

BLUETOOTH-SOVELLUKSEN KEHITTÄMINEN ANDROIDILLE



Ammattikorkeakoulututkinnon opinnäytetyö

HAMK Visamäki, Tietojenkäsittely

Syksy, 2017

Aatu Isaksson

Tietojenkäsittely
Visamäki

Tekijä	Aatu Isaksson	Vuosi 2017
Työn nimi	Bluetooth-sovelluksen kehittäminen Androidille	
Työn ohjaaja/t	Tommi Lahti, Lauri Salminen	

TIIVISTELMÄ

Tämän opinnäytetyön tavoitteena oli luoda kattava katsaus Bluetoothiin ja Android-mobiilikehitykseen. Työn toimeksiantajana oli Hämeen ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma.

Opinnäytetyön teoriaosuus jakautuu kahteen osaan. Ensimmäisessä osassa esitellään Bluetoothia ja vertaillaan sitä hieman muihin langattomiin tiedonsiirtotekniikoihin. Toisessa osassa kerrotaan olennaista tietoa Android-sovelluskehityksestä ja esitellään työssä käytettyjä työkaluja.

Työn tuloksena on kehitetty Androidille sovellus, jolla voi käydä viestikeskustelua kahden mobiililaitteen välillä Bluetoothia käyttäen. Käytännön osuudessa raportoitii sovelluksen kehitysprosessia ja luotiin syvempi katsaus ohjelmointiin.

Avainsanat Android, Bluetooth, mobiiliohjelmointi

Sivut 21 sivua

Degree Programme in Business Information Technology
Visamäki

Author	Aatu Isaksson	Year 2017
Subject	Bluetooth Software Development for Android	
Supervisors	Tommi Lahti, Lauri Salminen	

ABSTRACT

The goal of this Bachelor's thesis was to create a comprehensive overview of Bluetooth and Android mobile development. The thesis was commissioned by Häme University of Applied Sciences Degree Programme in Business Information Technology.

The theory in this Bachelor's thesis is divided into two sections. The first part includes the introduction of Bluetooth and its comparison to other wireless communication techniques. The second part includes essential information about Android app development, and the introduction of the tools that have been used in this working process.

As a result of this work, an app for Android was developed. The app can be used for communication between two mobile devices using Bluetooth. The practical part describes the app's developing process and takes a deeper look to the programming.

Keywords Android, Bluetooth, mobile development

Pages 21 pages

SISÄLLYS

1	JOHDANTO.....	1
2	BLUETOOTH	1
2.1	Bluetoothista yleisesti	1
2.2	Eri versiot.....	2
2.2.1	Ensimmäiset versiot	2
2.2.2	Bluetooth 4.0 + LE.....	2
2.2.3	Bluetooth 5	3
2.3	Vertailua muihin langattomiin tiedonsiirtotekniikoihin.....	3
2.3.1	WiFi.....	3
2.3.2	IrDA.....	4
3	ANDROID-SOVELLUSKEHITYS.....	4
3.1	Android.....	4
3.2	Java-ohjelmointikieli	5
3.3	Android SDK.....	6
3.4	Activityt	6
3.5	Android Studio	7
3.5.1	Gradle	8
3.5.2	Käyttöliittymä	9
4	BLUETOOTH-SOVELLUS	10
4.1	Projektin luonti.....	11
4.2	Ulkoasu	12
4.3	Bluetoothin käyttö	14
4.3.1	Laitteiden etsintä.....	15
4.4	Laitteeseen yhdistämien	16
4.5	Viestien lähetys ja vastaanottaminen	19
5	YHTEENVETO	VIRHE. KIRJANMERKKIÄ EI OLE MÄÄRITETTY.
	LÄHTEET.....	22

1 JOHDANTO

Langattomia tiedonsiirtotekniikoita on nykyään useita erilaisia, ja ne toimivat eri kantamilla. Yksi suosituimmista lyhyen kantaman tekniikoista on Bluetooth. Bluetoothin käyttö on yleistynyt vuosi vuodelta, ja tänä päivänä sitä käytetään erittäin useissa laitteissa, jopa kahvinkeitimissä on Bluetooth.

Valitsin tämä aiheekseni, koska mobiiliohjelmointi erityisesti Androidille kiinnostaa, ja Bluetooth on kiinnostava, ajankohtainen ja yleinen langaton tiedonsiirtoyhteys, joten päätin tehdä Bluetoothia käyttävän sovelluksen.

Opinnäytetyöni tavoitteena on kertoa lukijalle Bluetoothista ja hieman myös muista langattomista tiedonsiirtotekniikoista. Työn pääpaino on älypuhelimien välinen Bluetooth-yhteys, erityisesti kahden Android-puhelimen välillä.

Käytännön osuudessa tavoitteena on kehittää Android Studiolla puhelimelle sovellus, joka käyttää Bluetoothia. Sovelluksella voi etsiä Bluetooth-laitteita ja yhdistää toiseen Android-puhelimeen. Kun yhteys on saatu, niin käyttäjät voivat lähettää puhelimilla viestejä toisilleen Bluetoothin välityksellä.

2 BLUETOOTH

2.1 Bluetoothista yleisesti

Bluetooth mahdollistaa laitteiden langattoman kommunikoinnin lähietäisyydellä. Bluetooth-yhteyden välityksellä voi esimerkiksi lähettää tietoja, lähettää ja vastaanottaa ääntä ja jakaa internetyhteyttä. Se juontaa juurensa vuoteen 1994, jolloin ruotsalainen matkapuhelinyritys Ericsson rupesi tutkimaan, kuinka matkapuhelimien ja niiden oheislaitteiden välillä voitaisiin siirtää dataa langattomasti. Ericssonin tutkimusten ansiosta vuonna 1998 perustettiin Bluetooth SIG (Special Interest Group), johon kuuluu nykyisin yli 15 000 eri yritystä. Bluetooth sai nimensä viikinkikuninkaan, Harald Sinihampaan mukaan, ja logossa on yhdistetty skandinaaviset riimut Hagall ja Berkanan. (Absoluteastronomy n.d.)

Bluetooth käyttää radioaaltoja toimiakseen. Se toimii taajuusalueella 2400-2480 Mhz. Bluetoothin kantama vaihtelee noin viiden ja 100 metrin välillä lähettimen tason mukaan, joita on kolme (kuva 1). Lähettimen taso määräytyy lähetystehon mukaan. (Absoluteastronomy n.d.)

Class	Maximum permitted power		Range (m)
	(mW)	(dBm)	
Class 1	100	20	~100
Class 2	2.5	4	~10
Class 3	1	0	~5

Kuva 1. Bluetooth-lähettimen kantavuus eri tasoissa (Absoluteastronomy n.d).

