LAUREA
AMMATTIKORKEAKOULU
Yhdessä enemmän

Android-rahainventointisovellus

Vuorentie, Marlo

Laurea-ammattikorkeakoulu

Android-rahainventointisovellus

Marlo Vuorentie
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Tammikuu, 2017

Laurea-ammattikorkeakoulu **Tiivistelmä**
Tietojenkäsittelyn koulutusohjelma
Tradenomi (AMK)

Marlo Vuorentie

Android-rahainventointisovellus

Vuosi 2017 Sivumäärä 31

Tämän opinnäytetyön aiheena oli työelämälähtöinen kehittämistyö, jonka tavoitteena oli toteuttaa Yritys X:lle sähköinen (tietotekninen) rahan inventointijärjestelmä parantamaan inventoinnin hankalaksi ja hitaaksi koettua työprosessia. Järjestelmällä on tarkoitus laskea holvin kontteja ja niiden sisältämiä rahamääriä, sekä tulostaa laskelman lopputulos helppolukuiseen taulukkoon. Työn alussa perehdytään mobiilijärjestelmien standardeihin, ominaisuuksiin ja vaatimuksiin sekä soveltuvien laitteiden teknisiin ominaisuuksiin.

Varsinainen suunnittelutyö aloitettiin määrittelemällä käyttöjärjestelmä, jolle lopputuotoksen voisi järkevimmin toteuttaa. Työn tilaajan kanssa suunnitteluvaiheessa käydyissä keskusteluissa päädyttiin siihen, että järjestelmä toteutetaan mobiilisovelluksena loppukäyttäjien älypuhelimiin. Lopputuotoksena oleva järjestelmä toteutettiin Android-natiivisovelluksena Android Studio ohjelmaa kehitystyökaluna käyttäen. Myös varsinaiseen kehitystyöhön hyödynnettiin työn tilaajan kanssa koko työn elinkaaren ajan käytyjen keskustelujen sisältöä ja loppukäyttäjille toteuttamaa avoimen haastattelun tuottamaa aineistoa.

Työn suunnittelun pohjaksi muotoutui perusteellinen tarveselvitys. Selvitys toteutettiin pääsääntöisesti kvalitatiivisen tutkimuksen menetelmillä. Aineiston kerääminen toteutui tutkimalla kirjallisia ja verkosta saatuja lähteitä liittyen mobiilisovellusten kehittämiseen Android-käyttöjärjestelmään. Työprosessin yksityiskohtiin ja tavoitteisiin päästiin perehtymään lukuisissa henkilökunnan kanssa pidetyissä palavereissa ja katselmuksissa. Tarveselvityksen perusteella määriteltiin järjestelmän tarkoitus, laajuus ja tekniset tavoitteet.

Järjestelmä toteutettiin Android-mobiilisovelluksena sekä otettiin käyttöön kohdeosastolla. Saadun palautteen mukaan sovellus saavutti sille asetetut vaatimukset onnistuneesti. Suurin osa loppukäyttäjien toivomista lisäominaisuuksista onnistuttiin lisäämään sovellukseen, mutta yhtä toivottua ominaisuutta, eli konttien sisältämien rahamäärien muuttamista, ei ehditty implementoimaan.

Asiasanat: kehittämistyö, mobiilisovellus, Android

Laurea University of Applied Sciences **Abstract**
Degree Programme in Business Information Technology
Bachelor's Thesis

Marlo Vuorentie

Android monetary inventory application

Year 2017 Pages 31

The subject of this thesis was a workplace-oriented development project, the aim of which was to create an electronic (information technology) cash inventory system for Company X to improve their inventory management, which was perceived as a difficult and slow work process. The system is designed to calculate the vault containers and the money contained within them and to print the calculation result as an easy-to-read table. At the beginning of the work the main focus is on mobile systems, their standards and requirements, as well as the technical characteristics of suitable equipment.

The actual design began by defining the operating system on which the final result could be implemented in the most reasonable way. In the discussions with the contractor, it was concluded that the system was to be implemented as a mobile software to the end-users smartphones. The final product was created as an Android native application by using the Android Studio software as the tool of development. Also, with the whole life cycle of the product's development advantage was taken of the contents of discussions and the material produced with the open interview of the end-users.

The basis of the design work took shape as a feasibility study. The study was conducted with the methods of qualitative research. The data collection was done by studying written and online sources about the development of mobile applications for the Android operating system. It was possible to familiarize the details and objectives of the work process in a number of staff meetings and reviews. With the feasibility study the system's purpose, scope and technical goals were determined.

The system was created as a Android mobile application and was introduced in the target department. According to the feedback received, the application reaches it's set requirements successfully. Most of the end-user's requested additional features were added to the application, with the exception of one equally desired feature, ie the conversion of monetary amounts contained in the containers, was not implemented due to a lack of time.

Keywords: development project, mobile application, Android

Sisällys

1 Johdanto

Älypuhelimilla on kommunikoinnin lisäksi monia toimintoja ja niiden yleistymisen myötä käyttö myös työtehtävissä on lisääntynyt. Älypuhelimesta on tullut monitoimilaite, jolla voi käyttää laajasti eri mobiilisovellusten ansiosta mm. verkko-, video- ja karttapalveluita. Erilaisia mobiilisovelluksia on esimerkiksi Google Play -sovelluskaupassa yli 2 miljoonaa.

Opinnäytetyöni on työelämälähtöinen kehittämistyö, jonka tavoitteena on toteuttaa uusi sähköinen rahan inventointijärjestelmä Yritys X:n maksuvälineistä vastaavan osaston käyttöön. Toimeksiantajalla itsellään ei ollut edellytyksiä eikä resursseja toteuttaa järjestelmää haluttuun muotoon, joten se tilattiin osastolta, jossa työskentelin. Järjestelmän tavoitteena oli nopeuttaa aikaisempaa hidasta ja hankalaa manuaalista työprosessia. Tässä raportissa esitän opinnäytetyöni keskeisimmät käsitteet ja työssä käytetyt menetelmät yleisellä tasolla.

Työn lähtökohtana oli toimeksiantajan luoma inventointijärjestelmän prototyyppi, joka oli toteutettu tietokonesovelluksena ja josta selvisi järjestelmälle toivotut toiminnot pääpiirteittäin. Alussa tehtävänä oli selvittää järjestelmälle käytettävä alusta, ja päädyttiin siihen tulokseen, että järjestelmän voi toteuttaa mobiilisovelluksena osaston työntekijöiden älypuhelimiin. Työntekijöillä oli työsuhdepuheliminaan Android-älypuhelimet, joten järjestelmä toteutettiin Android-natiivisovelluksena käyttäen työkaluna Android Studio – kehitysympäristöä.

2 Työn lähtökohdat

Opinnäytetyön toimeksiantaja (jatkossa Yritys X) on merkittävä rahahuollon toimija Suomessa. Yritys X vastaa osaltaan käteisrahan saatavuudesta ja turvallisuudesta. Yrityksen maksuvälineistä vastaavan osaston inventointimenetelmiä haluttiin kehittää tehokkaammaksi. Käytössä olevan virheherkän ja hankalakäyttöiseksi koetun menetelmän sijaan tarve oli yksinkertaiselle ja vaivattomasti käyttöön otettavalle sähköiselle rahainventointijärjestelmälle. Vastuuosasto antoi kehittämistoimeksiannon IT-osastolle, jossa työskentelin. Toimeksiannon tavoitteena oli luoda sovellus, jolla pystyi tehokkaasti ja luotettavasti inventoimaan rahakontteja ja niiden sisältämiä rahamääriä.

Valtaosalla kohdeosaston työntekijöistä oli käytössään Android-älypuhelimet, joten oli luontevaa toteuttaa kehittämishanke Android-alustalle. Tämän lisäksi kehityshanke toteutettiin Windows Phone -käyttöjärjestelmälle, koska osalla työntekijöistä oli vielä käytössään Lumia-älypuhelimet.

2.1 Työn tavoitteet ja rajaus

Työn tilaaja määritteli, että kehittämistyö saavuttaa sille asetetut vaatimukset, kun suunnittelu- ja ohjelmointityön lopputuotteena saadaan vastuuosaston käyttöön rahainventointisovellus. Sovellus tuli toteuttaa Android- ja Lumia-älypuhelimille tehostamaan Yritys X:n rahahuoltotoiminnossa olemassa olevaa työprosessia.

Työssä pyrittiin rajaamaan opinnäytetyön aihe mahdollisimman yksinkertaiseksi, joten käsittelyn kohteena on tämän työn Android-sovelluskehityken teoriaa siinä määrin, mitä asiakkaalle kehitettävään sovellukseen käytettiin. Opinnäytetyöstä rajataan myös Windows Phonelle toteutettu vastaavanlaisen sovelluksen kehittäminen, siihen liittyvä teoria sekä mobiilialustoille kehittäminen yleisesti.

2.2 Keskeiset käsitteet

Android Studio

Android Studio on Googlen julkaisema ja tukema Androidin virallinen sovelluskehitystyökalu (David 2015).

API-taso

API (sovellusohjelmointirajapinta) on ohjelmoijan rajapinta laitteen ominaisuuksiin. API-taso on kokonaislukuarvo, joka identifioi sovellusohjelmointirajapinnan version. Uudemmat laitteet, joissa on korkeampi API-taso, ovat yhteensopivia aiempien tasojen kanssa, jolloin samaa sovellusta ei tarvitse tehdä uudelleen alemman API-tason laitteisiin. (Developer 2016.)

Dalvik

Muokattu Java-virtuaalikone, jossa Android-sovellukset suoritetaan
(Meier 2012, 16-17).

Google Play

Google Play, entiseltä nimeltään Android Market, on Android-laitteisiin esiasennettu sovelluskauppa, josta käyttäjät voivat ostaa, ladata ja asentaa Googlen ja kolmansien osapuolien sovelluksia ja jonne sovelluskehittäjät voivat julkaista omia sovelluksiaan. (Pcmag 2016.)

Hybridisovellus

Natiivisovellukset ja Web-sovellukset eivät ole toisiaan poissulkevia ja molempia tekniikoita yhdistelemällä voidaan luoda hybridisovelluksia. Hybridisovellusten toteutustapaa noudattamalla Web-sovellukset voidaan toteuttaa samankaltaisiksi kuin natiivisovellukset, jolloin valmis sovellus paketoidaan kohdealustalle sopivaan muotoon. Tämän jälkeen sovellusta voi jakaa esimerkiksi sovelluskaupoissa, josta loppukäyttäjä asentaa sen omaan laitteeseensa. (Budiu 2013.)

Java

Java on laitteistoriippumaton oliopohjainen ohjelmointikieli, jonka on julkaissut Sun Microsystems (Rouse 2016).

Lokalisointi

Ohjelman lokalisoinnilla tarkoitetaan ohjelmoinnissa käytettävän kielen sopeuttamista tietyn kielialueen kohdekulttuuriin, eli kirjotetaan ohjelma kohdekulttuurille ominaisella tyylillä (Rouse 2015).

Natiivisovellus

Natiivisovelluksella tarkoitetaan sovellusohjelmaa, joka on kehitetty jonkin tietyn laitteen tai alustan käyttöön. Älypuhelinsovelluksissa tämä tarkoittaa vaikkapa Android-älypuhelimelle luotua sovellusta. Muihin sovellustyyppeihin verrattuna natiivisovellukset tarjoavat käyttäjille nopeamman, luotettavamman ja responsiivisemman käyttökokemuksen. Ne myös pääsevät paremmin käsiksi laitteen ominaisuuksiin, kuten kameraan. (Budiu 2013.)

SQLite

SQLite on kevyt avoimen lähdekoodin relaatiotietokantajärjestelmä ja on upotettuna kaikkiin Android-laitteisiin (SQLite 2016).

Web-sovellus

Web-sovelluksiin voidaan luokitella mikä tahansa verkkosivun komponentti, joka toteuttaa jonkinnäköisen toiminnon loppukäyttäjälle, esimerkiksi Googlen hakukone. Web-sovellukset

ovat saatavilla kaikille laitteille, joissa on modernien verkkotekniikoiden kanssa yhteensopiva verkkoselain, sekä yhteys verkkoon. Web-sovelluksia ei siis tarvitse kehittää erikseen halutuille kohdealustoille, vaan ohjelman on tarkoitus toimia samalla tavalla jokaisen laitteen selaimella. Sovellus voi kuitenkin olla erilainen ulkonäöltään eri selaimilta katseltaessa, sillä selaimet eivät välttämättä näytä kaikkia elementtejä samalla tavalla. (Budiu 2013.)

WebView

WebView on Android järjestelmäkomponentti, jonka avulla Android-sovellukset voivat tuoda esille verkkosisältöä (Meier 2012, 201).

XML

XML, eli Extensible Markup Language on merkkauskieli. Androidissa käyttöliittymätiedostot ja muut resurssit kirjoitetaan XML:llä (Rouse 2014).

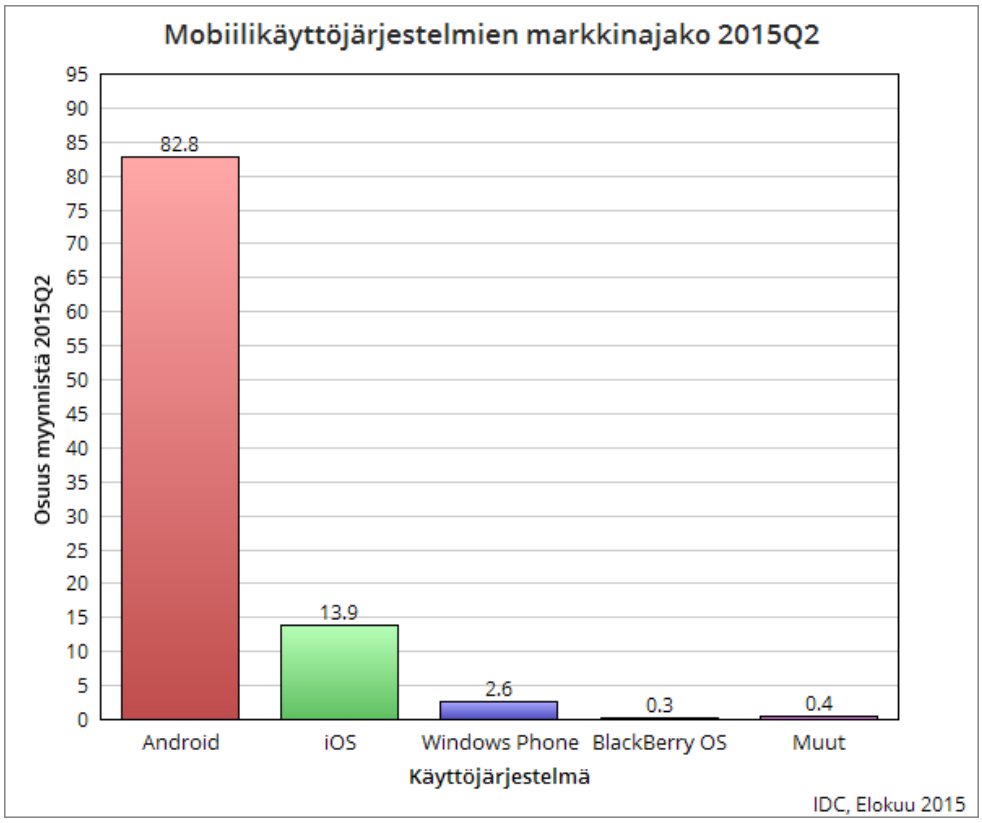
3 Android-käyttöjärjestelmä

Android on Googlen ja Open Handset Alliancen kehittämä ja vuonna 2008 julkaisema avoimen lähdekoodin ohjelmistopino mobiililaitteille, joka sisältää Googlen muokkaaman linux-käyttöjärjestelmäytimen. Sen lisäksi ohjelmistopino sisältää Bionic-C kirjaston ja muita järjestelmäkirjastoja, ohjelmistokehyksen sekä Dalvik-virtuaalikoneen, jossa suoritetaan Androidin järjestelmän prosessit ja muut sovellukset (Androidsuomi.fi 2016; Rantakeisu 2015; Rouse 2008). Selkeyden vuoksi tässä opinnäytetyössä ohjelmistopinoon viitatessa puhutaan käyttöjärjestelmästä. Älypuhelinten ja tablettien lisäksi Android-käyttöjärjestelmää käytetään myös älyrannekelloissa, televisioissa sekä Google Glass –älylaseissa. (Meier 2012).

Open Handset Alliance on 84 yrityksen ryhmittymä, jonka tavoitteena on mobiilialan avointen standardien ja Android-käyttöjärjestelmän kehittäminen. Ryhmittymä koostuu teleoperaattoreista, laitevalmistajista, sovelluskehittäjistä ja mainosalan yrityksistä. Kaikki ryhmittymän jäsenet ovat sitoutuneet tekemään Android-alustasta kaupallisesti menestyvän. Osa jäsenyrityksistä on lahjoittanut ryhmittymälle merkittäviä immateriaalioikeuksiaan, jotka julkaistaan avoimen lähdekoodin lisenssillä. Laitevalmistajat ja teleoperaattorit varmistavat osaltaan laitteiden ja piirisarjojen yhteensopivuuden alustan kanssa. Muutamia esimerkkejä ryhmittymän tunnetuimmista jäsenistä Googlen lisäksi ovat Dell, Sony Ericsson, Intel, Nvidia, Accenture, Vodafone ja T-Mobile. (Open Handset Alliance 2016.)

Android on tällä hetkellä maailman myydyin mobiilikäyttöjärjestelmä. Vuoden 2015 toisella neljänneksellä kaikista maailman myydyistä älypuhelimista 82,8 prosenttia toimivat Android-

käyttöjärjestelmällä. Menestyksekkäin Android-älypuhelimia valmistava yritys on tällä hetkellä Samsung, jonka mallisto tarjoaa puhelimia hintahaarukan molempiin päihin. Muita menestyviä Android-älypuhelin valmistajia ovat muun muassa LG, Motorola, Sony, Huawei, Xiaomi ja Lenovo (IDC 2015). Kuviossa yksi tuodaan esiin mobiilikäyttöjärjestelmien älypuhelinten markkinaosuudet vuodelta 2015.



Kuvio 1: Mobiilikäyttöjärjestelmien markkinajako 2015 (IDC 2015)

3.1 Androidin versiohistoria

Androidin ensimmäisen version julkaisun jälkeen vuonna 2008 siitä on kehitetty useita versioita, jotka versiosta 1.5 eteenpäin on nimetty makeisiin viittaavilla koodinimillä. Taulokussa yksi tuodaan esiin Androidin versiohistoriaa, versioiden julkaisupäivät ja Api-tasot sekä muutamia esimerkkejä tärkeimmistä uusista ominaisuuksista.

