

MOBIILISOVELLUKSEN LAAJENTAMINEN ERI ALUSTOILLE

Case: Onser Oy

Harri Luoma

Opinnäytetyö
Helmikuu 2012

Tietojenkäsittelyn koulutusohjelma
Luonnontieteiden ala





Tekijä(t) LUOMA, Harri	Julkaisun laji Opinnäytetyö	Päivämäärä 09.02.2012
	Sivumäärä 47	Julkaisun kieli Suomi
	Luottamuksellisuus () saakka	Verkojulkaisulupa myönnetty (X)
Työn nimi MOBIILISOVELLUKSEN LAAJENTAMINEN ERI ALUSTOILLE Case: Onser Oy		
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma		
Työn ohjaaja(t) TUIKKA, Tommi		
Toimeksiantaja(t) Onser Oy		
<p>Mobiililaitteet ovat käynnistäneet murroksen ohjelmistokehityksessä ja jatkossa niiden välityksellä käytetään yhä enemmän erilaisia sovelluksia. Tämä tarkoittaa sitä, että erilaiset älypuhelimet ja tablet-tietokoneet valtaavat alaa ja haastavat perinteiset tietokoneet. iPhone-, Android- ja Windows Phone -käyttöjärjestelmiä käyttävät puhelimet ovat esimerkkejä siitä, kuinka älypuhelimet kehittyvät huimaa vauhtia ja kuinka niihin kehitetään jatkuvasti uusia ja mullistavia sovelluksia. Tämän vuoksi mobiilisovelluskehitys on noussut paljon viime vuosina ja luo haasteita kehittäjille.</p> <p>Tällä hetkellä on valmiina iOS/iPhone-sovellus, joka toimii Video On Demand -toiminnalla. Se on kehitetty alihankintana, jota valvoo Onser Oy. Se on pyritty luomaan Applen esittämien sovelluskehityssääntöjen pohjalta, jotta se olisi mahdollisimman helppo käyttää ja kuluttaisi energiaa ja muistia mahdollisimman vähän. Sovellus koostuu kolmesta ominaisuudesta, joita ovat streaming, säilytystila ja maksujärjestely Applen App Storen kautta.</p> <p>Opinnäytetyön tavoitteena oli tutkia ja laajentaa iPhone-sovelluksen ominaisuudet muille mobiilialustoille, kuten Android ja Windows Phone 7. Tärkeimmät tutkimuskohteet olivat säilytystilan kehittäminen, streaming-teknologia ja tietoturva, joka on merkittävin osa applikaatiota.</p> <p>Päätutkimusmenetelmänä oli informaation kerääminen kirjoista ja verkkoartikkeleista. Lisäksi projektin aikana käytettiin kokeellista tutkimista. Työasemaan asennettiin testattavaksi sekä Androidin että Windows Phonen kehitysympäristöt ja muutamia esimerkkikoodeja tehtiin. Yksi osa tutkimusta oli myös se, kuinka sovelluksen julkaisu tehtiin, mitä se vaatii ja kuinka paljon se maksaa kehittäjälle.</p> <p>Opinnäytetyön tulokset auttavat Onser Oy:tä kehittämään sovelluksen Androidille ja Windows Phonelle.</p>		
Avainsanat (asiasanat) Mobiilisovellus, mobiililaitte, Android, Windows Phone, iOS, iPhone		
Muut tiedot		



Author(s) LUOMA, Harri	Type of publication Bachelor's Thesis	Date 09.02.2012
	Pages 47	Language Finnish
	Confidential () Until	Permission for web publication (X)
Title EXPANDING MOBILE SOFTWARE TO DIFFERENT KINDS OF MOBILE PLATFORMS Case: Onser Oy		
Degree Programme Business Information Systems		
Tutor(s) TUIKKA, Tommi		
Assigned by Onser Oy		
<p>Because of the fast improvement of mobile devices, there has been a breakthrough among the software development and in the future there will be many applications used via them. This means that different kinds of smart phones and tablets are taking over the industry and challenging the traditional computers. Mobile phones that use iOS, Android and Windows Phone operating systems are examples how smart phones are developing very fast and many applications are made for those devices. Thus the mobile software development has a great deal in a few years, which also creates challenges for the developers.</p> <p>There is a Video On Demand mobile software for iPhone and it has been developed as a subcontracted project, controlled by Onser Oy. The aim was to create proper software that fills Apple's development standards and rules, which means it would be easy to use and it would consume energy and memory as little as possible. The software already consists of three elements which are streaming, storage for data (controlled by streaming-provider) and billing, which is organized through Apple App Store.</p> <p>The goal of this thesis was to research and expand the features of iPhone to other mobile platforms, such as Android and Windows Phone 7. The most important features that had to be examined were storage for data and how it is built, streaming technology and security, which is a very remarkable part of the application.</p> <p>The primary research method was to collect information from books and web articles. Additionally experimental research was used during the project. Development tools for Android and Windows Phone 7 were installed and tested and a few code examples were made. One part of the research was also to find out how the publishing was implemented and what does it requires and how much it costs for a developer.</p> <p>The results of this thesis help Onser Oy to develop the software to Android and Windows Phone</p>		
Keywords Mobile application, mobile platform, Android, Windows Phone, iOS, iPhone		
Miscellaneous		

SISÄLTÖ

1 JOHDANTO	3
2 TUTKIMUSASETELMA.....	4
2.1 Opinnäytetyön taustateoria, tavoitteet ja rajaukset	4
2.2 Tutkimus-/kehittämismenetelmät	5
2.3 Tutkimuskysymykset.....	5
2.4 Käsitteitä.....	5
2.5 Toteutus	8
3 ÄLYPUHELIN JA MOBIILIJÄRJESTELMÄT	9
3.1 Älypuhelimet	9
3.2 Käyttöjärjestelmät	10
3.3 Mobiilisovellukset.....	12
3.4 Sovelluksien jakelukanavat	14
4 SOVELLUKSEN LÄHTÖTILANNE	17
4.1 iOS-videovuokraamosovellus	17
4.2 Sovelluksen streaming.....	18
4.3 Sovelluksen säilytystila	19
4.4 Maksujärjestely eli In-App purchase.....	20
5 ANDROIDILLE KEHITTÄMINEN	22
5.1 Kehitysympäristö	22
5.2 Säilytystila	24
5.3 Julkaisun tietosuoja	25
5.4 Sovelluksen julkaisu Android Marketiin.....	26
5.5 Maksujärjestely Androidissa	28
6 WINDOWS PHONE -ALUSTALLE KEHITTÄMINEN	30
6.1 Kehitysympäristö	30
6.2 Säilytystila eli Isolated Storage	34
6.3 Tiedon suojaus.....	36
6.4 Sovelluksen julkaisu Windows Phone Marketplaceen.....	37
6.5 Maksujärjestely Windows Phonessa	39
7 JOHTOPÄÄTÖKSET JA POHDINTA	41
LÄHTEET.....	43

KUVIOT

KUVIO 1. Mobiilikäyttöjärjestelmien markkinaosuudet toukokuussa 2011	12
KUVIO 2. Kahden suurimman älypuhelimien kehittyneet (Android ja Apple) markkinaosuudet Yhdysvalloissa	16
KUVIO 3. Sovelluksen arkkitehtuurikaavio	17
KUVIO 4. File server eli tiedostopalvelimen arkkitehtuuri	18
KUVIO 5. Videovuokraussovelluksen elokuvien selausvälilehti	19
KUVIO 6. Elokuvan tiedot välilehti	20
KUVIO 7. In-App Store- malli	21
KUVIO 8. Eclipsen kehitysympäristö Androidin lisäosatyökalulla.....	22
KUVIO 9. Hello Android -sovellus avattuna Android-emulaattoriin	23
KUVIO 10. Android Marketin rekisteröitymissivu.....	27
KUVIO 11. Uuden In-app-tuotteen julkaiseminen/tallentaminen.....	28
KUVIO 12. Microsoft Visual Studio 2010 Express 2010 for Windows Phone	31
KUVIO 13. XAML-koodin pätkä ja Click Me -painike	32
KUVIO 14. C#-koodin pätkä ja Click Me -painike	32
KUVIO 15. MediaElement API	33
KUVIO 16. Isolated Storage ja sovellus	35
KUVIO 17. Tiedostojen tallennus Windows Phone -puhelimeen.....	36
KUVIO 18. Salatun tietokannan luominen	37
KUVIO 19. Windows Phone Marketplace-sertifikaatioprosessi.....	38
KUVIO 20. MoVend-käyttöliittymän lisäys sovelluksen lähdekoodiin.....	39

TAULUKOT

TAULUKKO 1. Mobiilisovelluksien yleisimmät tyypit.....	13
--	----

1 JOHDANTO

Mobiililaitteet ovat käynnistäneet murroksen ohjelmistokehityksessä ja jatkossa niiden välityksellä käytetään yhä enemmän erilaisia sovelluksia. Tämä tarkoittaa sitä, että erilaiset älypuhelimet ja tablet-tietokoneet valtaavat alaa ja haastavat perinteiset tietokoneet ominaisuuksillaan ja käytettävyydellään. Nykyään sovelluksien käyttäminen on lähestulkoon ajasta ja paikasta riippumatonta.

iPhone- ja Android-käyttöjärjestelmää käyttävät puhelimet ovat esimerkkejä siitä, kuinka älypuhelimet kehittyvät huimaa vauhtia ja kuinka niihin kehitetään jatkuvasti uusia ja mullistavia sovelluksia. Täten mobiililaitteiden ohjelmistokehitys on noussut merkittäväksi osa-alueeksi. Eri alustoilla valmistettavat ohjelmistotekniikat eroavat merkittävästi toisistaan, joten tämä luo haasteita tehdä ohjelma usealle eri järjestelmälle.

Tämän opinnäytetyön aiheena on tutkia, kuinka Video On Demand -sovellus on toteutettavissa mm. Android- ja Windows-mobiilikäyttöjärjestelmiin, eli kyseessä on videovuokrausohjelma. Toimeksiantaja on saanut valmiiksi iOS/iPhone-alustalle tällaisen ohjelman, mutta tarkoitus olisi laajentaa sama sovellus lisäksi edellä mainituille alustoille.

Uskon tästä aiheesta olevan hyötyä ja antavan hyvää kokemusta tulevaisuuden tekniikasta. Opinnäytetyön tuloksena ei ole uuden sovelluksen ohjelmoiminen vaan tutkimus siitä, kuinka sellainen on tehtävissä ja mitä näkökulmia on otettava huomioon. Aihe on kiinnostava, ja kun projektin yksityiskohtia ja sitä, mitä on jo saatu aikaiseksi, käytiin lävitse, luotto siihen, että pystyn viemään tämän läpi, kasvoi huomattavasti.

2 TUTKIMUSASETELMA

2.1 Opinnäytetyön taustateoria, tavoitteet ja rajaukset

Tausta

Tutkimustyön tuloksena on tarkoitus saada aikaiseksi kehitysraportti, joka vastaa kysymyksiin annettujen ominaisuuksien toiminnallisuuden toteutumisesta. Raportin perusteella voidaan luoda videovuokrausohjelmisto Androidille, Windows-mobiilikäyttöjärjestelmälle.

Näkökulma ja perustelut

Onser Oy:n tarkoituksena on lanseerata jossain vaiheessa markkinoille älypuhelimissa toimiva videovuokraamo. Alustoina toimivat ensisijaisesti iPhone-, Android- ja Windows Phone -puhelimet.

Kehitystyö koskee näiden ominaisuuksien siirrettävyyttä muille alustoille ja sitä, kuinka mahdollista on rakentaa samanlaisia toiminnallisuuksia. Toinen merkittävä kehittämisen osa-alue on tietoturva ja sen kokoaminen.

Tutkittavan alueen rajaus

Tutkimus rajataan niin tutkimuksen, ohjelmistosuunnittelun, tarvittavien tekniikoiden kuin tarvittavan laitteistonkin osalta vain siihen laajuuteen, joka on tarpeen kehitettävän mobiilisovelluksen luomiseksi sitä käyttävien laitteiden kannalta.

Toimeksiantaja

Yritys: Onser Oy. Y-tunnus: 2322376-8. Työllistää 3 henkilöä.

Edustaja: Jouni Laveri, Hiihtäjäntie 4 C 2, 00810 Helsinki

puh. 050 545 7804 ja email jouni.laveri@onser.fi

Tehtävä ja tavoite

Kehittämistutkimuksen toteutuessa saadaan tietoa siitä, miten ja millä tekniikoilla haluttu mobiilisovellus on toteutettavissa eri alustoille. Projektin lopputuloksena syntyy kehitysraportti, joka vastaa kysymyksiin annettujen ominaisuuksien

toiminnallisuuden toteutumisesta.

2.2 Tutkimus-/kehittämismenetelmät

Teoriaosuus opinnäytetyöhön kerätään kirjallisista lähteistä, ja tutkimusmenetelmänä käytetään laadullista kehittämistutkimusta, jonka avulla luodaan kysymys, johon etsitään vastausta. Oleellisena tutkimuskysymyksenä on se, mitä asioita on otettava huomioon videovuokraamopalvelua kehitettäessä Android- ja Windows-alustoille ja lisäksi Video On Demand -toiminnalle. Toimeksiantajan alustavien toiveiden mukaan kartoitetaan osa-alueita, joiden kautta pyritään saamaan tuloksia mobiilisovelluksen toteuttamiseksi.