2.2 Eri versiot

2.2.1 Ensimmäiset versiot

Bluetoothin ensimmäinen versio oli Bluetooth 1.0. Siinä oli paljon ongelmia, ja valmistajilla oli vaikeuksia laitteiden ja Bluetoothin yhteensopivudessa. Nämä ongelmat korjattiin kuitenkin nopeasti, ja versiossa 1.2 tuli paljon parannuksia, kuten nopeampi laitteiden etsiminen ja laitteeseen yhdistäminen, korkeampi siirtonopeus (721 kbit/s) ja parempi äänenlaatu. (Ebay 2008.)

Bluetooth 2.0 + EDR julkaistiin vuonna 2004. Suurin ero edelliseen versioon oli Enhanced Data Rate (EDR), joka mahdollisti paljon korkeamman siirtonopeuden, jopa 2.1 Mbit/s. EDR toi mukanaan myös muita parannuksia, kuten pienemmän akunkulutuksen. (Ebay 2008.)

Bluetooth 3.0 + HS toi mukanaan erittäin korkean siirtonopeuden ja paremman virranhallinnan. Nopeus voi olla korkeimmillaan jopa 24 Mbit/s. (Majorhifi 2016.)

Specifications	1.1	1.2	2.0 + EDR	2.1 + EDR	3.0 +HS	4.0
Adopted	2002	2005	2004	2007	2009	2010
Transmission Rate	723.1 kbps	723.1 kbps	2.1 Mbps	3 Mbps	24 Mbps	25 Mbps
Standard PAN Range	10 m	10 m	10 m	10 m	10 m	50 m
Improved Pairing (without a PIN)				Yes	Yes	Yes
Improved Security		Yes	Yes	Yes	Yes	Yes
NFC Support			Yes	Yes	Yes	Yes

Kuva 2. Bluetooth versioiden vertailua (Quora 2016).

2.2.2 Bluetooth 4.0 + LE

Tämä versio toi mukanaan Low-energy (LE) -teknologian. Tämä tarkoittaa sitä, että esimerkiksi langattomissa kuulokkeissa akku kestää huomatta-

vasti kauemmin, jos se käyttää Low-energy -teknologiaa. Teknologia perustuu siihen, että kahden laitteen välinen Bluetooth-yhteys menee horrostilaan, jos isoa määrää dataa ei lähetetä. Bluetooth 4.0 on esimerkiksi Applen iPhone 4S -puhelimissa. (Laptopmag 2012.)

2.2.3 Bluetooth 5

Bluetooth 5 julkaistiin joulukuussa 2016. Se sisältää jopa neljä kertaa pidemmän kantaman, kaksi kertaa nopeamman tiedonsiirron ja kahdeksan kertaa suuremman lähetyskapasiteetin. Näiden parannusten tavoitteena on parantaa Bluetoothin käyttöä Internet of Things (IoT) -laitteissa. Versio on vielä niin uusi, ettei sitä ole vielä laitteissa, mutta luultavasti muutaman kuukauden kuluessa ensimmäiset Bluetooth 5 -laitteet julkaistaan. Bluetooth (2016.)

2.3 Vertailua muihin langattomiin tiedonsiirtotekniikoihin

2.3.1 WiFi

WiFiä eli langatonta lähiverkkoa alettiin tutkia jo 80-luvun puolivälissä, eli paljon aikaisemmin kuin Bluetoothia, mutta ne kuitenkin julkaistiin lähes samana vuonna. Kummassakin on paljon eri versioita, mutta versioiden nimeämistavat vaihtelevat. WiFi-versiot nimetään IEEE 802.11 standardin mukaisesti niin, että numero-osan loppuun lisätään kirjain, kuten 802.11.a. Bluetoothissa versiot taas nimetään lisäämällä vain version numero Bluetooth-sanan jälkeen, kuten Bluetooth 2.0. (Engineersgarage 2012.)

Kummatkin tekniikat käyttävät radioaaltoja toimiakseen, mutta taajuusalueet vaihtelevat hieman. Bluetooth toimii 2.4GHz taajuudella, mutta WiFi toimii 2.4, 3.6 ja 5 GHz taajuuksilla. Myös tiedonsiirtonopeudet vaihtelevat. Tällä hetkellä yleisimmät Bluetooth-versiot 4.0-4.2 saavuttavat parhaimmillaan 25mbps, kun taas Wifi Direct ylittää jopa 250mbps siirtonopeuteen. (Engineersgarage 2012.)

Tekniikoiden kantavuuden eroja on himan vaikea vertailla, sillä Bluetoothissa kantamaan vaikuttaa lähettimen luokka, ja WiFissä taas WiFi:n versio. Yleisesti ottaen kuitenkin WiFissä on parempi kantavuus. Tämä johtuu myös siitä, että WiFissä käytetään yleensä antennia, joka lisää kantavuutta. (Engineersgarage 2012.)

Bluetoothin välityksellä enintään seitsemän laitetta voivat kommunikoida keskenään. WiFissä taas laitteita voi olla useampia, riippuen reitittimestä. WiFissä turvallisuuteen on panostettu hieman enemmän. Paremman turvallisuuden ja pidemmän kantaman takia WiFi kuluttaa myös enemmän virtaa. (Engineersgarage 2012.)

Myös käyttötarkoituksessa on paljon eroja. WiFiä käytetään yleensä tietokoneiden, reitittimien, kameroiden, tulostimien ja pelikonsolien yhdistämisessä toisiinsa tai internetiin. Bluetoothia taas käytetään yleensä näppäimistön, hiiren ja kuulokkeiden yhdistämisessä tietokoneeseen, tai langattoman ohjaimen yhdistämisessä pelikonsoliin. (Engineersgarage 2012.)

2.3.2 IrDA

IrDA eli Infrared Data Association oli suosittu 90-luvun puolessavälissä, mutta suosio alkoi hiipua Bluetoothin ja WiFin yleistyessä. Tämä johtui siitä, että IrDA käyttää valoa toimiakseen, joten valon pitää osoittaa suoraan toiseen laitteeseen, jotta yhteys toimii. Tekniikassa on kuitenkin myös hyviä puolia. IrDA pystyy saavuttamaan Bluetoothia nopeamman tiedonsiirtonopeuden, jopa 1Gbps. Siinä on myös parempi suojaus kuin Bluetoothissa, sillä laitteiden pitää olla maksimissaan noin kahden metrin etäisyydellä toisistaan ja osoittaa toisiaan, jotta yhteys toimii. Tästä syystä ulkopuolisten on erittäin vaikea tunkeutua yhteyteen. IrDA kuluttaa myös hie- man vähemmän virtaa kuin Bluetooth, mutta IrDA-yhteydessä voi olla vain kaksi laitetta samaan aikaan, toisin kuin Bluetoothissa voi olla seitsemän laitetta samassa yhteydessä. IrDA:ta käytetään yleensä samoihin tarkoituksiin kuin Bluetoothia, eli kännyköiden, tulostimien ja tietokoneiden yhdistämiseen keskeen, mutta käyttömahdollisuuksia rajoittaa se, että laitteiden tulee olla todella lähellä toisiaan ja infrapuna-valon pitää osoittaa toiseen laitteeseen. (Rfwireless-world 2012.)

3 ANDROID-SOVELLUSKEHITYS

3.1 Android

Android on Googlen omistama avoimen lähdekoodin Linux-pohjainen käyttöjärjestelmä. Android-laitteita on monenlaisia, mutta yleisimpiä ovat puhelimet, tabletit ja rannekellot. Se on tänä päivänä suosituin puhelimen käyttöjärjestelmä. Monet valmistajat käyttävät Androidia puhelimissaan, ja tutuimpia näistä ovat Samsung, Huawei, Sony, Lenovo, HTC ja LG. Vaikka näissä puhelimissa kaikissa on Android, se ei tarkoita sitä, että kaikki olisivat samanlaisia. Android on avoimen lähdekoodin käyttöjärjestelmä, eli valmistajat voivat vapaasti muokata siitä oman näköisensä, joten Android näyttää erilaiselta esimerkiksi Samsungin ja Huawein puhelimissa. Google julkaisee 1–2 uutta Android-versiota joka vuosi. Versiot nimetään aakkosjärjestyksessä, ja ne saavat yleensä jonkun makeisen nimen (Kuva 3). (Androidpit 2016.)



Kuva 3. Androidin versiot (Recombu 2016).

Android on suosittu monesta syystä, mutta yksi yleisin syy on valinnanvapaus, jonka Android tarjoaa. Jos käyttäjä esimerkiksi haluaa puhelimen, jossa on iOS-käyttöjärjestelmä, sinun on pakko ostaa iPhone, mutta Android puhelimia saat monelta eri valmistajalta, joten voit valita kalliin tai halvan puhelimen. Käyttäjän on myös helppo muokata käyttöjärjestelmää omien tarpeidensa mukaisesti, sillä Android tarjoaa paljon erilaisia vaihtoehtoja, joilla voi vaikuttaa käyttöjärjestelmän ulkonäköön ja toimintaan. Androidin mukana tulee Google Play Store, josta käyttäjä voi ostaa sovelluksia, pelejä, kirjoja, elokuvia, musiikkia ja paljon muuta. Google Play Storea pidetäänkin Androidin yhtenä suurimmista vahvuuksista. (Androidpit 2016.)

3.2 Java-ohjelmointikieli

Java on Sun Microsystemsin kehittämä oliopohjainen ohjelmointikieli. Se sai alkunsa, kun James Gosling aloitti Java-projektin vuonna 1991. Projekti johti siihen, että Sun Microsystems julkaisi ensimmäisen Java-version vuonna 1995. Tänä päivänä Java on erittäin suosittu ohjelmointikieli, sillä voi ohjelmoida esimerkiksi Android-sovelluksia, pelejä ja Web-sovelluksia. (Tutorialspoint 2017.)

Kuten muillakin ohjelmointikielillä, myös Javalla on oma rakenne, syntaksisäännöt ja ohjelmointiparadigma. Java pohjautuu C-kieleen, joten sen

syntaksisäännöt muistuttavat paljon C:n sääntöjä. Javan mukana tulee myös virtuaalikone (Java Virtual Machine), joka ajaa ohjelmat. Sen avulla sama Java-ohjelma voidaan ajaa monella eri alustalla.

3.3 Android SDK

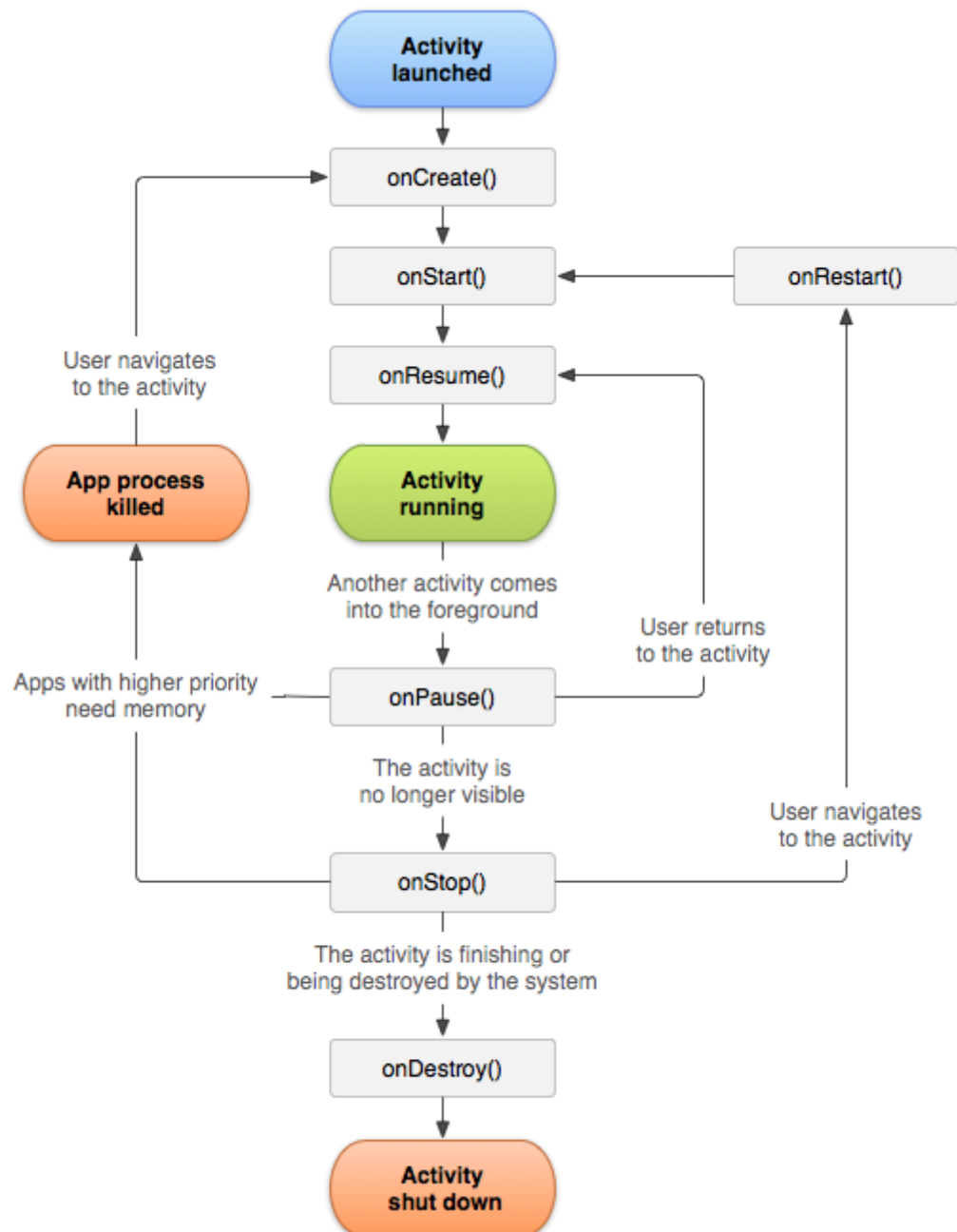
Android SDK on kehitystyökalu, jota tarvitaan, jotta Androidille pystytään tekemään sovelluksia. Jokaisen Android-version mukaan julkaistaan uusi SDK-versio. Se sisältää esimerkki-projekteja lähdekoodin kanssa, emulaattorin ja tarvittavia kirjastoja Android-sovellusten rakentamiseen. Sovellukset kirjoitetaan Java-ohjelmointikielellä. (Webopedia 2017.)

3.4 Activityt

Activityt ovat yksi Android-sovelluksen perusteista. Ne hoitavat käyttäjän vuorovaikutuksen sovelluksen kanssa. Activity tuottaa laitteen näytölle ikkunan, johon sovellus piirtää käyttöliittymän. Ikkuna täyttää yleensä laitteen koko näytön, mutta se voi olla myös pienempi. Tavallisesti yksi activity luo yhden kuvaruudun sovellukselle. Esimerkiksi yksi sovelluksen activity voi tuottaa asetukset-kuvaruudun, ja toinen activity voi tuottaa kirjautumis-kuvaruudun. (Developer Android n.d.a.)

Useimmat sovellukset omaavat monia kuvaruutuja, joka tarkoittaa, että niissä on myös monia activityjä. Yleensä yksi sovelluksen activity on määriteltä main activityksi, joka on ensimmäinen kuvaruutu, joka aukeaa, kun käyttäjä avaa sovelluksen. Jokainen activity voi käynnistää uuden activityn. Tällöin activityn tila usein muuttuu. (Developer Android n.d.a.)

Activityillä on myös oma elinkaari. Tämä tarkoittaa, että activityn tilaa voi vaihtaa, kun käyttäjä poistuu ja palaa activityyn. Activity-luokka tarjoaa metodeja, joiden avulla activitylle kerrotaan sen tilan vaihtumisesta. Näitä metodeja ovat onCreate, onStart, onResume, onPause, onStop ja onDestroy. Esimerkiksi sovelluksessa, joka toistaa videoita, voidaan näiden metodien avulla pysäyttää video, kun käyttäjä vaihtaa toiseen sovellukseen. Sitteen kun käyttäjä palaa sovellukseen, videota voidaan aloittaa toistamaan samasta kohtaa, johon se jäi, kun käyttäjä vaihtoi sovellusta. Tekemällä hyvää työtä activityn tilojen vaihdoksessa tulee sovelluksesta paljon vakaampi. Tällöin vältetään esimerkiksi siltä, että sovellus kaatuu, jos laitteeseen soitetaan samaan aikaan, kun sovellusta käytetään. (Developer Android n.d.b.)



Kuva 4. Activityn elinkaari (Developer Android n.d.c).

3.5 Android Studio

Android Studio on Googlen kehittämä virallinen kehitysympäristö Androidille. Se syrjäytti Eclipse Android Development Toolsin Googlen ensisijaisena kehitysympäristönä Androidille. Android Studio tarjoaa kattavan ja älykkään koodieditorin, nopean ja paljon ominaisuuksia sisältävän emulaattorin ja vakaan sekä monipuolisen build-työkalun. Android Studiolla pystyy kehittämään sovelluksia puhelimille, tableteille, Android Wear -laitteille ja Android TV -laitteille. Se tarjoaa myös emulaattorin näihin kaikkiin laitteisiin, joten koodin testaaminen käy nopeasti ja vaivattomasti. (Android developer 2016.)

3.5.1 Gradle

Gradle on suosittu avoimen lähdekoodin automatoitu build-työkalu, joka tekee työskentelemisestä helpompaa isojen projektien kanssa. Se julkaistiin vuonna 2007, ja pohjautuu sen edeltäjiin Apache Maveniin ja Apache Anttiin. Gradle käyttää Groovy ja Java -ohjelmointikieliä XML-kielen sijasta. Tästä syystä Gradlen koodi on paljon lyhyempää ja selkeämmän näköistä (Kuva 4). (Sitepoint 2016.)

Maven

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Gradle

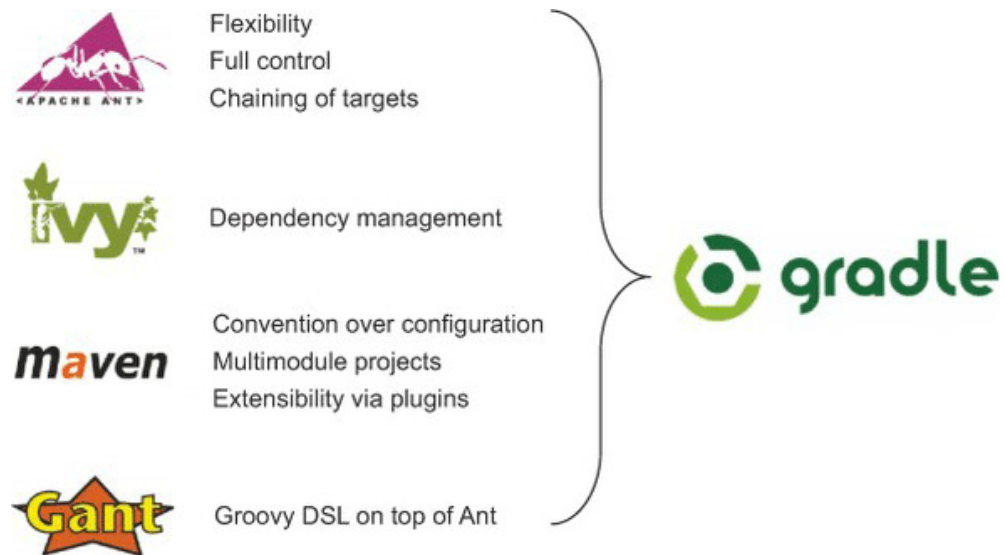
```
apply plugin: 'java'
group = 'com.mycompany.app'
archivesBaseName = 'my-app'
version = '1.0-SNAPSHOT'

repositories {
  mavenCentral()
}

dependencies {
  testCompile 'junit:junit:4.11'
}
```

Kuva 5. Koodin pituuden vertailua, kun samaan tavoitteeseen on päästy Gradlilla ja Mavenilla (Drdobbs 2014a).

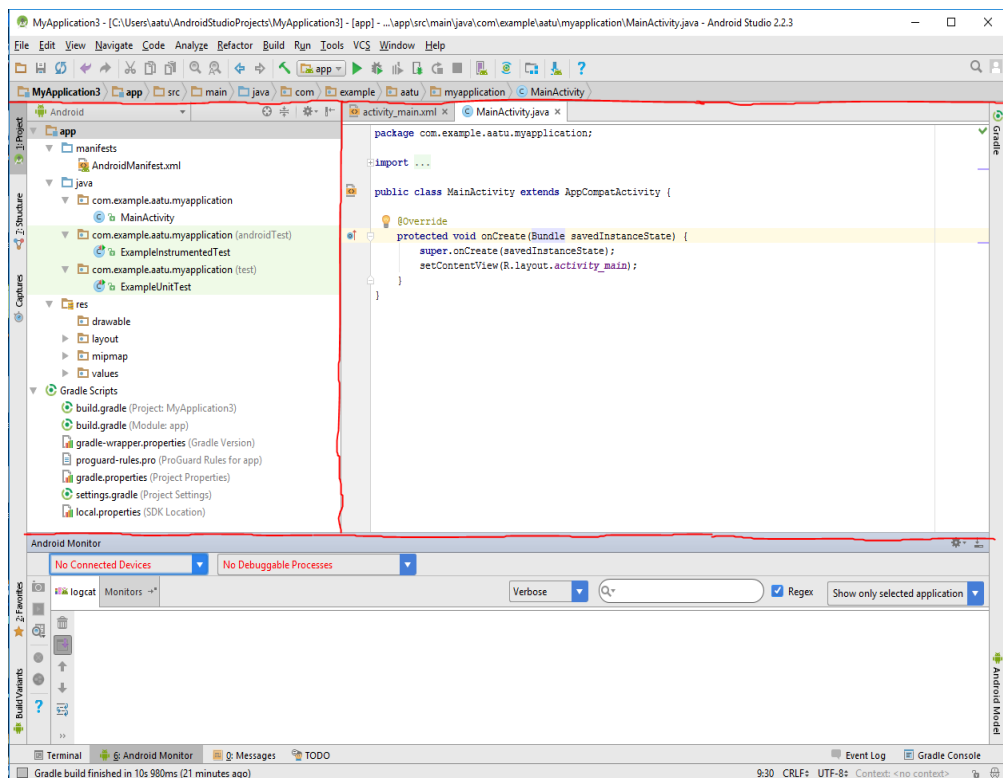
Gradle nopeuttaa projektin rakennus prosessia jättämällä väliin ajan tasalla olevien projektin osien uudelleen suorittamiset. Se myös helpottaa uusien kirjastojen lisäämistä projektiin käyttämällä vain yhtä riviä koodia. Monet pitävät Gradlea tällä hetkellä parhaana build-työkaluna, sillä siitä löytyy muiden työkalujen parhaat ominaisuudet (Kuva 5). (Sitepoint 2016.)



Kuva 6. Havainnollistava kuva Gradlen ja muiden build-työkalujen ominaisuuksista (Drdobbs 2014b).

3.5.2 Käyttöliittymä

Android Studio käyttöliittymä on melko samanlainen kuin muillakin kehitysympäristöillä. Vasemmalla puolella pystyy selailemaan tiedostoja ja siellä näkyy myös projektin rakenne (Kuva 6). Oikealla puolella puolestaan on koodieditori ja sovelluksen käyttöliittymän suunnittelutyökalu. Näiden osien alapuolelta löytyy monitori, johon saa esimerkiksi virheilmoitukset, jos sovelluksen ajossa löytyy virheitä. Ylhäältä löytyy navigointipalkki ja työkalurivi. Navigointipalkki näyttää polun, missä osassa projektin rakennetta olet. Työkaluriviltä onnistuu esimerkiksi SDK-versioiden valinta, emulaattorien valinta ja sovelluksen ajaminen.



Kuva 7. Android Studion käyttöliittymä.

4 BLUETOOTH-SOVELLUS

Opinnäytetyön käytännön osuudessa kehitettiin Bluetoothia käyttävä sovellus Android-puhelimelle Android Studiolla. Käytännön osuudessa seurattiin Hông Thái:n tekemää tutoriaalia (devexchanges 2016). Kun sovelluksen käynnistää, niin sovellus kysyy lupaa kytkeä laitteen Bluetooth päälle, jos luvan antaa, niin sovellus ja Bluetooth käynnistyvät. Tällöin aukeaa sovelluksen perusnäky, josta voi laittaa Bluetooth-näkyvyyden päälle, ja etsiä muita Bluetooth-laitteita. Kun Etsi laitteita -nappia painaa, niin aukeaa uusi ikkuna, johon listautuu kyseisen puhelimen laiteparit ja uudet löydetyt Bluetooth-laitteet. Näistä laitteista voi painamalla valita laitteen, johon Bluetooth-yhteys muodostetaan. Yhteyden muodostamisen jälkeen sovellus palaa perusnäkyyn, jossa voi kirjoittaa viestin ja lähettää sen nappia painamalla.

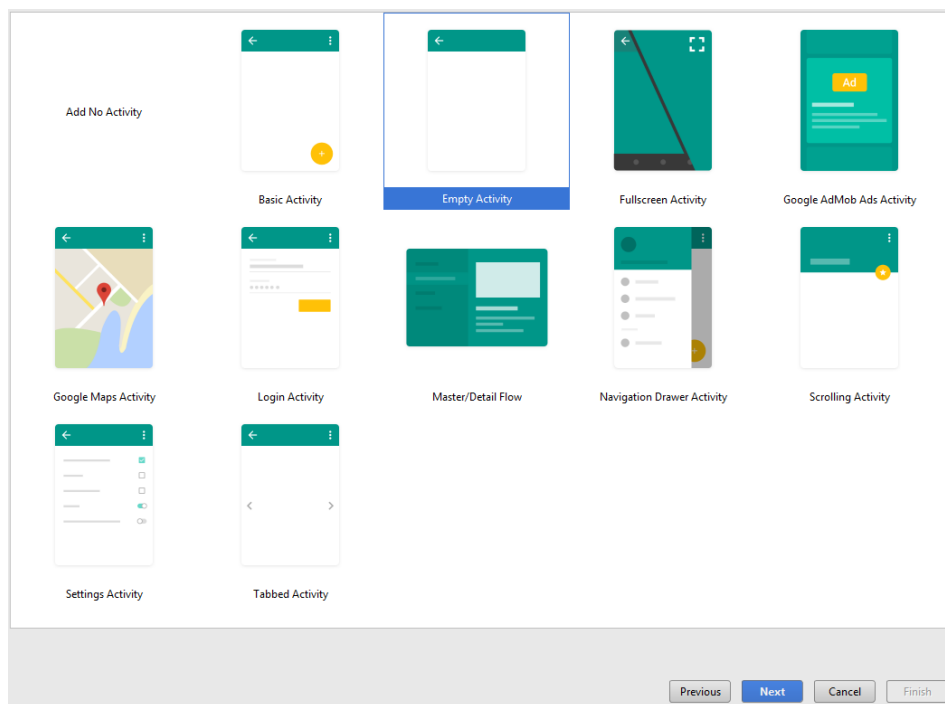
Sovelluksen luonnin yhteydessä valittiin alin Android-puhelimen versio, jossa sovellus toimii. Valitsin alimmaksi versioksi Android 4.0 (IceCreamSandwich), joten toimiakseen sovelluksen tulee olla kahdessa Android-puhelimessa, joissa käyttöjärjestelmänä on Android 4.0 tai uudempi versio. Testilaitteena sovelluskehityksessä oli Honor 7 -puhelin, jossa oli Android 6.0, ja HTC One X+, jossa oli Android 4.2.2.



Kuva 8. Sovelluksen perusnäkyvä.

4.1 Projektin luonti

Projekti luotiin valitsemalla Android Studiosta New project. Sitten projektille annetaan nimi, ja valitaan alin Android-versio, joka oli 4.0. Android Studio antaa arvion, kuinka monella Android-puhelimella sovellus toimii, kun alinta Android-versiota valitaan. Valitsemalla 4.0 Android Studio ilmoittaa, että sovellus toimii noin 97.4 % laitteista, jotka ovat aktiivisia Google Play Storessa. Tämän jälkeen valitaan ensimmäinen Activity sovellukselle. Android Studio tarjoaa muutamia valmiita activityja, kuten kirjautumis-activityn tai Google Maps -activityn (Kuva 8). Tähän sovellukseen valittiin tyhjä activity, jolloin sovelluksessa on valmiiksi oletusulkoasu Relative Layout, ja MainActivity-luokka luotu.



Kuva 9. Android Studion valmiit activity-pohjat.

4.2 Ulkoasu

Hyviä ulkoasu vaihtoehtoja olivat Linear Layout ja Relative Layout. Linear Layoutissa objektit linjataan yksitellen joko vaakasuuntaan tai pystysuuntaan. Relative Layoutissa taas objektit ovat suhteessa toisiinsa. Päädyn Linear Layoutiin, koska pienen kokeilun jälkeen napit ja tekstit asettuivat paremmin paikalleen. Niiden liikuttelu XML-tiedostosta oli myös helpompaa, kun käytössä oli Linear Layout.

Sovelluksessa on kaksi layout-tiedostoa. Toinen tulee näkyviin, kun sovelluksen käynnistää. Toinen taas tulee näkyviin, kun Etsi laitteita -nappia painaa. Tällöin aukeaa uusi ikkuna eli dialogi. Dialogilla on myös käytössään Linear Layout. Dialogin tehtävä sovelluksessa on listata laitteen laiteparit ja löydetyt Bluetooth-laitteet.



Kuva 10. Sovelluksen näkymä, kun dialogi on auki.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Laiteparit"
        android:id="@+id/textView" />

    <ListView
        android:id="@+id/list_parit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Löydetyt laitteet" />

    <ListView
        android:id="@+id/list_loydetyt"
        android:layout_width="match_parent"
        android:layout_height="200dp" />

    <Button
        android:id="@+id/ButtonPoistu"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Poistu" />

</LinearLayout>

```

Kuva 11. Dialog_layout.XML-tiedoston sisältö.

4.3 Bluetoothin käyttö

Jotta sovellus voi käyttää Bluetoothia, tulee sovellukselle antaa lupa AndroidManifest.XML-tiedostossa. Lupa BLUETOOTH antaa sovellukselle luvan käydä Bluetooth-kommunikointia, kuten lähettää viestejä Bluetoothiin välityksellä. Lupa BLUETOOTH_ADMIN antaa taas luvan esimerkiksi muiden Bluetooth-laitteiden etsimiseen.

```

<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>

```

Kuva 12. Bluetooth luvat AndroidManifest.XML-tiedostossa.

Käynnistyessään sovellus alustaa Bluetooth-adapterin ja tarkastaa tukeeko laite Bluetoothia. getDefaultAdapter()-metodi kutsuu laitteen Bluetooth-adapteria. Jos adapteria ei löydy, sovellus ei lähde käyntiin. Tarkistus sijoitetaan sovelluksen onCreate-metodiin, joka käynnistyy ensimmäisenä, kun sovellus käynnistetään.

```
bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
if (bluetoothAdapter == null) {
    Toast.makeText(this, "Bluetooth is not available!", Toast.LENGTH_SHORT).show();
    finish();
}
```

Kuva 13. Tarkistetaan tukeeko laite Bluetoothia.

Sovellus tarkastaa käynnistyksen jälkeen, onko laitteen Bluetooth päällä onStart-metodissa, joka käynnistyy heti onCreate-metodin jälkeen. isEnabled()-metodi tarkastaa onko Bluetooth päällä, jos Bluetooth ei ole päällä luodaan Intent, joka kysyy käyttäjältä halutaanko Bluetooth kytkeä päälle.

```
if (!bluetoothAdapter.isEnabled()) {
    Intent enableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(enableIntent, REQUEST_ENABLE_BLUETOOTH);
}
```

Kuva 14. Bluetoothin päälle kytkeminen.

Jotta puhelimet pystyvät löytämään toisensa Bluetoothin avulla, tulee Bluetoothin näkyvyyden olla päällä. Sovelluksessa BT näkyvyys-nappia painamalla luodaan Intent, joka asettaa laitteen näkyväksi. Laite on näkyvissä oletuksena 120 sekuntia, mutta aikaa pystyy pidentämään tai lyhentämään EXTRA_DISCOVERABLE_DURATION-ekstran avulla.

```
public void Näkyvyys (View v) {
    Intent discoverableIntent =
        new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
    discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 300);
    startActivity(discoverableIntent);
}
```

Kuva 15. BT näkyvyys -napin sisältö.

4.3.1 Laitteiden etsintä

Muiden Bluetooth-laitteiden etsintä aloitetaan yksinkertaisesti kutsumalla Bluetooth-adapterin startDiscovery()-metodia, jota kutsutaan painamalla Etsi laitteita-nappia. Jotta löydettyt laitteet saadaan näkyviin, sovelluksessa on BroadcastReceiver, joka vastaanottaa laitteen lähettämiä viestejä. Esimerkiksi kun Bluetooth-laitteiden etsintä on päällä ja uusi laite löytyy, niin sovellus saa ACTION_FOUND-viestin. Tämän jälkeen löydetyistä laitteesta voidaan luoda olio. Olion avulla voidaan kutsua getName() ja getAddress()

-metodeja, jotka palauttavat löydetyt laitteen nimen ja osoitteen. Sovelluksessa löydetyt laitteet lisätään taulukkoon, jos ne eivät ole jo laitepareja käytössä olevan laitteen kanssa, jos laitteita ei löydy yhtään, sekin ilmoitetaan taulukossa.

```
private final BroadcastReceiver discoveryFinishReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();

        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            if (device.getBondState() != BluetoothDevice.BOND_BONDED) {
                discoveredDevicesAdapter.add(device.getName() + "\n" + device.getAddress());
            }
        } else if (BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)) {
            if (discoveredDevicesAdapter.getCount() == 0) {
                discoveredDevicesAdapter.add("Ei löydettyjä laitteita");
            }
        }
    }
};
```

Kuva 16. BroadcastReceiver:in sisältämä koodi.

4.4 Laitteeseen yhdistämien

Sovelluksen ChatController-luokassa on kolme eri säiettä: AcceptThread, ConnectThread ja ReadWriteThread. AcceptThreadin työnä on odottaa, jos joku haluaa yhdistää laitteeseen. ConnectThreadin työnä on yrittää saada yhteys toiseen laitteeseen. ReadWriteThreadin työnä on ylläpitää laitteiden välistä kommunikointia.

Toiseen laitteeseen yhdistäminen sovelluksella vaatii sen, että myös kohde laitteessa on sama sovellus. Yhdistäminen alkaa, kun dialogista, johon laiteparit ja löydetyt laitteet ovat listattu, valitaan painamalla jokin laite. Listassa lukee kunkin laitteen kohdalla laitteen nimi ja MAC-osoite. Sovellus tekee näistä yhden stringin, ja sen jälkeen substringin, jossa jäljelle jää vain MAC-osoite. Tämän jälkeen kutsutaan ConnectToDevice-metodia, jossa on parametrina MAC-osoite.

ConnectToDevice-metodi luo uuden BluetoothDevice olion, jossa on kohde laitteen MAC-osoite. Sitten metodi kutsuu ChatController-luokasta connect-metodia ja antaa tälle parametrina olion.

```

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        bluetoothAdapter.cancelDiscovery();
        String info = ((TextView) view).getText().toString();
        String address = info.substring(info.length() - 17);
        connectToDevice(address);
        dialog.dismiss();
    }
});
private void connectToDevice(String deviceAddress) {
    bluetoothAdapter.cancelDiscovery();
    BluetoothDevice device = bluetoothAdapter.getRemoteDevice(deviceAddress);
    chatController.connect(device);
}

```

Kuva 17. Laitteen valinnan ja ConnectToDevice-metodin koodit.

Connect-metodi käynnistää ConnectThread-säikeen, joka yrittää yhdistää vapaana olevaan Bluetooth socketiin uniikin UUID-stringin avulla (kuva 18). Toisessa laitteessa on käynnissä tällöin AcceptThread-säie, joka kuuntelee tulevia yhteyspyyntöjä (kuva 19). Jos tulevassa yhteyspyynnössä on sama UUID-stringi, kuin vastaanottavassa laitteessa, yhteys hyväksytään. Tällöin yhdistävä laite yhdistää vastaanottavan laitteen serveri socketiin, ja molemmissa laitteissa käynnistyy ReadWriteThread-säie, jossa laitteiden yhteiseen socketiin alustetaan viestivirta molempiin suuntiin.

```

// Käynnistyy kun yritetään luoda yhteyttä
private class ConnectThread extends Thread {
    private final BluetoothSocket socket;
    private final BluetoothDevice device;

    public ConnectThread(BluetoothDevice device) {
        this.device = device;
        BluetoothSocket tmp = null;
        try {
            tmp = device.createInsecureRfcommSocketToServiceRecord(MY_UUID);
        } catch (IOException e) {
            e.printStackTrace();
        }
        socket = tmp;
    }

    public void run() {
        setName("ConnectThread");

        // Laitteiden etsintä otetaan pois päältä, jotta yhteys ei hidastu
        bluetoothAdapter.cancelDiscovery();

        // Yhdistetään bluetooth sockettiin
        try {
            socket.connect();
        } catch (IOException e) {
            try {
                socket.close();
            } catch (IOException e2) {
            }
            connectionFailed();
            return;
        }
    }
}

```

Kuva 18. ConnectThread-säie.

```

// On käynnissä, kun kuunnellaa tulevia yhteyspyyntöjä
private class AcceptThread extends Thread {
    private final BluetoothServerSocket serverSocket;

    public AcceptThread() {
        BluetoothServerSocket tmp = null;
        try {
            tmp = bluetoothAdapter.listenUsingInsecureRfcommWithServiceRecord(APP_NAME, MY_UUID);
        } catch (IOException ex) {
            ex.printStackTrace();
        }
        serverSocket = tmp;
    }

    public void run() {
        setName("AcceptThread");
        BluetoothSocket socket;
        while (state != STATE_CONNECTED) {
            try {
                socket = serverSocket.accept();
            } catch (IOException e) {
                break;
            }

            // Jos yhteys hyväksytään
            if (socket != null) {
                synchronized (ChatController.this) {
                    switch (state) {
                        case STATE_LISTEN:
                        case STATE_CONNECTING:
                            // Käynnistetään connected thread.
                            connected(socket, socket.getRemoteDevice());
                            break;
                        case STATE_NONE:
                        case STATE_CONNECTED:

                            try {
                                socket.close();
                            } catch (IOException e) {
                                // Ignored
                            }
                            break;
                    }
                }
            }
        }
    }
}

```

Kuva 19. AcceptThread-säie.

4.5 Viestien lähetys ja vastaanottaminen

Kun laitteet ovat yhteydessä keskenään, käyttäjät voivat lähettää niillä viestejä toisilleen. Viestin kirjoitus tapahtuu painamalla sovelluksen alalaidassa sijaitsevaa tekstikenttää, jonka jälkeen avautuu näppäimistö, jossa viestin voi kirjoittaa. Kun viesti on kirjoitettu, sen voi lähettää toiseen laitteeseen painamalla send-näppäintä, joka sijaitsee tekstikentän oikealla puolella. Viestit ilmestyvät tekstikentän yläpuolelle, niin että uusin viesti on aina alimmaisena.

Viestien vastaanottaminen tapahtuu ReadWriteThread-säikeessä, jossa syöttövirtaa luetaan jatkuvasti. Kun syöttövirta vastaanottaa viestin, se lähetetään MainActivity-luokkaan, jossa viesti lisätään taulukkolistaan näkyville käyttöliittymään.

```
public void run() {
    byte[] buffer = new byte[1024];
    int bytes;

    // Kuunnellaan InputStreamia
    while (true) {
        try {
            bytes = inputStream.read(buffer);
            // lähetetään vastaanotetut tavut MainActivity luokkaan nähtäväksi
            handler.obtainMessage(MainActivity.MESSAGE_READ, bytes, -1,
                buffer).sendToTarget();
        } catch (IOException e) {
            connectionLost();
            // käynnistetään palvelu uudelleen, jotta voidaan kuunnella InputStreamia
            ChatController.this.start();
            break;
        }
    }
}
```

Kuva 20. Viestien vastaanottaminen ReadWriteThread-säikeessä.

Myös viestien lähettäminen tapahtuu ReadWriteThread-säikeessä, jossa alustetaan ulostulovirta, joka lähettää kirjoitetut viestit toiseen laitteeseen.

```
// kirjoitetaan OutputStreamiin
public void write(byte[] buffer) {
    try {
        outputStream.write(buffer);
        handler.obtainMessage(MainActivity.MESSAGE_WRITE, -1, -1,
            buffer).sendToTarget();
    } catch (IOException e) {
    }
}
```

Kuva 21. Viestien lähettäminen toiseen laitteeseen ReadWriteThread-säikeessä.

5 JOHTOPÄÄTÖKSET

Työ olisi ollut liian laaja ja vaativa kaltaiselleni ohjelmoinnin aloittelijalle, mutta tutoriaalin seuraaminen teki siitä sopivan mittaisen ja hieman helpomman. Työssä eniten aikaa vei käytännön osuus, johtuen sen haastavuudesta.

Työn lopputuloksena syntyi toimiva sovellus, jolla pystyi lähettämään viestejä kahden Android-laitteen välityksellä Bluetoothia käyttäen. Sovellus on sinänsä melkoisen turha, sillä nykyisin internet on saatavilla lähes kaikkialla, ja sen välityksellä pystyy lähettämään viestejä vaikka maapallon toiselle puolelle. Sovelluksen kehitysprosessi oli kuitenkin todella opettavainen ja mielenkiintoinen. Opin työn aikana käyttämään Android Studioa melko sulavasti ja Java-koodikieli tuli paljon tutummaksi. Sovelluksen teko oli melko haastavaa ja tästä syystä en suosittelen ihan näin haastavaa työtä aloittelijoille.

Sovellusta voisi hyvin jatkokehittää lähettämään esimerkiksi tiedostoja toiseen laitteeseen. Toinen hyvä jatkokehitysidea olisi muuttaa sovellus käyttämään WiFiä ja yrittää sen avulla lähettää viestejä kahden Android-laitteen välityksellä.

LÄHTEET

Absoluteastronomy (n.d). Bluetooth. Viitattu 19.01.2017 osoitteesta <http://www.absoluteastronomy.com/topics/Bluetooth#encyclopedia>

Android developer (2016). Android Studio. Viitattu 01.02.2017 osoitteesta <https://developer.android.com/studio/index.html#features>

Androidpit (2016). What is android. Viitattu 25.01.2017 osoitteesta <https://www.androidpit.com/what-is-android>

Bluetooth (2016). Bluetooth 5 now available. Viitattu 20.01.2017 osoitteesta <https://www.bluetooth.com/news/pressreleases/2016/12/07/bluetooth-5-now-available>

Developer Android (n.d.a). Intro to Activities. Viitattu 07.03.2017 osoitteesta <https://developer.android.com/guide/components/activities/intro-activities.html>

Developer Android (n.d.b). Activity-lifecycle. Viitattu 07.03.2017 osoitteesta <https://developer.android.com/guide/components/activities/activity-lifecycle.html>

Developer Android (n.d.c). Activity-lifecycle (kuva 4). Viitattu 07.03.2017 osoitteesta <https://developer.android.com/guide/components/activities/activity-lifecycle.html>

Devexchanges (2016). Simple Bluetooth chat. Viitattu 22.11.2017 osoitteesta <http://www.devexchanges.info/2016/10/simple-bluetooth-chat-application-in.html>

Drdobbs (2014a). Why build your java projects with gradle (Kuva 4). Viitattu 03.02.2017 osoitteesta <http://www.drdobbs.com/jvm/why-build-your-java-projects-with-gradle/240168608>

Drdobbs (2014b). Why build your java projects with gradle (Kuva 5). Viitattu 03.02.2017 osoitteesta <http://www.drdobbs.com/jvm/why-build-your-java-projects-with-gradle/240168608>

Ebay (2008). Difference between bluetooth versions. Viitattu 20.01.2017 osoitteesta <http://www.ebay.com/gds/The-Difference-between-Bluetooth-1-1-1-2-2-0-2-1-Module-/10000000008782539/g.html>

Engineersgarage (2012). Difference between bluetooth and wifi. Viitattu 24.01.2017 osoitteesta <http://www.engineersgarage.com/contribution/difference-between-bluetooth-and-wifi>

Laptopmag (2012). What is Bluetooth 4.0. Viitattu 20.01.2017 osoitteesta <http://www.laptopmag.com/articles/just-what-is-bluetooth-4-0-anyway>

Majorhifi (2016). Difference between bluetooth versions. Viitattu 20.01.2017 osoitteesta <http://majorhifi.com/the-difference-between-bluetooth-3-0-4-0-4-1-4-2-explained>

Quora (2016). What are the latest features of Bluetooth 5 version Technology. Viitattu 20.01.2017 osoitteesta <https://www.quora.com/What-are-the-latest-features-of-Bluetooth-5-0-version-Technology>

Recombu (2016). what is android and what is an android phone. Viitattu 25.01.2017 osoitteesta https://recombu.com/mobile/article/what-is-android-and-what-is-an-android-phone_M12615.html

Rfwireless-world (2012). IrDA vs WLAN vs Bluetooth. Viitattu 25.01.2017 osoitteesta <http://www.rfwireless-world.com/Terminology/IrDA-vs-WLAN-vs-Bluetooth.html>

Sitepoint (2016) What is Gradle. Viitattu 03.02.2017 osoitteesta <https://www.sitepoint.com/quick-tip-what-is-gradle-and-how-does-it-work-with-android-studio/>

Tutorialspoint (2017). Java overview. Viitattu 31.01.2017 osoitteesta http://www.tutorialspoint.com/java/java_overview.html

Webopedia (2017). Android SDK. Viitattu 31.01.2017 osoitteesta http://www.webopedia.com/TERM/A/Android_SDK.html