Koodinimi	Versio	Esimerkkejä ominaisuuksista	Julkaisu	Api-tasot
	1.0 –1.1	- Sovellusten lataaminen ja päivittäminen Android Market sovelluksessa - Internet selain - Tuki Googlen sovelluksille, kuten Gmail, Maps, Contacts, Youtube	23.9.2008	1 – 2
Cupcake	1.5	- Virtuaalinäppäimistö tekstin ennustuksella - Videoiden nauhoitus ja katselu - Bluetooth tuki A2DP ja AVRCP profiileille	30.4.2009	3
Donut	1.6	- Gesture rajapinta - Vuoropohjainen navigointi	15.9.2009	4
Eclaire	2.0 – 2.1	- Päivitetty käyttöliittymä - Bluetooth 2.1 tuki - Uusia kameran ominaisuuksia	26.10.2009	5 – 7
Froyo	2.2 – 2.2.3	- Parannuksia nopeuteen - USB yhteyden jakaminen - Tiedostojen lähetys selaimessa	20.3.2010	8
Gingerbread	2.3 – 2.3.7	- Päivitetty käyttöliittymä - Paranneltu kopioi/liitä toiminto, virran ja näppäimistön käyttö - Tuki videopuheluille	6.12.2010	9 – 10
Honeycomb	3.0 – 3.2.6	- Tuki moniydin prosessoreille - Päivitetty 3D käyttöliittymä - Parempi tuki tableteille	22.2.2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	- Verkon datan hallinnointi - Kasvontunnistus lukituksen avaamiseen - Selaimeen tuki 16 välilehdelle - Käyttöliittymän rautakiihdytys (Hardware acceleration)	18.10.2011	14 – 15
Jelly Bean	4.1 – 4.3.1	- Google Now - Haku äänen avulla (Voice Search) - 4k resoluutio tuki - Parannuksia suorituskykyyn ja tietoturvaan	9.7.2012	16 – 17
KitKat	4.4 – 4.4.4	- Näytön nauhoitus - Uusi järjestelmän käyttöliittymä - Parannuksia suorituskykyyn	31.10.2013	19 – 20
Lollipop	5.0 – 5.1.1	- Uusi muotoilu (Material design) - Tuki useammalle sim-kortille - Paranneltu virrankäyttö	17.10.2014	21 – 22
Marshmallow	6.0 – 6.0.1	- USB Type-C tuki - Sormenjälkitunnistus	5.10.2015	23
Nougat	7.0	- Paranneltu moniajo (multitasking) - Saumattomat järjestelmäpäivitykset	22.8.2016	24

Taulukko 1: Androidin versiohistoria (SocialCompare 2016)

3.2 Sovelluskehitys

Mobiilisovelluksella tarkoitetaan sovellusohjelmaa, joka on kehitetty mobiililaitteelle, kuten älypuhelimelle tai tabletille, pöytätietokoneiden ja kannettavien tietokoneiden sijaan.

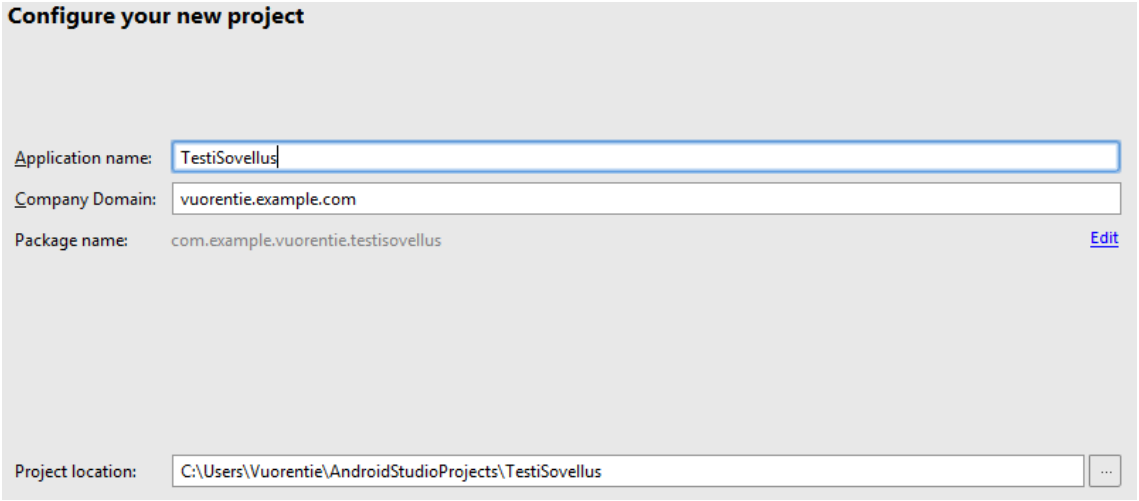
Mobiilisovellukset voidaan jakaa kolmeen eri kategoriaan sen perusteella, ovatko ne laitteille asennettavia natiivisovelluksia, selaimessa toimivia web-sovelluksia tai molempia tapoja ja tekniikoita yhdisteleviä hybridisovelluksia. (Budiu 2013; Rouse 2008.)

Mobiililaitteiden natiivisovellukset ovat tietylle mobiilialustalle, kuten Androidille tai iOS:lle kehitettyjä sovelluksia. Websovelluksia natiivisovelluksiin verrattaessa, on natiivisovelluksilla yleensä parempi suorituskyky ja ovat toimivuudeltaan luotettavampia. Natiivisovelluksilla on myös suora pääsy laitteen ominaisuuksiin, kuten GPS tai kamera. Tyypillisesti nämä sovellukset asennetaan jonkin sovelluskaupan, kuten Google Play kautta, mutta myös muita tapoja voi käyttää. (Budiu 2013; Technopedia 2016.)

Natiivisovellusten kehittäminen Androidille vaatii sovelluskehitystyökalun, joka tukee Androidille ohjelmointia eli sisältää Android SDK:n (software developement kit). Yleisimmät IDE:t (Integrated developement environtment) Androidille kehittämiseen ovat Eclipse ja Android Studio, mutta Android Studio on Googlen julkaisema ja tukema virallinen Android-kehitystyökalu. Androidin natiivisovellukset kirjoitetaan Javaa ja XML-merkkauskieltä käyttäen. (Ravencraft 2016.)

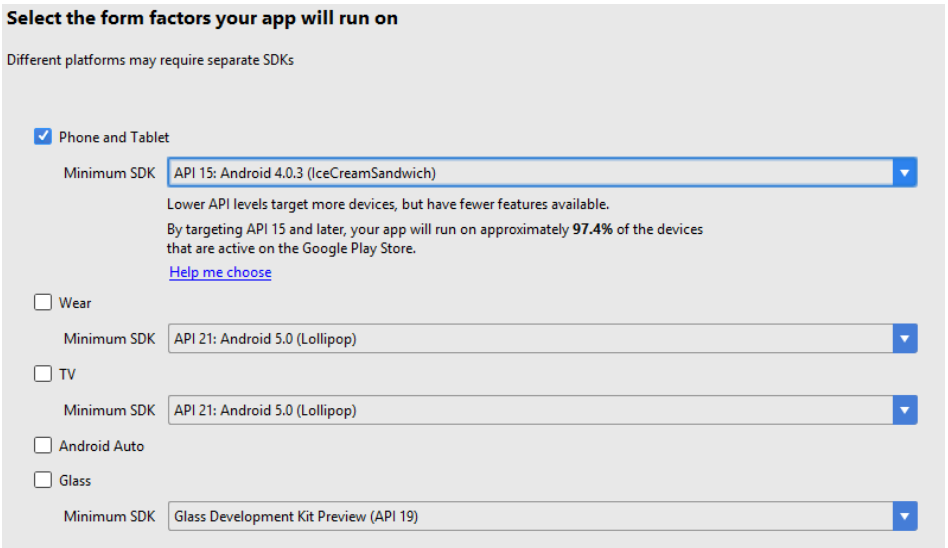
3.2.1 Sovelluksen luominen Android Studiossa

Android Studiossa uuden sovellusprojektin voi aloittaa navigoimalla ylävalikosta “File” ja “New Project” tai aloitusikkunasta valitsemalla “Start a new Android Studio project”. Tämän jälkeen aukeavassa ikkunassa valitaan “Application name” eli sovelluksen nimi, “Company Domain” eli yrityksen verkkotunnus sekä tiedostopolku projektille. Sovelluksen nimeksi voi asettaa minkä tahansa haluamansa nimen. Yrityksen verkkotunnusta käytetään sovelluksen tunnistamiseen. Ajatuksena on, että vaikka kahdella eri sovelluksella olisikin sama nimi, niiden verkkotunnukset eroavat toisistaan. Yksityishenkilö voi antaa yrityksen verkkotunnukseksi minkä tahansa fiktiivisen tunnuksen. Näillä asetuksilla on merkitystä vain, jos kehittäjä haluaa julkaista sovelluksensa Googlen julkaisualustassa Google Play:ssä (Jenkov 2014). Kuviossa kaksi näytetään uuden projektin konfigurointi-ikkuna.



Kuvio 2: Uuden projektin konfigurointi

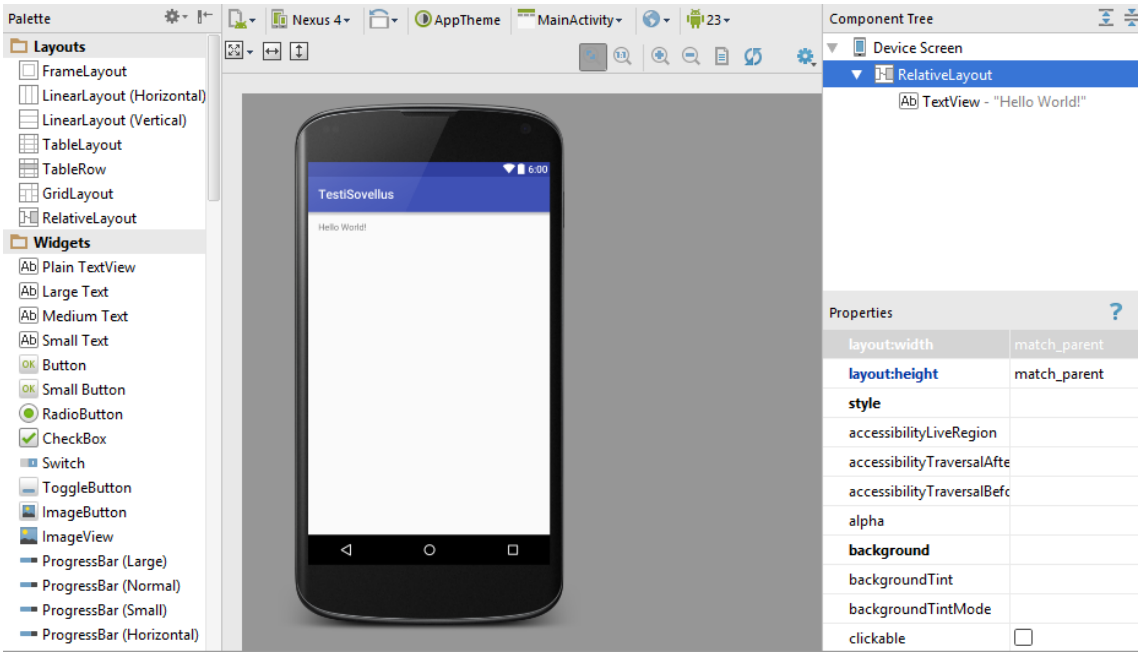
Seuraavassa ikkunassa valitaan sovelluksen kohdelaite sekä API-taso. Jos sovellus ei vaadi uudempien API-tasojen tuomia ominaisuuksia, on suotavaa valita aikaisempi versio, jolloin sovellukselle saadaan laajempi käyttäjäkunta, koostuen vanhemmista sekä uudemmista laitteista. Kolmannessa ikkunassa voidaan valita sovellukseen liitettävä aktiviteetti. Tämän jälkeen aktiviteetti pyydetään nimeämään, mutta oletusarvoja voi myös käyttää. Kuviossa kolme näkyy kohdelaitteen valintaikkuna.



Kuvio 3: Kohdelaitteen valinta

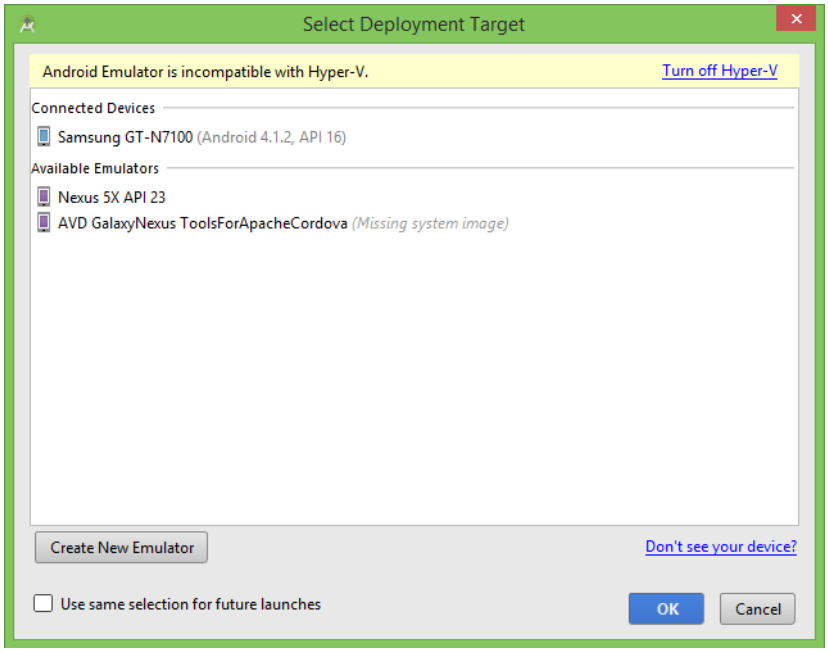
Aktiviteetin nimeämisen jälkeen Android Studio luo projektin. Projektin rakentamisen valmistuttua käyttäjälle aukeaa Android Studion työtila (kuvio 4), jossa on sovelluksen luontiin käytettäviä työkaluja. Sovelluksen ulkoasua voi katsella ja muokata käyttöliittymäeditorissa. “Design” näkymässä käyttöliittymään voidaan liittää haluttuja sovelluskomponentteja vetämällä ne vasemmalla olevasta sovelluskomponenttivalikosta

design näkymään, tai oikeasta yläkulmasta löytyvään “Component Tree” hierarkiaan. Käyttöliittymätiedostot voi myös kirjoittaa käsin XML-merkkauskieltä käyttäen “Code” näkymässä. Tällä tavalla voidaan vaikuttaa enemmän sovelluksen käyttöliittymän saavutettavuuteen ja parantaa ohjelman lokalisointia. Sovelluksen ohjelmakoodilliset toiminnallisuudet voi kirjoittaa projektihierarkian java-kansiosta löytyviin javaluokkiin. (Nurik 2010.)



Kuvio 4: Android Studion työtila

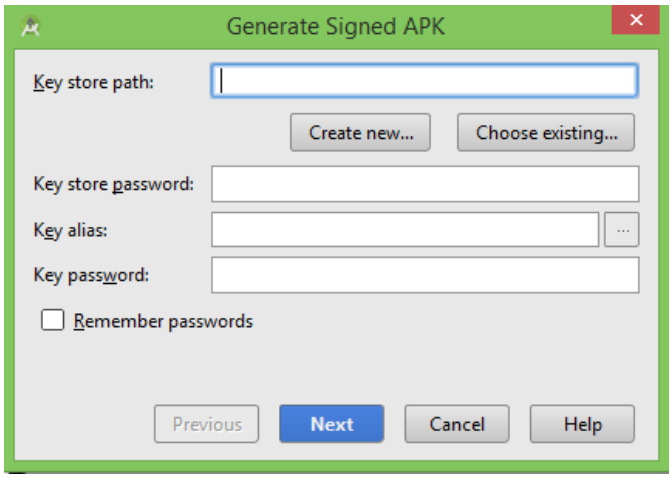
Sovellusta on syytä testata kehityksen aikana aina uusien muutosten myötä. Testaukseen sovellusta voi ajaa joko emulaattorissa tai yhteensopivassa laitteessa, joka on liitetty kehityskoneeseen USB-liitännällä. Sovelluksen ajaminen onnistuu valitsemalla ylävalikosta “Run” ja “Run App” tai painamalla työkalurivillä näkyvää virheää kolmiota. Tämän jälkeen aukeavasta ikkunasta valitaan alusta, jossa sovellus halutaan ajaa (kuvio 5). Sovellusta voi ajaa emulaattorissa, tai yhteensopivassa laitteessa, jos sellainen on kytketty kehityskoneeseen. Emulaattori on hyvä keino luoda toivotun kohdelaitteen omaava ympäristö, sillä esimerkiksi laitteen näytön koko, resoluutio ja saatavilla oleva muisti on vapaasti muokattavissa. Se ei kuitenkaan vastaa yhtä hyvin käyttäjän toimintoihin tai keskeytyksiin kuin oikea laite, eikä siinä ole mahdollisuutta käyttää esimerkiksi kameran ominaisuuksia. (Helppi 2013.)



Kuvio 5: Laitteen valinta

3.2.2 Sovelluksen jakaminen

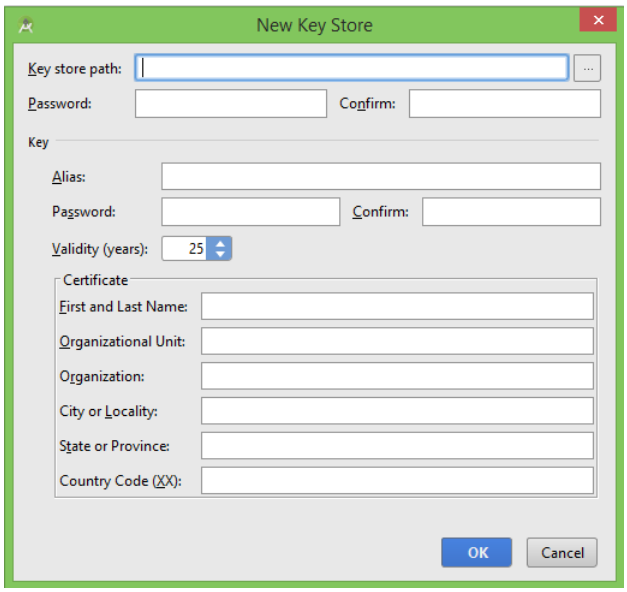
Kun sovellus on valmis, siitä voi generoida asennustiedoston, jota voi jakaa asennettavaksi laitteisiin esimerkiksi sähköpostitse liitetiedostona tai julkaisemalla sen Google Play:ssä. Google Play:ssä julkaiseminen on maksullista ja vaatii digitaalisesti allekirjoitetun asennuspaketin. Asennuspaketin luominen tapahtuu valitsemalla ylävalikosta “Build” ja “Build APK” allekirjoittamattomille asennuspaketeille tai allekirjoitetuille “Build” ja “Generate Signed APK” (Developer 2016). Kuviossa kuusi näkyy ikkuna uuden digitaalisesti allekirjoitetun asennuspaketin luomiseen.



Kuvio 6: Digitaalisesti allekirjoitetun asennuspaketin luominen

Avainsäiliö (Key store) on sovelluskehittäjän sertifiointi työkalu. Tyypillisesti avainsäiliön avulla todennetaan sovelluksen asennuspaketti, mutta sillä voi tehdä muitakin käyttäjän

todennusta vaativia toimintoja. Uuden avainsäiliön voi luoda pianamalla “Create new...” painiketta “Generate Signed APK” ikkunassa. Sen jälkeen aukeavaan ikkunan (Kuvio 7) ylimpiin kenttiin tulee syöttää uuden avainsäiliön tallennuspolku sekä haluttu salasana avainsäiliölle. Seuraavat kolme kenttää vaativat nimen ja salasanan avaimelle, sekä avaimen voimassaoloajan. Viimeisiin kenttiin syötetään sertifiointitiedot (Developer 2016.)



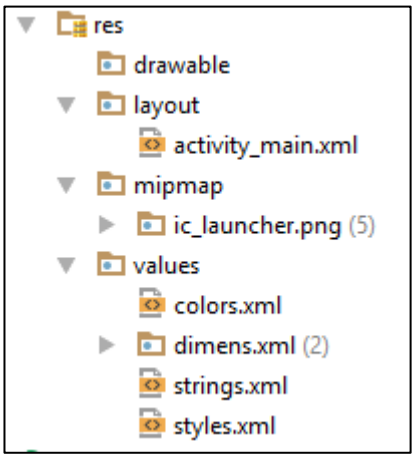
Kuvio 7: Uuden avainsäiliön luominen

3.3 Sovelluksen rakenne

Android-sovellukset koostuvat löyhästi toisiinsa yhdistyvistä komponenteista, joita sitoo manifest tiedosto. Manifest-tiedoston tehtävä on kuvailla järjestelmälle sovelluksen tiedot, luvat ja ulkoiset kirjastot, jotka vaaditaan ohjelmakoodin käynnistämiseen. Jokaisessa Android-projektissa on oma manifest-tiedostonsa, AndroidManifest.xml, tallennettuna projektihierarkian juureen. Manifest tiedoston lisäksi tyypillinen Android-sovellus sisältää Javaluokkia, käyttöliittymän tiedostoja ja ulkoisia resursseja (Meier 2012, 54-55). Kuviossa kahdeksan on esitetty Android-sovelluprojektin projektihierarkia.

Kuvio 8: Projektihierarkia

Android-ohjelmistokehittäjiä kehotetaan erottelemaan koodia sisältämättömät resurssit ohjelmakoodista erillisiin tiedostoihin, joihin voidaan viitata tarvittaessa ohjelmakoodissa. Ulkoisiin resursseihin kuuluu yksinkertaisemmat arvot, kuten merkkijonot ja värit, sekä monimutkaisemmat resurssit, kuten kuvat, animaatiot, teemat ja valikot. Merkittävimmät ulkoiset resurssit ovat kuitenkin layoutit eli käyttöliittymätiedostot (Meier 2012, 64). Resurssit tallennetaan projektihierarkiassa 'res'-kansion (kuvio 9) alle resurssin tyyppiä vastaavaan kansioon.



Kuvio 9: Res-kansio

Aktiviteetit (Activities) vastaavat Android-sovelluksessa tiedon asettelusta ja näyttämisestä, sekä käyttäjän toimintoihin vastaamisesta. Sovellusten käyttöliittymä rakennetaan yhden tai useamman aktiviteettiluokan ympärille. Aktiviteetit käyttävät fragmentteja (Fragments) ja näkymiä (Views) tiedon asetteluun ja näyttämiseen sekä käyttäjän toimintoihin vastaamiseen.

Palvelut (Services) ovat sovelluksen taustalla toimivia komponentteja, jotka toimivat ilman käyttöliittymää, päivittävät tiedonlähteitä ja aktiviteetteja, laukaisevat ilmoituksia ja lähettävät aikeita (Intents). Niitä käytetään pitkäkestoisten ja käyttäjän vuorovaikutusta vaatimattomien tehtävien suorittamiseen.

Sisällön tarjoajat (Content providers) hallitsevat ja säilyttävät sovelluksen dataa ja ja ovat tyypillisesti vuorovaikutuksessa SQL tietokantojen kanssa. Ne ovat myös ensisijainen tapa jakaa tietoa sovelluksien rajojen ulkopuolelle. Sisällön tarjoajat toimivat paljolti samalla tavalla kuin tietokannat johon voi tehdä kyselyitä sekä muokata, liäsätä tai poistaa sen siältöä.

Aikeet (Intents) ovat tehokas viestinvälitys rajapinta sovellusten välillä. Aikeita käytetään laajasti koko Android-käyttöjärjestelmässä. Aikeiden avulla voi käynnistää tai sulkea aktiviteetteja ja palveluita, lähettää viestejä järjestelmälaajuisesti tai tiettyyn aktiviteettiin,

palveluun tai lähetyksen vastaanottimeen sekä pyytää toimintoa suoritettavaksi tietyn tiedonpalan (datan) avulla.

Lähetysten vastaanottimien (Broadcast receivers) avulla sovelluksen voi määrittää kuuntelemaan viestejä muista sovelluksista tai järjestelmästä, jotka vastaavat asetettuja kriteereitä. Näitä viestejä kutsutaan tyypillisesti tapahtumiksi tai aikeiksi. Lähetysten vastaanottimet voivat esimerkiksi käynnistää sovelluksen reaktiona vastaanotettuun viestiin.

Ilmoituksilla (Notifications) voidaan ilmoittaa käyttäjälle sovelluksen tapahtumista tuomatta esiin sovelluksen aktiviteetteja, eli ilman täyttä käyttäjän huomion viemistä tai aktiviteetin keskeyttämistä. Ilmoitukset käsittelee erillinen Notification Manager, jonka avulla ilmoitus voi näyttää edistymispalkkia, vilkuttaa laitteen valoja tai LEDejä, värisyttää puhelinta, soittaa huomautusääniä, tulostaa tietoa ilmoitustaululle sekä lähettää aikeita. (Meier 2012, 54-55.)

3.4 SQLite

SQLite on kansainvälisen tiimin kehittämä avoimen lähdekoodin relaatiotietokantajärjestelmä. Toisin kuin useimmat SQL-tietokannat, SQLitellä ei ole erillistä serveriprosessia, vaan koko SQLite-järjestelmä liitetään sitä käyttävään sovellukseen. Se on toteutettu kompaktina C-kirjastona, jonka koko voi olla jopa alle 500 kilotavua ja pienen muistitarpeensa takia se on suosittu tietokantaratkaisu esimerkiksi puhelimissa ja mp3-soittimissa. (SQLite 2016.)

Android-sovelluskehityksessä SQLiten käyttöönotto ei vaadi minkäänlaista asennusta vaan se on sisällytetty kaikkiin android-laitteisiin. Sovelluksen tulee tyytyä siihen versioon SQLitestä, joka laitteeseen on upotettu. Api-tason 7 ja alemman tason laitteissa olevista SQLite versioista puuttuu osa sen nykyisistä ominaisuuksista. SQLiten käyttäminen sovelluksessa onnistuu “SQLiteOpenHelper” luokan avulla . (Rittmeyer 2013.)

4 Tutkimusmenetelmät

Empiirisessä, eli kokemuspohjaisessa tutkimuksessa, tutkimusmenetelmillä tarkoitetaan laadullisia (kvalitatiivisia) ja määrällisiä (kvantitatiivisia) tiedon keruu- ja analysointi menetelmiä. Niillä on tarkoitus kerätä konkreettista aineistoa ja selventää analyysimetodeja (Saukkonen 2003). Tutkimusta tehtäessä on syytä rajata tutkimusalue tutkimusongelmaan, jonka avulla tutkimus saadaan helpommin pysymään aiheessa ja kiteytettyä se, mitä tutkimuksessa halutaan tietää ja tutkia (KvaliMOTV 2006).

Tässä opinnäytetyössä kehittämisongelmana oli inventointisovelluksen kehittäminen mahdollisimman yksinkertaiseksi ja nopeasti käytettäväksi sekä sovelluksen jakaminen

loppukäyttäjille. Oli tarpeen myös arvioida sovellukselle asetettavat vaatimukset, millaisissa vaiheissa kehittämisessä edetään sekä millaiseen lopputuotokseen se johtaa. Jotta lopputuotos vastaisi sille asetettuja vaatimuksia, edellytettiin riittävän kattavaa käytännönläheisen tiedon hankintaa käyttäen havainnoinnin ja avoimen haastattelun tutkimuksellisia menetelmiä.

Opinnäytetyössä käytetty tutkimusmenetelmä on toimintatutkimus, jonka ajatuksena on muuttaa tai kehittää jotakin eli saada käytännössä aikaan todellista muutosta. Se on prosessi, joka tähtää asioiden muuttamiseen ja kehittämiseen paremmaksi. Toimintatutkimukseen kuuluu ongelma-analyysi, jonka avulla selvitetään kehittämisen tarpeet esimerkiksi kyselyä käyttäen, kehittämisprosessin tai -toiminnan arviointi sekä kehittämisestä syntyvä tuotos. (Edumondo 2016; Hyöky & Kyllönen. 2013; Virtuaaliammattikorkeakoulu 2007.)

Työ voidaan luokitella projektisuuntaiseksi laadulliseksi toimintatutkimukseksi. Projektin toteuttamista varten perehdyin seuraamalla ja havainnoimalla loppukäyttäjien työprosessia, sekä järjestin haastattelutilanteen, jossa haastateltavat kertoivat käyttökokemuksiaan ja huomioitaan mobiilisovelluksen silloiseen versioon liittyen. Haastattelua varten loppukäyttäjät testasivat työssään järjestelmän Windows Phone -mobiilisovelluksen varhaisversiota. Haastattelumateriaalia analysoimalla keräsin kehitysehdotuksia uutta mobiilisovelluksen versiota varten. Kehitysehdotusten perusteella pystyin muuttamaan loppukäyttäjälle tarpeettomia toimintoja sekä tuottamaan mielekkäämmän ja toiminnallisesti tehokkaamman ratkaisun mobiilisovellukseen.

Avoin haastattelu on aineistonkeruumenetelmä, jossa ollaan suorassa vuorovaikutuksessa tutkimuksen kohteen, eli haastateltavien henkilöiden kanssa. Tarkkojen kysymysten sijaan avoimessa haastattelussa on tarkoitus hankkia haastatteluaineistoa tutkimuksen kohteista keskustelunomaisesti ja antaen tilaisuuden haastateltavalle tuoda esiin hänen kokemuksiaan, mielipiteitään ja näiden perusteluita. (KvaliMOTV 2006.)

4.1 Kvalitatiivinen ja kvantitatiivinen tutkimusmenetelmä

Tutkimuksissa käytettävät menetelmät voidaan jakaa karkeasti laadullisiin eli kvalitatiivisiin sekä määrällisiin eli kvantitatiivisiin tutkimusmenetelmiin. Näiden tutkimussuuntausten eroja on pyritty havainnollistamaan, mutta käytännössä niitä on vaikea erottaa toisistaan, eikä tästä ole apua käytännön tutkimuksessa. Tutkimussuuntausten epäselvä määrittely on aiheuttanut sekaannusta, jossa esimerkiksi laadullinen metodologia yhdistetään tiettyihin aineistonkeruu tapoihin tai vaikeasti mitattavaan piirteeseen. (Hirsjärvi , Remes & Sajavaara. 2003, 123-124.)

Kvalitatiivisen tutkimuksen lähtökohtana on todellisen maailman havainnollistaminen. Siinä pyritään tutkimaan mahdollisimman kokonaisvaltaisesti tutkimuksen kohdetta, eli pyritään ymmärtämään, tulkitsemaan ja kuvaamaan sitä. Kvalitatiivisella tutkimuksella voidaan tutkia monimutkaisia ilmiöitä ja prosesseja, ilmiöitä, joita ei tunneta hyvin tai sitä koskevia muuttujia ei ole vielä tunnistettu. Laadullista tutkimusta voidaan kritisoida siitä, että se antaa kuvan jostakin laadukkaasta tai paremmasta. (Hirsjärvi ym. 2003, 124, 152-153; Niinisalo 2016.)

Kvantitatiivinen tutkimus tulkitsee ja kuvaa ilmiöitä numeerisen tutkimusaineistojen mittausmenetelmillä. Se perustuu mittaamiseen ja siitä saatavan havaintoaineiston analysointiin, eli on tärkeää, että aineisto soveltuu määrälliseen mittaamiseen. Kvalitatiivisen tutkimuksen tulosten luotettavuus voidaan varmistaa sillä, että käytettävä tutkimusaineisto on riittävän kattava ja edustava. (Vilpas 2013.)

Laadullisissa tutkimuksissa aineistonkeruumenetelmiin voidaan luokitella erilaiset haastattelut, suora ja osallistuva havainnointi, sekä valmiiden aineistojen ja dokumenttien käyttö, joista saadaan aikaan teoriaa. Kvantitatiivisen tutkimuksen aineistonkeruumenetelmiin kuuluu muun muassa mittaukset, testaukset ja havainnointi määrällisistä asioista, joista tulokseksi saadaan tilastoja tai taulukoita. (Pitkäranta 2010.)

Tässä työssä on käytetty aineistonkeruumenetelminä havainnointia ja avointa haastattelua. Havainnoinnista ja haastattelusta saadun aineiston analysoinnilla sain vaaditut tiedot inventointi järjestelmän luontiin ja jatkokehittämiseen. Käytetyt metodit voidaan luokitella laadullisiksi menetelmiksi. Kvantitatiivisia metodeja ei käytetty työssä, sillä siinä ei syntynyt tilastoja tai muita tilastollisesti käsiteltäviä tuloksia.

4.2 Reliabiliteetti ja validiteetti

Reliabiliteetilla tarkoitetaan tutkimuksessa käytetyn tutkimusmenetelmän tai mittarin luotettavuutta, eli näiden kykyä saavuttaa tarkoitettuja tuloksia. Hyvä reliabiliteetti saavutetaan tutkimustyön huolellisuudella ja se merkitsee käytännössä sitä, että tutkimuksen tuloksia voidaan pitää toistettavina eivätkä ne ole sattumanvaraisia. (Saukkonen 2013; Virtuaali Ammattikorkeakoulu 2007.)

Tutkimuksen validius tarkoittaa aineistoista tehtyjen johtopäätösten luotettavuutta, eli onko tutkimus pätevä ja onko saadut tulokset ja tehdyt päätelmät oikeita. Tutkimuksen kannalta validius on merkittävä laadun kriteeri, sillä siinä on kyse saatujen tulosten oikeellisuudesta. Validiteettia voidaan arvioida vertaamalla mittaustulosta todelliseen tietoon mitattavasta ilmiöstä. (Saukkonen 2003; Virtuaali Ammattikorkeakoulu 2007.)

Reliabiliteetin arvioinnissa selvitetään tutkimuksen instrumentin tarkkuutta ja verrataan sen objektiivisuutta ja subjektiivisuutta, sekä havainnon jatkuvaa samankaltaisuutta. Validiteetin arvioinnissa selvitetään, kuinka hyvin tutkimus ja siinä käytetyt menetelmät vastaavat tutkimuksen kohteena olevaa ilmiötä. Tutkimusmenetelmä voi olla reliaabeli, vaikka tutkimus ei olisikaan validi, jolloin tulokset eivät vastaa sitä mihin tutkimuksella tähdättiin. (Virtuaali Ammattikorkeakoulu 2007.)

4.3 Lähdekritiikki

Lähdekritiikillä tarkoitetaan tutkimuksessa käytettävien tietolähteiden käyttökelpoisuuden arviointia, eli selvitetään voiko tiedontuottajaan luottaa. Luotettavuutta on syytä arvioida koko tiedonhakuprosessin ajan. Arvioitaessa tulee kiinnittää huomiota tiedontuottajan lähteiden ajankohtaisuuteen sekä alkuperään, onko tiedontuottajan tai -tuottajien nimet mainittu, ja kenen julkaisema tai ylläpitämä lähde on. (Opetushallitus 2002; Tilastokeskus 2015.)

Internet on kattava tiedon julkaisukanava, mutta sinne on myös erittäin helppo tuottaa tietoa, eikä tiedon paikkansapitävyyttä välttämättä tarkisteta millään tavoin. Esimerkiksi kaupallisilla sivustoilla ei niinkään ole yhtä paikkansapitävää tietoa kuin tutkimuslaitosten sivustoilla. Painetut lähteet ovat luotettavampia käyttää myös siinä mielessä, että niitä ei voi verkkolähteiden tapaan muuttaa tai poistaa. (Tilastokeskus 2015.)

Tämän työn viitekehyksenä on käytetty enimmäkseen internetissä julkaistuja materiaaleja liittyen mobiilisovelluksiin ja niiden kehitykseen, sekä kirjallisena lähteenä “Professional Android 4 application development” -teosta. Lisäksi tiedonlähteenä on käytetty esimerkiksi koulutusorganisaatioiden sivustoja. Järjestelmää toteutettaessa on kirjallisuuden ja verkkolähteiden lisäksi käytetty omaa aikaisempaa osaamista.

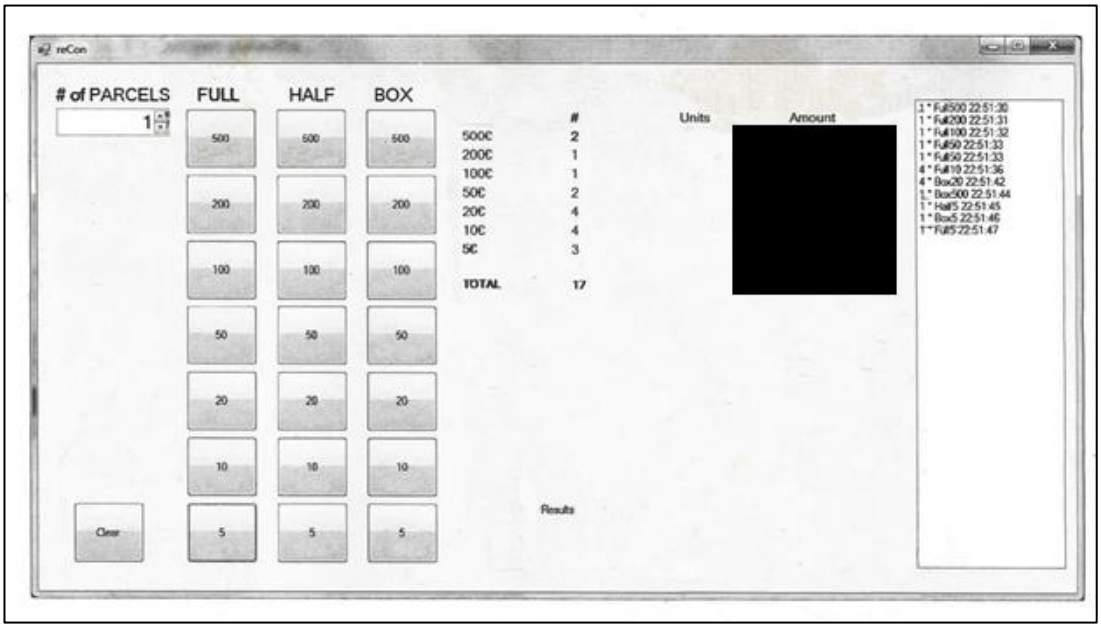
5 Android-mobiilisovelluksen kehittäminen

Työssä toteutettiin projektihankkeena rahanlaskentamobiilisovellus Android-alustalle rahainventointiin. Sovelluksen tilaajana oli Yritys X:n maksuvälineistä vastaava osasto. Sovelluksen avulla tulee kyetä inventoimaan rahakontteja ja niiden sisältämiä rahamääriä kappaleittain, mutta rahojen varsinaisilla arvoilla tai arvojen summilla ei ole käytännön tarvetta.

Sovellus toteutettiin asiakkaan Windows Formsilla tuottaman prototyypin pohjalta, josta selvisi sovelluksen käyttötarkoitus ja keskeisimmät toiminnot sekä suurpiirteinen kuva toivotusta käyttöliittymästä. Asiakkaan kanssa päätettiin, että sovelluksen voisi toteuttaa mobiilialustalle loppukäyttäjillä olevien työsuhdepuhelimien ja sovelluksen yksinkertaisuuden

vuoksi. Koska heillä oli käytössään sekä Windows Phone, että Android-älypuhelimia, sovellus tuli kehittää molemmille käyttöjärjestelmille.

Sovelluksen prototyyppi oli suunniteltu pöytä- ja kannettavalle tietokoneelle, joten sen käyttöliittymä oli suunniteltava uudelleen älypuhelinten pienemmille näytöille (kuvio 10). Sovelluksen toiminnallisuus oli kehitettävä kokonaan uudelleen Androidille, koska Windows ohjelman ominaisuudet eivät olleet luontevasti jäljiteltävissä toiseen alustaan. Windows Phonen osalta tämä ei ollut ongelma, vaan samankaltaisia ominaisuuksia oli käytettävissä sekä Windows Phone, että Windows Forms sovelluskehityksessä.



Kuvio 10: Järjestelmän prototyyppi

Sovellusten toteuttamiseen selvitettiin mahdollisia työkaluja ja Android-sovelluksen kehittämiseen päädyin käyttämään Android Studio -sovelluskehitystyökalua sille tarjotun tuen ja saatavilla olevan materiaalin vuoksi. Toisena mahdollisuutena pidin nykyisin Microsoftin omistamaa Xamarin Studio -ohjelmistokehitysalustaa. Xamarin Studio olisi mahdollistanut saman natiivisovelluksen yhtaikaisen kehittämisen sekä Android, Windows Phone ja tarvittaessa myös iOS-alustalle, mutta työkalun hinta ja sille saatava tuki tekivät siitä kehnon vaihtoehdon.

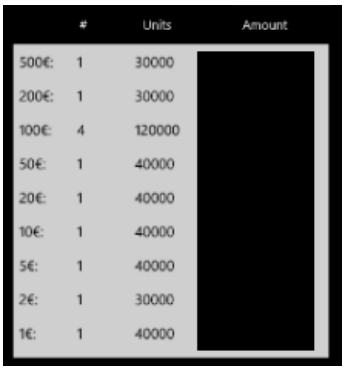
Loppukäyttäjien työtilat holvissa vaativat sovelluksen toimimaan ilman verkkoyhteyttä, joten oli luontevinta toteuttaa se laitteisiin asennettavana ja internet yhteyttä edellyttämättömänä natiivisovelluksena web-, tai hybridisovelluksen sijaan. Sovelluksen ensimmäiseen versioon asetettiin tavoitteeksi saada luotua päänäkymän käyttöliittymän alustava ulkoasu ja kontrollit eli painikkeet sekä dropdown-valikot. Tässä vaiheessa sovelluksesta puuttui vielä valintojen

mukaisen tiedon tallennus, listan rivien muokkaaminen ja poisto sekä lopputuloksen tulostus. Sovelluksen tuli olla helposti yhdellä kädellä käytettävä, joten sovelluskomponentit aseteltiin niin, että ne sopivat yhdelle näytölle ja kontrollit olivat peukalon ulottuvilla. Kuviossa 11 on esitelty näkymä sovelluksen kontrolleista.

Kuvio 11: Sovelluksen kontrollit

Valintojen mukainen tiedon tallennus tuli myös lisätä sovellukseen. Se missä Windows Phone -sovelluksessa kyseinen tieto oli helppo lisätä lista-sovelluskomponenttiin yksinkertaisena merkkirivinä sen omaa metodia käyttäen, oli Androidin lista hieman monimutkaisempi toteuttaa. Ratkaisuna ongelmaan tein sovellukseen SQLite-tietokannan, jonka sisällön mukaan lista täytettiin (populate list).

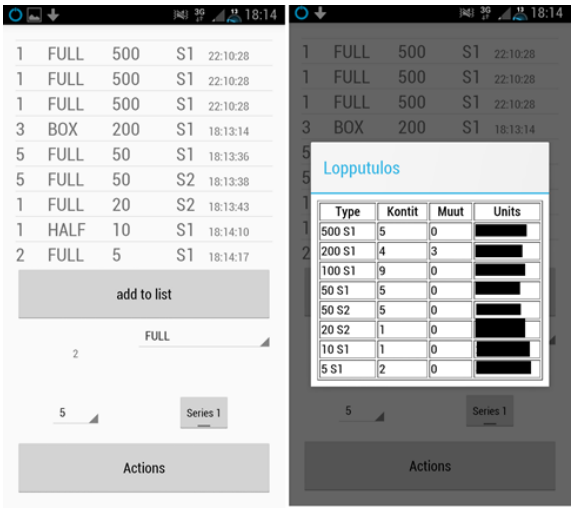
Viimeiseen loppukäyttäjille esitettävään sovellusversioon toteutettiin tiedon tallentaminen tietokantaan ja sen kautta näkyviin lista-komponenttin sekä tietokannan rivien mukaisen lopputuloksen tulostus näytölle. Tulostus toteutettiin WebView-sovelluskomponenttia käyttäen. WebView ikkuna avataan sovellukseen ja siihen kirjoitetaan pienimuotoinen HTML-dokumentti, jossa on taulukkoon tulostettuna rahatyyppien yhteenlasketut konttien ja yksittäisten kappaleiden määrät, sekä vielä tässä vaiheessa kehitystyötä rahojen arvojen summa. Kuviossa 12 on kuvakaappaus sovelluksen ensimmäisen version lopputuloksen tulostuksesta.



Kuvio 12: Lopputuloksen tulostuksen ensimmäinen versio

Sovellusta esiteltiin loppukäyttäjille, ja heidän kommenttiensa mukaan sovellukseen täytyi tehdä muutamia muutoksia. Lopputuloksen tulostuksessa oli vielä mukana rahojen arvojen summat, joka oli turhaa tietoa työprosessissa. Konttien määriä indikoivaan sarakkeeseen oli myös syytä tehdä muutos niin, että se jaettiin kahteen osaan, “kontit” ja “muut”, jossa kontit vastasivat täysinäisiä ja puolillaan olevia rahakontteja ja muut laatikoita, lavoja ja yksittäisiä kappaleita. Muita muutoksia toivottiin myös, kuten lisätyn rivin ilmestyminen listan kärkeen, listan lisäysten muokkaaminen, mahdollisuus muokata konttien sisältämä setelien lukumäärä ja tuhansien erottelu välilyönneillä.

Prototyyppiin nähden lopulliseen versioon järjestelmästä lisättiin uusia ominaisuuksia ja karsittiin turhia. Suurimmat muutokset prototyypin toimintoihin oli käyttöliittymän yksinkertaistaminen, kolikoiden inventoinnin lisääminen sekä turhan tiedon poistaminen. Viimeisin versio sovelluksesta vastasi paremmin loppukäyttäjien toiveita (kuvio 13).



Kuvio 13: Sovelluksen lopullinen ulkoasu

5.1 Suunnitelma ja toiminnan kuvaus

Sovellus antaa käyttäjän tallentaa listaan tietoa rahamääristä ja -konteista sekä tulostamaan tiedon perusteella lopputuloksen, josta selviää konttien määrät, sekä niiden sisältämien rahojen kappalemäärät. Jokaiseen listan objektiin tallennetaan tieto käyttäjän valitsemasta rahatyypistä, säilytysmuodosta, sarjanumerosta ja määrästä. Listan objekteihin tallennetaan myös tiedonsyötön kellonaika. Uusimmat lisäykset liitetään aina listan alkuun. Käyttäjä voi halutessaan poistaa tai muokata listasta virheellisiä tai tarpeettomia lisäyksiä. Listan voi myös tyhjentää yhdenaikaisesti, jolloin uuden laskelman voi aloittaa nopeasti.

Keskeisin toiminto sovelluksessa on listan lisäysten perusteella tehty tulostus, johon tulee tieto säilytystilojen ja niiden sisältämien seteleiden tai kolikoiden määrästä jokaisen

rahatyypin kohdalla. Säilytysmuodot tulostetaan loppukäyttäjien toiveesta kahteen sarakkeeseen, “kontit” ja “muut”, eli täydet ja puolikkaat kontit sekä laatikot. Työprosessissa loppukäyttäjät tarkistavat tulostuksen tiedot ja varmistavat että ne vastaavat sisäisen rahanhalliuntajärjestelmän tietoja.

5.2 Rakenne

Sovelluksen rakenne koostuu pääaktiviteetista ja sen sisällä näytettävistä dialogeista, eli pienemmistä aktiviteeteista, listasta ja sen riveistä, valikoista sekä kahdesta Java-luokasta, MainActivity.java ja DBAdapter.java. Pääaktiviteetti activity_main.xml on sovelluksen päänäkymä, jossa käyttäjä voi syöttää tiedot inventoitavista konteista tietokantaan ja sitä kautta näkyviin listaan, muokata tietokantaan lisäämiään tietoja ja poistaa ne yksitellen tai kaikki samanaikaisesti tarvittaessa. Käyttäjä voi tulostaa näytölle tekemiensä lisäysten perusteella taulukon, johon on listattu tieto jokaisen listatun rahatyypin yhteenlasketuista rahojen kappalemääristä. Varsinaiset toiminnallisuudet, tietokannan luontia ja sen muokkaamista lukuunottamatta, pää- ja muihin aktiviteetteihin tuodaan MainActivity Java-luokassa.

Dialogit ovat vain osittain näytön täyttäviä näkymiä layout-tiedostoissa (ulkoasu). Dialogissa dialog_listitem_options.xml voidaan poistaa tai siirtyä muokkaamaan yksittäistä listaan lisättyä riviä dialog_update_listitem.xml dialogiin. Dialog_numberpicker.xml avautuu kappalemääräkontrollia painettaessa, ja sen sisällä valitaan kappalemäärä NumberPicker -komponenttia käyttäen. Dialog_webview on näytön osittain täyttävä selainikkuna, jonne lopputulos tulostetaan taulukkoon HTML-kielellä.

Konttien valöörit, eli rahan tyypin ja säilytysmuodon sisältämät rahojen kappalemäärät on alunperin tallennettuna ulkoiseen resurssitiedostoon. Tiedot tallennetaan sovelluksen käynnistyessä ulkoisista resursseista ohjelmakoodiin, johon viitataan tarvittaessa. Muihin resursseihin kuuluu muun muassa valikoiden rivien tekstit ja sovelluksessa käytettyjen värien koodit.

Tiedon tallentaminen sovelluksessa toteutettiin Androidin sisäänrakennettua SQLite-tietokantaa käyttäen. Sovelluksen käynnistyessä tarkistetaan, löytyykö sen taustalta jo tietokannan nimeä vastaava tietokanta, ja jos tämä puuttuu, se luodaan. Tietokannassa on vain yksi taulu, johon tallennetaan käyttäjän syötteiden perusteella valittu säilytysmuoto, määrä sekä rahan tyyppi ja sarja. Jokaiseen tallennettuun riviin annetaan automaattisesti numeerinen tunnus ja viimeiseen kenttään tallennetaan myös tieto syötön kellonajasta. Tietokannasta löytyvät rivit tuodaan esille listaan aina jonkin tietokantaa koskevan muutoksen tapahtuessa.

5.3 Jakaminen

Sovelluksen jakaminen toteutettiin siten, että asennuspaketti lähetettiin loppukäyttäjille sähköpostilla. Kun sähköposti avattiin älypuhelimella, johon sovellus oli tarkoitus asentaa, antoi sähköpostisovellus asentaa paketin sisällön laitteseen. Sovelluksen olisi voinut jakaa myös muilla tavoin asennuspakettia käyttäen, kuten tietokoneelta puhelimeen siirtäen datakaapelin avulla tai sovelluskauppaan julkaisu, mutta yksinkertaisemmaksi ja nopeammaksi ratkaisuksi koitui sähköpostin käyttö.

Sovelluksen voi myös julkaista Googlen omaan mobiilisovellusjulkaisualustaan Google Playhin, mutta tässä tapauksessa oli suotavaa pitää sovellus vain organisaation saatavilla. Google Playssä julkaiseminen on myös maksullista ja vaatii käyttäjän rekisteröitymisen, sekä muuta ylimääräistä tietoa julkaistavaan sovellukseen liittyen. Android-sovelluksen rinnalle luotu Windows Phone -sovellus vaati julkaisun Microsoftin sovelluskaupassa, mutta siitä sai tehtyä yksityisen siten, että sovelluksen sai ladattua vain koodin avulla.

6 Yhteenveto ja johtopäätökset

Opinnäytetyölle asetetut tavoitteet saavutettiin onnistuneesti, kun inventointijärjestelmä toteutettiin mobiilisovelluksena Androidille ja Windows Phonelle. Järjestelmä antaa käyttäjän laskea suuria rahamääriä sisältäviä pakkauksia tehokkaasti niiden inventoinnissa. Sovellus otettiin käyttöön kohdeosastolla, ja saadun palautteen mukaan se helpotti ja nopeutti henkilöstön työprosessia. Projekti toteutui onnistuneesti kaikilta muilta osin, paitsi loppukäyttäjien toivomaa rahakonttien valöörien, eli niiden sisältämien rahamäärien muokkausta ei ehtinyt implementoimaan sovellukseen.

Työ eteni alussa hitaasti, sillä vielä siinä vaiheessa ei ollut varmaa, miten järjestelmä olisi luontevinta toteuttaa. Työn tahti nopeutui huomattavasti, kun toimeksiantajan kanssa oli päätetty, että järjestelmä toteutetaan sovelluksena älypuhelimelle. Ennen kuin Android Studion ensimmäinen stabiili versio julkaistiin vuonna 2014, standardi Android-ohjelmointityökalu oli Eclipse. Koska Android Studiolla on Googlen tuki ja olin jo itse tutustunut sen käyttöön, tuntui järkevimmältä ratkaisulta valita se sovelluksen kehitystyökaluksi. Saatuani tarvittavat työkalut sovelluksen toteuttamiseen, sen varsinaiseen kehittämiseen meni noin kaksi kuukautta.

Muita haasteita oli esimerkiksi sopivan alustan selvittäminen, työkalujen saanti ja toimintakuntoon laitto. Alussa työprosessin kulkua haittasi ohjelmointi- ja toteutustapavirheet. Jouduin aloittamaan sovelluksen kirjoittamisen uudelleen useampaan otteeseen tekemieni sovelluskomponentteihin ja käyttöliittymän rakenteeseen liittyvien

virhdeiden vuoksi. Suurin tekemäni virhe koodin kirjoittamisessa oli se, että tein ja kokeilin koodia ilman tarkkaa suunnitelmaa.

Projektissa olisi ollut hyvä käyttää enemmän aikaa suunnitteluun ja dokumentointiin. Epämääräinen aikataulutus ja puutteellinen suunnitelma vaikuttivat tehokkuuteen, sillä aikaa kului kokeilun ja erehdyksen kautta saavutettuihin tuloksiin. Toimeksiantaja ei myöskään saanut kirjallista palautetta projektin etenemisestä dokumentoinnin puutteesta johtuen.

Sovellusta kehittäessä kävin useaan otteseen tapaamassa työn tilaajaa. Tapaamisten aikana testattiin sovelluksen käyttöä työtilanteessa, sekä toteutettiin avoin haastattelu loppukäyttäjien kanssa. Haastattelussa loppukäyttäjät toivat esiin käyttökokemuksiaan ja toiveitaan sovelluksen toimivuuteen ja ominaisuuksiin liittyen. Työ toteutui sille asetetun aikarajan sisällä ja se otettiin käyttöön kohdeosastolla.

Opinnäytetyön aikana minulla oli mahdollista olla yhteistyössä Yritys X:n IT-osaston työntekijöiden kanssa. Heillä ei varsinaisesti ollut kokemusta mobiilisovellusten työskentelyn parissa, mutta sain heiltä tietoa sovelluksen tietokantaosuuden toteuttamiseen sekä sovellusten kehittämishankkeista yleisesti. Kehityshankkeeseen osallistuminen havainnollisti minulle konkreettisesti millaisista osa-alueista projekti koostuu.

Toimeksiannon toteuttamisen lisäksi, toisena tavoitteenani oli oman ammatillisen osaamiseni kehittäminen. Koen niiden tavoitteiden toteutuneen kohtuullisen hyvin. Tekemäni työ on saamani koulutuksen ohella ollut merkittävissä määrin edistämässä osaamistani. Mobiilisovellusten kehittäminen on mielestäni merkittävä osaamisen alue nykypäivänä, ja se voi tarjota minulle uusia mahdollisuuksia työelämässä. Työelämän projektissa työskentely oli itselleni uusi kokemus ja koen oppineeni siitäkin paljon.

Lähteet

Kirjallisuuslähteet

Meier, R. 2012. Professional Android 4 Application Development. 1. painos. Indianapolis: John Wiley & Sons.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2003. Tutki ja kirjoita. 6. - 9. painos. Helsinki: Tammi.

Sähköiset lähteet

Androidsuomi.fi. 2016. Mikä on Android?. Viitattu 25.4.2016.
http://blog.androidsuomi.fi/mika-on-android/

Budiu, R. 2013. Mobile: Native Apps, Web Apps, and Hybrid Apps. Viitattu 14.6.2016.
https://www.nngroup.com/articles/mobile-native-apps/

David, M. 2015. Learn more about the Android Studio IDE from Google. Viitattu 22.6.2016.
http://searchsoftwarequality.techtarget.com/feature/Learn-more-about-the-Android-Studio-IDE-from-Google

Developer. 2016. Android Keystore System. Viitattu 4.11.2016
https://developer.android.com/training/articles/keystore.html

Developer. 2016. Api Levels. Viitattu 20.11.2016
https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels

Developer. 2016. Sign Your App. Viitattu 3.11.2016
https://developer.android.com/studio/publish/app-signing.html

Edumendo. 2016. Tutkimus. Viitattu 20.10.2016.
http://edumendo.fi/page/tutkimus

Helppi, V. 2013. Emulator vs. Devices. Viitattu 11.10.2016
http://bitbar.com/rely-only-on-real-emulators-vs-devices/

Hyöky, A,. Kyllönen, E. 2013. Lukuhöperöksi kasvamassa. Viitattu 22.10.2016.
http://www.ouka.fi/c/document_library/get_file?uuid=23e10170-d394-414d-8102-791bf4ac3fd4&groupId=112792

IDC. 2015. Smartphone OS Market Share, 2015 Q2. Viitattu 23.4.2016.
http://www.idc.com/prodserv/smartphone-os-market-share.jsp

Jenkov, J. 2014. Your First Android App. Viitattu 23.4.2016.
http://tutorials.jenkov.com/android/your-first-android-app.html

KvaliMOTV. 2006. Tutkimusongelmat. Viitattu 13.10.2016.
http://www.fsd.uta.fi/menetelmaopetus/kvali/L2_3_1.html

KvaliMOTV. 2006. Avoin haastattelu. Viitattu 13.10.2016.
http://www.fsd.uta.fi/menetelmaopetus/kvali/L6_3_1.html

Niinisalo, T. 2016. Viitatu 14.10.2016.
https://aaltodoc.aalto.fi/bitstream/handle/123456789/20718/hse_ethesis_14393.pdf?sequence=1&isAllowed=y

Nurik, R. 2010. Viitattu 5.6.2016.
http://designbycode.tumblr.com/post/1079612795/why-dont-wysiwyg-android

Pitkäranta, A. 2010. Laadullisen tutkimuksen tekijälle. Viitattu 7.2.2016.
http://docplayer.fi/2847497-Laadullisen-tutkimuksen-tekijalle.html

Open Handset Alliance. 2016. Viitattu 6.10.2016.
http://www.openhandsetalliance.com/oha_faq.html

Pcmag. 2016. Google Play. Viitattu 23.11.2016.
http://www.pcmag.com/encyclopedia/term/64138/google-play

Rantakeisu, M. 2015. Android ohjelmistopino. Viitattu 4.7.2016.
http://www.mikarantakeisu.fi/internetsivut/content/mik%C3%A4-android

Ravencraft, E. 2014. Viitattu 3.9.2016.
http://lifehacker.com/i-want-to-write-android-apps-where-do-i-start-1643818268

Roman, N. 2010. Viitattu 5.6.2016
http://designbycode.tumblr.com/post/1079612795/why-dont-wysiwyg-android

Rouse, M. 2008. Android OS. Viitattu 23.4.2016.
http://searchenterpriselinux.techtarget.com/definition/Android

Rouse, M. 2016. Java. Viitattu 17.10.2016.
http://searchmicroservices.techtarget.com/definition/Java

Rouse, M. 2015. Localization. Viitattu 22.11.2016.
http://searchcio.techtarget.com/definition/localization

Rouse, M. 2014. XML (Extensible Markup Language). Viitattu 22.11.2016.
http://searchmicroservices.techtarget.com/definition/XML-Extensible-Markup-Language

Saukkonen, P. 2003. Tutkimusmenetelmät. Viitattu 7.2.2015.
http://www.mv.helsinki.fi/home/psaukkon/tutkielma/Tutkimusmenetelmat.html

SocialCompare. 2016. Android versions comparison. Viitattu 12.11.2016.
http://socialcompare.com/en/comparison/android-versions-comparison

SQLite. 2016. About SQLite. Viitattu 4.7.2016.
https://sqlite.org/about.html

SQLite. 2016. SQLite Android Bindings. Viitattu 15.9.2016.
https://www.sqlite.org/android/doc/trunk/www/index.wiki

Technopedia. 2016. Native mobile app. Viitattu 5.6.2016.
https://www.techopedia.com/definition/27568/native-mobile-app

Tilastokeskus. 2015. Kvantitatiivinen tutkimus. Viitattu 7.2.2015.
http://www.stat.fi/meta/kas/t_ktoiminta.html

Tilastokeskus. 2015. Lähdekritiikki. Viitattu 7.2.2015.
http://tilastokeskus.fi/virsta/thaku/02/02/

Vilpas, P. 2013. Kvantitatiivinen tutkimus. Viitattu 7.2.2016.
http://users.metropolia.fi/~pervil/kvantsu/Moniste.pdf

Virtuaali Ammattikorkeakoulu. 2007. Tutkimuksen validiteetti. Viitattu 7.2.2015.
http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/0709019/1193463890749/1193464185783/1194413809750/1194415367669.html

Kuviot

Taulukot