studio ja työssä käytetty älypuhelin

Ohjelmistokehityksessä käytettiin Android Studio 1.5 sovelluskehitysalustaa. Android Studio on Googlen kehittämä sovelluskehitysalusta Android mobiilisovelluksille ja se perustuu JetBrainsin IntelliJ IDEA ohjelmistoon. (Android Developer, Android Studio Overview, 2016). Kehitysalustana Android Studio on tyypillinen sovelluskehitysalusta, josta löytyy projektin build ja run ominaisuudet sekä SDK Manager jonka kautta ohjelmalle voi asentaa tarvittavia SDK paketteja sekä plugineja. Pääpainotteisena ohjelmointikielenä Android Studiolla tehdyissä applikaatioissa toimii Java.

Päätelaitteena sovelluskehitykselle käytettiin Samsung Galaxy S6 älypuhelin. Älypuhelin on kyseinen laitevaatimusten takia ja sen Android versio on 5.1.1 Lollipop.

### 4.2 Javasta yleisesti

Java on olio-ohjelmointiin perustuva ohjelmointikieli. Toisin kuin ohjelmointikielissä kuten C ja C++ jotka suorittavat toteutettavan koodin sillä alustalla mitä käytetään, Java suorittaa koodin virtuaalikoneessa alustasta riippumatta. Tämä tekee koodin suorittamisesta aikaakuluttavampaa, mutta helpommin siirrettävää yhdeltä alustalta toiselle, esimerkiksi Windowsilta Linuxille (Vaasan yliopisto, Pitkä harppaus taaksepäin: Javasta C++:n, 2016).

Java perustuu vahvasti C++:aan sillä molemmissa ohjelmointikielissä keskeisenä toimintaperiaatteena on oliopohjaisuus. Javassa on C++:aan verrattuna useita ominaisuuksia kuten kirjastot graafisille käyttöliittymille sekä rajapinnat jotka mahdollistavat Java sovelluksien toiminnan monissa eri käyttöjärjestelmissä. Javasta on tullut keskeinen ohjelmointikieli Android mobiilisovelluksien ohjelmointiin näiden ominaisuuksien ansiosta.

Javassa kaikki ohjelmointi toteutetaan luokkien sisäisesti. Javassa ei käytetä globaaleja funktioita tai tietoa, vaan näiden vastineena toimii static muuttujat ja metodit jonkin

luokan sisällä. Tässä korostuu Javan oliopohjaisuus; jokaisella luokalla on oma sisäinen rakenne ja suhde tiettyyn ympäristöön. Näiden ominaisuuksien kautta olioilla voidaan mm. suorittaa haluttaessa vain tiettyjä toimintoja luokan sisältä sekä taataan luokan sisältämien muuttujien arvojen suojaus (Oversti, Olio-ohjelmointi, 2016).

### **4.3 TAMK Indoor Positioning System – ohjelman päivittäminen**

TAMK:ssa aikaisemmin opiskellut henkilö on tehnyt koululle opinnäytetyönään sovelluksen joka perustuu myös IndoorAtlas teknologiaan. Tämä ohjelman jatkokehittäminen ei sellaisenaan ole ideaalista sillä IndoorAtlasin API ei enää tue 1.4.2 versiolle tehtyjä sovelluksia. Ohjelman käynnistäessä sovellus kaatuu siitä syystä että ohjelman kirjastot ovat 32-bittisiä eikä 64-bittisiä. Sovellus käyttää JAR-kirjastoja jotka on manuaalisesti tuotu osaksi projektia eikä kyseessä ole Gradle projekti.

Tämän takia omaa projektia varten käytettiin pohjana IndoorAtlasin tarjoamaa esimerkkiohjelmopakettia, jossa demonstroidaan erilaisia mahdollisuuksia IndoorAtlas teknologian käyttämiseksi omissa projekteissa. Esimerkkiohjelmapakettista karsittiin pois ominaisuudet joita tässä projektissa ei ole tarkoituksena käyttää ja tehtiin tarvittavia muutoksia ”ImageView” – esimerkkiohjelmaan, joka sovelluksena demonstroi oman IndoorAtlas venuen käyttämistä omissa projekteissa.

### **4.4 Gradle ja Maven projektit**

Android sovelluskehityksessä on käytetty pitkään Ant ja Maven työkaluja projektien rakennusvaiheessa. Nämä työkalut ovat itsessään mahdollistaneet älypuhelinsovelluskehityksen, mutta ne ovat puutteellisia jos sovelluksessa halutaan käyttää useampia erilaisia kirjastoja joita Java projekteissa usein käytetään. Vastauksena Ant ja Maven työkalujen puutteille on kehitetty Gradle, joka yksinkertaistaa riippuvuuksien ja sisältöjen asettamista omille projekteille. Gradle hyödyntää Antin ja Mavenin parhaimpia ominaisuuksia ja tekee sovelluksen rakennusskriptistä yksinkertaisemman. Ant ja Maven projekteissa projektin rakennusskripti on XML muodossa kirjoitettu, mikä tekee suurempien skriptien käsittelystä kohtuuttoman monimutkaista (Dr.Dobbs, Why Build Your Java Projects with Gradle Rather than Ant

or Maven, 2016). Gradle on nykyään Android Studion lähtökohtainen projektin rakennusala ja sitä käytetään tämän opinnäytetyön projektissakin.

#### 4.5 IndoorAtlas SDK

IndoorAtlas tarjoaa SDK-paketin joka tulee liittää osaksi omaa sovellusprojektia. SDK-paketti on kokoelma kirjastoja jotka sisältävät omia kutsuttavia funktioita.

Tämän opinnäytetyön alkuperäisenä tarkoituksena oli jatkokehittää edellisen opiskelijan IndoorAtlas-sovellusta. Aikaisempi ohjelma on rakennettu IndoorAtlas SDK 1.4.2 versiota käyttäen. Opinnäytetyötä tehdessä uusin versio oli 2.1.0, joka eroaa huomattavasti 1.4.2 versiosta. Taulukko 1:ssä on listattu kaikki 1.4.2 ja 2.1.0 SDK pakettien luokat.

<b>IndoorAtlas Android SDK 1.4.2</b>	<b>IndoorAtlas Android SDK 2.1.0</b>
CalibrationEvent	IAFloorPlan
CalibrationState	IALatLng
ErrorCodes	IALocation
FloorPlan	IALocation.Builder
FutureResult	IALocationListener
GeoPoint	IALocationListenerSupport
ImagePoint	IALocationManager
IndoorAtlas	IALocationRequest
IndoorAtlasException	IALocationService
IndoorAtlasFactory	IARegion
IndoorAtlasListener	IARegion.Listener
MetricPoint	IAResourceManager
PositionerParameters	IAResult
ProgressCallback	IAResult.Error
ResultCallback	IAResult.Error.Category
ServiceState	IAResultCallback
	IATask

Taulukko 1: IndoorAtlas SDK kirjastojen luokat listattuna. (IndoorAtlas API, 2016).

Kuten taulukosta näkyy, IndoorAtlas Android SDK 2.1.0 versiossa luokat rakentuvat täysin eri tavalla kuin 1.4.2 versiossa.

IndoorAtlas SDK 2.X versioissa on siirrytty AAR Maven riippuvuuksien käyttöön ja IndoorAtlas suosittelee niiden käyttöä helpomman 64-bittisyyden tukemisen vuoksi, mutta tarjoaa mahdollisuuden tuoda kirjasto projektiin ladattavana JAR-tiedostona. (IndoorAtlas, Android SDK 1.X to 2.X migration guide, 2016). JAR-kirjastojen käyttäminen omissa projekteissa ei ole suositeltua sillä se tekee IndoorAtlasin SDK:n kehittyessä kirjastojen integroimisesta huomattavasti työläämpää, kun AAR Maven riippuvuuksilla käytettävät kirjastot vain määritetään omassa build.gradle tiedostossa, joka sitten lataa ne Internetin kautta projektin build-vaiheessa ja asettaa ne osaksi projektia.

IndoorAtlas Android SDK 2.X versioissa älypuhelimien kalibrointi suoritetaan IALocationManager luokassa aiempien CalibrationEvent ja CalibrationState luokkien sijaan. Kalibrointi ei ole välttämätöntä ennen sijaintipäivityksiä varten kuten se on ollut SDK 1.3.X versioissa (IndoorAtlas, IndoorAtlas Android SDK 1.x to 2.x Migration Guide, 2016). Tämä muutos mahdollistaa sovelluskehittäjän itse päättävän minkälainen kalibrointi on tarpeellista sovellukselle. SDK 2.X paketteihin perustuvat sovellukset jatkavat älypuhelimien kalibrointia samaan aikaan kun sovellus on käynnissä.

#### **4.6 Projektin avaaminen ja asetusten kohdistaminen**

Android Studiolla työskennellessä on hyvä käyttää vähintään Android Studio 1.5 versiota sekä uusimpia SDK paketteja. Android SDK Managerin kautta voidaan asentaa puuttuvia SDK paketteja sekä päivittää olemassaolevia. Muun muassa Android SDK Build Tools, Android SDK Tools ja Android SDK Platform-Tools työkalut on oltava päivitetty uusimpaan versioon jotta uusien projektien synkronointi onnistuu.

Projektin avaamalla Android Studioissa saattaa Android Studio antaa virheilmoituksen ”Please select Android SDK”. Tämä tarkoittaa sitä että projektille ei ole osoitettu SDK pakettia. Ongelma tulee korjata navigoimalla Android Studioissa File > Project Structure > Project Settings > Modules > Main > Dependencies, ja täältä asettamalla asianmukaisen Module SDK:n, tässä tapauksessa Android API 23 Platform.

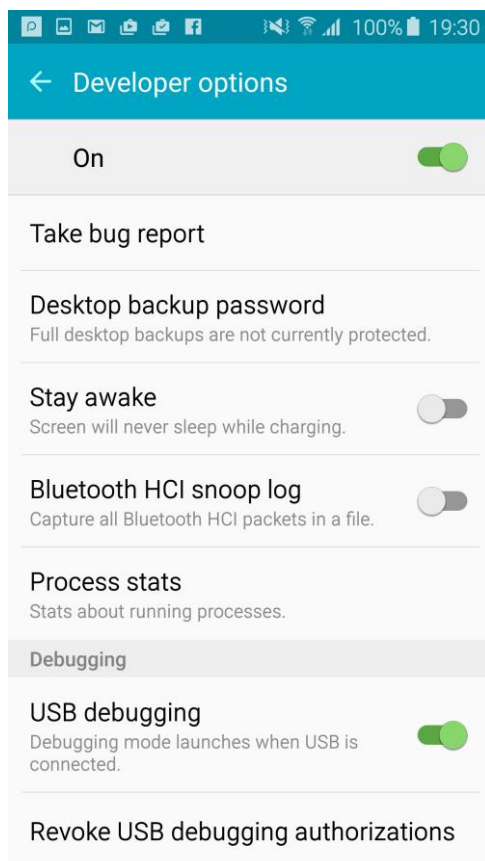
Projektia tehdessä on tärkeää käyttää oikeaa älypuhelimia alustana jolla projektia testataan emulaattorin sijaan, sillä älypuhelimien anturien arvoilla on keskeinen merkitys ohjelman toimimisen kannalta ja emulaattorilla ei luonnollisesti näitä antureita ole.



### 4.6.1 Älypuhelimien asetukset

Älypuhelimesta on aktivoitava Developer options mahdollisuus jotta älypuheliminta voi käyttää Android-sovelluksien testausalustana. Android 4.2 versiossa ja uudemmissa versioissa Developer options on piilotettu ja sen saa auki navigoimalla puhelimesta Settings > About device ja painamalla Build number kenttää 7 kertaa. Älypuhelin kysyy tämän jälkeen halutaanko Developer options ottaa käyttöön. Tämän jälkeen Developer options ilmestyy Settings valikkoon.

Developer options valikossa päällä tulee olla ”On” ja ”USB Debugging” kohdat (Kuva 4). Kun nämä kohdat on valittu päälle ja älypuhelin kytketty USB:n kautta tietokoneeseen, Android kysyy käyttäjältä haluaako se luoda yhteyden tietokoneeseen. Tähän vastataan kyllä.



Kuva 4: Androidin Developer options valikko.

## 4.6.2 Projektin suorittaminen

Projektin suoritusasetukset määritetään Android Studioissa navigoimalla Run > Run/Debug Configurations. Keskeisiä asetuksia ovat Module johon asetetaan suoritettava moduuli ja Deployment Target Options johon määritetään USB Device.

Android Studioissa projektin ”Run” toiminto saattaa antaa virheilmoituksen INSTALL\_FAILED\_UPDATE\_INCOMPATIBLE. Tämä tarkoittaa vain sitä että puhelimelle on asennettu aikaisempi versio ohjelmasta joka tulee poistaa. Sen voi poistaa komentorivillä navigoimalla Android SDK kansioon, sieltä kansioon platform-tools, ja sitten antaa käskyn adb uninstall ”ohjelma-tähän”. Uusimmat Android Studion versiot osaavat suorittaa saman komentorivikäskyn käyttäjän suostumuksella ongelman ilmetessä.

## 4.7 Tehty sovellus

Toimivan Android-sovelluksen aikaansaamiseksi käytettiin apuna IndoorAtlasin omaa esimerkkisovelluspakettia, joka on ladattavissa IndoorAtlasin Developer alueelta. Linkki sivulla vie GitHubiin josta esimerkkisovelluksen voi ladata. Paketti sisältää Simple Example, ImageView, Google Maps – Basic ja Google Maps - Overlay aliohjelmat.

Esimerkkisovelluksesta karsittiin pois turhat esimerkit ja työssä keskityttiin ImageView-aliohjelman kehittämiseen. Tämä ohjelma lataa kartoitetun pohjapiirroksen automaattisesti ApiKey ja ApiSecret arvojen mukaan IndoorAtlasin pilvipalvelusta. Kuva tiedostoista jota projekti sisältää on liitteenä (Liite 1).

### 4.7.1 Java-tiedostot

Ohjelman käynnistyessä Android-älypuhelimella ovat ensimmäisenä suoritettavina Java-tiedostoina tiedostot ListExamplesActivity ja SdkExample. ListExamplesActivity toimii päätiedostona sovellukselle, josta käyttäjä valitsee aliohjelman jonka haluaa suorittaa. SdkExample tiedosto toimii Java interfacena listattavien aliohjelmien otsikolle ja selitykselle.

ImageView aliohjelma koostuu kahdesta Java tiedostosta; ImageViewActivity ja BlueDotView. ImageViewActivity tiedosto toimii tiedostona joka käsittelee karttasovelluksen perustoiminnallisuutta ja BlueDotView sisältää kartalle piirrettävän reaaliaikaisen sijainnin toiminnallisuuden. Kuvissa 5 ja 6 kuvataan Java tiedostojen toimintaa.

```

private IALocationListener mLocationListener = new IALocationListenerSupport() {
    @Override
    public void onLocationChanged(IALocation location) {
        Log.d(TAG, "location is: " + location.getLatitude() + "," + location.getLongitude());

        TextView textView = (TextView) findViewById(R.id.lowerText);
        textView.setText(" Latitude: " + location.getLatitude() + "\n Longitude: " + location.getLongitude());

        if (mImageView != null && mImageView.isReady()) {
            IALatLng latLng = new IALatLng(location.getLatitude(), location.getLongitude());
            PointF point = mFloorPlan.coordinateToPoint(latLng);
            mImageView.setDotCenter(point);
            mImageView.postInvalidate();
        }
    }
};

```

Kuva 5: ImageViewActivity luokan IALocationListener Interface joka suoritetaan aina kun älypuhelimien sijainti muuttuu. Log.d – funktiota käytetään debuggaamiseen joka näyttää muuttuneet pituus- ja leveysasteet Android Studio monitorissa. TextView – funktioilla viedään samaiset arvot Android-älypuhelimien näytön alareunaan. If-loopissa suoritetaan varsinainen sijainnin muuttaminen näytöllä näkyvälle sijainnille. mImageView on instanssi BlueDotView – luokasta joka on määritetty samannimisessä Java-tiedostossa.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_image_view);
    // prevent the screen going to sleep while app is on foreground
    findViewById(android.R.id.content).setKeepScreenOn(true);

    TextView textView = (TextView) findViewById(R.id.upperText);
    textView.setText("A3 Floorplan"); // draws this text to the upper layer of the screen

    mImageView = (BlueDotView) findViewById(R.id.imageView);

    mDownloadManager = (DownloadManager) getSystemService(Context.DOWNLOAD_SERVICE);
    mIaLocationManager = IALocationManager.create(this);
    mFloorPlanManager = IAResourceManager.create(this);

    /* optional setup of floor plan id
       if setLocation is not called, then location manager tries to find
       location automatically */
    final String floorPlanId = "";
    if (!TextUtils.isEmpty(floorPlanId)) {
        final IALocation location = IALocation.from(IARegion.floorPlan(floorPlanId));
        mIaLocationManager.setLocation(location);
    }
}

```

Kuva 6: onCreate metodi joka suoritetaan ImageViewActivity luokan alkuvaiheessa yhden kerran. Metodissa luodaan mm. teksti näytön yläreunaan TextView funktioilla, IALocationManager ja IAResourceManager instanssit ja mahdollisuutena olisi vielä tuoda haluttu pohjapiirros manuaalisesti ohjelmaan floorPlanId muuttujan arvoksi. Koska nykyinen sovellus osaa hakea pohjapiirroksen automaattisesti, tämän kohdan voi jättää huomiotta.

BlueDotView luokkaan sisällytetään SubsamplingScaleImageView kirjasto jonka on tehnyt Dave Morrissey. Kirjasto mahdollistaa karttakuvan zoomauksen ja kirjastoon sisältyy myös kuvan pyörittämisominaisuus jota ei käytetä tässä sovelluksessa.

#### 4.7.2 XML-tiedostot

XML-tiedostoilla määritetään Android-sovelluksen ulkonäkö ohjelmaa suoritettaessa. Sovelluksessa käytetään kahta XML-tiedostoa, activity\_main ja activity\_image\_view. Activity\_main tiedostossa luodaan lista käytettävistä aliohjelmista ja activity\_image\_view tiedostossa määritetään ulkonäkö karttaohjelmalle. Tiedostoissa asetetaan leveydet, korkeudet ja sijainnit käytettäville layouteille, tekstin koot ja fontit sekä etäisyydet näytön reunoista.

### 4.7.3 Gradle skriptit

Projektikansion juureen on asetettu tarvittavat Gradle skriptit joita projekti käyttää sen rakennusvaiheessa. Näillä skripteillä asetetaan ohjelmalle mm. järjestys missä rakennus suoritetaan, kirjastot joita projektiin sisällytetään sekä muut keskeiset asetukset.

Gradle.properties tiedosto sisältää jokaiselle IndoorAtlas sovellukselle identtisen ApiKey:n ja ApiSecret:n jotka on luotu IndoorAtlasin käyttöönottaessa. Local.properties asettaa polun Android Studion käyttämille SDK paketeille. Projektitiedoston avattaessa saattaa Android Studio huomauttaa siitä että polku tässä tiedostossa SDK paketteihin on eri kuin mihin Android Studion SDK paketit ovat asennettu. Tässä tapauksessa Android Studio kehottaa muutoksen tekemiseen tässä tiedostossa joka ei koidu ongelmaksi. Kuvissa 7 ja 8 esiintyy projektin build.gradle tiedostojen sisältöä.



```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:2.1.0'  
    }  
}  
  
allprojects {  
    repositories {  
        jcenter()  
        mavenCentral()  
        maven {  
            url "http://indooratlas-ltd.bintray.com/mvn-public"  
        }  
    }  
}
```

Kuva 7: Build.gradle tiedosto koko projektikansion juuressa. Varsinaiselle ohjelmamoduulille on oma build.gradle tiedostonsa Basic kansiossa. Tiedostossa määritetään käytettävä Gradlen versio sekä Maven riippuvuudet.

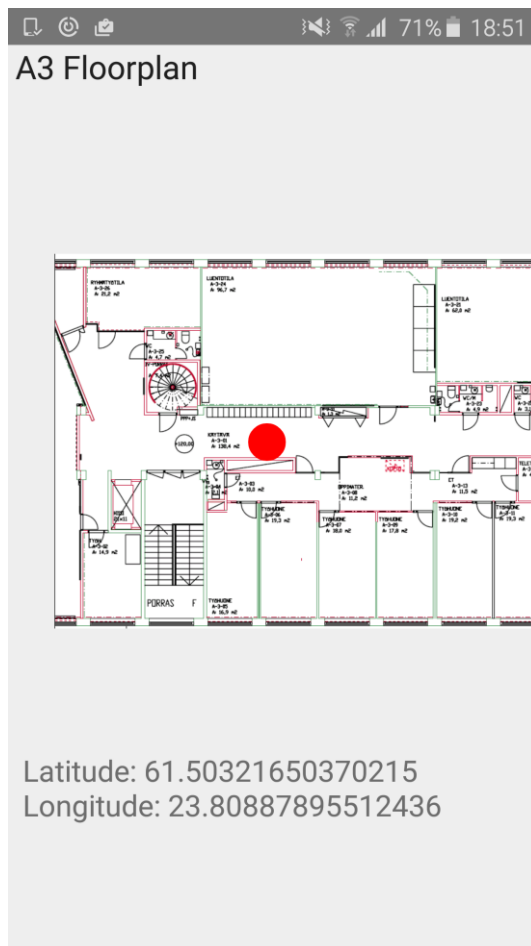
```
dependencies {  
    compile 'com.indooratlas.android:indooratlas-android-sdk:2.1.0@aar'  
    compile 'com.android.support:appcompat-v7:23.1.0'  
    compile 'com.google.android.gms:play-services:8.1.0'  
    compile 'com.squareup.picasso:picasso:2.5.2'  
    compile 'com.davemorrissey.labs:subsampling-scale-image-view:3.2.0'  
    compile 'com.pubnub:pubnub-android:3.7.5'  
    compile 'com.android.support:design:23.0.1'  
}
```

Kuva 8: Build.gradle tiedoston riippuvuudet Basic moduulin juuressa. Tässä tuodaan projektiin tarvittavat kirjastot, mm. IndoorAtlas Android SDK 2.1.0.

#### 4.7.4 Sovelluksen toiminnallisuus

Älypuhelimella suoritettaessa sovellus näyttää käyttäjän sijainnin punaisena pallona kartalla. Ohjelmaa varten kartoitettiin TAMK:n A siiven 3. kerroksen käytävä. Kun sovelluksen käynnistää, pitää käyttäjän olla paikanpäällä kyseisellä käytävällä tai vähintään samassa kerroksessa, sillä Android sovellus ei muuten lataa kyseisen käytävän pohjapiirrosta näkyviin.

Sovellus näyttää käyttäjän sijainnin parhaimmillaan 0-3 metrin tarkkuudella oikeasta sijainnista. Välillä virhe saattaa nousta noin 5 metriin ja punainen piste liikkuu pidemmällä edessä kuin mitä oikeasti ollaan. Tämä johtuu todennäköisimmin siitä että kartoituksessa on käytetty koulun langatonta verkkoa 3G:n tai 4G:n sijasta. IndoorAtlas itse suosittelee kartoitukseen käytettävän juurikin mobiiliverkkoa (IndoorAtlas, Troubleshooting, 2016). 3G verkkoa ei ollut käytettävissä kartoitusta tehdessä sillä liittymä ei sisältynyt koululta saatuun älypuhelimkeen. Kuvassa 9 näytetään sovelluksen toimintaa.



Kuva 9: Sovellus toiminnassa. Punainen pallo näyttää käyttäjän sijainnin ja alla näkyvät puhelimen kartoittavat pysty ja vaaka-asteet. Pohjapiirrosta voi zoomata lähemmäksi tai kauemmaksi sormiliikkeillä.

Punainen pallo voi myös jäädä paikoilleen yhteen paikkaan, jolloin kartoitus lähtee jälleen kulkemaan kun pallon kohtaan liikutaan. Tämä tapahtuu useimmiten silloin kun sovelluksen käynnistää ensimmäistä kertaa. Pallo ilmestyy kartalle usein käytävän eteläpäätyyn, josta paikantaminen on siten hyvä aloittaa. Virheestä huolimatta sovellus antaa vähintäänkin karkean arvion käyttäjän sijainnista.

Paikannuksessa ohjelma on tarkimmillaan kun liikutaan käytävän eteläpäädyistä pohjoispäätä kohti. Aivan käytävän päässä paikannus saattaa mennä päätyyn aiemmin kuin mitä siellä oikeasti ollaan. Tämä voi johtua siitä, että ympäröivät metalliesineet kuten hissit ja kaapit sekä muut elektroniset laitteet häiritsevät geomagneettista kenttää rakennuksen sisällä ja aiheuttavat siten virhettä paikannukseen.

## 4.8 Kehitysmahdollisuudet

Sovelluksella on paljon erilaisia kehitysmahdollisuuksia. Alkuperäisenä tarkoituksena oli kehittää navigointimahdollisuus paikannussovellukseen, mutta tämä jäi tekemättä koska suurin osa työhön kuluneesta ajasta kului toimivan paikannussovelluksen aikaansaamiseen. Toteutettu sovellus toimii osaltaan hyvänä pohjana mahdolliselle tulevaisuuden sovelluskehitykselle.

Nykyinen sovellus toimii vain yhdellä pohjapiirroksella, johon olisi mahdollista tuoda manuaalisesti aina uusia pohjapiirroksia. Täten pohjapiirroksen kuuluisi vaihtua sovelluksessa jota varten olisi kartoitettu esimerkiksi useita kerroksia ja osia rakennuksessa.

Koko TAMK:n sisältävä sisätilapaikannussovellus olisi hyvä toteuttaa suuremmilla pohjapiirroskuvilla kuin mitä tässä työssä on käytetty. Haasteeksi tämänlaisessa sovelluksessa saattaisi tulla vaihtelut eri kerrosten välillä ja se miten tämän vaihtelun saa sujuvasti toimimaan omassa sovelluksessa. Kerroksia ja siipiä on TAMK:n kaltaisessa rakennuksessa suhteellisen paljon joten tämän ominaisuuden luominen sovellukseen on keskeistä.

Navigointiominaisuutta varten olisi myös hyödyllistä luoda ensin mahdollisuus lisätä sovellukseen yksittäisten tilojen tunnistamismenetelmä. Sovellus voisi näyttää ylä- tai alareunassa tilan jossa ollaan, eli esimerkiksi B1 käytävä, luokka A3-21, juhlasali D1-04 ja niin edelleen. Tilat ja niiden sijaintitiedot tulisi listata tietorakenteeseen josta niitä voidaan selata ja siten valita haluttu määränpää.

Yksi mielenkiintoinen kehitysmahdollisuus mainittujen lisäksi voisi olla kahden älypuhelimien käyttäminen samassa sovelluksessa ja yhden käyttäjän sijainnin näkyminen toisen käyttäjän älypuhelimessa. Tällaista ominaisuutta voi kuitenkin olla haasteellista kehittää TAMK:lla sillä koululla on vain niukasti älypuhelimia käytettäväksi tämänkaltaiseen sovelluskehitykseen.



## 5 YHTEENVETO

Magneettinen sisätilapaikannus on innovatiivinen ratkaisu tulevaisuuden sisätilapaikannussovelluksia varten. Se on Wi-Fi verkosta riippumaton tapa saada luotettavaa ja tarkkaa tietoa päätelaitteen sijainnista kartoitetussa rakennuksessa. Sisätilapaikannukselle on tarvetta mm. kauppakeskuksissa ja IndoorAtlasin tarjoama teknologia tekee sovellusten käytöstä helppoa kuluttajalle. Yrityksellä voi olla oma sovellus omiin käyttötarkoituksiin jonka kuluttaja voi ladata omaan älypuhelimensa vaikkapa Google kaupasta.

Opinnäytetyötä tehdessä ilmaantui ongelmia jotka pidentivät työn valmistumista. Opinnäytetyön tekoa varten koululle tilattiin ensin Samsung Galaxy S5 Mini – älypuhelin, jonka sensorit eivät toimineet ohjelmaa testattaessa. IndoorAtlas MapCreatorin kalibroituvaiheessa älypuhelin ei saanut kalibroituksi itseään 100 % asti vaan luku vaihteli 0 % - 92 % välillä. Samsung Galaxy S6:lla kalibrointi onnistuu sekunneissa vaivattomasti.

Oma sovellus saatiin toimimaan älypuhelimella ja sovellus näyttää 1-5 metrin tarkkuudella käyttäjän sijainnin kartalla. Alkuperäisenä tarkoituksena opinnäytetyössä oli jatkokehittää olemassa olevaa sovellusta, mutta tämän sovelluksen toimimattomuuden takia aikaa kului huomattavasti oman toimivan Android Studio projektin aikaansaantiin. Aikaansaatu projekti on kuitenkin hyvä mahdolliselle tulevaisuuden jatkokehitykselle jota TAMK:ssa saatetaan tehdä.

## LÄHTEET

Android Developer. Android Studio Overview. Luettu 20.4.2016  
<http://developer.android.com/tools/studio/index.html>

Community Health Maps. NIH. Luettu 25.4.2016  
<https://communityhealthmaps.nlm.nih.gov/2014/07/07/how-accurate-is-the-gps-on-my-smart-phone-part-2/>

Dr.Dibbs. Why Build Your Java Projects with Gradle Rather than Ant or Maven. Luettu 4.5.2016.  
<http://www.drdoobs.com/jvm/why-build-your-java-projects-with-gradle/240168608>

IndoorAtlas. About us. Luettu 10.4.2016.  
<https://www.indooratlas.com/about-us/>

IndoorAtlas. FAQ. Luettu 1.3.2016.  
<https://www.indooratlas.com/faq/>

IndoorAtlas. Developer Guide. Luettu 13.5.2016.  
<http://docs.indooratlas.com/android/dev-guide/displaying-location-google-maps.html>

IndoorAtlas. IndoorAtlas Android SDK 1.x to 2.x Migration Guide. Luettu 13.5.2016  
<http://docs.indooratlas.com/android/dev-guide/sdk-20-migration-guide.html>

IndoorAtlas Blog. Troubleshooting. Luettu 13.5.2016.  
<https://blog.indooratlas.com/2015/09/24/troubleshooting/>

Oversti. Olio-ohjelmointi. Luettu 26.5.2016.  
<http://oversti.org/wiki/java/oliot>

Vaasan yliopisto. Pitkä harppaus taaksepäin: Javasta C++:n. Luettu 25.5.2016.  
<http://lipas.uwasa.fi/~h79423/c/materiaalit/c02.html>

Wikipedia. Indoor Positioning System. Luettu 27.4.2016  
[https://en.wikipedia.org/wiki/Indoor\\_positioning\\_system](https://en.wikipedia.org/wiki/Indoor_positioning_system)

Wikipedia. GPS. Luettu 27.4.2016  
<https://fi.wikipedia.org/wiki/GPS>

Wikipedia. Magnetic Positioning. Luettu 10.4.2016  
[https://en.wikipedia.org/wiki/Magnetic\\_positioning](https://en.wikipedia.org/wiki/Magnetic_positioning)

## LIITTEET

### Liite 1. Projektin sisältämät tiedostot