Toimintatapojen ja -mallien läpikäyminen toimeksiantajan kanssa luo hyvät edellytykset projektin suorittamiselle. Tässä työssä ilmenevät kehittämisprojektille tyypilliset piirteet, joita ovat, ongelmalähtöisyys, tavoitteellisuus, osallistuvuus, suunnitelmallisuus, kertaluonteisuus ja tietyt rajalliset resurssit.

Tavoitteena on tuottaa hyödyllistä tietoa yrityksen käyttöön suunniteltaessa toimivaa ohjelmistoa eri mobiilialustoille. Lisäksi kehitysraportin tuloksena on ohjelmisto, joka on kehitettävissä eteenpäin tulevaisuuden tarpeisiin.

2.3 Tutkimuskysymykset

Tutkimusongelma:

- Kuinka toteutetaan eri alustoilla (Android, Windows Phone) toimiva videovuokraamo?

Alaongelmat:

- Miten taataan tietoturva, joka on tärkeä osa sovellusta?
- Minkälainen salaus on suositeltava?
- Kuinka streamaus eli suoratoisto voidaan toteuttaa?

2.4 Käsitteitä

Video On Demand (VOD)

Ladattava bittimuotoinen videotiedosto, jota levitetään mm. Internetin välityksellä.

Streaming

Suomenkielinen termi on suoratoisto. Yhteisnimitys jakelutavalle, jossa tiedostosta ei jää käyttäjälle kopiota, kun hän käyttää sitä.

iTunes / iTunes Store

iTunes on Applen julkaisema maksuton musiikkisoitinohjelma, joka on saatavilla sekä Mac OS X - että Microsoft Windows –käyttöjärjestelmille, ja iTunes Store on huhtikuussa 2003 julkaistu verkkomusiikkikauppa, jossa myydään nykyään myös elokuvia ja TV-sarjoja

HLS (HTTP Live Streaming)

Applen kehittämä protokolla, joka jakaa suoratoistossa olevan tiedoston pieniin HTTP-pohjaisiin lataustiedostoihin.

WLAN

Wireless Local Area Network eli langaton lähiverkkotekniikka.

API

Application programming interface eli ohjelmointirajapinta, jonka avulla eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään.

SDK

Software Development Kit eli kehitystyökalu, jonka avulla luodaan ohjelmistoja eri alustoille.

ADT

Android Development Tool eli lisäosa Androidin kehitysympäristöön.

Linux kernel

Ydin eli kerneli on tietokoneen käyttöjärjestelmän alin osa, joka mahdollistaa kaikkien muiden tietokoneen ohjelmien toiminnan.

Java

Sun Microsystemsin kehittämä laaja teknologiaperhe ja ohjelmistoalusta, johon kuuluu muun muassa laitteistoriippumaton oliopohjainen ohjelmointikieli sekä ajoaikainen ympäristö virtuaalikoneineen ja luokkakirjastoineen.

Dalvik

Googlen Androidin virtuaalikäyttöjärjestelmä

.apk-tiedosto

Android application package file -tiedostomuoto on Android-käyttöjärjestelmälle tarkoitettu sovellusten asennus- ja jakelupaketti.

IPC

Inter-process communication on joukko menetelmiä, joissa vaihdetaan tietoa monissa tietoketjuissa yhdessä tai useammassa prosessissa.

C# -kieli

Englanniksi lausuttuna C sharp on Microsoft-yhtiön .NET-konseptia varten kehittämä ohjelmointikieli.

.NET Framework

Englanniksi lausuttuna dot net. Ohjelmistokehys, jota ajetaan pääosin Microsoft Windows -käyttöjärjestelmässä.

Visual Basic

Microsoftin kehittämä BASIC-sukuinen yleiskäyttöinen ohjelmointikieli

RIA

Rich Internet Application on web-applikaatio, jolla on monia samoja piirteitä verrattuna työpöydän ohjelmiin.

HTML

Hypertext Markup Language on avoimesti standardoitu kuvauskieli, jolla voidaan kuvata hyperlinkkejä sisältävää tekstiä eli hypertekstiä.

I/O –operaatiot

Input/output-operaatiot viittaavat tietokoneen ja käyttäjän tai toisen järjestelmän väliseen kommunikointiin.

SQL

Structured Query Language on IBM:n kehittämä standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä.

LINQ

Language Integrated Query on Microsoft .NET Framework -komponentti, joka mahdollistaa datakyselyt .NET-ohjelmointikielessä.

2.5 Toteutus

Tutkimus toteutetaan useassa vaiheessa. Ensimmäisessä vaiheessa sovitaan projektin yksityiskohdista eli siitä, mitä asioita on tutkittava. Lisäksi käydään lävitse, mitä ominaisuuksia on jo luotu ja mitä asioita halutaan tutkittavan.

Toisessa vaiheessa saatujen tietojen perusteella selvitetään erilaisia alustoja ja niiden toiminnallisuuksia. Lisäksi tutkitaan miten eri alustojen toiminnot eroavat iOS/iPhone-alustan toiminnoista, kuten esimerkiksi miten sama ominaisuus on toteutettu toisella alustalla.

Kolmannessa vaiheessa tutkitaan selvitettyjen järjestelmien tietoturvakäytänteitä. Tarkoituksena on saada selville, miten ohjelmisto rakennetaan ja kuinka suojataan yhteys ja puhelimen käyttö.

Viimeisessä vaiheessa saatujen tietojen perusteella kirjoitetaan kehitysraportti, jonka perusteella voidaan luoda videovuokrausohjelma iOS/iPhone-alustan lisäksi Android- ja Windows-järjestelmille. Projektiin tutustutaan aiheeseen liittyvän kirjallisuuden sekä Internetin suomien tietovarastojen ja lisäksi kehitysympäristöjen avulla.

3 ÄLYPUHELIN JA MOBIILIJÄRJESTELMÄT

Tässä luvussa käydään läpi lyhyesti mitä ovat älypuhelimet ja valaistaan hiukan raportissa esille tulevia yleisimpiä mobiilijärjestelmiä.

Nykyaikaisille mobiilijärjestelmille on ominaista laajennettavuus ja käyttäjän tarpeisiin mukautuminen. Käyttäjille pyritään tarjoamaan mahdollisuus parantaa mobiililaitteen toiminnallisuutta siten, että järjestelmä palvelee jokaista käyttäjää aiempaa yksilöllisemmin. Tämä on myös yksi tärkeä ero sulautettuihin järjestelmiin, kuten työasemiin, verrattuna. Sulautetut järjestelmät toteutetaan yleensä yhtä tarkoitusta varten, kun taas mobiililaitteet ovat laajennettavissa. (Mobiiliohjelmointi 2004, 3, 8.)

3.1 Älypuhelimet

Älypuhelimien määritelmä on karkeasti ottaen seuraavanlainen. Se on laite, jolla on kehittyneemmät ominaisuudet verrattuna tavalliseen matkapuhelimeen.

Älypuhelimessa on käyttöjärjestelmä, joka tukee tietynlaista käyttöliittymää, ja johon sovelluskehittäjät voivat suunnitella erilaisia ohjelmia. Yleisesti ottaen älypuhelimissa ovat isommat näytöt ja tehokkaammat prosessorit kuin matkapuhelimissa. (Phone Scoop n.d.)

Ensimmäinen älypuhelimeksi kutsuttava laite IBM Simon lanseerattiin vuonna 1994 sen tullessa myyntiin amerikkalaisen teleoperaattori BellSouthin toimesta. Hintaa tällä laitteella oli silloin 899 dollaria. IBM Simon sisälsi juuri ne ominaisuudet, joita sanakirjan mukaan älypuhelin tarkoittaa. Puhelimessa oli mm. kalenteri, laskin, faksi, sähköposti ja kosketusnäyttö. IBM Simon oli kuitenkin kooltaan iso ja painava ja kärsi paljon teknisistä vioista, minkä vuoksi se vedettiin pois markkinoilta jo seuraavana vuonna. (RetroCom n.d.)

Nykyisin älypuhelimia on markkinoilla monia ja ne valtaavat alaa tavallisten matkapuhelimien tieltä. Suurin murros tapahtui vuonna 2007 Applen julkistaessa iPhone-puhelimen, joka oli täysin kosketuksella toimiva laite eikä sisältänyt fyysisiä näppäimiä. Tämän jälkeen ne ovat levinneet yhä enenevässä määrin. iPhoneen lisäksi Samsung, HTC ja Nokia valmistavat älypuhelimia. Älypuhelimille on kehitetty monia eri käyttöjärjestelmiä, joista tunnetuimmat ovat Applen iOS, Googlen Android,

Nokian Symbian ja Microsoftin Windows Phone. Edellämainituista iOS ja Android ovat vallanneet markkinoita ja vieneet esimerkiksi Nokian osuutta matkapuhelinliiketoiminnassa.

3.2 Käyttöjärjestelmät

Mobiililaitteiden ohjelmistot toteutetaan useimmiten mobiiliohjelmistoalustojen avulla. Valmistajia on useita, ja alustojen tekniset ominaisuudet poikkeavat toisistaan joissakin suhteissa. (Mobiiliohjelmointi 2004, 12.) Seuraavassa esitellään tunnetuimmat käyttöjärjestelmät, joihin myös opinnäytetyön alustakartoitus keskittyy. Kuviossa 1 on havainnollistettu, kuinka järjestelmien markkinaosuudet jakautuivat toukokuussa 2011.

iOS/iPhone

iOS eli aiemmin iPhone OS on Applen kehittämä käyttöjärjestelmä, joka perustuu Darwin BSD -käyttöjärjestelmään ja löyhästi joihinkin Mac OS X -jakelun komponentteihin. Ensimmäinen versio iOS-järjestelmästä lanseerattiin ja julkaistiin kesäkuussa 2007. Järjestelmä on suljettu, eikä siihen voi asentaa kolmannen osapuolen sovelluksia. Kuitenkin vuonna 2008 avatun App Storen yhteydessä julkaistiin uusi versio, joka mahdollisti ulkopuolistien sovelluksien kehittämisen tälle alustalle. Nykyään iOS on levinnyt muihinkin Applen tuotteisiin, kuten iPad-tablettiin ja iPod Touch -mediasoittimeen. (iPad & iPhone 2011, 4.)

Android

Android on Googlen julkaisema avoimen lähdekoodin käyttöjärjestelmä älypuhelimille ja mobiililaitteille. Alun perin Androidia kehitti Android Inc., jonka Google sitten osti itselleen ja nykyään kehittämisestä vastaa Open Handset Alliance. (OHA). OHA perustettiin vuonna 2007 ja samalla Android julkistettiin. OHA on 84 teknologia- ja mobiiliyrityksen muodostama ”liittouma”, joka kehittää Androidia ja luo avoimia standardeja mobiililaitteille. Androidille kehittäminen ja sen käyttäminen on ilmaista. Tämän vuoksi sovelluksia on todella paljon, ja niitä on ladattavissa Android Marketin kautta. (Mikä on Android? n.d.) Matkapuhelinvalmistajista muun muassa LG, HTC ja Samsung kehittävät älypuhelimia Androidille. Jokaisella valmistajalla on Androidille oma käyttöliittymänsä, jotta ne eroaisivat toisistaan.

Ensimmäinen Androidia käyttävä puhelin oli HTC Dream. Android on noussut lyhyessä ajassa suosituksi käyttöjärjestelmäksi ja onkin kivunnut älypuhelinmarkkinoiden johtajaksi vuoden 2010 viimeisellä kvartaalilla.

Symbian OS

Symbian OS on Symbian Ltd:n ohjelmistoalusta, jonka historia juontaa juurensa Psion Software -ohjelmistoyritykseen. Yritys kehitti vuonna 1994 EPOC32-järjestelmän eli 32-bittisen käyttöjärjestelmän, ja siitä julkaistiin vuonna 1997 ensimmäinen virallinen versio. Kesäkuussa 1998 Psion Softwaresta tuli yhteisyritys nimeltä Symbian, kun matkapuhelinvalmistajat Nokia, Panasonic, Motorola ja SonyEricsson perustivat sen. (Mobiiliohjelmointi 2004, 12.) Tämän jälkeen ensimmäiset Symbiania käyttävät matkapuhelimet tulivat markkinoille vuosituhannen alkupuolella ja tämän avulla Nokia saavutti markkinajohtajuuden.

Windows Phone

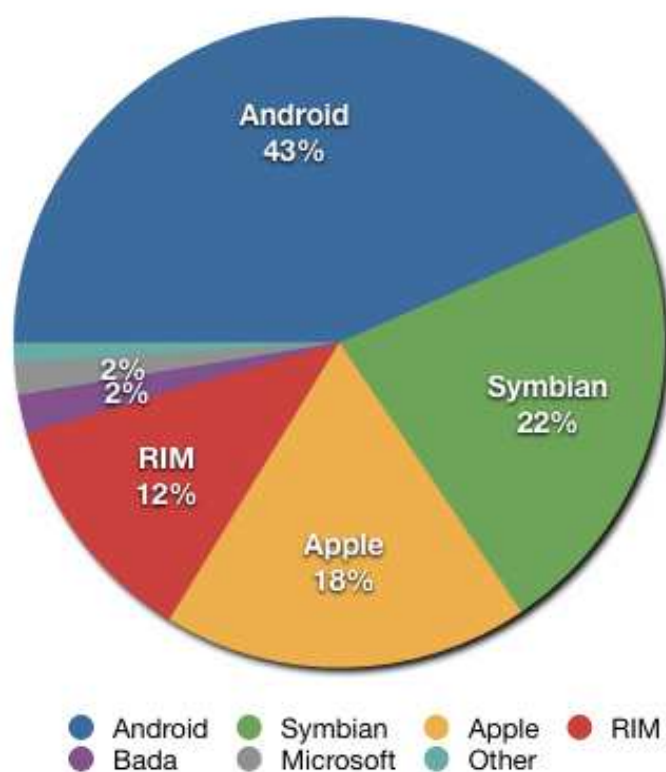
Windows Phone on Microsoftin kehittämä mobiilikäyttöjärjestelmä. Sen taival alkoi tavallaan vuonna 2004 koodinimellä ”Photon”, mutta kehitystyö eteni hitaasti, kunnes se lopulta keskeytettiin. Vuonna 2008 Microsoft uudelleenorganisoi Windows Mobile-ryhmän ja alkoi työstää uutta järjestelmää. Vuonna 2009 tuotos oli tarkoitus julkaista kokonaan uudenaikaisena Windows Phone -käyttöjärjestelmänä, mutta useiden viivästysten johdosta Microsoft julkaisi väliaikaisena versiona Windows Mobile 6.5:n. Barcelonassa toukokuussa vuonna 2010 järjestetyssä Mobile World Congressissa Microsoft paljasti muutamia yksityiskohtia tulevasta Windows Phone-mobiilikäyttöjärjestelmästä. Samassa kuussa se julkisti lehdistötiedotteen jossa listattiin yritykset, jotka auttaisivat ja operoisivat Windows Phone -järjestelmää. Vuonna 2011 matkapuhelinvalmistaja Nokia solmi yhteistyösopimuksen Microsoftin kanssa ja ryhtyi valmistamaan älypuhelimia alustanaan Windows Phone. Ensimmäiset Nokia Windows -puhelimet julkistettiin lokakuussa 2011. (Windows Phone n.d.)

Muita käyttöjärjestelmiä

Pienempiä mobiilialustoja on muun muassa Samsungin valmistama Bada, jonka se julkaisi 10. marraskuuta 2009. Ensimmäinen Bada-puhelin Wave S8500 julkistettiin vuonna 2010 samaisessa Mobile World Congress –tapahtumassa, jossa Microsoft

ilmoitti omista suunnitelmistaan. Eri tahot miettivät Badan tarpeellisuutta, koska markkinoilla on monia samanlaisia alustoja ja lisäksi Samsung valmistaa puhelimia Androidille. Tästä huolimatta Samsung aikoo kehittää Bada-puhelinten valikoimaa jatkossakin.

Kanadalaisen Research In Motionin (RIM) kehittämä BlackBerry on langaton mobiililaitte, joka on erityisen suosittu Pohjois-Amerikassa. Yksi suosion takeista on hyvä tietoturva. Ensimmäiset puhelimet tulivat markkinoille vuonna 1997, mutta ne toimivat ainoastaan hakulaitteina, joilla pystyi lähettämään tekstiviestejä. Vuonna 2002 RIM julkaisi uudemman mallin, kun ominaisuuksien joukkoon lisättiin tärkeä puhelukyky. Nykyään RIM:llä on monia älypuhelimia.



KUVIO 1. Mobiilikäyttöjärjestelmien markkinaosuudet toukokuussa 2011
(Gartner 2011)

3.3 Mobiilisovellukset

Mobiilisovellus on älypuhelimissa ja muissa mobiililaitteissa ajettava ohjelma. Monet mobiilisovellukset, kuten SMS/MMS-viestimet, -selaimet ja -soittimet ovat valmiiksi

asennettuina matkapuhelimissa, kun toiset ohjelmat ovat taas ladattavissa ilmaiseksi tai maksua vastaan. Käyttäjä voi siis ladata ohjelman langattomasti ja asentaa sen. Vaihtoehtoisesti ne voi saada myös ladattuina ja asennettuina mobiilioperaattorin palvelusta. Älypuhelimista puhuttaessa käyttöjärjestelmän saa omasta sovelluskaupasta. Huolimatta siitä, kuinka sovelluksia on saatavilla loppukäyttäjille, mobiilisovelluksia on todella paljon, ja niiden markkinat ovat jatkuvassa kasvussa, koska sovelluskehittäjien ja julkaisijoiden määrä nousee koko ajan. (Mobile Applications n.d.)

Kuviossa 2 on taulukoitu mobiilisovelluksien yleisimpiä tyyppisiä eli kommunikaatio-sovelluksia, pelejä, multimediaa ja tuottavuutta.

TAULUKKO 1. Mobiilisovelluksien yleisimmät tyypit

Table 2: Mobile Application Types	
Communications:	
<ul style="list-style-type: none"> - E-mail Clients - IM Clients - Mobile Web and Internet Browsers - News/Information Clients - On-Device Portals (Java Portals) - Social Network Clients 	
Games	
<ul style="list-style-type: none"> - Puzzle/Strategy (e.g., Tetris, Sudoku, Mah-jong, Chess, Board Games) - Cards/Casino (e.g., Solitaire, Blackjack, Roulette, Poker) - Action/Adventure (e.g., Doom, Pirates of the Caribbean, Role-Playing Games) - Sports (e.g., Football, Soccer, Tennis, Basketball, Racing, Boxing, Skiing) - Leisure Sports (e.g., Bowling, Pool, Darts, Fishing, Air Hockey) 	
Multimedia:	
<ul style="list-style-type: none"> - Graphics/Image Viewers - Presentation Viewers - Video Players - Audio Players - Streaming Players (Audio/Video) 	
Productivity:	
<ul style="list-style-type: none"> - Calendars - Calculators - Diary - Notepad/Memo/Word Processors - Spreadsheets - Directory Services (e.g., yellow pages) - Banking/Finance 	

(Mobile Application n.d.)

3.4 Sovelluksien jakelukanavat

Älypuhelimien kehittyessä ja kasvattaessa suosiotaan verrattuna tavallisiin matkapuhelimiin on myös mobiilisovelluspalveluiden suosio noussut huomattavasti. Applen julkistaessa ensimmäisenä kyseisen palvelun tapahtui tällä saralla iso murros ja mobiilisovelluksia kehitetään yhä enenevässä määrin. Internetissä mainostetaan paljon eri verkkosivuilla sitä, että nyt on kehitetty myös mobiiliversio palvelusta. Lisäksi televisiossa on nykyisin runsaasti mainontaa tähän liittyen. Esimerkkinä toimii Elisa Viihde, jonka mobiilisovellusta mainostetaan eri alustoille saatavaksi. Seuraavassa on esitelty lyhyesti yleisimmät mobiilisovelluksien jakelukanavat järjestelmittäin.

App Store

App Store on Applen kehittämä palvelu, joka julkaistiin heinäkuun 10. päivä 2008 samalla kun iTunes-ohjelmaan tuli päivitys. Seuraavana päivänä iPhone 3G tuli markkinoille ja se oli varustettu iOS 2.0.1 -järjestelmällä, joka tuki myös App Storea. Palvelussa käyttäjät voivat selailta ja ladata sovelluksia iTunes Store -kaupasta, joka on kehitetty iOS SDK:lla (Software Development Kit) eli kehitystyökalulla ja julkaistu/hyväksytty Applen toimesta. Ohjelmasta riippuen se voi olla joko ilmainen tai maksullinen. Apple perii 30 %:n palkkiota sovelluksen myynnistä, ja loput 70 % jäävät kehittäjälle. Applen App Store oli ensimmäisiä mobiilisovelluskauppoja, ja sen hyvästä menestyksestä johtuen muut kilpailijat käynnistivät vastaavanlaisia palveluita. Sana ”app store” alkoi myös tarkoittaa yleisesti palvelua mobiililaitteille. Apple haki tavaramerkkiä termille App Store vuonna 2008, ja se hyväksyttiin vuonna 2011. Tammikuun 22. päivä 2011 oli 10 miljardin ladatun sovelluksen rajapyykki rikottu. (Apple 2012.)

Android Market

Android Market on Googlen kehittämä sovelluksien jakelu-/ostoskanava Android-älypuhelimille, ja se on App Storea vastaava palvelu. Android Market julkistettiin elokuussa 2008 ja avattiin käyttäjille muutama kuukausi myöhemmin. Seuraavan vuoden keväällä 2009 mahdollistettiin tuki maksullisille sovelluksille. Jokaisessa Android-puhelimessa ohjelma on esiasennettuna, ja käyttäjä tarvitsee luonnollisesti internetyhteyden sovellusten lataamiseksi. Sovellukset voivat olla ilmaisia tai

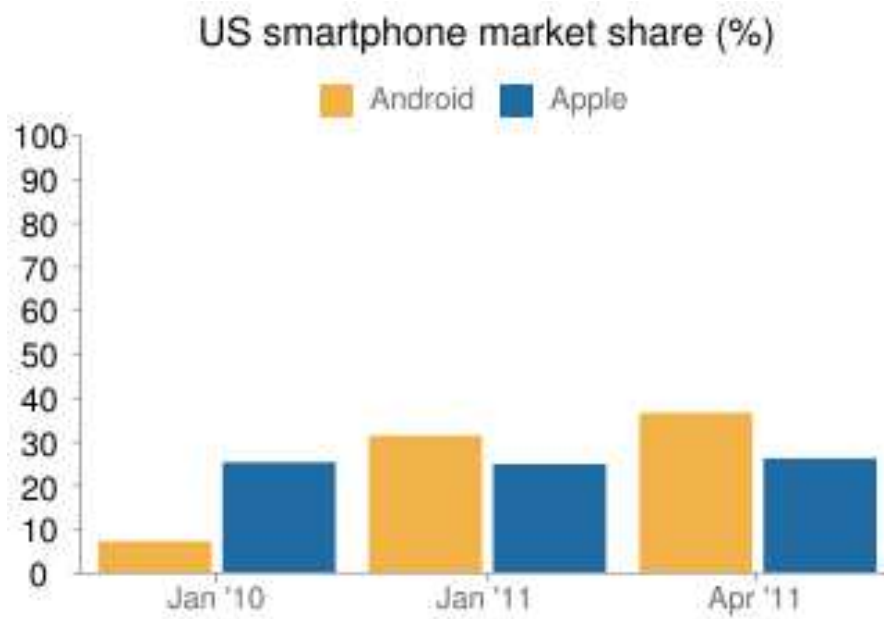
maksullisia. Ostaakseen sovelluksen Android Marketista käyttäjällä on oltava Googlen käyttäjätunnus ja luottokortti. (Android Suomi n.d.) Maksupolitiikka hoituu Google Checkoutin avulla. Vuonna 2011 Google esitteli selainpohjaisen palvelun, joka mahdollistaa Android Marketin sovelluksien lataamiseen tietokoneen kautta. Käyttäjällä on oltava Android-puhelin kytkettynä koneeseen ladatakseen ja asentaakseen ohjelman. (Engadget 2011.) Helmikuussa 2011 verkkokauppa Amazon avasi mobiilisovelluskauppa Amazon Appstoren Googlen Androidille, mutta se on tällä hetkellä saatavilla Yhdysvaltain käyttäjille.

Windows Phone Marketplace

Windows Phone Marketplace on Microsoftin kehittämä palvelu luonnollisesti Windows Phone 7-alustalle. Palvelu on toimintaperiaatteiltaan samanlainen kuin Apple App Store ja Android Market eli käyttäjät voivat selata ja ladata sovelluksia, joita ovat kehittäneet kolmannet osapuolet. Phone Marketplace käynnistettiin Windows Phone 7:n yhteydessä lokakuussa 2010 useimmissa maissa, eli sovelluskauppa on nuorin kaikista palveluista. Tammikuussa 2011 Marketplace sisälsi jo noin 55 000 tuhatta sovellusta. Kun Mango (Windows Phone 7.5) tuotiin markkinoille, verkkosivusto Marketplace julkistettiin. Marketplace tarjoaa sovelluksille kokeiluversion, jossa käyttäjällä on vaihtoehto ladata koekäyttö tai demo maksullisesta ohjelmasta. Jos käyttäjällä on Xbox Live -tunnukset, niin ne ovat myös yhdistettävissä sovelluskauppaan. (Mr. Windows Phone Finland 2012.)

Muita

Kahden edellä mainitun mobiilisovelluskaupan lisäksi on olemassa RIMin BlackBerry App World. Se on täysin samanlainen palvelu kuin App Store ja Android Market. Sovellusympäristö tarjoaa BlackBerryn käyttäjille mahdollisuuden selailuun, latailuun ja kolmansien osapuolien ohjelmiin. Palvelu julkaistiin huhtikuussa 2009, ja sillä on suurin palkkio yksittäisestä sovelluksesta verrattuna toisiin kauppoihin.



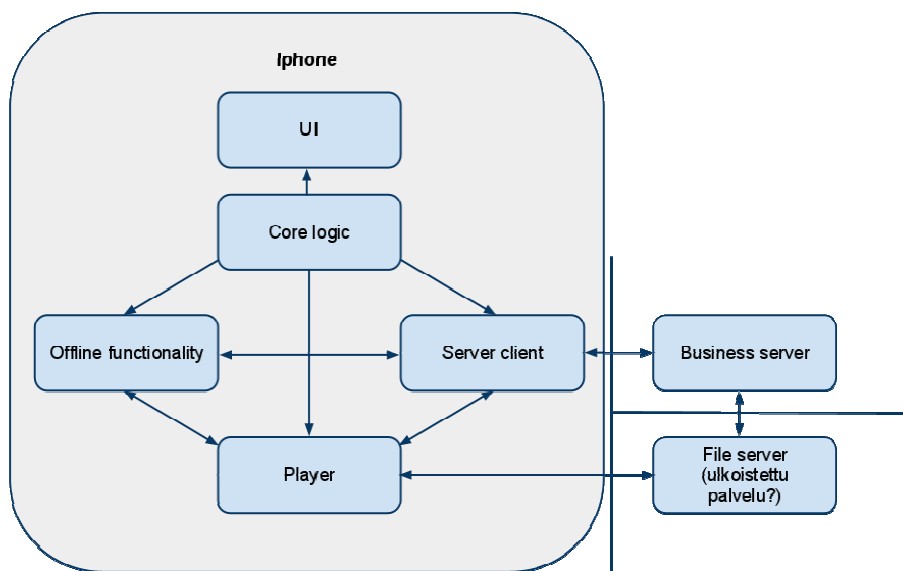
KUVIO 2. Kahden suurimman älypuhelimien kehittyneet (Android ja Apple) markkinaosuudet Yhdysvalloissa

(Geek With Laptop 2011)

4 SOVELLUKSEN LÄHTÖTILANNE

4.1 iOS-videovuokraamosovellus

Tällä hetkellä on valmiina iOS/iPhone-sovellus, joka toimii Video On Demand-toiminnalla. Se on kehitetty alihankintana, jota valvoo Onser Oy. Se on pyritty luomaan Applen esittämien sovelluskehityssääntöjen pohjalta, jotta se olisi mahdollisimman helppo käyttää. Huomioon on tämän vuoksi myös otettu se, että ohjelma veisi mahdollisimman vähän virtaa ja muistia. Kuviossa 3 on iPhone-sovelluksen arkkitehtuurikaavio, josta käy ilmi, kuinka kaikki elementit ovat toisiinsa yhteydessä. (Onser Oy tiedonanto 2011.)



KUVIO 3. Sovelluksen arkkitehtuurikaavio

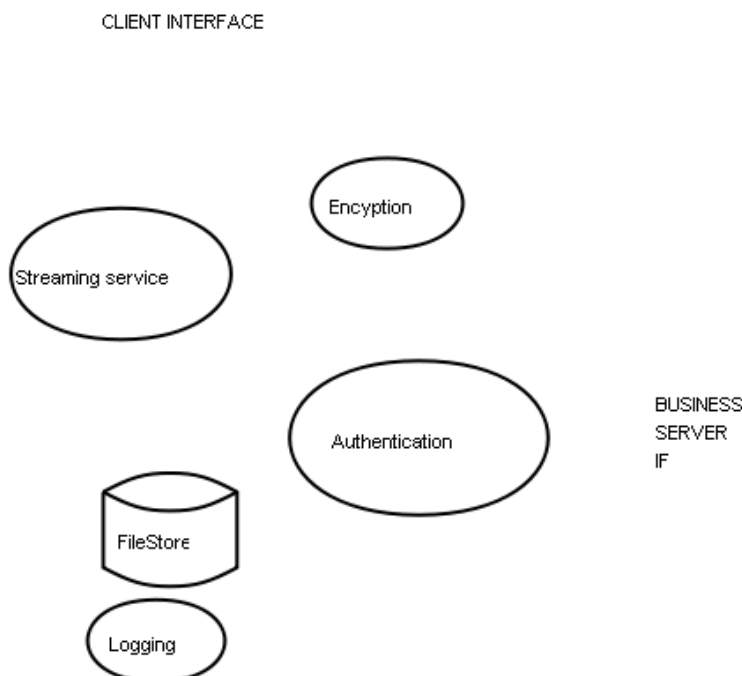
(Onser Oy tiedonanto 2011)

Sovelluksen ominaisuudet ovat tällä hetkellä seuraavat:

- Streaming-tekniikka on ulkoistettu ja kehitetty alihankkijan kanssa.
- Elokuviin säilytystila on myös ulkoistettu ja sitä hallinnoi streaming-toimittaja.
- Maksujärjestely tapahtuu Applen App Storen välityksellä.

4.2 Sovelluksen streaming

Streaming-tekniikka on ulkoistettu ja sitä kehitetään yhteistyössä alihankkijan kanssa. Streaming-tekniikalla voidaan jakaa kokonaisia video- ja äänitiedostoja. Tiedostoa jaettaessa se hinnoitellaan koon ja käyttötavan mukaan. Apple vaatii yli kymmenen minuuttia tai yli 5 megatavua sisältävien elokuvien suoratoistoa HLS-tekniikalla. Lyhenne tulee termistä HTTP Live Streaming, joka on Applen kehittämä protokolla, joka jakaa suoratoistossa olevan tiedoston pieniin HTTP-pohjaisiin lataustiedostoihin. Alla olevassa kuviossa on havainnollistettu, kuinka tiedostopalvelimen streaming on toteutettu. (Onser Oy tiedonanto 2011.)



KUVIO 4. File server eli tiedostopalvelimen arkkitehtuuri
(Onser Oy tiedonanto 2011)

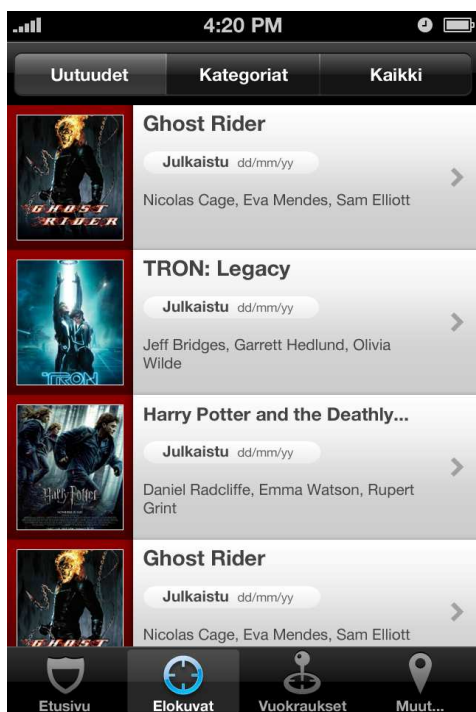
- Client interface: on sovelluksen käyttöliittymä.
- Encryption: moduuli on vastuussa tiedonsiirron salauksesta suoratoiston aikana.
- Streaming service: palvelu on vastuussa suoratoistosta, kun tiedosto siirtyy kovalevyiltä asiakasohjelman pyytäessä sitä.
- Authentication: moduuli on vastuussa siitä että käyttäjällä on oikeus katsoa pyydetty elokuva, ja se toimii lisäksi yhteytenä tehtäväpalvelimen ohjelmointirajapintaan (API).

- Filestore: tiedostojen säilytys ja Logging: lokitiedosto

4.3 Sovelluksen säilytystila

Sovelluksella on kaksi eri säilytystilaa. Elokuviin eli tiedostojen tiedot on ulkoistettu ja tätä tilaa hallinnoi streaming-toimittaja, koska on teknisesti helpompaa ja vähentää yhteensopivuusongelmia, kun säilytystila ja suoratoistopalvelu ovat samalla yhtiöllä. Toinen säilytystila tarkoittaa puhelimen omaa muistia, johon elokuvan voi ladata ja jossa se säilyy 24 tuntia. Tallennustila on mukautuva eli se muotoutuu tiedoston koon mukaisesti kantaan. Muistista sen voi katsoa haluamanaan ajankohtana. Tämä ominaisuus on luotu sen vuoksi, että on mahdollista käyttää WLAN-verkkoa tiedoston lataamiseen. Toteutustapa on varmempi kuin aiemmin mainittu HLS, joka on alttiimpi verkon kuormittumiselle ja muille teknisille seikoille. (Onser Oy tiedonanto 2011.)

Kuvioissa 5 ja 6 ovat kuvankaappaukset iPhoneen videovuokrausohjelmasta. Kuviossa 5 on otos Elokuvat-välilehdestä muutamine elokuvaesimerkkeineen. Kuviossa 6 on valitun elokuvan lisätietoja, ja kuten kuvasta näkee, ohjelman avulla voi katsoa muun muassa elokuvan trailerit ja lukea tiivistetyn kuvauksen juonesta.



KUVIO 5. Videovuokraussovelluksen elokuvien selausvälilehti
(Onser Oy tiedonanto 2011)



KUVIO 6. Elokuvan tiedot välilehti

(Onser Oy tiedonanto 2011)

4.4 Maksujärjestely eli In-App purchase

In-App purchase eli vapaasti suomennettuna applikaation sisäinen osto on sovellus, jonka avulla myydään mobiilisovelluksen digitaalista sisältöä. Tämä osanen on liitettävissä sekä ilmaiseen että maksulliseen versioon. In-App purchase -sovelluksen avulla kauppa upotetaan suoraan applikaatioon. Liitännäisyys iOS-ohjelmaan luodaan käyttämällä Store Kit -kehystä, joka ottaa yhteyden App Storeen applikaation puolesta taatakseen turvallisen maksuprosessin käyttäjän suunnalta. Ostaminen voidaan tehdä vain sovelluksen sisällä ja ne vahvistetaan iTunes store -tilin avulla. (In-App Purchase Programmig Guide 2011.)

Valmista sovellusta rekisteröitäessä App Storeen on sille valittava ostopohja neljän vaihtoehdon joukosta. Malleja ovat consumable, non-consumable, auto-renewable subscription ja subscription. Consumable tarkoittaa sitä, että on käyttöoikeus vain yhteen kertaan. Non-consumable on myös ostettavissa kerran, mutta sitä voidaan käyttää uudestaan ja millä tahansa laitteella käyttäjän iTunes-tilin mukaisesti. Auto-renewable subscription on malli, joka on tarkoitettu kaikille laitteille, ja nimensä mukaisesti se uudistaa tilauksen automaattisesti, ellei sitä keskeytetä. Subscription on malleista kaikkein vanhin käytössä oleva ja tarkoittaa sitä, että tilaus tehdään tietyksi

ajaksi. (Onser Oy tiedonanto 2011.)

Videovuokraussovelluksen iPhone-versiossa on mietitty consumable-mallin käyttöä eli käyttöoikeutta vain yhteen kertaan. Ongelmana kuitenkin tässä on se, että jos stream katkeaa kesken kaiken, niin kuinka käyttöoikeutta jatketaan. Tähän on kuitenkin varauduttu, ja sovelluksen streaming-toimittaja on kehittämässä jonkinlaista ratkaisua.

In-App purchasen -sovelluksen toiminnot ovat tapahtumatasolla seuraavat. Sovellus pyytää palvelimelta listauksen tuotteista ja näyttää ne asiakkaalle. Asiakas voi valikoimasta valita haluamansa tai sitten ei. Mahdollisen valinnan jälkeen sovellus pyytää ostoksen Applen palvelimelta käyttäen Store Kit:ä. Store Kit lähettää vastauksen transaktion. Tämän jälkeen In-App purchase lähettää palvelimelleen ns. kuitin. Sovelluksen palvelin toimittaa validoidun kuitin Applen palvelimelle. Kuviossa 7 on havainnollistettu perustasolla toimintaperiaate. (Onser Oy tiedonanto 2011.)



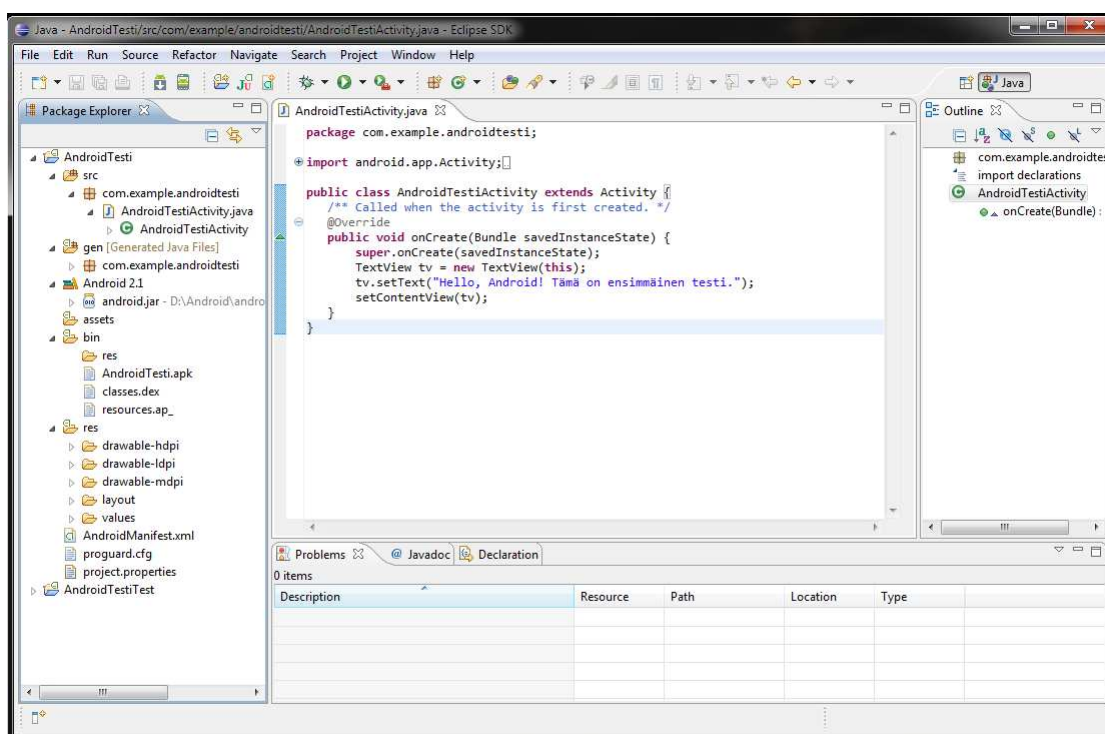
KUVIO 7. In-App Store- malli

(Apple Developer 2011)

5 ANDROIDILLE KEHITTÄMINEN

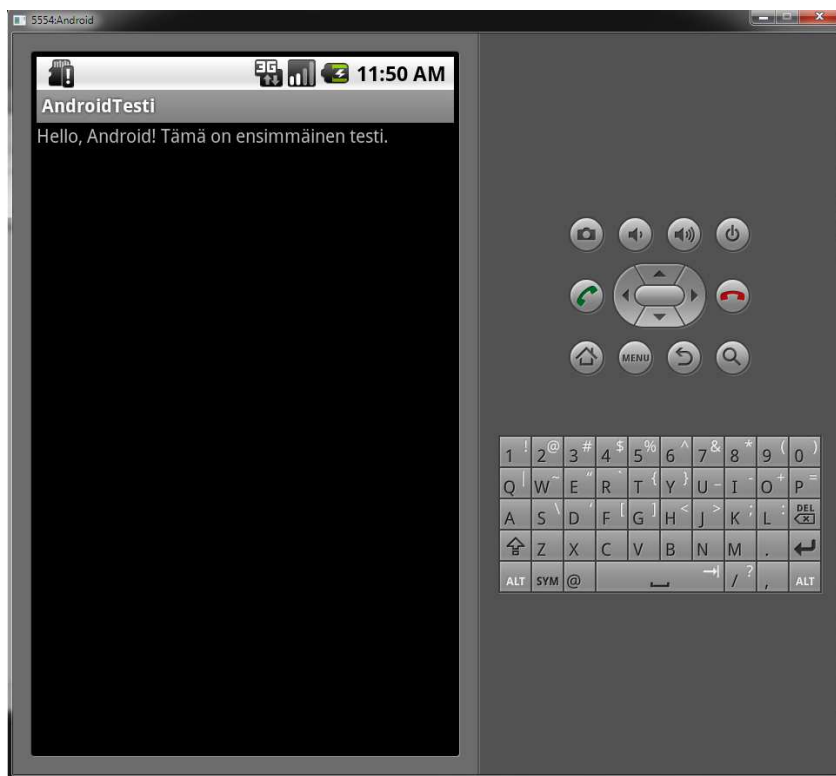
5.1 Kehitysympäristö

Androidille kehittämisen alkuun pääsee vaivattomasti ja kustannuksettomasti. Kehittäjä tarvitsee itselleen Eclipse-kehitysympäristön, joka on avoimen lähdekoodin ohjelma. Lisäksi Eclipseen tarvitaan Android Software Development Kit (SDK)-lisäosa, jonka avulla pystytään kehittämään sovelluksia Androidille. Kolmantena osana on vielä Android Development Tools (ADT), joka tarjoaa kehittämistä helpottavia työkaluja (kuvio 8). Kummatkin tiedostot ovat ladattavissa developer.android.com-sivustolta. (Android Kehitys 2011.)



KUVIO 8. Eclipsen kehitysympäristö Androidin lisäosatyökalulla

Android SDK sisältää kehitys- ja paketointityökalut sekä emulaattorin. Emulaattoriin on ladattavissa eri virtuaalisia puhelinmalleja, jotka toimivat sen kautta ja näin voidaan testata käytännössä, miltä ohjelma näyttää ajettaessa (kuvio 9.) Toinen vaihtoehto on ladata testiohjelma Android-puhelimeen, kunhan vain ensiksi määritellään asetukset oikeiksi puhelimen tiedoista. (Android Kehitys 2011.)



KUVIO 9. Hello Android -sovellus avattuna Android-emulaattoriin

Android-järjestelmä perustuu Linux kerneliin. Useimmat sovellukset on kirjoitettu Java-ohjelmointikielellä ja niitä ajetaan Dalvik-virtuaalikoneen kautta. Sovellukset on asennettu yksittäiseen tiedostoon, joka on .apk-päätteinen. Android-sovellus rakentuu neljään elementtiin, jotka ovat seuraavat:

AndroidManifest.xml: hallintatiedosto, joka kertoo järjestelmän kaikille päätason komponenteille (palvelut, aktiviteetit ja lähetysvastaanottajat), mitä kunkin on tehtävä sovelluksessa. Tämä myös määrittelee, mitkä oikeudet tarvitaan.

Aktiviteetit (Activities): koodi yksittäiselle, käyttäjäorientoituneelle tehtävälle. Usein sisältää käyttöliittymän näyttämistä käyttäjälle.

Palvelut (Services): koodin runko, jota ajetaan taustalla. Tapahtuu joko omana prosessina tai toisen ohjelman ajamana.

Lähetysvastaanottajat (Broadcast receivers): objekti, joka on toteutettu kun IPC (inter-process communication) -mekanismia käytetään käyttöjärjestelmän tai toisen sovelluksen toimesta. Esimerkkinä viesti vähäisestä akun tilasta.

(Android Security Overview 2011.)

5.2 Säilytystila

Android tarjoaa useita vaihtoehtoja pysyvän sovellustiedon tallentamiselle. Valinta riippuu siitä, mitkä ovat tiedon tallennuksen tarpeet. Pitäisikö tiedon olla salaista sovellukselle tai vastaavasti saatavissa muiden ohjelmien käyttöön ja kuinka paljon data tarvitsisi tilaa? Datan varastointivaihtoehdot ovat seuraavat:

- Jaetut asetukset (Shared Preferences): varastoi yksityisen primitiivitiedon avainarvo-pareihin.
- Sisäinen tallennustila (Internal Storage): varastoi yksityisen tiedon laitteen muistiin.
- Ulkoinen tallennustila (External Storage): varastoi julkisen tiedon jaettuun ulkoiseen varastoon.
- SQLite-tietokannat (SQLite Databases): varastoi tietuedatan tietokantaan.
- Internetyhteys (Network Connection): varastoi tiedon verkkoon omalle palvelimelle.

Videovuokraussovellusta kehitettäessä Androidille ovat huomioon otettavia tallennustekniikoita sisäinen tallennustila ja tietokanta. Elokuvatiedostot ovat tallennettuina erilliselle verkkopalvelimelle ja tilattu elokuva tallentuu puhelimen muistiin ja säilyy siellä 24 tuntia. Tärkeintä on siis tutkia, miten nämä vaihtoehdot toteutetaan. (Android Developers 2011.)

Tiedostot voidaan tallentaa suoraan puhelimen sisäiseen muistiin. Oletuksena on, että tallennetut tiedot ovat ainoastaan sovelluksen käytettävissä, joten muut tahot eivät pääsee käsiksi niihin. Tämä suojaus on upotettu Androidin toimesta ja on riittävä useimmille sovelluksille. Kun käyttäjä poistaa asennuksen, nämä tiedostot pyyhkiytyvät pois. Android tarjoaa täyden tuen SQLite-tietokannoille. Jokainen kanta, joka luodaan on saatavilla sen nimellä applikaation missä tahansa luokassa kun sitä pyydetään, mutta ei sovelluksen ulkopuolella. Suositelluin tapa luoda uusi tietokanta on tehdä alaluokka SQLiteOpenHelper-luokasta. (Android Developers 2011.)

Kehityssivustolla suositellaan ja kerrotaan ns. sisällöntuottajista (engl. content providers). Niiden avulla dataa varastoidaan, palautetaan ja tehdään saatavilla olevaksi muille kaikille sovelluksille. Sisällöntuottajat ovat ainoa tapa jakaa dataa applikaatioiden välillä, koska ei ole olemassa yhteistä tiedon tallennustilaa, johon

kaikki Android-paketit pääsisivät käsiksi. Android käsittelee lukuisia eri määriä sisällön tuottajia, joita on tehty yleisimmille datatyypeille (audio, video, kuva ja yhteystiedot yms.). Kehittäjä voi suorittaa kyselyitä näihin tuottajiin sisältötyypin mukaisesti, mutta joihinkin tietoihin on oltava kunnan oikeudet päästääkseen dataa lukemaan. (Android Developers 2011.)

Sisällöntuottajista oli pitkä listaus, ja videovuokraussovellusta silmällä pitäen ja siihen liittyen löytyi muutamia esimerkkejä: `MediaStore.Video`, `MediaStore.Video.Media`, `MediaStore.Video.VideoColumns` ja `MediaStore.Video.Thumbnails`. Ensimmäinen on ns. käyttöliittymäobjekti, joihin kolme seuraavaa sisältyy. Toiseksi viimeinen on sisällöntuottaja, jonka avulla voidaan tuoda esille tietoa videon halutuista tiedoista, kuten vaikka kuvaus. Viimeinen esimerkki on sisällöntuottaja, jonka avulla kehittäjä voi tehdä videosta thumbnailin eli pienen näytekuvan kahdella eri koolla. (Android Developers 2011.)

5.3 Julkaisun tietosuoja

Edellisessä luvussa puhuttiin sovelluksen julkaisusta ja maksutavan valitsemisesta, joten on tärkeää sen vuoksi myös tutkia tietosuojan ja suunnittelun ohjenuoria. Perusohjeita noudattamalla kuka vain sovelluskehittäjä pystyy käyttämään sovelluksen sisäisen ostomallin palveluita. Lähteessä kerrottiin muutamia pääotsikoin, mitä on otettava huomioon:

- Suorita allekirjoituksen vahvistus.
- Suojaa lukitsematon sisältö.
- Piilota lähdekoodisi ja käytä sattumanvaraisia numerosarjoja.
- Tärkein eli suojaa Android Marketin valtuuttama julkinen avain.

Ensimmäinen kohta tarkoittaa käytännössä sitä, että allekirjoituksen vahvistaminen pitäisi suorittaa palvelimeen eikä laitteeseen. Kun vahvistusprosessin sulauttaa palvelimen hoidettavaksi, hyökkääjien on vaikeampi murtaa vahvistustapahtuma takaisinmallintamalla eli tutkimalla läpi ostomallin lähdekoodin rakennetta. (Android Developers 2011.)


Toisessa kohdassa tarkoitetaan sitä, että estääksesi haittakäyttäjiiä käyttämästä suojaamaton sisältöä hyödykseen, on mieluusti käytettävä reaaliaikaista palvelua tai

etäpalvelinta sisällön tuottamiseksi. Se, että sisältöä tuotetaan jommankumman tavan kautta, auttaa pitämään tiedon ”puhtaana”. Lataamalla tietoa näiden kahden välityksellä voidaan lukiematon sisältö varastoida laitteen muistiin tai muistikortille, kuten esimerkkinä videovuokraussovelluksen iPhone-versiossa on todettu puhelimen muistin käytöstä. (Android Developers 2011.)

Ostomallin koodin piilottamalla vaikeutetaan hyökkääjien pyrkimyksiä etsiä suojaprotokollia ja muita sovelluksen komponentteja. Kehityssivusto suosittelee, että käytetään edes jonkinlaista piilotustyökalua hyväksi lähdekoodissa, kuten esimerkiksi Proguard-ohjelmaa. Lisäksi kun ajetaan piilotusohjelmaa, suositellaan käyttämään seuraavia tekniikoita piilottaakseen ostomallin koodin. (Android Developers 2011.)

5.4 Sovelluksen julkaisu Android Marketiin

Kun sovellus on saatu sellaiselle asteelle, että se on julkaisukelpoinen, se voidaan julkaista yleiseen levitykseen muiden käyttäjien saataville. Saadakseen valmiin applikaation Android Marketiin, on ensiksi rekisteröidyttävä Android-kehittäjäksi. Rekisteröityminen vaatii google-tilin, johon kehittäjyys liitetään. Tämä prosessi maksaa 25 dollaria eli n. 17 euroa, ja se on maksettava luottokortilla. Tili on ikuinen eikä vaadi muita toimenpiteitä. Rekisteröityminen onnistuu osoitteen <http://market.android.com/publish> kautta. Kuviosta 10 näkyy rekisteröitymissivun alku, jossa pyydetään tarvittavia tietoja ja jossa esimerkkinä toimii opinnäytetyön tekijän google-tilin osoite. Kuten aiemmin on mainittu, maksu hoituu Google Checkoutin välityksellä. (Android Suomi n.d.)


harri.luoma86@gmail.com | [Home](#) | [Help](#) | [Android.com](#) | [Sign out](#)

Getting Started

Before you can publish software on the Android Market, you must do three things:

- Create a developer profile
- Pay a registration fee (\$25.00) with your credit card (using Google Checkout)
- Agree to the [Android Market Developer Distribution Agreement](#)

Listing Details
Your developer profile will determine how you appear to customers in the Android Market

Developer Name
Will appear to users under the name of your application

Email Address

Website URL

Phone Number
Include plus sign, country code and area code. For example, +1-650-253-0000. [why do we ask for this?](#)

Email Updates Contact me occasionally about development and Market opportunities.

[Continue »](#)

© 2011 Google - [Android Market Terms of Service](#) - [Privacy Policy](#)

KUVIO 10. Android Marketin rekisteröitymissivu

Verrattuna Applen App Storeen Android Marketissa sovelluksien julkaiseminen on helpompaa, koska kuten aiemmin In-App Purchase -luvussa todettiin, Apple on hyvin tarkka App Storessa julkaistavista materiaaleista. Apple käy hyvin tarkasti läpi sovelluksen sisällön, ja jos se kokee sisällön sopimattomaksi tai koodista löytyy yksikin virhe, se voi olla heti hylkäämisperuste. Kuvioista 11 nähdään mitä kohtia on täytettävä ennen sovelluksen julkaisemista. Tärkeää sovelluksen ominaisuuksien valinnassa on muistaa se, ettei valittua ostomallia voi enää jälkeinpäin muuttaa. Muutoin täytettävät kohdat ovat hyvin selkeitä kielen valitsemisesta aina tuotteen hinnoitteluun.

ANDROID market

Home | Help | Android.com | Sign out

Home » MyTestApp2 » In-app Products »

Create New In-app Product

MyTestApp2
Version: 1.0
Application State: ✔ **Published**

Questions about in-app products? Click [here](#) to learn more!

In-app Product ID
0 characters (100 max)

Purchase Type Managed per user account Unmanaged
Please note that purchase type cannot be modified in the future.

Publishing State Unpublished

Language | *English (en) |
[add language](#) The **default language** is inherited from the owning application. Click [here](#) to change default language!

Title
0 characters (best viewed under 25, max 55)

Description
0 characters (80 max)

Price USD

Publish **Save**

KUVIO 11. Uuden In-app-tuotteen julkaiseminen/tallentaminen

5.5 Maksujärjestely Androidissa

Samoin kuin iPhonessa, myös Androidissa on eri maksujärjestelyjä julkaistavalle sovellukselle. Samanlainen applikaation sisäinen ostotapa on mahdollista valita. Androidille on kuitenkin vain kaksi eri vaihtoehtoa maksutyypeille ja nämä ovat ”managed per user account” ja ”unmanaged”. Ensin mainitussa sovelluksen voi ostaa kerran käyttäjätiliä kohden. Android Market säilyttää pysyvästi tapahtumatiedon tästä tilauksesta. Tämä mahdollistaa sen, että komennolla RESTORE_TRANSACTIONS

pyydetään Android Marketista tiedot sovelluksen tiedoista, jotka käyttäjä on ostanut. Jos käyttäjä yrittää tilata uudestaan jo tilattua sovellusta Android Market antaa herjan ”Item already purchased” -virheen eli tilaus on jo tehty. Kyseinen maksujärjestelytyyppi on käytännöllinen, jos myydään pelitasoja tai sovellukseen lisäominaisuuksia. (Android Developers 2011.)

Unmanaged-tyyppiä käyttävät sovellukset eivät tallennu Android Marketin tietoihin, mikä tarkoittaa sitä, ettei tilaustietoja voi kysyä kannasta. Näitä sovelluksia voi tilata rajattomasti Android Marketista. Sovelluksen kehittäjä on tässä tapauksessa vastuussa siitä, kuinka monta kertaa tällainen on mahdollista. Unmanaged-tyyppi soveltuu parhaiten kertatilauksellisille ohjelmille, ja esimerkkeinä lähteessä mainitaan mm. polttoaine tai loitsut. Tällaiset asiat ovat kulutustavaraa ja yleensä monta kertaa tilattavissa. (Android Developers 2011.)

Videovuokraussovelluksen iPhone-version ostomallina toimii kertaostoinen consumable, joten luonnollisesti myös Androidin versioon valitaan sama menetelmä. Tässä tapauksessa se on unmanaged-malli, joka sopii tähän, koska tarkoitus on maksaa palvelusta yhden kerran ja siitä ei jää muistimerkintää Android Marketiin.

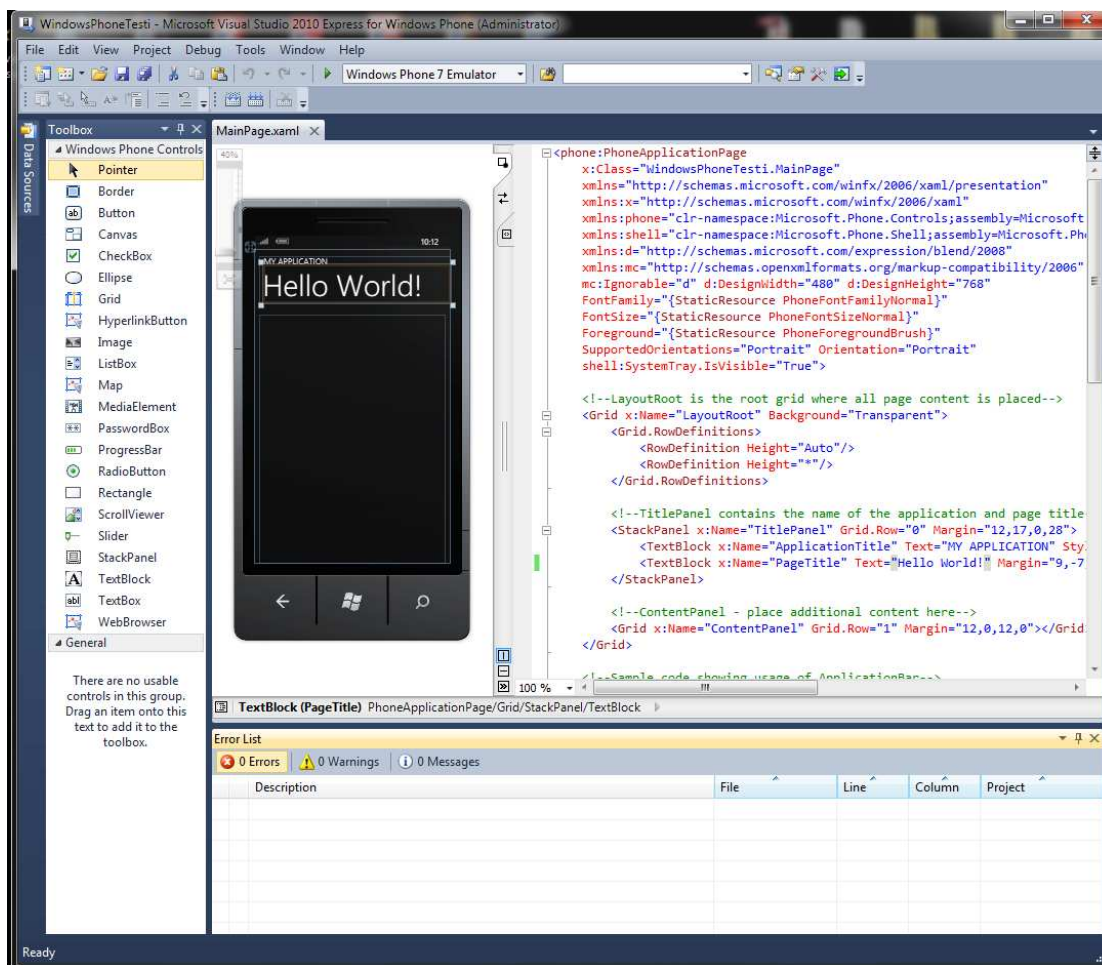
6 WINDOWS PHONE -ALUSTALLE KEHITTÄMINEN

6.1 Kehitysympäristö

Kehitettäessä Windows Phone -alustalle sovellusta on mahdollista valita kehitystyökalu kahdesta eri ohjelmointialustasta. Valittavana on C#- tai Visual Basic -ohjelmointikieli. Ensiksi on ladattava latausohjelma, joka hoitaa kehitysympäristön lataamisen ja asennuksen. Kokonaisuudessaan Windows Phone Developer Tools sisältää Visual Studio 2010 Express -ohjelman ja Windows Phone -emulaattorin. Lisäksi erityisesti pelikehitystä varten on XNA Game Studio -työkalu ja hyötyohjelmia varten Silverlight for Windows Phone. Videovuokraussovellusta kehitettäessä Silverlight on luonnollinen vaihtoehto. (The Silverlight and XNA Frameworks for Windows Phone 2011.)

Silverlight for Windows Phone

Microsoft Silverlight oli alun perin Adobe Flashin kaltainen web-ohjelmointiympäristö ja se soveltui videoiden suoratoistoon. Ohjelman kehittyessä siihen tuli mukaan lisää käyttöominaisuuksia ja tuki .NET-kielille ja kehitystyökaluille. Silverlight on yhteensopiva Microsoft Windows- ja Mac OS X -käyttöjärjestelmillä toimivien selaimien kanssa. Nyt kun myös Windows Phone-mobiilialustan kehitys on alkanut, Silverlight-tuki on laajentunut myös tähän järjestelmään. (Microsoft Silverlight 2012.) Kehityksessä on kuitenkin otettava huomioon, mitä ominaisuuksia mikäkin tukee, koska Silverlight- ja Silverlight for Windows Phone -työkalujen välillä on eroja, tai toisin sanoen jälkimmäisestä on jätetty pois ominaisuuksia. Kuviossa 11 on kehitysympäristön kuva.



KUVIO 12. Microsoft Visual Studio 2010 Express 2010 for Windows Phone
 Kun videovuokraussovellusta aletaan kehittää Windows Phone -alustalle, on Silverlight kahdesta alkukappaleesta mainituista kehitystyökalusta sopivin. Silverlight-työkalua käytetään, jos sovellukseen halutaan jokin seuraavista ominaisuuksista:

- XAML-pohjainen (Extensible Application Markup Language) sisällönkuvauskieli
- Rich Internet Application (RIA) käyttöliittymänä
- Sovelluksessa käytetään Windows Phone -alustan kontrolleja.
- Sovellukseen upotetaan videoita.
- Halutaan käyttää HTML-selainkontrollia.

XAML-sisällönkuvauskielillä luodaan sovelluksen käyttöliittymä, kuten kontrollit, muodot, tekstit ja muut näytöllä esiintyvät sisällöt. XAML on samantapainen ohjelmointikieli kuin HTML ja sisältää samoja elementtejä, mutta on paljon

tehokkaampi. Listatuista ominaisuuksista oleellisin kehitettävälle sovellukselle on tietenkin videoiden toistaminen. XNA-työkalu ei ole missään tapauksessa poissuljettu vaihtoehto, koska nämä kaksi työkalua voidaan myös yhdistää toisiinsa, jos esimerkiksi halutaan tehdä tehokas sovellus. Kuviossa 12 on pieni esimerkki siitä, kuinka XAML-kielen avulla luodaan painike. Lisäksi kuviossa 13 on havainnollistettu, kuinka XAML kirjoitetaan C#-kielellä. (Microsoft Silverlight 2011.)

```
XAML
<Grid x:Name="LayoutRoot" Background="White">

    <Button Width="60" Height="30">Click Me</Button>

</Grid>
```



KUVIO 13. XAML-koodin pätkä ja Click Me -painike

(Microsoft Silverlight 2011)

```
C#
// Initialize the button
Button myButton = new Button();

// Set its properties
myButton.Width = 60;
myButton.Height = 30;
myButton.Content = "Click Me";

// Attach it to the visual tree, specifically as a child of
// the Grid object (named 'LayoutRoot'). In other words, position
// the button in the UI.
LayoutRoot.Children.Add(myButton);
```

KUVIO 14. C#-koodin pätkä ja Click Me -painike

(Microsoft Silverlight 2011)

Tutoriaaleissa oli myös hyvä esimerkki siitä, kuinka saadaan aikaiseksi videotiedoston toisto Windows Phone -alustalla, koska kysymyksessä on videovuokraussovelluksen kehitys, joten nämä ovat hyviä ja tarpeellisia malleja. Esimerkissä oli kaksi eri vaihtoehtoa: MediaPlayerLauncher ja MediaElement API. Jälkimmäinen on tarkoitettu kokeneemmille ohjelmoijille, ja tämän elementin käyttö havainnollistettiin harjoituksessa. Huomioon on otettava myös se, mitä tiedostotyyppejä Windows Phone tukee. (Microsoft Silverlight 2011.)

Ensimmäisessä kohdassa näytettiin, kuinka paikallinen tiedosto soitetään alustalla komennon mukaisesti. Avataan valmis projekti ja lisätään haluttu tiedosto painamalla hiiren oikealla näppäimellä eli Add -> Existing Item. Tämän jälkeen täytyi vielä valita tiedoston tyyppi Resource. Sovelluksen lähdekoodin piti lisätä seuraavaksi kuvan näyttämä palanen koodia. Tässä tapauksessa "test.wmv":n tilalla on oma nimi tiedostolle. (Microsoft Silverlight 2011.) Yritin kokeilla tätä ominaisuutta itse, mutta kyseinen MediaElement -luokka ei ole tuettu emulaattorissa, joten tulos jäi siltä osin näkemättä ja en omista Windows Phone -puhelinta.

Toinen esimerkki on siitä, kun tiedosto haetaan palvelimelta suoraan. Tämä onnistuu samalla kaavalla, mutta erillistä tiedostoa ei tarvitse lisätä projektiin.

Videovuokraussovelluksen yhtenä ominaisuutena iPhone-versiossa on se, että tiedostojen säilytystila ja streaming on ulkoistettu ja näin ollen erillisellä palvelimella, joten tämä olisi mahdollisesti sopiva ratkaisu siihen tarkoitukseen. (Microsoft Silverlight 2011.)

```
XAML
<MediaElement Source="test.wmv" AutoPlay="True"/>

XAML
<MediaElement Source="http://mschannel9.vo.msecnd.net/o9/mix/09/wmv/key01.wmv" AutoPlay="True"/>
```

KUVIO 15. MediaElement API

(Microsoft Silverlight 2011)

Sovelluksen rakentaminen on melko lailla selkeää, rakenteen muokkaaminen on helppoa ja valmiin koodin testaaminen on kätevää ohjelmassa olevan emulaattorin

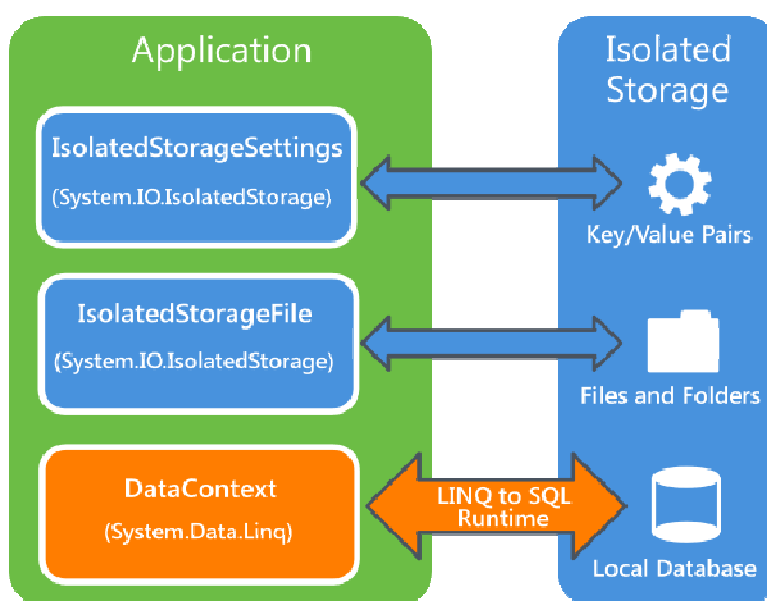
kautta. Ainut ongelma tulee vastaan silloin, kun halutaan ajaa sovellusta puhelimesta. Silloin on rekisteröitävä puhelin testikäyttöön App Hub -portaalissa, ja tämä ei onnistu ilman maksettua jäsenyyttä.

6.2 Säilytystila eli Isolated Storage

Isolated Storage mahdollistaa hallittavien sovelluksien luoda ja ylläpitää omaa tietovarastoa. Windows Phone -applikaatioissa kaikki I/O (input/output) -operaatiot on eristetty. Applikaatioilla ei ole suoraa yhteyttä perustana olevaan käyttöjärjestelmän tiedostojärjestelmään tai muiden sovelluksien omiin tietovarastoihin. Tämä parantaa turvallisuutta, vähentää ulkopuolisia haittoja ja tiedon korruptoitumista. (MSDN 2011.)

Tiedon säilytyksellä on muutamia tallennustapoja riippuen tiedon tyypistä. Windows Phone -sovellukset käyttävät varastointia puhelimeen seuraavin tavoin. Kuviossa 14 se on havainnollistettu tarkemmin.

- Asetukset (Settings): varastoi datan avain-/arvopareina käyttäen `IsolatedStorageSettings`-luokkaa.
- Tiedostot ja kansiot (Files and folders): varastoi tiedostot ja kansiot käyttäen `IsolatedStorageFile`-luokkaa.
- Relationalinen data (Relational data): varastoi datan tietokantaan käyttäen LINQ (Language Integrated Query) -komponenttia SQL:ään (Structured Query Language).



KUVIO 16. Isolated Storage ja sovellus

(MSDN 2011)

Puhelimen varastointiresurssit ovat kuitenkin rajalliset. Windows Phone -applikaatioilla ei ole säädettävää tilakiintiötä, koska jokaisen sovelluksen vaatimukset ovat erilaiset. Jos levytilan kokoa säättää sovellusta kohden, sillä voi olla epämieluisia lopputuloksia käyttäjille. Tämän vuoksi sovelluksien pitäisi säilöä tietoa vain, kun se on tarpeellista. Tiedon säilömisessä kannattaa käyttää muutamia hyviä tekniikoita, kuten lähdemateriaali neuvoi. (MSDN 2011.)

Jos sovellus luo mitä tahansa väliaikaista tietoa Isolated Storage -varastoon, on pidettävä huoli, että tieto pyyhitään pois, kun sitä ei enää tarvita. Siksi on hyvä luoda sovellusta kehitettäessä kansio välimuistille, jonka voi sitten säännöllisin väliajoin puhdistaa. Videovuokraussovellus sisältää mitä todennäköisimmin myös tietoa tehdyistä tilauksista, joten on hyvä poistaa niitä tietoja, jotka ovat vähemmän tärkeitä. (MSDN 2011.) Kuviossa 15 on ohjelmakoodiesimerkki siitä, kuinka System.IO.IsolatedStorage-tehtävä toimii.

```

C#
using System.IO;
using System.IO.IsolatedStorage;

private void btnWrite_Click(object sender, RoutedEventArgs e)
{
    // Obtain the virtual store for the application.
    IsolatedStorageFile myStore = IsolatedStorageFile.GetUserStoreForApplication();

    // Create a new folder and call it "MyFolder".
    myStore.CreateDirectory("MyFolder");

    // Specify the file path and options.
    using (var isoFileStream = new IsolatedStorageFileStream("MyFolder\\myFile.txt", FileMode.OpenOrCreate, myStore))
    {
        // Write the data
        using (var isoFileWriter = new StreamWriter(isoFileStream))
        {
            isoFileWriter.WriteLine(txtWrite.Text);
        }
    }
}

// This code opens and reads the contents of myFile.txt.
private void btnRead_Click(object sender, RoutedEventArgs e)
{
    // Obtain a virtual store for the application.
    IsolatedStorageFile myStore = IsolatedStorageFile.GetUserStoreForApplication();

    try
    {
        // Specify the file path and options.
        using (var isoFileStream = new IsolatedStorageFileStream("MyFolder\\myFile.txt", FileMode.Open, myStore))
        {
            // Read the data.
            using (var isoFileReader = new StreamReader(isoFileStream))
            {
                txtRead.Text = isoFileReader.ReadLine();
            }
        }
    }
    catch
    {
        // Handle the case when the user attempts to click the Read button first.
        txtRead.Text = "Need to create directory and the file first.";
    }
}

```

KUVIO 17. Tiedostojen tallennus Windows Phone -puhelimeen (MSDN 2011.)

Kuvassa yksinkertaisesti luodaan applikaatio, johon voidaan syöttää tekstiä, tallennetaan teksti tiedostoon ja lopuksi luetaan tiedoston sisältö.

6.3 Tiedon suojaus

Luottamuksellisen tiedon tallentaminen puhelimen Isolated Storage -varastoon ei ole täysin turvallista. Tiedon salaaminen ei nosta turvallisuutta, jos salauksen purkuavain on puhelimesta ja lisäksi ei ole väliä, kuinka hyvin se on piilotettu. Data Protection API (DPAPI) ratkaisee tämän ongelman tarkasti luomalla ja varastoimalla salausavaimen käyttäjän ja puhelimen tunnistetietoja. DPAPI salaa ja purkaa tiedon tunnistetietoja hyödyntäen. ProtectedData-luokkaa soveltaen pääsee yhteyteen DPAPI:iin ja käyttämään Protect- ja Unprotect- menetelmiä. Windows Phone -laitteissa jokainen applikaatio saa oman salausavaimen, joka luodaan silloin

kun sovellus ensimmäisen kerran ajetaan. Avain pysyy samana päivitystenkin jälkeen. (MSDN 2011.)

Kun käytössä on paljon tietoa ja se on varastoituna tietokannassa Windows Phone -applikaatiota varten, on järkevintä salata koko tietokanta kuin yksittäinen tiedonjyvä. Kuten aiemmin mainittiin, sovellukset voivat käyttää LINQ-komponenttia tallennettaessa tietoa SQL:n avulla kantaan. Ennen kuin tietokanta luodaan, voidaan käyttää ns. yhteysmerkkijonoa (engl. connection string) määrittelemään tietokannan konfigurointiarvot. Esimerkkinä se, että merkkijono voi määrittellä sen, salataanko tietokanta salausavaimella vai salasanalla. Alla olevassa kuviossa 16 luodaan salattu tietokanta. (MSDN 2011.)

```
C#  
  
// Create the data context.  
MyDataContext db = new MyDataContext("Data Source='isostore:/mydb.sdf';Password=  
'securepassword'");  
  
// Create an encrypted database after confirming that it does not exist.  
if (!db.DatabaseExists()) db.CreateDatabase();
```

KUVIO 18. Salatun tietokannan luominen

(MSDN 2011)

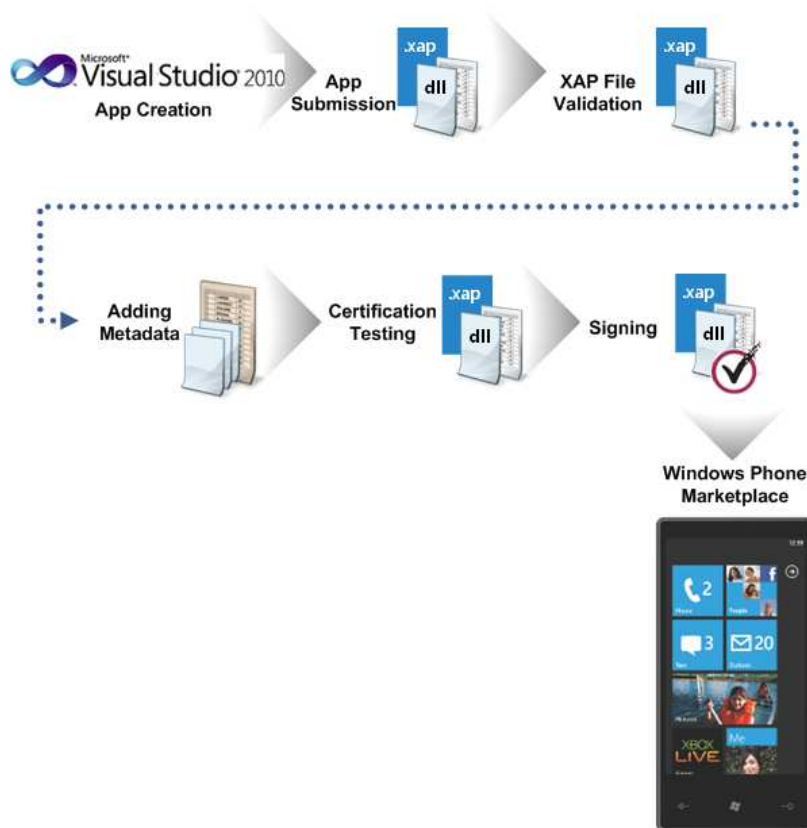
6.4 Sovelluksen julkaisu Windows Phone Marketplaceen

Pääperiaatteet eheän sovelluksen julkaisulle ovat hyvin loogiset ja yksinkertaiset, ja kun ne ovat kunnossa, hyväksytyin applikaation kehittäminen ei ole suuri ongelma. Ominaisuudet ovat seuraavat:

- Sovelluksen on oltava luotettava.
- Sovellus käyttää tehokkaasti hyödykseen resursseja.
- Sovellus ei häiritse puhelimen toiminnallisuutta.
- Sovellus ei sisällä minkäänlaista haittaohjelmaa.

Julkaistaakseen valmiin sovelluksen kehittäjän on rekisteröidyttävä App Hub -portaalissa Windows Phone -kehittäjäksi. Liittymismaksu palveluun on vuosittain 99 dollaria. Rekisteröitymisprosessi kestää muutaman viikon, jona aikana varmistetaan

kehittäjän oikea henkilöllisyys siltä varalta ettei tule haittaohjelmia. Rekisteröityessä kehittäjä tarvitsee Windows Live ID:n, yhteystiedot ja luottokortin numerot. Käyttäjätilejä on tarjolla kolme, joista on valittava yksi oikeaan tarkoitukseen: yritys, yksilö tai opiskelija. Näistä jokainen sisältää hiukan eri ominaisuuksia. App Hub-sivuston kautta hallinnoidaan kehitettyjä sovelluksia ja julkaistaan ne Windows Phone Marketplace -sovelluskauppaan. Julkaistava sovellus lähetetään hyväksyttäväksi, ja se käy läpi sertifiointiprosessin. Microsoft lupaa, että prosessi kestää normaalisti viisi työpäivää, mutta joissain tapauksissa on jouduttu odottamaan jopa kymmenisen työpäivää hyväksymistietoa. (MSDN 2011.) Kuviossa 19 on havainnollistettu sertifiointiprosessi yksinkertaistettuna.



KUVIO 19. Windows Phone Marketplace-sertifikaatioprosessi
(MSND 2011)

6.5 Maksujärjestely Windows Phonessa

Windows Phone -alustalle ei ole tällä hetkellä kehitettynä omaa In-App purchase -APIa. Foorumeilla ja uutispalstoilla oli alustavia tietoja siitä, että tämä ominaisuus olisi tulossa jossain vaiheessa ja kuuluu Microsoftin kehitettävien asioiden listalle. (LiveSide.net 2011.) Tämän takia sovelluksille ei ole saatavilla eri vaihtoehtoja maksujärjestelyistä, kuten vastaavasti on iPhonelle ja Androidille.

Windows Phone -sovellukselle on tehty muutamia kolmannen osapuolen maksujärjestelyohjelmia. Yksi vaihtoehto on MoVend, joka on alun perin suunniteltu Androidille mutta on sitten laajennettu myös Windows-puolelle. Palvelu mahdollistaa rahansiirtoliikenteen ja maksut tapahtuvat saumattomasti ilman, että pitäisi poistua sovelluksesta. MoVend tarjoaa kehittäjälle myös käyttäjä- ja lisenssihallinnoinnin, myyntien seuraamisen ja analysoinnin. MoVendille on oma kehitysympäristönsä, joka ”yhdistetään” kehitettyyn sovellukseen, ja sen avulla määritellään kaikki tarvittavat asetukset. Työkalut tähän ovat saatavilla osoitteesta www.movend.com ja sieltä löytyvät luonnollisesti myös ohjeet kehitykselle. Kuviossa 20 on näytetty, kuinka MoVend-osa lisätään. (WMPoweruser 2011.)

Step 1: Implement *IMoVend* Interface

Your class must implement *IMoVend* and override the necessary call back methods.

```
public partial class MainPage : PhoneApplicationPage, IMoVend
{
    MoVendAPI api;

    // Post-purchase call-back methods
    public void PurchaseSuccessCallback(NavigationService source)
    {
        // Navigate to the app.
        source.Navigate(new Uri("/MainPage.xaml", UriKind.RelativeOrAbsolute));
    }

    public void PurchaseFailureCallback(
        NavigationService source, ResultType resultType, string errorMessage)
    {
        // Navigate to the app.
        source.Navigate(new Uri("/MainPage.xaml", UriKind.RelativeOrAbsolute));
    }
}
where ResultType is an enum,
public enum ResultType { CLIENT_ERROR, SERVER_ERROR, OK, INFO }
```

Step 2: Instantiate the MoVendAPI object

Retrieve an API instance by calling in your Page_Loaded Event Handler:

```
private void PhoneApplicationPage_Loaded(object sender, RoutedEventArgs e)
{
    // Instantiate MoVend
    api = MoVendAPI.GetInstance(this, "YourApiKey", "Your Application ID");
}
```

KUVIO 20. MoVend-käyttöliittymän lisäys sovelluksen lähdekoodiin
(MoVend Developers Resource 2011)

MoVend tarjoaa myös eri vaihtoehtoja haluttavalle maksujärjestelytyypille. Nämä ovat consumable items, subscription-based items ja activation-based items. Kaikki edellä mainitut ovat siis kutakuinkin samoja, mitä muiden käyttöjärjestelmien vastaavat tarjoavat. Ensimmäinen malli on kertaluonteinen tyyppi, toinen malli on tarkoitettu tietyksi ajaksi (päivittäinen tai kuukausittainen) ja viimeinen malli on lisenssipohjainen eli siinä on rajoittamaton käyttöaika. Tässäkin tapauksessa videovuokraussovelluksen toimintatavaksi valitaan consumable items kertaluontoisen maksun vuoksi. (MoVend Developers Resource 2011.)

Windows Phone -sovelluksien hinnoittelu on minimissään 0,99 dollaria ja tämä suhteutettuna haluttuun valuuttaan. Vastaavasti maksimihinta saa applikaatiolle olla 499,99 dollaria ja se muutettuna haluttuun valuuttaan. Hinnoittelulla ja verotuksella on maakohtaisia variaatioita, mutta keskimäärin kaikille alueille ovat samat osuudet eli 70 % valitusta hinnasta jää kehittäjälle ja loput 30 % menevät Microsoftille. (MSDN 2011.)

7 JOHTOPÄÄTÖKSET JA POHDINTA

Opinnäytetyön tuloksena saatiin päällisin puolin tietoa siitä kuinka on kehitettävissä mobiilisovellus muihin alustoihin, kun pohjalla on jo valmis videovuokraussovellus ominaisuuksineen. Tutkimustulokset antoivat lähinnä perustasolla vastauksia kehitettäviin asioihin, mutta kehitystyökalut tulivat hyvin selväksi allekirjoittaneelle, jolla ei ole juuri yhtään kokemusta ohjelmoinnista. Lähtökohta oli täten hiukan hataralla pohjalla, ja työn teon aloittaminen viivästyinkin tämän takia, koska ei ollut juuri minkäänlaista käsitystä mistä kaiken aloittaisi. Tässä opinnäytetyössä ei toki tehty riviäkään itse lähdekoodia, mikä helpotti tutkimustyötä ja prosessia.

Valmiin videovuokraussovelluksen laajentaminen iPhone-alustalta Androidille ja Windows Phonelle ei ole loppujen lopuksi kovinkaan vaikeaa tai vaikeaselkoista. Lähtiessäni selvittämään Android- ja Windows Phone-järjestelmien kehitysympäristöjä ja miten valmiit ominaisuudet voitaisiin siirtää, asiat lähtivät etenemään hitaasti ja pala kerrallaan. Tilannetta helpotti se, että iPhone-sovellus oli valmiina ja sain käyttööni kaiken mahdollisen dokumentaation tähän liittyen. Ensiksi oli tutustuttava tähän applikaatioon ja sen toiminnallisuuksiin. Vähitellen pääsin jyvälle asioista ja aloin kartoittaa muiden kehitysympäristöjen ohjeita, miten samat ominaisuudet onnistuisi näiden avulla toteuttamaan.

Ensimmäisenä tietenkin oli asennettava kummankin alustan kehitysympäristöt ja niiden kanssa ei tullut ongelmia. Syynä oli lähinnä se, että tarvittavat ohjelmistot lisäosatyökaluineen oli helposti saatavilla ilman erillisiä rekisteröitymisiä ja maksuja. Androidin kehitysympäristö Eclipse ja siihen asennettava liitännäinen olivat sinänsä selkeitä, ja nämä avoimen lähdekoodin ohjelmat tarjosivat hyvät puitteet aloittelevallekin kehittäjälle. Windows Phonelle tuli vastaan sellaisia asioita, että oli oikeastaan tiedettävä, mitä etukäteen haluaa. Microsoftilla on tietenkin takanaan pidempi historia kehitystyössä, joten valittavia ympäristöjä oli sen mukaisesti. Silverlight-työkalusta oli eri versioita ja valittavana on maksullinen versio tai ilmainen, josta puuttuu sitten joitain ominaisuuksia.

Tutkittaessa kehitettyjen sovelluksien lopullista julkaisua sovelluskauppaan vaativat Android ja Windows Phone keskenään hieman eri tavalla sen mitä sovellukselta ja kehittäjältä vaaditaan. Androidin ollessa avoimeen lähdekoodiin perustuva järjestelmä

sovelluksien kehittäminen ja julkaisu olivat huomattavasti helpompia. Toki näissäkin tapauksissa sovellus käy läpi seulan, mutta ei niin tarkkaa kuin verrattuna Microsoftiin tai aiemmin mainittuun Appleen. Tästä syystä Android Marketiin julkaistaan huomattavan paljon applikaatioita.

Kartoitettaessa haluttuja ominaisuuksia, kuten tutkimuskohteita, tietoturvaa ja säilytystilaa, näistä kahdesta osasta ohjeet vaihtelivat huomattavasti. Windows Phone -alustalle oli hyvin selkeästi kerrottu, miten on kannattavaa suojata applikaation tiedot ja kuinka tallentaminen onnistuu. Vastaavasti Androidin puolella asiat olivat vaikeammin selitetty ja vaihtoehtoja oli enemmän. Ongelmia tuli siinä, miten tällaiset elementit olisi toteutettavissa. Toisin sanoen materiaalia oli kyllä riittävästi, mutta asian ytimen löytäminen tuotti välillä vaikeuksia.

Lopputuloksena oli tietoa kahdesta eri kehitystyökalusta ja -ympäristöstä. Hiljalleen kiiri mieleen ajatus myös siitä, että olisin voinut kokeilla käytännössä, kuinka sovelluksen kehittäminen tapahtuisi. Tein muutamia koodin pätkiä vain havainnollistaakseni ohjelmien ja ohjelmointikielien käyttöä. Huomasin työn edetessä sen, ettei koodaaminen tai erilaisten osien luominen ole niin epäloogista ja vaikeaa, kuin aiemmin olin kuvitellut.

Pohdintaosuuden tekoaikana selailin samalla uutisia, ja vastaan tuli tuore älypuheliin liittyvä uutisointi. Siinä kerrottiin, kuinka Android olisi vähitellen nousemassa suosituimmaksi alustaksi kehittäjien parissa, ja tämä muutos tapahtuisi joskus vuoden sisällä. Toki iOS-kehittäjiä on vielä enemmistö, mutta tutkimuksien mukaan noin 27 % aikoi tukea sitä vielä jatkossa ja 29 % kannattaisi Androidia. Artikkelissa huomioitiin myös Windows Phonen tuleva markkina-arvo ja oltiin sitä mieltä, että Microsoft on onnistunut vakuuttamaan kehittäjät mobiilijärjestelmän hyvydestä. Lopuksi mainittiin web-standardien muuttuvasta kentästä eli Java ja Flash olisivat jäämässä tulevaisuuden tekniikan HTML5:n jalkoihin. (Tietokone 2012.)

LÄHTEET

- About Silverlight. N.d. Microsoft Silverlight. Viitattu 6.2.2012.
<http://www.microsoft.com/silverlight/what-is-silverlight/#what-is-silverlight>
- Android Developers. 2011. Android-kehittäjien sivusto. Viitattu 9.12.2011.
<http://developer.android.com/>
- Android Open Source Project. 2011. Android-sovelluskehitys. Viitattu 24.11.2011
<http://source.android.com>
- Apple Developer - iOSDev Center. 2011. Viitattu 8.12.2011.
<Http://developer.apple.com/devcenter/ios/index.action>
- BellSouth – IBM Simon PDA cellph. N.d. RetroCom. Viitattu 1.11.2011.
http://www.retrocom.com/bellsouth_ibm_simon.htm
- damaster. 16.6.2011. In-app purchase coming to Windows Phone "Mango".
liveside.net. Viitattu 8.1.2011. <http://www.liveside.net/2011/06/16/in-app-purchase-coming-to-windows-phone-mango/>
- Kaarlo. 7.2.2011. Näin pääset Android kehityksessä alkuun. Android sovelluskehitys.
Viitattu 17.11.2011 <http://androidkehitys.fi/sovelluskehitys/nain-paaset-android-kehityksessa-alkuun/>
- Hirsjärvi, S., Remes & P., Sajavaara, P. 2007. Tutki ja kirjoita. 13. uud. p. Helsinki: Tammi.
- iTunes. 2012. iTunes A:sta Ö:hön. Apple. Viitattu 6.2.2012.
<http://www.apple.com/fi/itunes/features/>
- Masalin, T. 2011. iPad & iPhone käsikirja. Jyväskylä: WSOYpro.
- Microsoft Silverlight Team. 21.6.2010. XAML: Silverlight:NET. Viitattu 14.12.2011.
[http://www.silverlight.net/learn/overview/what-is-silverlight/xaml-\(silverlight-quickstart\)](http://www.silverlight.net/learn/overview/what-is-silverlight/xaml-(silverlight-quickstart))
- Mikkonen, T. 2004. Mobiiliohjelmointi. Helsinki: Talentum.
- Mikä on Android? N.d. Android Suomi. Viitattu 9.11.2011.
<http://blog.androidsuomi.fi/mika-on-android/>
- MoVend Developer's Resource. 18.7.2011. In-app mobile commerce platform. Viitattu 8.1.2011. <http://developer.movend.com>
- MSDN – Explore Windows, Web, Cloud, and Windows Phone Software Development. 2011. Microsoft-kehitysympäristö. Viitattu 15.12.2011.
<http://msdn.microsoft.com/en-us/default.aspx>
- Onser Oy. 2011. Tiedonanto, dokumentit ja kuvat.

Paananen, V. 31.1.2012. Windows Phone FAQ eli usein kysytyjä kysymyksiä. Mr. Windows Phone Finland | Vesku Paanasen mobiiliblogi Windows Phonesta ja muustakin. Viitattu 6.2.2012. <https://mrwpf.wordpress.com/windows-phone-faq-eli-usein-kysytyj-kysymyksi/>

Radu. 13.6.2011. Apple vs Google: The Battle has Two Winners. Geek with Laptop. Viitattu 10.11.2011 <http://www.geekwithlaptop.com/apple-vs-google-the-battle-has-two-winners>

Saarelainen, A. 24.1.2012. Android ohittaa iOS:n kehittäjien tärkeimpänä alustana. