



Toni Heikkinen ja Aki Nikula

ANDROID - VIDEONHALLINTA

ANDROID - VIDEONHALLINTA

Toni Heikkinen ja Aki Nikula
Opinnäytetyö
Syksy 2012
Tietojenkäsittelyn koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma

Tekijä(t): Toni Heikkinen ja Aki Nikula

Opinnäytetyön nimi: Android - Videonhallinta

Työn ohjaaja(t): Pekka Ojala

Työn valmistumislukukausi ja -vuosi: Syksy 2012

Sivumäärä: 42

Opinnäytetyön tavoitteena oli luoda Android-käyttöjärjestelmässä toimivaan ResultCodeMobile-ohjelmaan videonhallintamoduuli, jolla voidaan katsoa otteluvideoleikkeitä tietyistä videon kohdista. Työ tehtiin, koska videonhallintamoduulin uskotaan tuovan käyttäjälle suurta hyötyä analysoitaessa ottelua. Lisäksi videonhallintamoduuli antaa ResultCodeMobile-ohjelmalle suuren lisämarkkina-arvon. Työn toimeksiantajana oli Code4m Oy.

Työn tuloksena syntyi toimiva videonhallintamoduuli, jolla käyttäjä voi katsella haluamiaan otteluvideoleikkeitä. Videonhallintamoduulissa käyttäjä voi rajata näkyvillä olevia tapahtumia tapahtuma-, pelaaja- sekä aikarajauksella.

Raportin tietoperusta pohjautuu lähinnä Internetissä oleviin lähteisiin, koska Android on niin uusi asia, ettei siitä ole vielä monia kirjoja kirjoitettu. Tietoperustana käytettiin myös muutamia kirjoja, jotka kertoivat esimerkiksi Java-kielestä. Videonhallintamoduuli ohjelmoitiin työaikana Code4m Oy:n tiloissa. Ohjelman ohjelmoimiseen käytettiin Java-, SQL-, SQLite-, PHP- ja XML-kieliä.

Opinnäytetyö onnistui hyvin. Työhön sisällytettiin kaikki toiminnallisuudet, jotka toimeksiantaja halusi. Lisäksi tulevaisuutta ajatellen tuli paljon uusia kehitysideoita, joita jatkossa tullaan lisäämään videonhallintamoduuliin.

Asiasanat: Android, ResultCode, Video, Jalkapallo, Videonhallinta

ABSTRACT

Oulu University of Applied Sciences
Degree Programme In Business Information Systems

Author(s): Toni Heikkinen and Aki Nikula

Title of thesis: Android - Videonhallinta

Supervisor(s): Pekka Ojala

Term and year when the thesis was submitted: Autumn 2012

Number of pages: 42

The goal of this thesis was to create a video control module for ResultCodeMobile program which works using Android Operation System. In the program there is possibility to watch certain video events from video of the football game. The project was done, because module is expected to bring great new benefits when analyzing the match. Module will also give a great additional market value for ResultCodeMobile application. Employer of this thesis was Code4m Oy.

The result of the project was a working video control module, where user can watch events from the video. In this video control module user can limit the events displayed by type, player or time of the event.

The information of the report is based on the Internet sources, because Android is such a new thing and there is not many books available yet. A few books has been used, which contained information for example about Java programming language. Module was created on work time in office of Code4m Oy. Application was programmed by using Java, SQL, SQLite, PHP and XML programming languages.

The thesis was successful. Video control module achieved every function what employer wanted. There were also many new development ideas for the future, and they will be added later in to the module.

Keywords: Android, ResultCode, Video, Football, Video control.

Sisällys

1 JOHDANTO	6
2 CODE4M OY	7
2.1 ResultCode	7
2.2 ResultCodeMobile	8
3 ANDROID	9
3.1 Historia	9
3.2 Versiot	10
3.3 Ympäristöt	12
3.4 Rakenne	16
4 VIDEONHALLINTA	30
4.1 Ohjelman kuvaus	30
4.2 Ohjelman toteutus	33
4.3 Käyttöliittymä	34
4.4 Ohjelmakoodi ja tietokannat	35
5 POHDINTA	38

1 JOHDANTO

Opinnäytetyön aiheena on Android - Videonhallinta. Tavoitteena oli luoda Android-käyttöjärjestelmässä toimivaan ResultCodeMobile-ohjelmaan videonhallintamoduuli, jolla voidaan katsella videoitujen otteluiden videoleikkeitä tietyistä ottelun kohdista.

ResultCodeMobile-ohjelma on laaja jalkapallovalmentajan työkalu, jolla voidaan hallita oman jalkapallojoukkueen asioita nopeasti sekä luotettavasti. Tätä opinnäytetyötä vasten ajateltuna ResultCodeMobilessa on myös toimintoja ottelutapahtumien sekä pelaajaseurantatilastojen tallentamiseen, joita opinnäytetyönä tehdyllä videonhallintamoduulilla voidaan katsella.

Videonhallintamoduulin päätarkoituksena on, että käyttäjä voi katsoa videolta haluamansa videoleikkeet. Käyttäjän ei myöskään tarvitse itse etsiä videosta kohtia, jolloin esimerkiksi maalit on syntynyt, vaan ohjelma etsii tapahtumat automaattisesti käyttäjän asettamien rajausten mukaan. Opinnäytetyö tehtiin Code4m Oy:lle ja tulevaisuudessa se otetaan käyttöön osana ResultCodeMobile-ohjelmistoa.

Opinnäytetyön ensimmäinen luku sisältää johdannon. Luvussa kaksi kerrotaan Code4m Oy:stä, sen perustiedoista sekä tuotteista. Siinä mainitaan myös Code4m Oy:n tuotteista, ResultCodesta ja ResultCodeMobilesta. Luvussa kolme on tietoa Androidista, sen historiasta sekä eri versioista ja versioiden tuomista uusista päivityksistä ja ominaisuuksista. Lisäksi kerrotaan tärkeimmistä Androidin kehitysympäristöistä, kuten Eclipse ja NetBeans. Loppupuolella tarkastellaan ohjelmointirakennetta ja sitä, mistä rakenne koostuu sekä tärkeimmistä komponenteista. Luvussa kerrotaan myös ohjelmointikielistä ja tietokannoista. Neljännessä luvussa käsitellään videonhallintamoduulia ja viimeisessä on pohdinta.

2 CODE4M OY

“Code4m Oy on vuoden 2003 alussa perustettu teknologiayritys, josta löytyy liki 30:n vuoden kokemus tietotekniikasta, sen asiakaslähtöisestä soveltamisesta sekä nykyaikaisen teknologian viemisestä lähelle ihmisiä” (Code4m Oy, hakupäivä 11.10.2012).

Maaliskuussa 2013 Code4m Oy täyttää kymmenen vuotta. Code4m Oy on oululainen pienyritys, jonka alaisuudessa työskentelee tällä hetkellä kolme työntekijää, mukaan lukien toimitusjohtaja itse. Code4m Oy:n päätuote on tulos- ja tietopalvelu ResultCode. (Code4m Oy, hakupäivä 11.10.2012.)

2.1 ResultCode

ResultCode on Code4m Oy:n mukaan markkinoiden monipuolisin ja edullisin Internetissä toimiva liitto-, piiri-, seura- sekä joukkue- ja tulos- ja tietopalvelu. ResultCode on tällä hetkellä käytössä jalkapallossa ja futsalissa. Tuote on kehitetty yhteistyössä Palloliiton piiriorganisaatioiden kanssa. ResultCode tehostaa viestintää ja vähentää merkittävästi päivittäisiä rutiineja sekä niihin kuluva-aikaa. (Code4m Oy, hakupäivä 11.10.2012.)

ResultCode-tulospalvelu ottaa vastaan tekstiviestinä lähetetyt ottelun tulosilmoitukset, laskee sarjataulukot ja julkaisee ne Internet-sivuille automaattisesti. Tulospalvelun avulla ottelutulosten käsittely tapahtuu hetkessä. Tulospalvelu sisältää myös muun muassa otteluohjelmat sekä otteluihin liittyvät kenttä- ja erotuomaritiedot. (Code4m Oy, hakupäivä 11.10.2012.)

2.2 ResultCodeMobile

ResultCodeMobile on Android-käyttöjärjestelmälle toteutettu jalkapallovalmentajan työkalu. Ohjelma pohjautuu ResultCode-palveluun. Pääosin ResultCodeMobile on tarkoitettu käytettäväksi taulutietokoneilla eli tabletilaitteilla, mutta ohjelmaa on mahdollista käyttää muillakin Android-käyttöjärjestelmän omaavilla mobiililaitteilla.

ResultCodeMobilen kehitystyö aloitettiin vuonna 2011. Sen ensimmäinen versio (0.9) julkaistiin 2.11.2012 beta-versiona. Sovellus on toteutettu ja ohjelmoitu käyttäen ohjelmointikielenä Javaa. ResultCodeMobile on käännetty kolmelle eri kielelle, jotka ovat englantia, suomi ja ruotsi.

ResultCodeMobilella käyttäjä voi hoitaa oman joukkueensa asioita nopeasti ja luotettavasti. Sovelluksella voi luoda, muokata ja tarkastella ottelun kokoonpanoja, pelijärjestelmiä, taktiikoita sekä pelikirjoja milloin ja missä tahansa. Sovelluksen täydellisen toiminnan takaamiseksi on suositeltavaa käyttää langatonta verkkoyhteyttä. Käyttäjällä on mahdollisuus taltioida otteluista ottelutapahtumat, pelaajaseurantatilat ja ottelun joukkueiden väliset pallonhallintatiedot. Näistä tallennetuista tiedoista sovellus muodostaa joukkue- ja pelaajakohtaiset otteluraportit. Kaikki ResultCodeMobilessa tallennettu tieto on mahdollista tallentaa ulkoiseen tietokantaan, jota myös ResultCode-palvelu käyttää. Tämä mahdollistaa tiedon hyödyntämisen myös tulevaisuudessa.

Valmentajan näkökulmasta katsottuna ResultCode-palvelun tietokannasta saatavan tiedon uudelleen hyödyntäminen on tehokasta. Jokaisen ottelun analysointi onnistuu huomattavasti aikaisempaa helpommin, mikäli ottelusta on kerätty ResultCodeMobile-sovelluksella tietoja. Sovelluksella muodostetut otteluraportit ovat kattavia ja selkeitä, joten niistä näkee heti millä osa-alueella joukkueet ovat hallinneet. Videonhallintamoduulin avulla kaikki kerätty tieto saa aivan uuden merkityksen, koska kerättyyn tietoon voidaan liittää helposti myös liikkuvaa kuvaa.

3 ANDROID

Android on mobiililaitteille suunniteltu ohjelmistopino, joka sisältää käyttöjärjestelmän lisäksi väliohjelmistoja ja käyttäjän perusohjelmia. Se perustuu avoimen lähdekoodin GPLv2 lisensoituun Linux-käyttöjärjestelmäyttimeen. (Wikipedia: Android 2012, hakupäivä 11.10.2012.)

Androidia kehitti alun perin Android Inc., jonka Google myöhemmin osti itselleen. Nykyisin Androidin kehittämisestä vastaa Open Handset Alliance, joka on vuonna 2007 perustettu Googlen johtama liittouma (Wikipedia: Open Handset Alliance 2012, hakupäivä 11.10.2012). Android-älypuhelimia kehittävät useimmat suuret valmistajat, kuten HTC, LG, Samsung, ja Sony sekä joukko pienempiä yrityksiä. Myös monet perinteisemmät valmistajat, kuten esimerkiksi Asus, ovat alkaneet kehittää Android-laitteita. (Androidsuomi: Mikä on Android?, hakupäivä 11.10.2012.)

Android-ohjelmoinnissa käytetään Java-kieltä ja Googlen kehittämiä Java-kirjastoja. Tiedon tallentamiseen ohjelmia varten käytetään SQLite-relaatiotietokantaa. Kehitysympäristössä on laite-emulaattori, virheiden jäljittämistyökaluja, muistia ja suorituskykyä mittaavia työkaluja sekä liitännäinen Eclipse-ohjelmointiympäristölle. (Wikipedia: Android 2012, hakupäivä 11.10.2012.)

3.1 Historia

Android-käyttöjärjestelmä julkistettiin 5. marraskuuta 2007 (Wikipedia: Android 2012, hakupäivä 11.10.2012). NPD Group -tutkimusyhtiön mukaan Android nousi Yhdysvaltain toiseksi myydyimmäksi mobiilialustaksi vuoden 2010 ensimmäisen neljänneksen aikana (Melanson D 2010, hakupäivä 11.10.2012). Vuoden 2010

viimeisellä neljänneksellä Android nousi älypuhelinten markkinajohtajaksi (Virki & Carew 2011, hakupäivä 11.10.2012).

Markkinatutkimusyhtiö IDC:n mukaan vuoden 2012 kolmannella vuosineljänneksellä kolme neljästä ostetusta älypuhelimesta oli Googlen Android-käyttöjärjestelmään pohjautuvia. Yhteensä Android-käyttöjärjestelmän sisältäviä älypuhelimia ostettiin samalla neljänneksellä 136 miljoonaa kappaletta. (Koivunen 2012, hakupäivä 11.10.2012.)

3.2 Versiot

Ensimmäinen kaupallinen Android-versio (1.0) julkaistiin markkinoille 23. syyskuuta 2008 (Morrill, hakupäivä 11.10.2012). Androidista on julkaistu useita uusia versioita tämän julkistuksen jälkeen. Yleensä uusiin versioihin korjataan aiempien versioiden virheitä ja lisätään uusia ominaisuuksia. Tässä luvussa kerrotaan Androidin eri versioista ja niiden tuomista päivityksistä, jotka ovat videonhallinnan ohjelmointityön tai käyttämisen kannalta tärkeitä.

Virallinen 1.5 (Cupcake) -versiopäivitys Androidille julkistettiin 30. huhtikuuta 2009. Päivitys sisälsi uusia ominaisuuksia ja käyttöliittymäpäivityksiä, muun muassa videoiden tallentamisen ja katsomisen videokameramoodilla, uuden tekstin ennustamista hyödyntävän ohjelmistonäppäimistön, automaattisen Bluetooth-yhteyden muodostamisen tietyllä välimatkalla, uusia widget-elementtejä (käyttöliittymäelementtejä), kansioita kotiruutuun ja animoituja ruutusiiirtymiä. (Wikipedia: Android 2012, hakupäivä 11.10.2012.)

Versiopäivitys 1.6 (Donut) -ohjelmistokehityspaketti julkistettiin 15. syyskuuta 2009. Päivitys sisälsi parannetun Android Market -ohjelmakaupan, integroidun kameran, videokameran sekä kuvagallerian käyttöliittymän. Päivitys tuki WVGA-

näyttöresoluutioita ja sen haku- ja kameran sovellukset olivat nopeampia. (Wikipedia: Android 2012, hakupäivä 11.10.2012.)

Versiopäivitys 2.0 (Eclair) -ohjelmistokehityspaketti julkistettiin 27. lokakuuta 2009. Päivitys sisälsi optimoidun laitteistokiihdytyksen, tuen useimpiin näytön kokoihin ja resoluutioihin, uuden muokatun käyttöliittymän sekä HTML5-tuen selaimelle, sisäisen välähdystuen kameralle, digitaalisen zoomauksen ja MotionEvent-luokan, joka pystyy käsittelemään monikosketustapahtumia. Lisäksi päivityksen mukana tuli parannettu virtuaalinen näppäimistö. (Wikipedia: Android 2012, hakupäivä 11.10.2012.)

Versiopäivitys 2.2 (Froyo) -ohjelmistokehityspaketti julkistettiin 20. toukokuuta 2010. Päivityksen mukana tuli yleisiä Androidin nopeus-, muisti- ja suorituskykyoptimointeja, Just-in-time -toteutus, joka vauhdittaa sovelluksia sekä parannettu sovelluskäynnistäjä, jossa on linkkejä puhelin- ja selainsovellukseen. Se sisälsi myös parannetun Market-sovelluksen, jossa oli ohjelmien automaattinen päivitys, nopean siirron eri näppäimistön kielten välillä sekä tuen numeerisille ja aakkosnumeerisille salasanoille. (Wikipedia: Android 2012, hakupäivä 11.10.2012.)

Versiopäivitys 4.0 (Ice Cream Sandwich) julkaistiin 19. lokakuuta 2011. Päivitys toi mukanaan paremman tekstinsyötön ja oikoluvun. (Wikipedia: Android 2012, hakupäivä 11.10.2012.) Versiopäivitys 4.1 (Jelly Bean) julkaistiin 27. kesäkuuta 2012. Muutoksia tässä päivityksessä olivat muun muassa kielituki, käyttäjän asennettavissa olevat näppäimistökartat, parannettu kameran sovellus, monikanavaäänet, USB-ääni (ulkoisille äänilähteille) ja äänien tauoton toistaminen. (Wikipedia: Android 2012, hakupäivä 11.10.2012.)

3.3 Ympäristöt

Eclipse IDE

IDE on lyhenne sanoista "Integrated Development Environment". Suomeksi tämä tarkoittaa ohjelmointiympäristöä. Yksinkertaisimmillaan ohjelmointiympäristö sisältää tekstieditorin ja ohjelmointikielen kääntäjän. Lähdekoodia tuotetaan tekstieditorilla, joka käännetään komentorivikäntäjällä. (Wikipedia: Ohjelmointiympäristö 2012, hakupäivä 11.10.2010.)

Eclipse IDE on ohjelmointiympäristö, joka tukee eri ohjelmointikieliä. Eclipsen tukemia ohjelmointikieliä ovat muun muassa Java, C, C++ ja PHP. IBM alkoi kehittää Eclipseä vuonna 1993. Vuonna 2001 se julkaistiin avoimen lähdekoodin lisenssillä. Vuodesta 2004 lähtien ohjelman kehityksestä on vastannut Eclipse Foundation säätiö. (Wikipedia: Eclipse (IDE) 2012, hakupäivä 11.10.2010.)

Eclipse on toteutettu pääasiassa käyttäen Java-ohjelmointikieltä. Eclipse ei kuitenkaan käytä samoja käyttöliittymäkirjastoja kuin Java, vaan sille on toteutettu oma kirjasto. Tästä syystä Eclipse ei ole täysin alustariippumaton sovellus. Eclipsen kehitysympäristöstä on olemassa useita erilaisia jakelupaketteja, jotka on räätälöity eri kehittäjien tarpeisiin. (Wikipedia: Eclipse (IDE) 2012, hakupäivä 11.10.2010.)

Eclipsen valmistelu Android-ohjelmointiin on käyttäjäystävällinen prosessi. Ensimmäiseksi tietokoneelle ladataan ja asennetaan Eclipse IDE -sovellus. Tämän jälkeen ladataan Android SDK (Software Development Kit), jonka jälkeen asennetaan Eclipsen tarvitsema ADT-lisäosa (Android Development Tools). Viimeiseksi ladataan uusimmat SDK-työkalut, -päivitykset sekä -alustat käyttämällä SDK-hallintatyökalua (SDK Manager). Vasta tämän jälkeen Eclipse on valmis Android-ohjelmointiin. (Android Developers: Building Your First App, hakupäivä 11.10.2012.)

NetBeans IDE

NetBeans-projekti alkoi Prahassa vuonna 1996 Kaarlen yliopistossa, josta se kehittyi myös kaupalliseksi tuotteeksi. Sun Microsystems -yhtiö osti NetBeansin kehittämiseen keskittyneen yrityksen ja julkaisi sen avoimen lähdekoodin projektina vuonna 2000. NetBeans on ohjelmointiympäristö ohjelmointikielille Java, JavaScript, PHP, C ja C++. NetBeans on kehitetty Java-ohjelmointikielellä, joten sen käyttö vaatii Javan ajoympäristön (JRE, Java Runtime Environment). Java-ohjelmien kehittämiseen tarvitaan myös Java-kehitystyökalut (JDK, Java Development Kit). (Wikipedia: NetBeans 2012, hakupäivä 11.10.2010.)

NetBeans on helposti muutettavissa Android-ohjelmointia varten. Ensin ladataan ja asennetaan NetBeans IDE -sovellus. Tämän jälkeen ladataan ja asennetaan lisäosat käyttäen hyväksi NetBeansin omaa lisäosan lataus- ja asennusosiota. Sitten lisätään Android-ohjelmointialusta ja sen työkalut NetBeansiin, käyttäen Android SDK- ja AVD-hallintatyökaluja. (Jackson, hakupäivä 11.10.2012.)

Debugger

Debugger on työkalu konekielisen ohjelman vianetsintään. Sen avulla voidaan käydä läpi ohjelmaa komento kerrallaan tai tiettyyn kohtaan asti, tutkia rekistereitä ja keskusmuistin sisältöä. Sillä voidaan myös täyttää tai tyhjentää muistialueita. (Järvinen 2004, 42.)

Android SDK

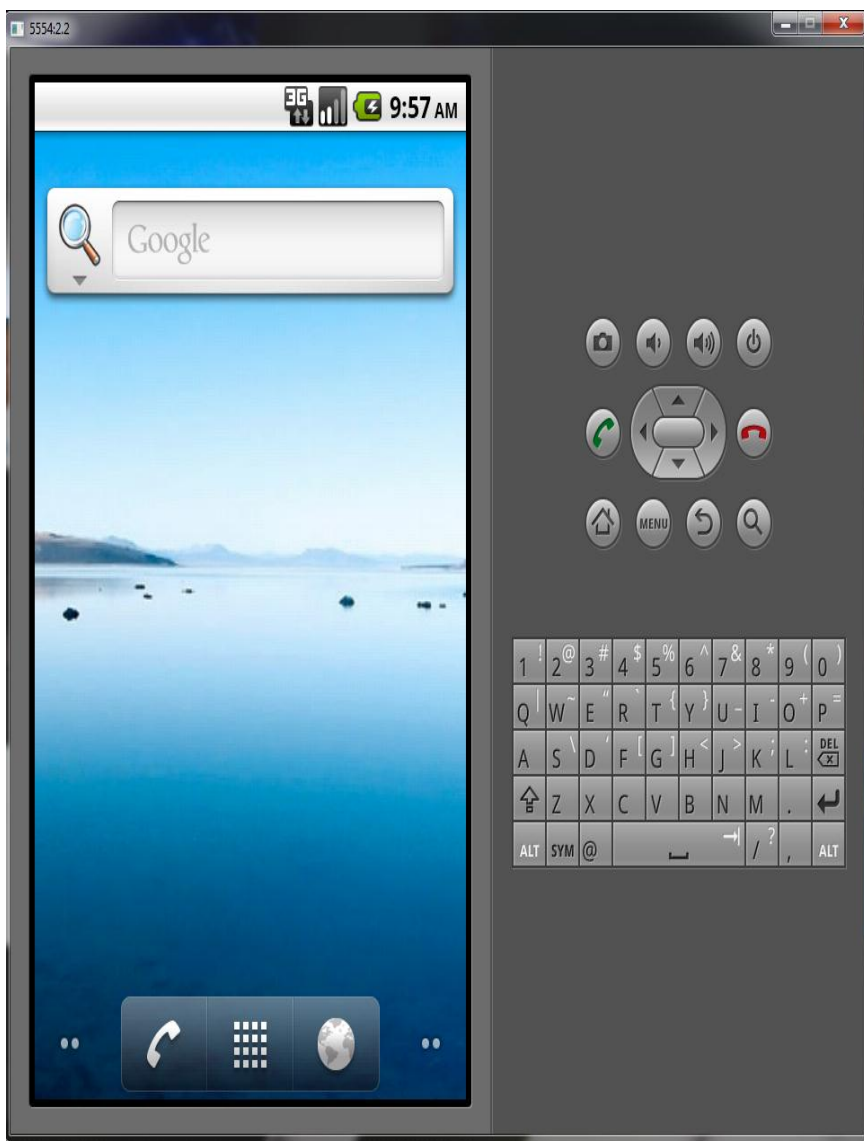
Android SDK tarjoaa ohjelmointikehityksessä tarvittavat työkalut, kuten esimerkiksi API-kirjastot, debuggerin ja dokumentteja Android-ohjelmoinnista (Wikipedia: Android software development 2012, hakupäivä 11.10.2010). Näitä tarvitaan Android-ohjelmien rakentamisessa, testaamisessa ja virheiden etsinnässä (Android Developers: Get the Android SDK, hakupäivä 11.10.2010).

AVD

AVD (Android Virtual Device) on emulaattorikokoonpano, jolla voidaan mallintaa oikeaa laitetta. Emulaattorille määritellään laitteisto ja ohjelmistot, joita se sitten jäljittelee. (Android Developers: Managing Virtual Devices, hakupäivä 11.10.2012.)

AVD:lle voidaan asettaa sen jäljittelemä laitteisto. Sille voidaan esimerkiksi määrittää, onko emulaattorilla kameraa, onko laitteella fyysistä QWERTY-näppäimistöä ja kuinka paljon muistia sillä on käytössä. AVD:lle voidaan valita ajettavan Android-käyttöjärjestelmän versio, asettaa näytön koko ja ulkomuoto, sekä mahdollinen muistikortti ja sen koko. AVD-kokoonpanolla voidaan myös tallentaa sen tiedot ohjelmoinnissa käytettävälle koneelle. (Android Developers: Managing Virtual Devices, hakupäivä 11.10.2012.)

AVD-kokoonpanojen lukumäärää ei ole rajoitettu. Niitä voi olla sekä luotuna että käytössä niin monta kuin niitä tarvitaan. (Android Developers: Managing Virtual Devices, hakupäivä 11.10.2012.) Kuviossa yksi on näkymä Android-käyttöjärjestelmästä, jota ajetaan emulaattorilla.



KUVIO 1. Android-käyttöjärjestelmä emulaattorissa

ADT Plugin

ADT (Android Development Tools) on Eclipse IDE:n lisäosa, joka on suunniteltu antamaan tehokas kehitysympäristö, jossa voidaan kehittää Android-ohjelmistoja. Se laajentaa muun muassa Eclipsen kykyjä, mikä auttaa tuottamaan nopeasti uusia Android-projekteja, lisäämään uusia API-paketteja, sekä etsimään virheitä ohjelmista käyttämällä hyväksi Android SDK työkaluja. ADT:n avulla voidaan myös luoda allekirjoitettuja tai allekirjoittamattomia .apk-asennuspaketteja, joiden avulla

voidaan levittää tuotettua Android-ohjelmaa eteenpäin. (Android Developers: ADT Plugin, hakupäivä 11.10.2012.)

Android-ohjelmistojen kehitys ADT:tä apuna käyttäen on suositeltava ja nopea tapa päästä alkuun ohjelmoimisessa. ADT sisältää esimerkiksi ohjatun uuden projektin luomisen, editorin graafiseen XML-tiedostojen muokkaamiseen sekä tulostusikkunan debuggaukseen. (Android Developers: ADT Plugin, hakupäivä 11.10.2012.)

3.4 Rakenne

Tässä luvussa perehdytään Android-ohjelman perusrakenteeseen. Luvussa käsitellään käyttöliittymää, ohjelmakoodia ja niiden ohjelmointiin tarvittavia ohjelmointikieliä.

3.4.1 Käyttöliittymä

Androidissa käyttöliittymämäärytykset tehdään XML-kielellä. Extensible Markup Language (XML) on yksinkertainen ja joustava tekstiformaatti, joka on johdettu metakielestä SGML (Standard Generalized Markup Language, ISO 8879). XML on alun perin suunniteltu vastaamaan laajojen sähköisten julkaisujen ongelmiin, mutta sillä on myös yhä suurempi rooli laaja-alaisessa tiedon vaihdossa Internetissä ja muualla. (W3C: Extensible Markup Language (XML) 2012, hakupäivä 11.10.2012.)

XML

XML esiteltiin ensimmäisen kerran vuonna 1997. Se perustuu jo 1960-luvulla kehitettyyn SQML-kieleen (Simplest Quantified Modal Logic). Se on rakenteisten dokumenttien kuvaamiseen tarkoitettu kuvauskieli. Sillä tarkoitetaan dokumenttia, joka sisältää sekä asiasisältöä että tietoa eri dokumentin osien suhteista toisiinsa.

XML-dokumenttien on tarkoitus olla helposti ihmisen ja koneen luettavissa (W3C: Extensible Markup Language (XML) 1.0 (Fifth Edition) 2012, hakupäivä 11.10.2012). XML on laajennettava kieli, joten dokumentin laatija voi halutessaan lisätä dokumenttiin omia elementtejä tarpeen mukaan. (Paananen 2004, 326.)

Yleisiä ulkoasukomponentteja

Android-ohjelmien ulkoasujen luomisessa erilaisten ulkoasukomponenttien (widget-elementtien) käyttö on nopea ja helppo tapa luoda uusia ohjelmien ulkoasuja. Seuraavassa listataan lyhyesti yleisiä komponentteja, joita käytetään Android-ohjelmoimisessa.

TextView on komponentti, joka näyttää sille asetettua tekstiä, jota voidaan muuttaa ohjelmallisesti. Komponenttiin asetettua tekstiä ei voida muokata suoraan komponentissa. (Android Developers: TextView, hakupäivä 11.10.2012.)

EditText on samankaltainen komponentti kuin TextView-komponentti. Sen erona on, että siihen asetettua tekstiä voidaan muokata suoraan itse komponentissa. (Android Developers: EditText, hakupäivä 11.10.2012.)

CheckBox (asetusnappi) on kaksiarvoinen komponentti. Se voi olla arvoltaan joko "checked" (valittu) tai "unchecked" (ei valittu). (Android Developers: CheckBox, hakupäivä 11.10.2012.) Käyttäjä voi vaihdella CheckBox-komponentin arvoa rajattomasti.

RadioButton (valintanappi) on komponentti, joka voi olla arvoltaan "checked" (valittu) tai "unchecked" (ei valittu). Kun yksittäistä, *ei valittua* RadioButtonia painetaan, sen arvo muuttuu *valituksi*. Käyttäjä ei voi muuttaa sitä enää takaisin *ei valituksi*. Yleensä RadioButtoneita käytetään yhdessä RadioGroupin kanssa. RadioGroup voi sisältää useita RadioButtoneita. Jos tässä tilanteessa painetaan *ei valittua* RadioButtonia, sen arvo muuttuu *valituksi* ja kaikkien muiden

RadioButtonien arvo muuttuu *ei valituiksi*. (Android Developers: RadioButton, hakupäivä 11.10.2012.) Toisin sanoen, käyttäjä voi valita, mikä RadioGroupin sisällä olevista RadioButton-komponenteista on *valittu*.

ListView-komponentti on listanäkymä, jossa voidaan näyttää erilaisia tietoja listanomaisesti. Komponentissa olevat tiedot näytetään pystysuunnassa. (Android Developers: ListView, hakupäivä 11.10.2012.)

ImageView on komponentti, joka näyttää sille asetetun kuvan. Kuvan koko skaalautuu automaattisesti ImageViewin kokoiseksi. (Android Developers: ImageView, hakupäivä 11.10.2012.)

Button-komponentti kuvastaa painallusnappia. Käyttäjä voi painaa painallusnappia kerran tai painaa sen pohjaan, jolloin napille asetettu toiminto toteutetaan. (Android Developers: Button, hakupäivä 11.10.2012.)

AlertDialog on dialog-ikkuna, johon voidaan asettaa maksimissaan kolme erilaista toimintanappia. Toimintanapit ovat PositiveButton, NeutralButton ja NegativeButton. AlertDialog-ikkunaa käytetään yleensä ilmoittamaan käyttäjälle virheet virhetilanteissa, mutta niitä voidaan käyttää myös muunlaisissa tilanteissa. (Android Developers: AlertDialog, hakupäivä 11.10.2012.)

AlertDialogiin voidaan asettaa otsikon lisäksi joko tekstiä, listoja tai näkymä (View), johon voidaan asettaa useita erilaisia komponentteja. Jälkimmäisillä tavoilla voidaan muokata AlertDialog toimimaan toisenlaisissa tehtävissä, eikä pelkästään virheilmoitusten tarkoituksissa. (Android Developers: AlertDialog, hakupäivä 11.10.2012.)

3.4.2 Ohjelmakoodi

Activity

Activity-komponentti on sovelluskomponentti, joka huolehtii näytöstä, minkä kanssa käyttäjä voi olla vuorovaikutuksessa (Android Developers: Activities, hakupäivä 11.10.2012). Melkein kaikki Activity-komponentit ovat vuorovaikutuksessa käyttäjänsä kanssa. Ne huolehtivat näkymien asettamisesta, jotka on luonut kyseiselle Activity-komponentille. Vaikka Activity-komponentti näytetään käyttäjälle yleensä koko näytön kokoisilla ikkunoilla, se voi olla myös AlertDialog-ulkoasukomponentin tapainen kelluva ikkuna tai sisältää myös useita toisia Activity-komponentteja. (Android Developers: Activity, hakupäivä 11.10.2012.)

Järjestelmän Activity-komponentteja hallinnoi niin sanottu "Activity-pakka". Kun uusi Activity-komponentti avataan, se asetetaan pakan päällimmäiseksi ja näin ollen siitä tulee käynnissä oleva komponentti. Edellinen Activity-komponentti joutuu aina pakan toiselle sijalle ja pysähtyy. (Android Developers: Activity, hakupäivä 11.10.2012.)

Activity-komponentilla on pääsääntöisesti neljä eri tilaa. Jos Activity-komponentti on ruudulla näkyvissä (pakan päällimmäisenä), komponentti on *käynnissä*. Jos Activity-komponentti on näkyvissä, mutta se ei ole vuorovaikutuksessa käyttäjän kanssa, se on *keskeytetty*. Keskeytetty Activity-komponentti on "hengissä", mutta se voidaan lopettaa käyttöjärjestelmän puolesta, jos järjestelmä tarvitsee lisää muistia. (Android Developers: Activity, hakupäivä 11.10.2012.)

Jos Activity-komponentin on syrjäyttänyt toinen Activity-komponentti, se on *pysäytetty*. Se on täysin piilossa käyttäjältä, mutta säilyttää silti kaiken tietonsa. Tässä tilassa oleva Activity-komponentti yleensä lopetetaan, jos järjestelmä tarvitsee lisää muistia. Jos Activity-komponentti on keskeytetty tai pysäytetty, järjestelmä voi pyytää komponenttia lopettamaan toimintansa tai lopettaa sen itse,

jos se tarvitsee lisää muistia. Kun Activity-komponentti näytetään käyttäjälle uudestaan, se tulee käynnistää kokonaan uudelleen. (Android Developers: Activity, hakupäivä 11.10.2012.) Activity-komponentin elinkaari voidaan jakaa seuraaviin metodeihin:

onCreate()

Metodia onCreate() kutsutaan, kun Activity-komponentti luodaan. Tässä metodissa luodaan muun muassa näkymät. Tämä metodi myös luo Bundlen, joka sisältää Activity-komponentti tilan, jos komponentti vain avataan uudelleen. Tästä seuraava metodi on aina onStart(). (Android Developers: Activity, hakupäivä 11.10.2012.)

onRestart()

Metodia onRestart() kutsutaan silloin, kun Activity-komponentti on pysäytetty, jotta se voidaan käynnistää uudelleen. Tästä seuraava metodi on aina onStart(). (Android Developers: Activity, hakupäivä 11.10.2012.)

onStart()

Metodia onStart() kutsutaan silloin, kun Activity-komponentti tulee näkyväksi käyttäjälle. Sitä seuraa metodi onResume(), jos komponentti tulee etualalle ja onStop(), jos komponentti piilotetaan näkyvistä. (Android Developers: Activity, hakupäivä 11.10.2012.)

onResume()

Metodia onResume() kutsutaan silloin, kun Activity-komponentti alkaa toimia käyttäjän kanssa. Tällöin komponentti on "Activity-pakan" päällimmäisenä ja se on vuorovaikutuksessa käyttäjän kanssa (esimerkiksi näytön kosketus tai kirjoitus

tekstikenttään). Tätä metodia seuraa aina metodi onPause(). (Android Developers: Activity, hakupäivä 11.10.2012.)

onPause()

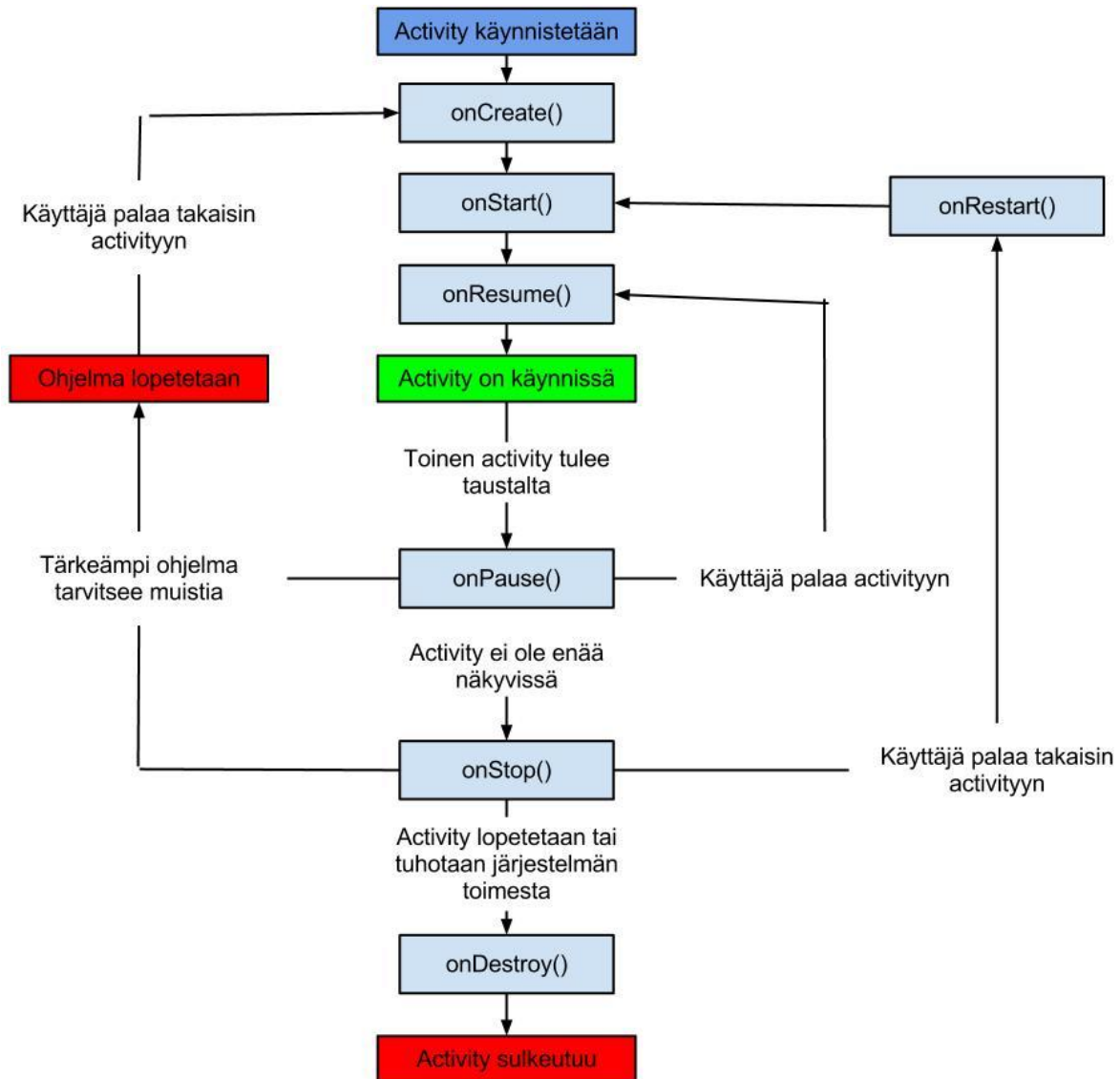
Metodia onPause() kutsutaan, kun järjestelmä aikoo palauttaa edellisen Activity-komponentin. Metodia käytetään yleensä tallentamaan tallentamattomat tiedot ja lopettamaan animaatiot sekä muut mahdolliset keskusmuistia vaativat tapahtumat. Metodi tulee kuitenkin olla nopeasti ajettava, sillä seuraava Activity-komponentti ei voi tulla näkyviin ennen kuin tämä on suoritettu. Tätä metodia seuraa joko onResume() -metodi, jos komponentti palaa näkyväksi tai onStop() -metodi, jos komponentti piilotetaan käyttäjältä. (Android Developers: Activity, hakupäivä 11.10.2012.)

onStop()

Metodia onStop() kutsutaan, kun Activity-komponentti ei ole enää käyttäjälle näkyvissä toisen komponentin vuoksi. Tämä voi johtua joko siitä, että toinen Activity-komponentti on asetettu näkyväksi tai nykyinen on tuhottavana. Tätä metodia seuraa joko metodi onRestart(), jos Activity-komponentti aloittaa vuorovaikutuksen käyttäjän kanssa, tai onDestroy(), jos komponentti tuhoetaan. (Android Developers: Activity, hakupäivä 11.10.2012.)

onDestroy()

Metodia onDestroy() kutsutaan, kun Activity-komponentti tuhoetaan. Tämä johtuu siitä, että komponentti lopetetaan käyttäjän tai järjestelmän toimesta. Tämän metodin jälkeen ei enää kutsuta metodeja. (Android Developers: Activity, hakupäivä 11.10.2012.) Activity-komponentin elinkaarta on havainnollistettu kuviossa kaksi.



KUVIO 2. Activity-komponentin toimintaprosessi

Intent

Intent on luokka, joka sisältää Activity-komponentin abstraktin kuvauksen. Käytännössä siinä lähetetään kutsuvan Activity-komponentin tiedot kutsuttavalle Activity-komponentille. Uudelle komponentille lähetetty Intent luodaan seuraavasti:

- Intent intent = new Intent(<context>, <uudenActivityn luokka>);

Intentin välityksellä on mahdollista siirtää arvoja kahden eri Activity-komponentin välillä.

Arvot tallennetaan Extra-laajennuksen luotuun Intentiin:

- `intent.putExtra(<String arvon nimi>, <String arvo>);`

Intentin Extrat voidaan vastaanottaa Bundleen:

- `Bundle bundle = new Bundle();`

- `bundle = getIntent().getExtras();`

Metodi `getIntent()` hakee Activity-komponentille lähetetyn Intentin ja `getExtras` hakee laajennukset haetun Intentin sisältä. (Hyvän käyttöliittymän valmistaminen GUIDe-prosessimallilla, hakupäivä 11.10.2012.)

AndroidManifest

Jokaisella Android-sovelluksella on oltava sen juuressa tiedosto `AndroidManifest.xml`. Tämä tiedosto välittää tärkeimmät tiedot sovelluksesta käyttöjärjestelmälle, ennen kuin järjestelmä voi suorittaa mitään sovelluksen koodista. (The `AndroidManifest.xml` File, hakupäivä 11.10.2012.)

`AndroidManifest`-tiedosto muun muassa nimeää Java-paketin sovellukselle, jonka nimi toimii uniikkina tunnuksena ohjelmalle. Se kuvaa sovelluksen komponentit (Activity-komponentit ja palvelut) ja nimeää kaikki luokat, jotka otetaan käyttöön eri komponenteissa ja julkaisee niiden ominaisuudet. Näiden julkaisujen avulla käyttöjärjestelmä tietää, mikä komponentti on kyseessä ja missä tapauksissa se tulee käynnistää. (Android Developers: The `AndroidManifest.xml` File, hakupäivä 11.10.2012.)

`AndroidManifest` myös päättää, missä prosessissa komponentit ajetaan. Se ilmoittaa käyttöjärjestelmälle, mitä oikeuksia ohjelmalla on. Näin se voi käyttää suojeltuja osia API-kirjastosta ja kommunikoida toisten ohjelmien kanssa. Samalla se ilmoittaa, mitä oikeuksia tarvitaan, jotta toiset ohjelmat voivat myös kommunikoida sovelluksen komponenttien kanssa. `AndroidManifest` ilmoittaa laitteelle myös sovelluksen suorittamiseen tarvittavan minimi API-kirjaston. Lisäksi `AndroidManifest`-tiedosto listaa kirjastot, joiden pitää olla linkitettyinä ohjelmaan. (Android Developers: The `AndroidManifest.xml` File, hakupäivä 11.10.2012.)

Java

Java on Sun Microsystemsin kehittämä laaja teknologiaperhe ja ohjelmistoalusta, johon kuuluu muun muassa laitteistoriippumaton oliopohjainen ohjelmointikieli sekä ajoaikainen ympäristö virtuaalikoneineen ja luokkakirjastoineen. Java-ohjelmointikielen kehittivät Bill Joy ja James Gosling kollegoineen Sun Microsystemsillä 1990-luvun alussa. (Wikipedia: Java 2012, hakupäivä 11.10.2010.)

Java kuuluu ohjelmointikieliin, joissa on käytössä niin sanottu vahva tyyppitys. Tämä tarkoittaa sitä, että jokaisella muuttujalla on tyyppi, ja muuttujat voivat saada ainoastaan tyyppinsä mukaisia arvoja. Toisin kuin tavanomaisissa kielissä, Javan lähdekoodia ei käännetä suoraan konekielille, vaan tavukoodiksi, joka suoritetaan virtuaalikoneessa. Muihin ohjelmointikieliin verrattuna (esimerkiksi C tai C++), Java sisältää runsaasti ominaisuuksia, kuten graafisen käyttöliittymäkirjaston, rinnakkaisuuden hallinnan, verkko-ominaisuudet ja rikkaat rajapinnat. Nämä rajapinnat ovat toisissa kielissä käyttöjärjestelmäriippuvaisia tai kolmansien osapuolten kirjastojen varassa. (Wikipedia: Java 2012, hakupäivä 11.10.2010.)

Javan perusominaisuuksiin kuuluu se, että suoritettavien lauseiden ja laskettavien lausekkeiden muoto on peräisin C-kielistä. Koska Java on olio-ohjelmointikieli, sen ohjelmoinnissa käytetään alusta alkaen luokkia ja niiden ilmentymiä, joita kutsutaan olioiksi. Javalla on vahva tyyppitys, eli muuttujille sijoitetaan vain niiden tyyppin mukaisia arvoja. Ohjelmointikielenä Java on määritelty erittäin täsmällisesti, minkä tarkoituksena on, että ohjelma toimii samoin missä tahansa riippumatta siitä, missä se on laadittu. Java käyttää merkkitiedon esittämiseen koodausta, joka tarjoaa useimpien maailman kielten merkkivalikoimat. (Wikka 2003, 9–10.)

Vaikka Java on suhteellisen pieni kieli (varattuja sanoja ja syntaktisia rakenteita on "vähän"), on se erittäin ilmaisuvoimainen. Kieli tarjoaa välineet rinnakkaisohjelmointiin ja poikkeusten käsittelyyn. Muistialueiden vapauttaminen

on automaattista, sillä Java-ohjelman suorittamisen yhteydessä niin sanottu automaattinen roskienkerääjä tutkii, ovatko muistialueiden varaukset käyneet tarpeettomiksi. Javan kirjastot ovat hyvin monipuolisia ja täsmällisesti määriteltyjä (muun muassa tietoliikenne ja graafinen käyttöliittymä). (Wikka 2003, 9–10.)

Java-ohjelmat käännetään tietokoneriippumattomalle Bytecode-välikielelle, jota tulkitsemalla suoritetaan itse ohjelmat. Tavallisten sovellusten (application) lisäksi Javalla voidaan laatia www-sivuille sijoitettavia, niin sanottuja sovelmia (applet). Näiden avulla www-selain noutaa välikielisen ohjelmatiedoston palveluntarjoajan tietokoneesta. Selaimen oma tulkki suorittaa välikielisen ohjelman käyttäjän tietokoneessa. Turvallisuussyistä sovelman oikeudet ovat rajoitettuja verrattuna sovelluksiin. Esimerkiksi sovelma ei yleensä voi lukea tai kirjoittaa tiedostoja. (Wikka 2003, 9–10.)

Java on ensimmäinen kieli, jolla voidaan toteuttaa täysin laite- ja ohjelmistoriippumattomia vuorovaikutteisia sovelluksia. Laitteistoriippumattomuus tarkoittaa toimivuutta kaikissa käyttöjärjestelmä- ja suoritinympäristöissä. (Paananen 2004, 322.)

PHP

PHP (lyhenne sanoista *PHP: Hypertext Preprocessor*) on Perl-kielen kaltainen ohjelmointikieli, jota käytetään erityisesti web-palvelinympäristöissä dynaamisten web-sivujen luonnissa. Ohjelmointikielen lisäksi PHP-ympäristössä on laaja luokkakirjasto. PHP on komentosarjakieli, jossa ohjelmakoodia tulkitaan vasta ohjelman suorituksen vaiheessa. PHP:tä voidaan käyttää useilla erilaisilla alustoilla ja käyttöjärjestelmillä. Ensimmäinen versio PHP:stä julkaistiin vuonna 1995. Nykyisin PHP on vertailuissa johtava dynaamisten web-palveluiden tuottamiseen tarkoitettu kieli. (PHP 2012, hakupäivä 11.10.2010.)

Useimmiten PHP:tä käytetään upotettuna HTML-sivujen sisälle. PHP:n muuttujat ovat heikosti tyyditettyjä, eli ne voivat saada minkä tahansa tyyppisen arvon riippuen kontekstista. PHP:ssä käytettyjä muuttujatyyppejä ovat totuusarvo (boolean), kokonaisluku (integer), liukuluku (float), merkkijono (string), joukko (array), olio (object), ulkoinen resurssi (resource) ja tyhjä (NULL). PHP-ohjelmoinnissa voidaan myös tallentaa tietoa tietokantoihin. PHP:ssa on tietokantatuki esimerkiksi MySQL- ja SQLite-tietokannoille. (PHP 2012, hakupäivä 11.10.2010.)

3.4.3 Tietokannat

Tietokanta-termiä käytetään tietotekniikassa tietovarastolle. Se on kokoelma tietoja, jotka ovat yhteydessä toisiinsa. Tietokannat esitetään tietotekniikassa taulukkorakenteella. Esitettävät tiedot ovat tauluissa. Jokaisella eri sarakkeella on aina jokin tietotyyppi, joka kertoo minkälaista tietoa kyseiseen sarakkeeseen voi tallentaa, lisäksi voidaan määritellä voiko sarakkeen solu saada tyhjän arvon (NULL). (Wikipedia: Tietokanta 2012, hakupäivä 11.10.2010.)

Tallentaminen, haku ja päivitykset ovat tietokannan hallinnan ydintoimintoja. Tietokannan datan sekä dataan kohdistuvien muutosten on noudatettava tiettyjä eheysehtoja, joilla tarkoitetaan dataan ja muutoksiin liittyviä oikeellisuus- ja ristiriidattomuussääntöjä. Tietokantajärjestelmä tarjoaa mekanismit, jotka mahdollistavat usean samanaikaisen käyttäjän päivittämään tietokannan tietoja virheettömästi. Päivitysten pitää eristyä riittävästi muiden samanaikaisten käyttäjien päivityksistä. (Paananen 2004, 395–404.)

Transaktio on käsittelykomentojen sarja, joka siirtää tietokannan eheästä tilasta toiseen eheään tilaan. Ellei eheää tilaa saavuteta, tietokantajärjestelmän on kyettävä peruuttamaan koko transaktio ja kaikki sen komennot. Transaktio on kokonaisuutena suoritettava (commit) tai se on jätettävä kokonaan suorittamatta (rollback). (Paananen 2004, 395–404.)

Tietokantoja on monenlaisia, ja niitä käytetään moniin tarkoituksiin. Tietokanta-alan virallinen kansainvälinen ISO-standardointi sisältyy pääasiassa SQL-standardin (Structured Query Language) alle. Tietokantajärjestelmä perustuu aina johonkin tietomalliin, joista tärkeimmät ovat: hierarkkinen tietomalli, verkkomalli, relaatiomalli ja oliomalli. (Paananen 2004, 395–404.)

Hierarkkisessa mallissa asioiden väliset suhteet ovat pelkästään sisältyvyys- ja alistussuhteita. Erityiskäytössä hierarkiaa käytetään paljon, esimerkiksi Windows-käyttöjärjestelmät tallettavat sisäisiä kokoonpanotietojaan hierarkkisessa registry-tietokannassa. (Paananen 2004, 395–404.)

Relaatiomalli

Relaatiomallin mukaisen määrämuotoisen tiedon käsittelyn laajennuspiirteet näyttävät perustuvan johtavissa järjestelmätuotteissa aivan erilaisiin tekniikoihin. Tällaisia laajennuspiirteitä ovat esimerkiksi multimedian käsittely, monimuotoisten dokumenttien hallinta sekä vapaan tekstidatan, karttatiedon ja aikasarjojen käsittelytekniikat. (Paananen 2004, 395–404.)

Yleisin ja yleispätevin ajatusmalli tietokannan rakenteelle on sellainen, että tietokannan ajatellaan koostuvan joukosta luetteloita, joissa on rivejä ja sarakkeita. Luetteloita sanotaan tauluiksi. Relaation ilmentymä on taulu eli taulukko, joka on tiedon tallentamiseen käytetty kokonaisuus. Taulu koostuu riveistä ja sarakkeista. Tauluina esitetyn tiedon käsittely ja etsiminen on hyvin selkeätä. Eri tauluissa olevien tietojen väliset yhteydet perustuvat tietosisältöjen mukaisiin viittauksiin. Taulu on tavallisimman tietokantatyypin eli relaatiomallisen tietokannan ydinrakenne, sillä kaikki relaatiotietokannan sisältämä tieto on tallennettu tauluihin. (Paananen 2004, 406–407.)

Taulujen väliseltä yhteydeltä vaaditaan enemmän kuin satunnainen yhtäläisyys parin tiedon välillä. Viittauksen pitää joka hetki kohdistua täsmälleen yhteen

kohdetaulun riviin. Viittauksen kohteena on oltava taulun avain. Avaimella tarkoitetaan yksilöivää saraketta tai sarakkeiden yhdistelmää. (Paananen 2004, 406–407.)

Vuonna 1970 E.F. Codd julkaisi relaatiomallin, jossa hän määritteli miten tiedot tulisi tallentaa tietokantaan ja kuinka näitä tietoja tulisi käsitellä. IBM lähti kehittämään relaatiomallin mukaista prototyyppitietokantaa. Sen tietokantakieleksi valittiin SEQUEL. Myöhemmin tätä projektissa käytettävää kieltä alettiin kutsua nimellä SQL. (Hovi 1999, 9–10.)

SQL

Nimi SQL on peräisin sanoista Structured Query Language eli strukturoitu kyselykieli (Hovi 1999, 9–10). SQL-standardi hallitsee tietokantatekniikkaa laajasti. Sen osa, jolla ohjelmistot luovat tietokantayhteyksiä, on toteutunut ODBC-tekniikkana. (Paananen 2004, 395–404.) SQL ei kuitenkaan nimestään huolimatta ole pelkästään kyselykieli, vaan se kattaa myös tietokannan rakenteen määrittelyn ja sen muokkaamisen, kyselyt, päivitykset (tietokannan lisäykset, muutokset ja poistot), valtuuksien ja turvallisuuden hoidon sekä tapahtuman käsittelyn ohjaamisen. (Hovi 1999, 9–10.) Käytännössä kaikki relaatiotietokannat ymmärtävät SQL-kieltä. SQL ei määritä verkkoprotokollaa tai standardia, jolla SQL-komentoja välitetään. (Wikipedia: SQL 2012, hakupäivä 11.10.2010.)

SQLite

SQLite on relaatiotietokantajärjestelmä. Se on toteutettu pienenä, alle 225 kilotavun C-kirjastona. (Wikipedia: SQLite 2012, hakupäivä 11.10.2010.)

Toisin kuin monet muut relaatiotietokannat, koko SQLite-järjestelmä linkitetään suoraan sitä käyttävään sovellukseen, joten erillistä ODBC-yhteyttä, tietokannan hallintaohjelmaa tai tietokantapalvelinta ei tarvita. Itse tietokanta voidaan tallentaa

kokonaan tietokoneen muistiin, tai tallentaa tiedostoon, joka lukitaan transaktioiden ajaksi. Tästä syystä SQLite soveltuu erityisesti tietokantajärjestelmän lisäämiseen sovellusohjelmiin. (Wikipedia: SQLite 2012, hakupäivä 11.10.2010.)

4 VIDEONHALLINTA

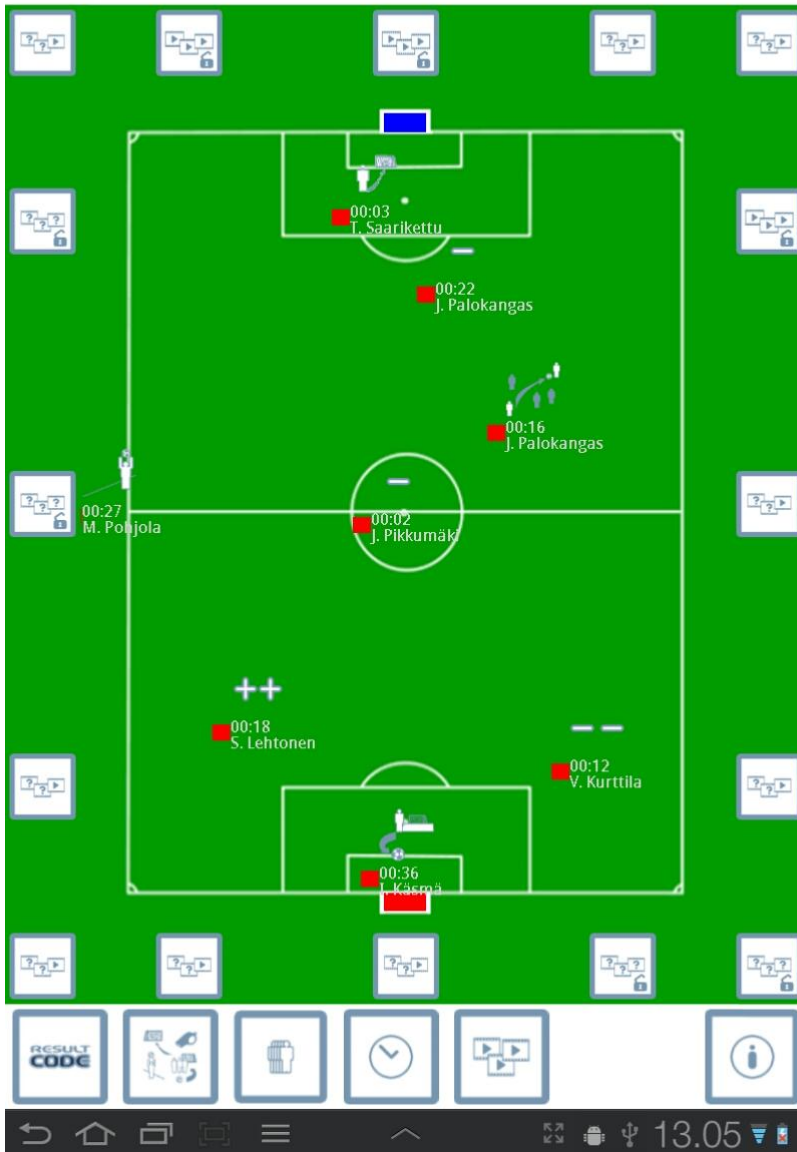
4.1 Ohjelman kuvaus

Videonhallintamoduuli liitetään tulevaisuudessa osaksi ohjelmaa ResultCodeMobile ja sillä voidaan analysoida tilastoitujen ja videoitujen otteluiden tietoja. Videonhallintamoduulilla mahdollistetaan käyttäjälle ohjelmalla ResultCodeMobile tallennettujen ottelutapahtumien ja pelaajaseurantatilastojen katsominen videolta siten, että käyttäjän itse ei tarvitse etsiä videolta oikeaa toistokohtaa, vaan se haetaan automaattisesti valitun tapahtuman tietojen avulla.

Videonhallintamoduulin tulee hakea tarvittava videoleike käyttäjän näkökulmasta katsottuna mahdollisimman vähällä tiedonsiirrolla ja mahdollisimman nopeasti. Videonhallintamoduulissa näytetään ottelun tilastoidut ottelutapahtumat ja pelaajaseurantatilastot. Ohjelmassa ResultCodeMobile käytettyjä pelaajaseurantatilastoja ovat: peliä paljon edistävä asia (++), peliä vähän edistävä asia (+), peliä vähän heikentävä asia (-) ja peliä paljon heikentävä asia (--).

Kun käyttäjä haluaa katsoa videolta tapahtuman tai useita tapahtumia, hän valitsee ensin kamerapaikan, josta videot toistetaan. Käyttäjä voi halutessaan rajata näytettäviä tapahtumatyyppejä ja pelaajia. Lisäksi käyttäjä voi tehdä aikarajauksen, jolloin näytetään ainoastaan käyttäjän valitsemalla aikavälillä tapahtuneet tapahtumat. Käyttäjä voi ohjelmassa valita, katsooko hän yksittäisen tapahtuman vai kaikki rajauksien sisälle jäävät tapahtumat peräkkäin. Kuviossa kolme on näkymä videonhallinnasta.

Tervarit-jussin Pallo - HauPa 3



KUVIO 3. Videonhallintamoduuli

Käynnistettäessä VideontoistoActivityä ohjelma hakee ResultCoden tietokannasta kaikki otteluun liitetyt videot ja sijoittaa ne niille määrätyille kamerapaikoille. Kamerapaikkoja on määritetty 16 kappaletta ja ne on sijoitettu eri puolille kentän laitoja. Yhtä kamerapaikkaa kohden voi olla nolasta rajattomaan määrä videoita.

Mikäli käyttäjä valitsee kamerapaikan, johon ei ole liitetty yhtään videota, ohjelma avaa selaimella PHP-nettisivun, jolla lisätään uusi video tietokantaan. Sivulla käyttäjä syöttää muun muassa videon verkko-osoitteen, kuvauspaikan kohdan kentän reunalla ja aloituspotkun ajankohdan videolla. Nämä tiedot lisätään ResultCode-kantaan, josta ohjelma pystyy ottamaan ne käyttöön.

Videonhallintamoduulissa on myös mahdollisuus kommentoida jokaista tapahtumaan liittyvää videota erikseen. Kommentit sekä kommentoija tallennetaan tietokantaan ja tiedot on mahdollista saada myöhemminkin ja myös eri käyttäjien käyttöön. Kommentoija voi valita, onko hänen kommenttinsa julkinen vai yksityinen. Julkinen kommentti näkyy kaikille, yksityinen vain hänelle itselleen. Oletusarvona on, että kommentti liitetään tapahtumaan, jota juuri katsotaan, mutta jos käyttäjä haluaa niin hän voi kommentoida videoleikettä. Tapahtumaan liitetty kommentti näkyy videontoistossa missä tahansa kamerapaikassa, kun taas tiettyyn videoon liitetty kommentti näkyy vain tietyn kamerapaikan videontoistossa. Kuviossa neljä näkyy videonhallinnan yksittäisen videoleikkeen videontoistoikkuna.



KUVIO 4. Videontoistoikkuna

4.2 Ohjelman toteutus

Videonhallintamoduuli on ohjelmoitu pääosin Java-kielellä, ja se toimii vain Android-käyttöjärjestelmillä toimivilla älypuhelimilla sekä tableteilla. Kaikkiaan

videonhallintamoduulin tekemisessä käytettiin seuraavia kieliä: Java, XML, PHP, SQL ja SQLite.

Videonhallintasovelluksen rakenne koostuu kahdesta osasta, ulkoasusta ja ohjelmakoodista. Ulkoasu on toteutettu XML-kielillä. Ohjelmakoodi, joka sijaitsee Activity-komponentissa, on ohjelmoitu käyttäen Javaa. Käytettäessä laitteen omaa tietokantaa, kutsutaan ohjelmakoodissa metodeja, joissa käytetään SQLiteä. Kun halutaan yhteys ulkoiseen palvelimeen, kutsutaan palvelimella olevia PHP-tiedostoja, joissa käytetään PHP:tä ja SQL:ää. Nämä tiedostot hoitavat SQL-kannan avaukset ja kyselyt.

4.3 Käyttöliittymä

Videonhallinnan käyttöliittymän muodostamiseen käytetään kahta XML-tiedostoa. Käyttöliittymätiedostoja on kaksi, koska ulkoasu muuttuu hieman kun vaihdetaan pystynäytöstä vaakanäkymään. Perusulkoasultaan XML-tiedostot ovat suurin piirtein samoja. Ulkoasun määrittelyssä on käytetty komponentteja LinearLayout, TextView ja ImageView.

XML-tiedostossa asetetaan pääasiassa komponenttien sijainnit ja osalle asetetaan niiden koot. Ohjelmakoodissa asetetaan komponentit muuttujiin, joista niiden muokkaaminen onnistuu helposti. Ohjelmakoodissa lasketaan ja asetetaan tarvittaessa tarkemmin toisten komponenttien koot, ja lisätään uusia komponentteja.

AndroidManifest

Tiedostossa AndroidManifest määritellään koko ohjelman ominaisuuksia, käyttöoikeuksia sekä muita käyttöön liittyviä tietoja. Tässä ohjelmassa ollaan määritetty, että käyttäjällä täytyy olla Android SDK Versio 8 tai sitä uudempi versio.

Käytännössä tämä tarkoittaa sitä, että Android-käyttöjärjestelmän, jolla ohjelmaa suoritetaan, on oltava versio 2.2 tai uudempi.

Tiedostossa `AndroidManifest` määritellään myös ohjelman tiedot, ja koska videonhallintamoduuli on osa sovellusta `ResultCodeMobile`, niin ohjelman tiedot ovat sen mukaiset. Tärkeimpinä ominaisuuksina Activity-komponentin käytöstä on tieto, voidaanko komponenttia kääntää vaakatasoon. Videonhallintamoduulin ominaisuuksiin kuuluu, että Activity-komponenttia voidaan käyttää sekä vaaka- että pystyasennossa.

Videonhallintamoduulin käyttö vaatii vähintään seuraavat oikeudet: **Internet**, joka sallii ohjelman avata verkkoyhteyden, **Access wifi state**, joka sallii ohjelman käyttää tietoja langattomista verkkoyhteyksistä, **Access network state**, joka sallii ohjelman käyttää tietoja verkkoyhteyksistä ja **Vibrate**, joka sallii ohjelman käyttää laitteen värinätoimintoa.

4.4 Ohjelmakoodi ja tietokannat

Videon lisäys – PHP

Videon lisäys `ResultCode`-palvelun tietokantaan suoritetaan Internetissä PHP-tiedoston avulla. Itse videota ei lisätä palvelimelle, vaan ainoastaan sen tiedot lisätään tietokantaan, kuten verkko-osoite, kuvauspaikka kentän reunalla ja aloituspotkun ajankohta videolla. Tietokantaan asetetun videon verkko-osoitteen avulla ohjelma noutaa videon sen sijainnista.

Videon lisäys on tehty PHP-tiedostona siitä syystä, että videoiden lisäämistä voidaan tarpeen mukaan rajoittaa. Tämä onnistuu helpommin muuttamalla yhtä PHP-tiedostoa kuin ohjelman lähdekoodia, joka vaatisi käyttäjien ohjelmien päivittämisen.

Tietokantahaut – PHP

Video itsessään voi sijaita millä palvelimella tahansa, mutta sen tiedot tulee olla linkitettyinä ResultCode-palvelun tietokannassa. Ohjelmisto hakee tietokannasta videon tiedot, kuten videon verkko-osoitteen, kuvauskohtan kentänlaidalta, kuvaajan ja missä kohdassa videolla aloituspotku tapahtuu. Näiden tietojen avulla ohjelma voi näyttää käyttäjän haluaman pelitapahtuman tarkan kohdan.

Kyseiset tiedot haetaan ResultCode-palvelun tietokannasta PHP-tiedostojen avulla. Ohjelma ottaa yhteyden PHP-tiedostoon, joka hakee tiedot palvelimelta. Tämän jälkeen tiedot siirretään PHP:stä ohjelmaan. Tämä on tehty siitä syystä, että PHP-tiedostoa on helpompi hallita erikseen kuin ohjelman lähdekoodia. Jos tiedon hakua tulee jostain syystä rajoittaa tai täsmentää, tämä onnistuu helposti muokkaamalla PHP-tiedostoa.

Laitteen sisäinen tietokanta

Videohallinta käyttää sisäisestä tietokannasta videoiden kommenttitaulua, johon tallennetaan ja josta haetaan kaikki videoihin liittyvät kommentit. Lisäksi tietokannasta haetaan ottelusta taltioidut ottelutapahtumat ja pelaajaseurantatilastot.

Activity

Videohallintamoduulin Activity-komponentissa ensimmäisessä osassa on Activity-komponentin muuttujien määrittämiset. Tätä seuraa metodin onCreate() määrittäminen ja viimeisenä muiden käytettävien metodien määrittelemiset.

Metodeista, joihin Activity-komponentin elinkaari voidaan jakaa, tässä opinnäytetyössä käytetään ainoastaan onCreate() -metodia. Tässä metodissa määritellään muun muassa komponentin painikkeet, kuvat, layoutit ja painikkeiden

kuuntelijat sekä haetaan sisäisestä ja ResultCode-palvelun tietokannoista tarvittavat tiedot ohjelman toimimiseen.

5 POHDINTA

Onnistuimme opinnäytetyössämme hyvin, saimme sisällytettyä kaikki toimeksiantajalta tulleet vaatimukset työhön. Lisäksi kesken opinnäytetyön tekemisen otimme moduliin mukaan myös kommentoinnin, joka tuli pyyntönä toimeksiantajalta. Videonhallintamoduulista tuli lopulta juuri sellainen kuin siitä haluttiinkin, ja jatkokehittelyä varten saimme useita kehitysideoita ja mahdollisuuksia. Tulevaisuudessa videonhallintamoduuli otetaan käyttöön osana ResultCodeMobilea.

Alkuperäinen aikataulumme oli kiireinen. Koska ResultCodeMobilen beta-version julkaisu oli ajankohtainen ja vei meiltä erittäin paljon työaikaa, myöhästyimme aikataulusta noin kuukauden verran.

Myöhemmin tulemme vielä tutkimaan tarkemmin videonhallintamoduulista kohtaa, jossa video-ikkuna muutetaan koko näytölle. Tämän hetken ohjelmassa video ladataan aina uudestaan, jolloin lataaminen kestää hieman. Tutkimisen tarkoituksena on selvittää, saammeko jollakin tapaa ladattua tiedostoa laitteen muistiin ja helposti käytettäväksi, ettei sovelluksen tarvitsisi joka kerta ladata videota uudestaan. Tällöin myös tiedoston poistaminen laitteen muistista pitää hallita, ettei muistin määrä lopu kesken käyttäjien laitteilta.

Videonhallintamoduulin tekemisessä opimme paljon uusia asioita Android-ohjelmoinnista ja miten uuden ohjelmamoduulin saa otettua käyttöön osana ResultCodeMobilea. Täysin uutena asiana meille tuli muun muassa Videoview-komponentti, jota emme olleet koskaan aikaisemmin käyttäneet. Lisäksi opimme työn aikana ajankäytön hallintaa.

LÄHTEET

Android Developers: Activities. Hakupäivä 11.10.2012,
[Http://developer.android.com/guide/components/activities.html](http://developer.android.com/guide/components/activities.html).

Android Developers: Activity. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/app/Activity.html](http://developer.android.com/reference/android/app/Activity.html).

Android Developers: ADT Plugin. Hakupäivä 11.10.2012,
[Http://developer.android.com/tools/sdk/eclipse-adt.html](http://developer.android.com/tools/sdk/eclipse-adt.html).

Android Developers: AlertDialog. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/app/AlertDialog.html](http://developer.android.com/reference/android/app/AlertDialog.html).

Android Developers: Building Your First App. Hakupäivä 11.10.2012,
[Http://developer.android.com/training/basics/firstapp/index.html](http://developer.android.com/training/basics/firstapp/index.html).

Android Developers: Button. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/widget/Button.html](http://developer.android.com/reference/android/widget/Button.html).

Android Developers: CheckBox. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/widget/CheckBox.html](http://developer.android.com/reference/android/widget/CheckBox.html).

Android Developers: EditText. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/widget/EditText.html](http://developer.android.com/reference/android/widget/EditText.html).

Android Developers: Get the Android SDK. Hakupäivä 11.10.2012,
[Http://developer.android.com/sdk/index.html](http://developer.android.com/sdk/index.html).

Android Developers: ImageView. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/widget/ImageView.html](http://developer.android.com/reference/android/widget/ImageView.html).

Android Developers: ListView. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/widget/ListView.html](http://developer.android.com/reference/android/widget/ListView.html).

Android Developers: Managing Virtual Devices. Hakupäivä 11.10.2012,
[Http://developer.android.com/tools/devices/index.html](http://developer.android.com/tools/devices/index.html).

Android Developers: RadioButton. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/widget/RadioButton.html](http://developer.android.com/reference/android/widget/RadioButton.html).

Android Developers: TextView. Hakupäivä 11.10.2012,
[Http://developer.android.com/reference/android/widget/TextView.html](http://developer.android.com/reference/android/widget/TextView.html).

Android Developers: The AndroidManifest.xml File. Hakupäivä 11.10.2012,
[Http://developer.android.com/guide/topics/manifest/manifest-intro.html](http://developer.android.com/guide/topics/manifest/manifest-intro.html).

Androidsuomi: Mikä on Android?. Hakupäivä 11.10.2012,
[Http://blog.androidsuomi.fi/mika-on-android/](http://blog.androidsuomi.fi/mika-on-android/).

Code4m Oy. 2012. Hakupäivä 11.10.2012, [Http://www.code4m.com/](http://www.code4m.com/).

Hovi, A. 1999. SQL-ohjelmointi Pro Training. Helsinki: Suomen Atk-kustannus Oy.

Hyvän käyttöliittymän valmistaminen GUIDe-prosessimallilla. Hakupäivä
11.10.2012, http://some.lappia.fi/wiki/images/d/dc/Android_ohjelmointi_05.pdf.

Jackson, G. S. How to Make an Android App With Netbeans. Hakupäivä
11.10.2012, [Http://www.ehow.com/how_12191131_make-android-app-netbeans.html](http://www.ehow.com/how_12191131_make-android-app-netbeans.html).

Järvinen, P. 2004. NÖRTTISANAKIRJA. Helsinki: WSOY.

Koivunen, S. 2012. Googlen Androidin markkinaosuus nousi jo 75 prosenttiin. Hakupäivä 11.10.2012, [Http://www.puhelinvertailu.com/uutiset.cfm/2012/11/03/googlen_androidin_markkinaosuus_nousi_jo_75_prosenttiin](http://www.puhelinvertailu.com/uutiset.cfm/2012/11/03/googlen_androidin_markkinaosuus_nousi_jo_75_prosenttiin).

Melanson, D. 2010. NPD: Android ousts iPhone OS for second place in US smartphone market. Hakupäivä 11.10.2012, [Http://www.engadget.com/2010/05/10/npd-android-ousts-iphone-os-for-second-place-in-us-smartphone-m/](http://www.engadget.com/2010/05/10/npd-android-ousts-iphone-os-for-second-place-in-us-smartphone-m/).

Morrill, D. 2008. Announcing the Android 1.0 SDK, release 1. Hakupäivä 11.10.2012, [Http://android-developers.blogspot.fi/2008/09/announcing-android-10-sdk-release-1.html](http://android-developers.blogspot.fi/2008/09/announcing-android-10-sdk-release-1.html).

Paananen, J. 2004. Tietotekniikan peruskirja. Docendo Finland Oy.

Virki, T. & Carew, S. 2011. Google topples Nokia from smartphones top spot. Hakupäivä 11.10.2012, [Http://uk.reuters.com/article/2011/01/31/oukin-uk-google-nokia-idUKTRE70U1YT20110131](http://uk.reuters.com/article/2011/01/31/oukin-uk-google-nokia-idUKTRE70U1YT20110131).

W3C: Extensible Markup Language (XML). 2012. Hakupäivä 11.10.2012, [Http://www.w3.org/XML/](http://www.w3.org/XML/).

W3C: Extensible Markup Language (XML) 1.0 (Fifth Edition). 2012. Hakupäivä 11.10.2012, [Http://www.w3.org/TR/REC-xml/](http://www.w3.org/TR/REC-xml/).

Wikipedia: Android. 2012. Hakupäivä 11.10.2012, [Http://fi.wikipedia.org/wiki/Android](http://fi.wikipedia.org/wiki/Android).

Wikipedia: Android software development. 2012. Hakupäivä 11.10.2012,
[Http://en.wikipedia.org/wiki/Android_software_development#Android_SDK](http://en.wikipedia.org/wiki/Android_software_development#Android_SDK).

Wikipedia: Eclipse (IDE). 2012. Hakupäivä 11.10.2012,
[Http://fi.wikipedia.org/wiki/Eclipse_%28IDE%29](http://fi.wikipedia.org/wiki/Eclipse_%28IDE%29).

Wikipedia: Java. 2012. Hakupäivä 11.10.2012, [Http://fi.wikipedia.org/wiki/Java](http://fi.wikipedia.org/wiki/Java).

Wikipedia: NetBeans. 2012. Hakupäivä 11.10.2012,
[Http://fi.wikipedia.org/wiki/NetBeans](http://fi.wikipedia.org/wiki/NetBeans).

Wikipedia: Ohjelmointiympäristö. 2012. Hakupäivä 11.10.2012,
[Http://fi.wikipedia.org/wiki/Ohjelmointiymp%C3%A4rist%C3%B6](http://fi.wikipedia.org/wiki/Ohjelmointiymp%C3%A4rist%C3%B6).

Wikipedia: Open Handset Alliance. 2012. Hakupäivä 11.10.2012,
[Http://en.wikipedia.org/wiki/Open_Handset_Alliance](http://en.wikipedia.org/wiki/Open_Handset_Alliance).

Wikipedia: PHP. 2012. Hakupäivä 11.10.2012, [Http://fi.wikipedia.org/wiki/PHP](http://fi.wikipedia.org/wiki/PHP).

Wikipedia: SQL. 2012. Hakupäivä 11.10.2012, [Http://fi.wikipedia.org/wiki/SQL](http://fi.wikipedia.org/wiki/SQL).

Wikipedia: SQLite. 2011. Hakupäivä 11.10.2012, [Http://fi.wikipedia.org/wiki/SQLite](http://fi.wikipedia.org/wiki/SQLite).

Wikipedia: Tietokanta. 2012. Hakupäivä 11.10.2012,
[Http://fi.wikipedia.org/wiki/Tietokanta](http://fi.wikipedia.org/wiki/Tietokanta).

Wikla, A. 2003. Ohjelmoinnin perusteet Java-kielillä. Espoo: OtaDATA ry.