
AVOIMEEN LÄHDEKOODIIN PERUSTUVAN MOBIILIN OHJELMISTOALUSTAN TESTAAMINEN MITTAUSSOVELLUKSESSA

Jari-Pekka Pitkänen

Opinnäytetyö

Ammattikorkeakoulututkinto

| | |
|---|--------------------------|
| Koulutusala Tekniikan ja liikenteen ala | |
| Koulutusohjelma Tietotekniikan koulutusohjelma | |
| Työn tekijä(t) Jari-Pekka Pitkänen | |
| Työn nimi Avoimeen lähdekoodiin perustuvan mobiilin ohjelmistoalustan testaaminen mittaussovelluksessa | |
| Päiväys 7.5.2011 | Sivumäärä/Liitteet 34 |
| Ohjaaja(t) Lehtori Jussi Koistinen | |
| Toimeksiantaja/Yhteistyökumppani(t) Mega Elektroniikka Oy, tekninen johtaja Arto Holopainen & Marko Juntunen | |
| Tiivistelmä <p>Tämän opinnäytetyön tavoitteena oli tutustua uuteen Android-mobiilialustaan ja toteuttaa mobiililaitteessa toimiva mittaussovellus. Työn tarkoituksena on selvittää alustan soveltuvuus EKG-mittaukseen.</p> <p>Sovelluksen kehityskielenä oli Java ja kehitysympäristönä Eclipse. Sovellukselle asetettuja vaatimuksia olivat tiedon lukeminen Bluetooth-anturilta, EKG-signaalin piirtäminen näytölle, sykelukeman laskeminen sekä tiedon tallentaminen muistikortille. Toivotunlaisesta käyttöliittymästä saatiin valmiit kuvat, eikä sitä tarvinnut suunnitella erikseen. Työn tekeminen aloitettiin tutustumalla Android-alustaan ja sen uudenlaisiin Java-kirjastoihin, minkä jälkeen toteutettiin sovellus.</p> <p>Työn tuloksena toteutettiin EKG:n mittaukseen soveltuva sovellus Android-alustalle. Jatkokehityksellä tästä sovelluksesta voidaan saada toimiva sovellus mobiilimarkkinoille. Tässä dokumentissa on esitelty sovelluksen rakenne sekä kehityksen vaiheet.</p> | |
| Avainsanat Android, Bluetooth, EKG, Java, mobiilisovellus | |
| julkinen | |

| | | | |
|---|------------|------------------|----|
| Field of Study Technology, Communication and Transport | | | |
| Degree Programme Degree Programme in Information Technology | | | |
| Author(s) Jari-Pekka Pitkänen | | | |
| Title of Thesis Testing Open Source Mobile Operating System in Measurement Application | | | |
| Date | 7 May 2011 | Pages/Appendices | 34 |
| Supervisor(s) Mr. Jussi Koistinen, Lecturer | | | |
| Project/Partners Mega Electronics Ltd, Mr. Arto Holopainen, Technical Director & Mr. Marko Juntunen | | | |
| <p>Abstract</p> <p>The main purpose of this thesis was to explore the new Android mobile platform and implement a measurement application for a mobile device. The study was done in order to find out how suitable the platform is for ECG measurements.</p> <p>Java was used as the programming language in the application and Eclipse was used as the IDE. Demands set for the application were reading data sent by a Bluetooth sensor, drawing the ECG signal on the screen, calculating heart rate and saving received data to the memory card. Pictures of the desired user interface were provided by the commissioner and there was no need to do any separate planning. The study was started by exploring the Android platform and its new Java libraries. The planning of the application was started after exploring the platform and ultimately the programming was started.</p> <p>The result of this application development project was an application suitable for ECG measurement on the Android mobile platform. With some further development this application would have potential for mobile application markets.</p> | | | |
| Keywords Android, Bluetooth, ECG, Java, mobile application | | | |
| public | | | |

ALKUSANAT

Tämän työn tekemisessä oli sekä hyvät että huonot hetkensä. Työn edetessä tuli hyvä tunne ja vastaavasti ongelmia kohdatessani motivaatio oli koitoksella. Tätä opinnäytetyötä oli kuitenkin varsin miellyttävä tehdä suunnittelusta itse ohjelman ja tämän dokumentin kirjoittamiseen. Työ opetti paljon suunnittelun tärkeydestä ja mobiiliohjelmoinnista sekä työmäärästä, joka tarvitaan toimivan sovelluksen kehittämiseen.

Työssä ohjaajana toimi lehtori Jussi Koistinen. Haluan kiittää häntä ohjauksesta ja opastuksesta sekä kärsivällisyydestä tätä pitkittänyttä projektia kohtaan. Kiitokset myös elektroniikkasuunnittelija Timo Ollikaiselle Savonia AMK:n IT-kehitysyksiköstä tämän työn aiheesta ja Android-puhelimen lainaamisesta. Lopuksi vielä kiitokset tekninen johtaja Arto Holopaiselle Mega Elektroniikka Oy:stä sekä toimitusjohtaja Marko Juntuselle tytäryhtiö MegaKoto Oy:stä kaikesta opastuksesta sekä kärsivällisyydestä työtä kohtaan sovellusta tehdessäni.

SISÄLTÖ

| | | |
|-------|---|----|
| 1 | JOHDANTO | 7 |
| 2 | TYÖKALUT | 8 |
| 2.1 | Eclipse..... | 8 |
| 2.2 | Android SDK..... | 8 |
| 3 | BLUETOOTH | 9 |
| 4 | EKG | 10 |
| 5 | ANDROID..... | 11 |
| 5.1 | Androidin arkkitehtuuri..... | 12 |
| 5.2 | API-tasot..... | 14 |
| 6 | MITTAUSSOVELLUS..... | 15 |
| 6.1 | Sovelluksen rakenne | 15 |
| 6.2 | Activity..... | 17 |
| 6.2.1 | Activity:n elinkaari..... | 17 |
| 6.2.2 | AndroidEKG-luokka..... | 19 |
| 6.3 | Sovelluksen toiminnot ja valikot | 20 |
| 6.3.1 | Sovelluksen toiminnot..... | 20 |
| 6.3.2 | Valikot sekä komentojen kuuntelu | 24 |
| 6.4 | Bluetooth-yhteys..... | 27 |
| 6.4.1 | Laitteiden etsiminen ja niihin yhdistäminen..... | 27 |
| 6.4.2 | Mittauksen aloittaminen ja tiedon lukeminen | 28 |
| 6.5 | Näytölle piirtäminen | 29 |
| 7 | TULOSTEN TARKASTELU | 31 |
| 7.1 | Työprosessin tarkastelua..... | 31 |
| 7.2 | Ongelmia ja niiden tarkastelua..... | 31 |
| 7.3 | Tavoitteiden täytyminen..... | 33 |
| 7.4 | Jatkokehitys ja tulevaisuudennäkymät..... | 33 |
| | LÄHTEET | 34 |

1 JOHDANTO

Työn tarkoituksena on testata Googlen uuden Android-mobiilialustan soveltuvuutta reaaliaikaisen mittauksen suorittamiseen. Työ tehtiin kuopiolaiselle Mega Elektroniikka Oy:lle, joka on biosignaalin tarkkailuun neurologiassa, kuntoutukseen, työterveyteen sekä urheilulääketieteeseen erikoistunut yritys. Mega Elektroniikka Oy tuottaa monenlaisia antureita sekä sovelluksia näiden alueiden mittausten ja testien suorittamiseen.

Työssä testataan Android-puhelimen soveltuvuutta EKG-mittaukseen, jossa mobiililaitte lukee anturin Bluetoothin välityksellä lähettämää dataa. Luettu data piirretään näytölle reaaliajassa ja sen avulla lasketaan myös käyttäjän syke. Alustan kaikista ominaisuuksista ei työtä aloitettaessa ollut varmuutta ja työssä tutustuttiin mittaukseen tarvittaviin ominaisuuksiin ja toteutettiin mittaussovellus. Vastaavanlainen EKG:n mittaussovellus Mega Elektroniikka Oy:llä on jo Symbian-alustalle, jossa ohjelma toimii ongelmitta.

Uusi Android-mobiilikäyttöjärjestelmä on kiinnostava alusta, jonka käyttö lisääntyy nopeasti. Koska Android-alusta on vielä varsin uusi, sille ei ole vielä monia kaupallisia sovelluksia EKG:n mittaukseen. Alustan suuri levinneisyys tekee siitä varteenotettavan markkinointialueen sovelluksille.

2 TYÖKALUT

2.1 Eclipse

Opinnäytetyön sovelluksen kehitystyökaluna käytettiin Eclipseä, joka on virallisesti tuettu kehitysympäristö. Eclipse on ilmainen avoimen lähdekoodin integroitu kehitysympäristö, ja se tukee Java-sovellusten kehittämistä. Android-sovelluksia on mahdollista kehittää myös muilla ympäristöillä, kuten Javaan keskittyneellä NetBeans IDE:llä sekä IntelliJ IDEA:lla. Erilaisten liitännäisten (plug-in) kautta Eclipseen saa tuen myös monille muille ohjelmointikielille, kuten esimerkiksi C:lle sekä C++:lle. Android-sovellusten kehittämiseen tarvittava liitännäinen on nimeltään ADP (Android Development Tools), ja se voidaan asentaa Eclipsestä käsin.

2.2 Android SDK

Eclipsen lisäksi Android-sovelluksen kehittämiseen tarvittiin Android SDK (Software Development Kit). SDK:lla ladataan Android-sovellusten kehittämiseen tarvittavat paketit, joita ovat esimerkiksi sovelluksen kehittämiseen käytettävä API-versio ja tämän dokumentointi. Myös sovelluksien testaamisessa käytettävien Android-emulaattoreiden luominen ja hallinta tapahtuvat SDK:n avulla. Emulaattorit eivät vielä tue kaikkia puhelinten ominaisuuksia. Näitä ovat esimerkiksi multi-touch ja Bluetooth.

3 BLUETOOTH

Bluetooth on avoin langaton teknologia, joka mahdollistaa datan lähettämisen ja lukemisen lyhyillä kantomatkoilla. Sen kantomatka on noin 10 m ja tiedon välittämiseen käytetään radioviestiliikennettä, joten yhteydessä olevien laitteiden ei tarvitse olla näköyhteydessä toisiinsa. Bluetooth tarjoaa turvallisen keinon yhdistää ja välittää tietoa laitteiden välillä, ja sitä käytetään esimerkiksi puhelimissa, GPS-vastaanottimissa ja handsfree-laitteissa. Teknologia on nykyään käytössä lähes kaikissa mobiililaitteissa sekä kannettavissa tietokoneissa. (Bluetooth: Facts.)

Bluetoothin kehitti matkapuhelinvalmistaja Ericsson vuonna 1994, ja myöhemmin syyskuussa 1998 perustettiin Bluetooth Special Interest Group (SIG), joka valvoo Bluetoothin kehitystä ja lisensoimista. Sitä perustamassa olivat Ericsson, IBM, Intel, Toshiba, Motorola ja Nokia ja myöhemmin siihen liittyi monia muita yrityksiä. Jotta jokin laitetta voi markkinoida Bluetooth-laitteena, sen täytyy täyttää SIG:n määrittämät standardit. (Bluetooth: Facts.)

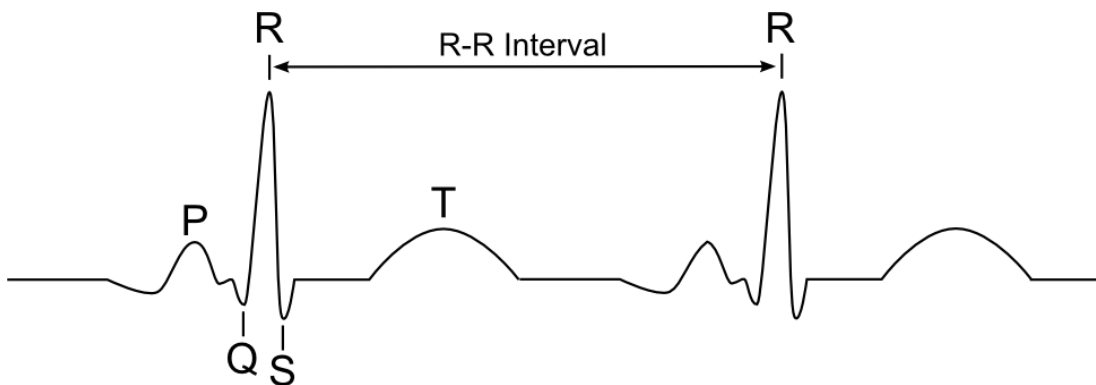
Bluetoothin toimintataajuus on 2 402 – 2 480 MHz, joka on maailmanlaajuisesti lisensoimaton lyhyen matkan taajuus. Bluetooth käyttää taajuushyppelytekniikkaa, jossa lähettäjä vaihtelee nopeasti lähetystaajuutta ja käyttää tiedon vaihtoon 79:ää 1 MHz:n levyistä kanavaa. Bluetoothin tiedonsiirtonopeus on nykyään noin 3 Mb/s, mutta uuden 3.0-version pitäisi nostaa se 24 Mb:iin/s. (Bluetooth: Facts.)

4 EKG

Lyhenne EKG tulee sanasta elektrokardiogrammi, joka tarkoittaa sydänsähkökäyrää. Käytössä on myös termi sydänfilmi, vaikka tutkimuksessa ei rekisteröidä mitään filmille. Tutkimustekniikka on nykyään digitaalinen ja käyrät tulostetaan tarvittaessa paperille. Yleensä EKG:tä rekisteröidään kiinnittämällä elektrodit kumpaankin ranteeseen ja nilkkaan ja kuuteen kohtaan rintakehällä. (EKG, Suomen Sydänliito ry.)

EKG:tä rekisteröimällä nähdään sydämen rytmi, jonka normaalimuotoa kutsutaan sinusrytmiksi. Tällöin jokaista P-aaltoa seuraa QRS-kompleksi, jonka jälkeen tulee T-aalto. (EKG, Suomen Sydänliito ry.)

Syke voidaan laskea R-R-välien pituuksista. Ihmisen normaali syke lepotilassa on noin 70 eli sydän lyö 70 kertaa minuutissa. Kuvassa 1 on esitetty normaali EKG-käyrän osa, jossa näkyvät käyrän komponentit. (EKG, Suomen Sydänliito ry.)



KUVA 1. EKG-käyrä. (Biomedical Engineering | ECG Assignment)

5 ANDROID

Vuonna 2005 Internetin käytön suosion suuren kasvun myötä Google osti Androidia alun perin kehittäneen Android Inc. -yrityksen, jotta se voisi keskittyä mobiilialustan kehittämiseen. Applen vuonna 2007 julkaiseman iPhoneen uudet ideat, kuten multi-touch ja avoin markkinapaikka sovelluksille, sisällytettiin nopeasti Androidiin ja näiden lisäksi käyttöjärjestelmään lisättiin myös mahdollisuus ajaa useaa sovellusta yhtä aikaa. (Steele & Nelson 2010, 1.)

Android on avoimen lähdekoodin sulautettu käyttöjärjestelmä, joka rakentuu Linux-ytimen päälle, mutta ei kuitenkaan ole sulautettu versio Linuxista. Sovelluksien ohjelmoinnissa Androidille käytetään Javan ohjelmistokehystä (framework), mutta se ei kuitenkaan ole täysin samanlainen kuin Java-ohjelmointikielessä. Esimerkiksi yleisiä Java-kirjastoja, kuten esimerkiksi Swing, ei tueta ollenkaan. Osa Javan kirjastoista on korvattu Androidin omilla kirjastoilla, joita on optimoitu käytettäväksi vähemmällä resursseilla. (Steele & Nelson 2010, 2.)

5.1 Androidin arkkitehtuuri

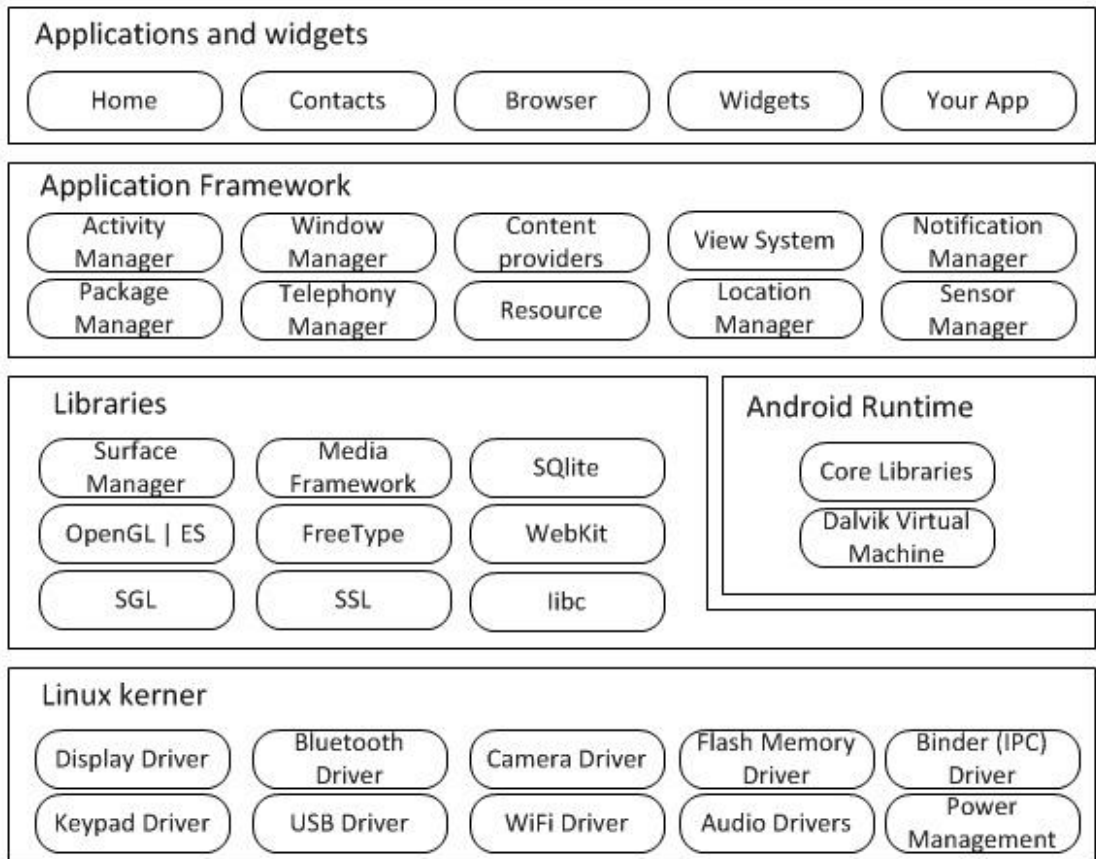
Androidin arkkitehtuurin pohjalla on Linux-ydin, jota käytetään muistin sekä prosessin hallintaan, verkon käyttämiseen ja muihin käyttöjärjestelmän palveluihin. Androidin käyttäjä ei suoraan näe Linuxia milloinkaan, eivätkä sovellukset myöskään tee suoria kutsuja siihen. (Burnette 2008, 30 - 31.)

Seuraavalla tasolla ytimen päällä ovat Androidin omat kirjastot, jotka on käännetty puhelimen käyttämälle laitteistoarkkitehtuurille. Tärkeitä kirjastoja ovat esimerkiksi Surface Manager, joka hallitsee näytön piirtämistä, sekä 2D ja 3D grafiikan piirtämiseen käytettävä kirjasto. (Burnette 2008, 31 - 32.)

Ytimen päällä ovat myös sovellusten suoritukseen liittyvät toiminnot. Kaikki Android-sovellukset kirjoitetaan Java-ohjelmointikielellä ja ne ajetaan virtuaalikoneen päällä. Androidissa tämä virtuaalikone on nimeltään Dalvik. Pohjimmiltaan se on Java-virtuaalikone, joka on suunniteltu toimimaan laitteiden rajallisilla resursseilla. Dalvik eroaa perinteisestä Javan virtuaalikoneesta kahdella tärkeällä tavalla. Ensimmäiseksi Dalvik suorittaa .dex-tiedostoja, jotka muutetaan siihen muotoon kääntämisvaiheessa normaaleista .jar ja .class -tiedostoista. Toiseksi Dalvikin ydin Java-kirjastot eroavat Java Standard Edition (Java SE) sekä Java Mobile Edition (Java ME) -kirjastoista. (Burnette 2008, 32 - 33.)

Kirjastojen yläpuolella on sovellusten framework, joka sisältää korkean tason palikoita, joita käytetään sovelluksien rakentamiseen. Nämä tulevat Androidin mukana, mutta niitä on mahdollista kehittää tarpeen vaatiessa lisää. Oleellisia osia ovat muun muassa Activity Manager, joka hallitsee sovellusten elinkaarta ja Notification Manager, joka ilmoittaa saapuneista viesteistä ja muista tapahtumista. (Burnette 2008, 33 - 34.)

Hierarkian huipulla ovat sovellukset ja widgetit. Käyttäjät näkevät vain tällä tasolla olevat ohjelmat. Seuraavalla sivulla olevassa kuvassa 2 esitetään Androidin arkkitehtuuri.



KUVA 2. Androidin arkkitehtuuri. (Burnette 2008, 31)

5.2 API-tasot

API-taso (Application Programming Interface, ohjelmointirajapinta) on kokonaislukuarvo, joka ilmoittaa Android-alustan version. Jokainen Android alusta tukee tasan yhtä versiota, joskin API:n uudempi versio tukee myös kaikkia vanhempia API:n versioita. Päivitykset rajapintaan toteutetaan siten, että uusi versio rajapinnasta on yhteensopiva vanhan version kanssa, mutta ne kuitenkin lisäävät tai korvaavat toimintoja. Päivitykset eivät koskaan poista mitään rajapinnasta vaan ne jättävät vanhat osat paikoilleen vanhoja tasoja käyttäville sovelluksille, mutta nämä osat merkitään vanhentuneiksi. (Android API Levels)

Taulukossa 1 on esitetty kirjoitushetkeen mennessä julkaistut Android-käyttöjärjestelmän versiot.

TAULUKKO 1. API-tasot

| Alustan versio | API-taso |
|----------------|----------|
| Android 3.0 | 11 |
| Android 2.3.3 | 10 |
| Android 2.3 | 9 |
| Android 2.2 | 8 |
| Android 2.1 | 7 |
| Android 2.0.1 | 6 |
| Android 2.0 | 5 |
| Android 1.6 | 4 |
| Android 1.5 | 3 |
| Android 1.1 | 2 |
| Android 1.0 | 1 |

6 MITTAUSSOVELLUS

Mittaussovellus on itsenäinen sovellus, joka asennetaan käyttäjän Android-laitteeseen. Jotta mittaussovelluksesta olisi jotain hyötyä, tarvitaan kuitenkin oikeanlainen Bluetooth-anturi EKG:n mittaamiseen ja laitteen täytyy myös tukea Bluetoothia. Sovellus sisältää kaikki tarvittavat toiminnot yksinkertaiseen EKG-mittaukseen.

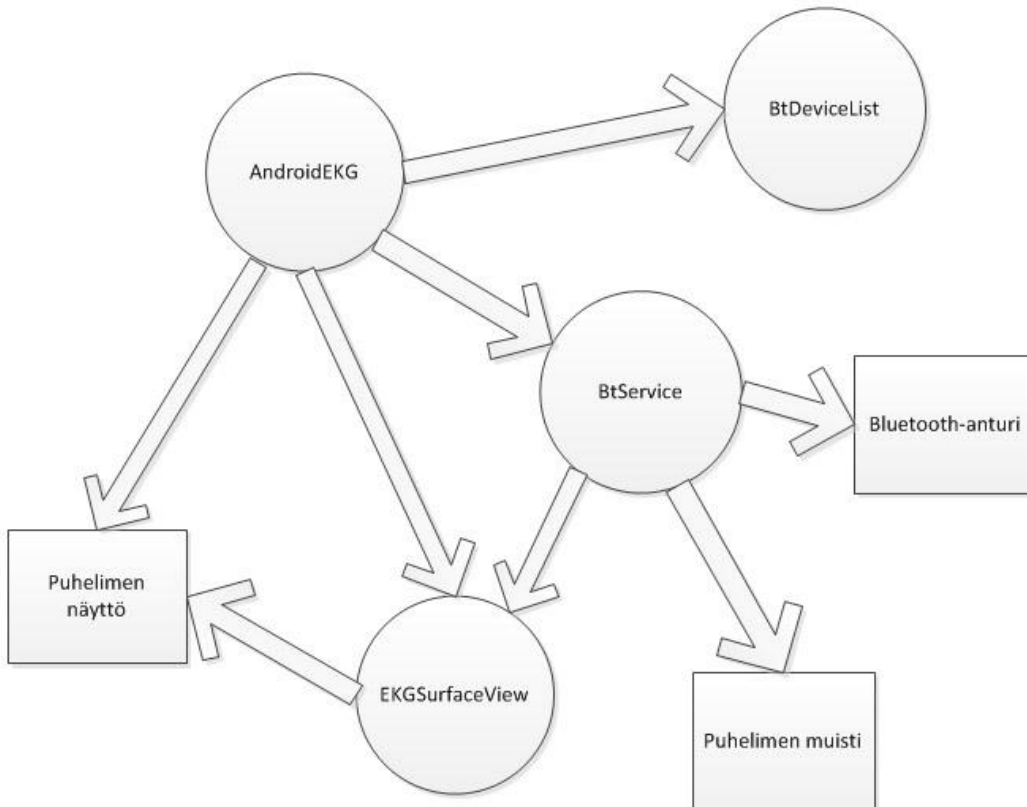
6.1 Sovelluksen rakenne

Android-sovelluksen, jossa käyttäjän on tarkoitus olla vuorovaikutuksessa sovelluksen kanssa, täytyy sisältää vähintään yksi activity-luokka. Tämä activity-luokka vastaa J2ME:n MIDlet-luokkaa ja, se sisältää metodit, joita käyttöjärjestelmä kutsuu tarpeen vaatiessa ohjatakseen sovelluksen toimintaa. Tällainen tapahtuma on esimerkiksi puhelun saapuminen, jonka seurauksena käyttöjärjestelmä kutsuu activity-luokan metodia, jolla sovellus laitetaan taukotilaan. Puhelun päättyessä käyttöjärjestelmä kutsuu toista metodia, joka palauttaa sovelluksen jälleen aktiiviseksi.

AndroidEKG-sovellus sisältää seuraavat luokat:

- **AndroidEKG.java** on activity ja vastaa sovelluksen käyttöliittymästä.
- **BtDeviceList.java** on myös activity ja vastaa asetukset-näytön päälle piirretävistä Bluetooth-laitteiden listasta sekä niiden etsimisestä.
- **BtService.java** hallitsee Bluetooth-yhteyksiin liittyvät toiminnot.
- **EKGSurfaceView.java** hoitaa mittauksen tietojen piirtämisen näytölle.

Seuraavalla sivulla kuvassa 3 on esitetty sovelluksen luokat ja niiden hierarkia.



KUVA 3. Sovelluksen luokat

6.2 Activity

Activity on sovelluksen osa, joka antaa sovellukselle näytön, jonka kautta käyttäjä voi olla vuorovaikutuksessa sovelluksen kanssa. Jokaiselle activity:lle annetaan oma ikkuna, jolle käyttöliittymä piirretään. Kun uusi activity aloitetaan, edellisen activity:n suoritus pysäytetään mutta se pidetään tallessa activity:iden pinossa. (Activity | Android Developers)

6.2.1 Activity:n elinkaari

Activity sisältää sovelluksen elinkaaren ja muut toiminnan kannalta oleelliset metodit, joita laitteen käyttöjärjestelmä kutsuu luodessaan sovelluksen, sovelluksen käynnistyessä ja uudelleenkäynnistyessä, laittaessaan sovelluksen taukotilaan ja palatesaan takaisin sekä pysäyttäessään ja tuhotessaan sovelluksen. Näiden metodien nimet ovat onCreate(), onStart(), onRestart(), onPause(), onResume(), onStop() ja onDestroy(). Näistä onCreate()-metodia kutsutaan, kun activity aloitetaan, ja kaikki alkutoiminnot, kuten esimerkiksi ensimmäisen näytön näyttäminen, pitäisi laittaa tänne. (Activity | Android Developers)

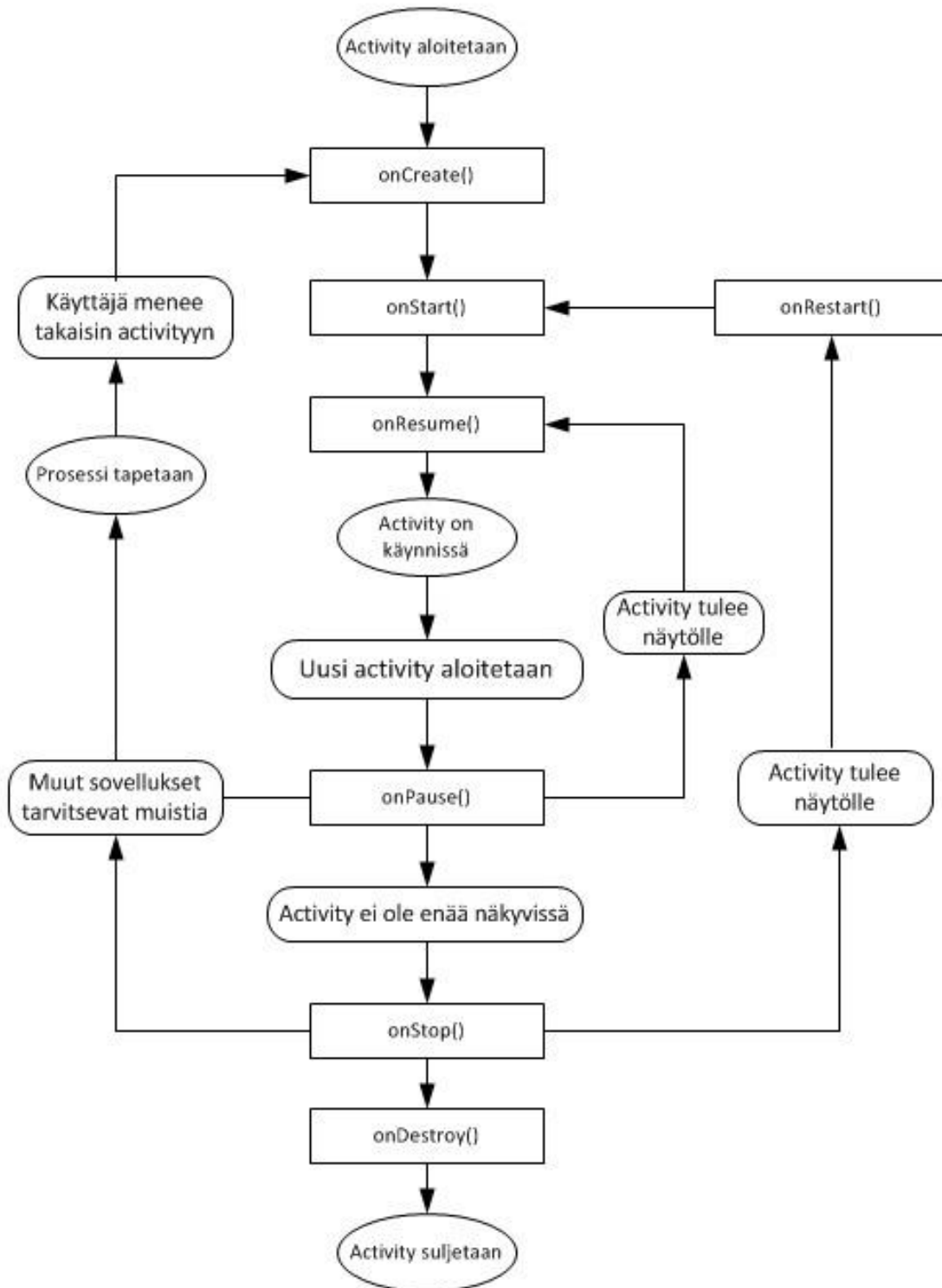
Kahdesta seuraavasta metodista onStart()-metodia kutsutaan, kun sovellus tulee näkyviin käyttäjälle. Tämä tapahtuu heti onCreate()-metodin tai onRestart()-metodin jälkeen. onRestart()-metodia kutsutaan sovelluksen pysäyttämisen jälkeen, jolloin sen suoritus aloitetaan uudelleen. (Activity | Android Developers)

Seuraavista kahdesta metodista onResume()-metodia kutsutaan, kun activity aloittaa vuorovaikutuksen käyttäjän kanssa. onPause()-metodia kutsutaan, kun järjestelmä aloittaa siirtymisen toiseen activity:yn. (Activity | Android Developers)

onStop()-metodia kutsutaan, kun käyttäjä sovellus ei enää näe sovellusta jonkun muun activity:n ollessa sen päällä. Tämä tapahtuu esimerkiksi käynnistettäessä toinen sovellus tai vaihtaessa jo käynnissä oleva toinen sovellus aktiiviseksi. (Activity | Android Developers)

Sovellusta lopetettaessa järjestelmä kutsuu onDestroy()-metodia. Tämä voi seurata siitä, että sovelluksessa joku kutsuu finish()-metodia tai järjestelmä väliaikaisesti tuhoaa instanssin säästääkseen tilaa. (Activity | Android Developers)

Seuraavalla sivulla olevassa kuvassa 4 on esitetty activity:n elinkaari.



Kuva 4. Activity:n elinkaari

6.2.2 AndroidEKG-luokka

AndroidEKG-sovelluksen pää-activity on AndroidEKG-luokka, joka perii Activity-luokan ja implementoi OnClickListener-rajapinnan. Sovelluksen kaikki perustoiminnot on toteutettu AndroidEKG-luokassa.

Sovelluksen käynnistyessä AndroidEKG-luokan onCreate()-metodissa tarkastetaan, tukeeko puhelin Bluetoothia. Jos Bluetoothille ei löydy tukea, tästä ilmoitetaan käyttäjälle ja sovellus suljetaan. Tuen löytyessä luodaan alkuvalikko ja käyttäjä voi aloittaa sovelluksen käyttämisen. Alla on esimerkkikoodia Bluetooth-tuen tarkastamisesta sovelluksessa.

```
public class AndroidEKG extends Activity implements OnClickListener {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Haetaan laitteen Bluetooth-adapteri
        mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

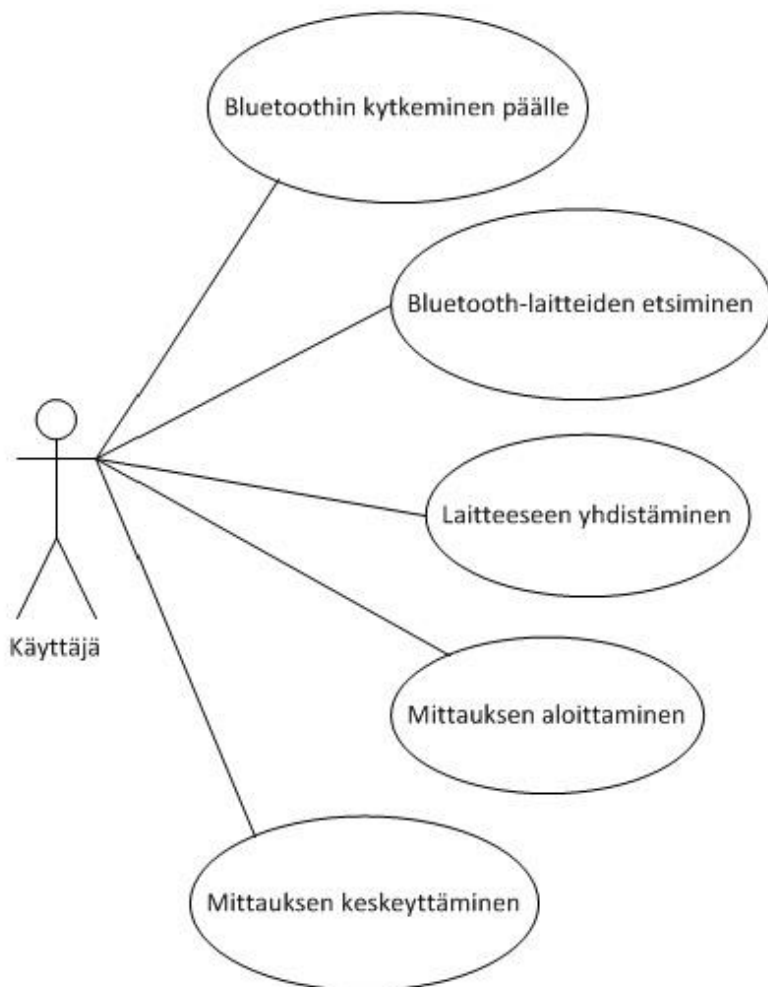
        // Jos adapteri on null, niin Bluetoothia ei tueta
        if (mBluetoothAdapter == null) {
            Toast.makeText(this, "Laitte ei tue Bluetoothia", Toast.LENGTH_LONG).show();
            finish();
            return;
        }

        // Muu koodi
    }
}
```

6.3 Sovelluksen toiminnot ja valikot

6.3.1 Sovelluksen toiminnot

Käyttäjälle näkyvistä toiminnoista oleellisimpia ovat Bluetoothin kytkeminen päälle sovellusta käynnistäessä, Bluetooth-laitteiden etsiminen ja laitteeseen yhdistäminen sekä mittauksen aloittaminen ja keskeyttäminen.



KUVA 5. Sovelluksen oleellimmat toiminnot käyttötapauskaviona

Tarkemmat käyttötapauskuvaukset kuvan 6 käyttötapauksista ovat seuraavilla sivuilla (kuvat 6 - 10).

Käyttötapausten nimi: Bluetoothin kytkeminen päälle

Osallistajat: Käyttäjä

Tuloehdot: Sovellus käynnistetään Bluetoothin ollessa kytkettynä pois päältä.

Kuvaus: Käyttäjä käynnistää sovelluksen, kun Bluetooth on kytkettynä pois päältä, jolloin sovellus kysyy laitetaanko Bluetooth päälle.

Poikkeukset: Laitteessa ei ole Bluetoothia tai käyttäjä ei kytke sitä päälle. Sovellus suljetaan.

Lopputulokset: Sovellus on alkuvalikossa ja Bluetooth on kytkettynä päälle.

KUVA 6. Käyttötapauskuvaukset Bluetoothin kytkemisestä päälle

Käyttötapausten nimi: Bluetooth-laitteiden etsiminen

Osallistajat: Käyttäjä

Tuloehdot: Käyttäjä on avannut laitteiden etsimiseen käytettävän ikkunan.

Kuvaus: Käyttäjä aloittaa laitteiden etsimisen tai valitsee paritetun laitteen.

Poikkeukset: Bluetooth on kytketty pois päältä sovelluksen käynnistämisen jälkeen.

Lopputulokset: Listalla näkyvät löydetyt EKG-anturit.

KUVA 7. Käyttötapauskuvaukset Bluetooth-laitteiden etsimisestä

Käyttötapausten nimi: Anturiin yhdistäminen

Osallistajat: Käyttäjä

Tuloehdot: Anturi näkyy löydettyjen laitteiden listalla.

Kuvaus: Käyttäjä valitsee anturin.

Poikkeukset: Virhe yhdistämisen aikana tai anturi on suljettu. Sovellus ilmoittaa virheestä käyttäjälle.

Lopputulokset: Sovellus muodostaa yhteyden anturiin.

KUVA 8. Käyttötapauskuvaus anturiin yhdistämisestä

Käyttötapausten nimi: Mittauksen aloittaminen

Osallistajat: Käyttäjä

Tuloehdot: Sovellus on alkuvalikossa.

Kuvaus: Käyttäjä painaa aloita-nappia.

Poikkeukset: Käyttäjä ei ole muodostanut yhteyttä anturiin, sovellus ilmoittaa tästä käyttäjälle.

Lopputulokset: Sovellus on mittausnäkyssä ja mittaus alkaa.

KUVA 9. Käyttötapauskuvaus mittauksen aloittamisesta

Käyttötapaus: Mittauksen keskeyttäminen

Osallistajat: Käyttäjä

Tuloehdot: Sovellus on mittausnäkyssä.

Kuvaus: Käyttäjä painaa keskeytä-nappia.

Poikkeukset: -

Lopputulos: Sovellus lopettaa mittaamisen ja palaa päävalikkoon.

Muut vaatimukset: -

KUVA 10. Käyttötapauskuvaus mittauksen keskeyttämisestä

6.3.2 Valikot sekä komentojen kuuntelu

Sovelluksen päävalikko luodaan sovellusta käynnistettäessä onCreate()-metodissa. Nappien luomisessa käytetään Androidin Button-widgetiä, ja kaikki nappien sijaintiin ja ulkonäköön liittyvät asetukset ovat omassa XML-tiedostossaan, joka määrittää näkymän sisällön asettelun. Näkymän sisällön luomisen jälkeen jokaiselle napille asetetaan kuuntelija käyttäjän painalluksille. Valikon luomisen jälkeen jäädään odottamaan käyttäjän komentoja, joita OnClickListener-rajapinta kuuntelee. Komennot käsittelee onClick()-metodi, jossa verrataan painetun napin id-arvoa käyttöliittymässä käytettyjen nappien id-arvoihin. Jos verrattavat arvot ovat samat, suoritetaan nappiin määritetyt toiminnot. Alla olevassa kuvassa 11 näkyy sovelluksen alkuvalikko.



KUVA 11. Alkuvalikko

Käyttäjä voi valita haluamansa toiminnon laitteen omilla näppäimillä tai käyttää Androidin tukemaa kosketusnäyttöä. OnClickListener-rajapinta tunnistaa molemmat tavat eikä kosketusnäytön käyttämiselle tarvita mitään erillisiä määrytyksiä.

Asetukset-nappia painettaessa kutsutaan metodia, joka luo Asetukset-näkymän. Näkymä luodaan vaihtamalla näkymän sisällön asettelu `setContentView()`-metodilla, joka saa parametrina käytettävän XML-tiedoston nimen. Näytön sisältö piirretään XML-tiedostosta ja kaikille komponenteille määritetään omat kuuntelijat sekä asetetaan sisällöt alaspäinvalikoihin. Alla olevassa kuvassa 12 on esitetty Asetukset-näkymä.



KUVA 12. Asetukset-näkymä

Android tukee XML-tiedostoja (Extensible Markup Language), joiden avulla objektien sijoittelu näytölle käyttöliittymää tehdessä on yksinkertaista. Jokaiselle näytölle voi tehdä oman XML-tiedoston, josta sisällön tiedot ladataan. Seuraavalla sivulla on esimerkki Asetukset-napin sijoittamisesta näytölle XML-tiedostoa käyttäen.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/layoutMain" >
    <Button
        android:id="@+id/asetukset"
        android:layout_width="150dip"
        android:layout_height="60dip"
        android:layout_marginTop="60dip"
        android:layout_centerHorizontal="true"
        android:text="Asetukset" />
</RelativeLayout>
```

6.4 Bluetooth-yhteys

Sovellus muodostaa Bluetooth-yhteyden anturiin ja kaikki kommunikointi laitteiden välillä tapahtuu tämän yhteyden kautta. Bluetooth-ominaisuuksien käyttäminen alkaa Bluetooth-antureiden etsimisellä ja mahdolliseen löydettyyn laitteeseen yhdistämisellä.

6.4.1 Laitteiden etsiminen ja niihin yhdistäminen

Sovelluksessa laitteiden etsimisestä ja niihin yhdistämisestä vastaa BtDeviceList-luokka. Käyttäjän painettua Parita EKG -nappia Asetukset-näkymässä, sen päälle avautuu pieni ikkuna, joka on sovelluksen toinen activity. Kuvassa 13 on esitetty laitteiden etsimiseen käytettävä ikkuna.



KUVA 13. Laitteiden etsiminen

Parita EKG -nappia painettaessa AndoridEKG-luokasta aloitetaan uusi Intent, joka luo BtDeviceList-luokan ilmentymän ja näyttää sen näytöllä. Intent on viesti järjestel-

mälle, että nyt halutaan tehdä jotain, tässä tapauksessa aloittaa uusi activity. Luokka etsii ensin laitteen kanssa jo paritetut laitteet `getBondedDevices()`-metodilla ja näyttää nämä listalla. Tämän jälkeen odotetaan käyttäjän komentoa, joka voi olla jo paritetun laitteen valitseminen tai Bluetooth-laitteiden haku. Etsi laitteita -nappi suorittaa `startDiscovery()`-metodin, joka etsii kaikki kantomatkalla olevat Bluetooth-laitteet. Kaikki löydetty laitteet otetaan talteen `ArrayAdapter`:iin ja laitteet, joiden nimestä löytyy tietynlainen merkkiyhdistelmä, näytetään löydettyjen laitteiden listalla.

Käyttäjän valitessa laite listalta, luokka palauttaa valitun laitteen MAC-osoitteen (Media Access Control) `AndroidEKG`-luokkaan, jossa se otetaan talteen. `AndroidEKG`-luokasta kutsutaan `BtService`-luokan `connect()`-metodia, joka saa parametrina laitteen, johon yhteys halutaan luoda. `Connect()`-metodissa aloitetaan uusi säie, jossa luodaan `RFCOMM` (Radio Frequency Communication) Bluetooth socket, jonka jälkeen laitteeseen yritetään yhdistää luodun socket:in kautta. Yhdistämisen onnistuessa aloitetaan uusi säie, jossa hallitaan kaikki laitteiden välinen kommunikointi.

6.4.2 Mittauksen aloittaminen ja tiedon lukeminen

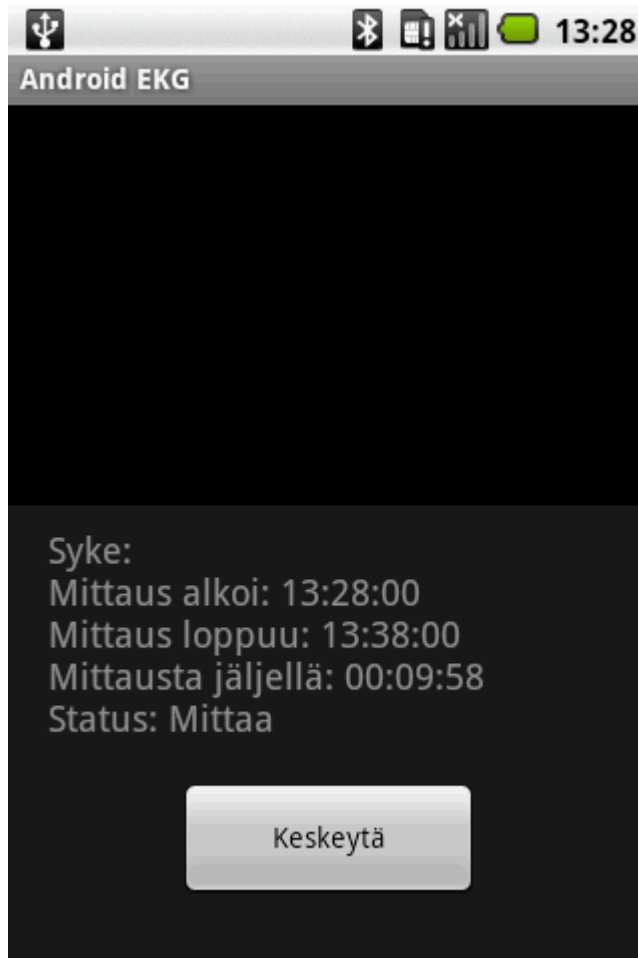
Aloita-nappia painettaessa tarkastetaan, onko Bluetooth-anturiin luotu yhteyttä. Yhteyden löytyessä anturille lähetetään ensin mittauksen asetuskomennot ja näiden jälkeen aloituskomento. `AndroidEKG`-luokasta kutsutaan `BtService`-luokan `write()`-metodia, joka saa parametrina lähetettävän komennon tavutaulukkona. Komento lähetetään tavu kerrallaan käyttäen avattua `OutputStream`-datavirtaa.

Anturin lähettämät datapaketit otetaan vastaan avatun `InputStream`-datavirran kautta. Puskuriin luetaan tietty määrä tavuja, joista käydään läpi tavu kerrallaan haluttujen arvojen löytämiseksi. Ensin datan lukeminen synkronisoidaan, jotta yksittäinen datapaketti saataisiin luettua kerralla kokonaan. Synkronointi tehdään vertaamalla luettujen tavujen arvoja toivottuihin arvoihin ja arvojen ollessa kohdallaan siirrytään lukemaan mittausarvot sisältävät tavut. Kun mittausarvot on luettu, ne annetaan `EKGSurfaceView`-luokalle, jossa ne piirretään näytölle. Anturin rajapinta on salaista tietoa, joten sitä ei käsitellä tämän tarkemmin.

6.5 Näytölle piirtäminen

Androidissa näytölle piirtämiseen on mahdollista käyttää joko Canvas-luokkaa tai SurfaceView-luokkaa. SurfaceView-luokka on View-luokan alaluokka ja se on täysin omistettu piirtämiselle. Sovelluksessa päädyttiin käyttämään SurfaceView-luokkaa, joka on parempi vaihtoehto, jos näyttöä joudutaan päivittämään jatkuvasti. Toinen syy SurfaceView-luokan valitsemiseen oli se, että Canvas-luokassa ei saa laitettua suoraan nappeja tai muita widgeteja. SurfaceView-luokan näytölle voidaan lisätä nappeja ja tekstilaatikoita ja määrittää näiden sijoittelu XML-tiedostossa.

SurfaceView hoitaa piirtämisen omassa säikeessään, joten sen ei tarvitse odottaa näkymähierarkian valmiutta piirtämiseen. SurfaceView-luokassa piirtäminen toteutetaan kutsumalla `getHolder()`-metodia, joka palauttaa `SurfaceHolder`-rajapinnan. Tämän rajapinnan kautta päästään käyttämään SurfaceView-luokan näytön alla olevaa pintaa. Tämän jälkeen luodaan uusi säie, jolle rajapinta annetaan parametrina. Säikeessä hoidetaan EKG-datan piirtäminen sekä mittauksen tietojen päivitys näytölle. Kuvassa 14 näkyy sovelluksen mittausnäyttö.



KUVA 14. Mittausnäyttö

Säikeessä piirtäminen tehdään kutsumalla `lockCanvas()`-metodia, joka palauttaa näytön pinnan, jota voidaan käyttää piirtämiseen. EKG-käyrä piirretään ottamalla vastaan `BtService`-luokan lukemat arvot ja piirtämällä viivat näiden pisteiden väliin. X-akselin arvoa kasvatetaan aina seuraavalle y-akselin arvolle ja edellisen viivan päätepisteen arvoa käytetään uuden viivan alkupisteenä. X-akselin arvon kasvettua suureksi se palautetaan takaisin alkuarvoon, jolloin piirtäminen näytölle alkaa alusta. Kun piirtäminen näytölle on tehty, kutsutaan `unlockCanvasAndPost()`-metodia, joka päivittää muutokset näytölle. `lockCanvas()`- ja `unlockCanvasAndPost()`-metodeita kutsutaan jokaiselle piirrettävälle arvolle.

7 TULOSTEN TARKASTELU

7.1 Työprosessin tarkastelua

Aiheen varmistumisen jälkeen työprosessi alkoi Android-käyttöjärjestelmään tutustumisella. Työtä aloitettaessa tietoa kyseisestä käyttöjärjestelmästä tai sen toiminnasta ei juuri ollut. Kokemusta mobiiliohjelmoinnista minulla ei juuri ollut, ja Java-kielikin vaati hieman muistiin palauttamista. Android-ohjelmoinnissa käytetty Java-kielen variaation opetteleminen tuotti alussa jonkin verran ongelmia, mutta sovellusesimerkkiä kautta kieleen tutustuminen onnistui melko helposti.

Työn alussa Mega Elektroniikka Oy antoi dokumentin, joka sisälsi halutut toiminnot sekä toiveet sovelluksen käyttöliittymästä. Tämä vähensi suunnitteluun tarvittua aikaa.

7.2 Ongelmia ja niiden tarkastelua

Jälkeenpäin voidaan sanoa, että sovelluksen suunnitteluvaihe jäi hieman liian vaajaaksi ja toteutusvaiheessa joutui tekemään muutoksia luokkiin, jotta toiminnot saatiin toimimaan. Myös uudenlaisen ohjelmointikielen käyttö tuotti alussa jonkin verran ongelmia. Vaikka Androidin ohjelmointikieli onkin Java-kieltä, sen muokattuihin kirjastoihin tutustuminen vaati paljon totuttelua. Myöskään Androidin tukemat XML-tyylitiedostojen käytöstä ei ollut ennestään kokemusta, mutta näiden käyttö oli varsin yksinkertaista ja käyttöliittymän rakentaminen näiden avulla oli helppoa.

Teknisellä tasolla suurimmat ongelmat olivat sovelluksen toteuttamisessa sekä Bluetoothin käyttämisessä. Ensimmäinen ongelma ilmeni Eclipsessä jo heti sovelluksen ohjelmoinnin aloittamisen jälkeen. Jostain vieläkin tuntemattomasta syystä Eclipse silloin tällöin kieltäytyi siirtämästä sovelluksen uutta versiota emulaattoriin. Normaalisti Eclipse asentaa sovelluksen uuden version ajohetkellä emulaattorin puhelimeen ja suorittaa sen automaattisesti, mutta toisinaan tämä ei tapahtunut. Ainoa toimiva ratkaisu, jonka löysin, oli projektin luominen uudelleen. Kaikkien tiedostojen sisällön kopioimisen jälkeen uusi projekti toimi normaalisti emulaattorissa.

Ongelmat Bluetoothin kanssa alkoivat, kun huomattiin, että emulaattori ei tuekaan Bluetoothia. Ainoa mahdollisuus oli saada Android-laite, jossa sovellusta voisi testata. Hieman myöhemmin sain Android Developer Phonen lainaan Savonia AMK:n IT-kehitysyksiköstä, mutta myöhemmin huomasin, että tässä Dev Phonessa oli käytössä

liian vanha alustan versio. Projektia luotaessa Eclipsessä valitaan API-taso, jolle sovellus tehdään. Puhelimessa oli käytössä tason 3 API, josta kyllä löytyy tuki Bluetoothille, mutta tämän tason API:ssa ei ollut saatavilla Bluetoothin paketteja käytettäviksi ohjelmoinnissa. Ensimmäinen versio, josta löysin Bluetooth-paketit, oli 6 mutta Dev Phoneen ei ollut mahdollista päivittää näin uutta versiota. Ratkaisu tähän löytyi pitkän etsimisen jälkeen, kun vahingossa löysin Backport-paketin Bluetoothille. Tämä paketti lisäsi Bluetooth-toiminnot matalammalle API-tasolle, jolloin sovelluksen pystyi tekemään käyttämällä API-tasoa, jota Dev Phone tuki.

Bluetoothin toimimaan saamisesta minulla oli jo ennestään pahoja muistoja ja aivan helpolla sen toimimaan saaminen ei onnistunut Androidissakaan. Suuri osa ohjelmointiin käytetystä ajasta meni yrittäessä saada Bluetooth-yhteys toimimaan, mutta myös tiedon lukemisen ja piirtämisen toimimaan saamiseen meni paljon aikaa. Bluetooth-laitteiden etsimisen sai toimimaan suhteellisen helposti valmiin esimerkin pohjalta, mutta laitteeseen yhdistämisessä tuli ongelmia, sillä Android ei antanut keinoa syöttää PIN-koodia sovelluksen kautta. Kiertotie tähän löytyi kuitenkin parittamalla laitteet puhelimen asetuksista, jolloin yhdistäminen sovelluksessa onnistui tarvitsematta syöttää PIN-koodia.

Suurin ongelma tuli vastaan Bluetooth-anturin lähettämän datan lukemisessa oikealla tavalla. Oikeiden tavujen lukeminen ei tahtonut onnistua yhdelläkään yritetyllä tavalla, vaikka pystyin katsomaan esimerkkiä toisesta sovelluksesta, joka luki ne oikein.

Työprosessista ja ongelmista oppi paljon seuraavalle kerralle. Suunnitteluvaiheeseen pitäisi käyttää enemmän aikaa, jotta myöhemmin toteutusvaiheessa ei tarvitsisi tehdä suuria muutoksia. Ennen toteutusvaihetta olisi hyvä myös tietää, mitä tekniikoilla on mahdollista toteuttaa ja miten näiden toteutus tapahtuu. Esimerkkinä tästä on piirtämisen toteuttaminen, joka ensin toteutettiin Canvas-luokkaa käyttämällä, mutta loppuvaiheessa se vaihdettiin SurfaceView-luokkaan sen paremmin soveltuvien toimintojen seurauksena. Myös kehitysympäristöön olisi hyvä tutustua paremmin, jotta siihen liittyvät mahdolliset ongelmat saataisiin minimoitua.

7.3 Tavoitteiden täyttyminen

Opinnäytetyön tavoitteena oli toteuttaa Android-laitteessa toimiva EKG-mittaussovellus, joka hyödyntää Mega Elektronikka Oy:n Bluetooth-anturia. Sovelluksella oli seuraavat tavoitteet:

- sovellus vastaanottaa anturin lähettämää EKG-raakadataa käyttäjän valitsemalla näytteistyksellä.
- sovellus piirtää EKG-signaalin näytölle reaaliaikaisena ja näyttää R-R-arvoista lasketun sykelukeman.
- sovellus tallentaa vastaanotetun datan muistikortille.

Sovellukselle asetetut tavoitteet toteutuivat tyydyttävästi. Sovellus tarjoaa käyttäjälle yksinkertaisen ja selkeän käyttöliittymän, jonka käyttäminen on helppoa.

Luetun datan näytölle piirtäminen ja tallentaminen muistikortille on vielä parannuksen tarpeessa. Alkuvaiheessa tavoitteena oli tallentaa data EDF+ -formaattissa, mutta myöhemmin sovittiin, että normaali tekstitiedosto riittäisi. Nämä puutteet on kuitenkin helppo korjata mahdollisessa jatkokehityksessä.

7.4 Jatkokehitys ja tulevaisuudennäkymät

Jatkokehityksellä tästä sovelluksesta on mahdollista saada markkinoille toimiva ja kaupallisesti potentiaalinen Android-sovellus. Android-laitteet jatkavat yleistymistään ja myös sykemittareiden suosio on kasvussa, mikä tekee sykemittausohjelman kehittämisestä Android-alustalle varteen otettavanmahdollisuuden.

Työn tuloksena syntynyt sovellus kaipaa vielä jonkin verran kehitystyötä, ennen kuin se on valmis markkinoille. Varsinkin piirtäminen ja tiedostoon tallentaminen tarvitsevat jonkin verran kehittämistä. Myös virnehallinnasta löytyy varmasti parannettavaa.

Sovellusta oli varsin mielenkiintoista tehdä, ja projekti opetti paljon ohjelmistokehityksestä sekä mobiiliohjelmoinnista ja erilaisista ongelmatilanteista.

LÄHTEET

Bluetooth: Facts [verkkosivu]. [viitattu 16.2.2011]. Saatavissa:

<http://www.absoluteastronomy.com/topics/Bluetooth>

EKG - Suomen Sydänliitto ry [verkkosivu]. [viitattu 16.2.2011]. Saatavissa:

<http://www.sydanliitto.fi/ekg>

Biomedical Engineering | ECG Assignment [verkkosivu]. [viitattu 16.2.2011]. Saata-

vissa: <http://eleceng.dit.ie/tburke/biomed/assignment1.html>

Steele, J. & Nelson, T. 2010, The Android Developer's Cookbook. Crawfordsville, Indiana: Pearson Education, Inc.

Burnette, E. 2008, Hello, Android - Introducing Google's Mobile Development Platform December 2008, The Pragmatic Bookshelf.

Android API Levels | Android Developers [verkkosivu]. [viitattu 18.3.2011]. Saatavis-

sa: <http://developer.android.com/guide/appendix/api-levels.html>

Activity | Android Developers [verkkosivu]. [viitattu 24.3.2011]. Saatavissa:

<http://developer.android.com/reference/android/app/Activity.html>

www.savonia.fi

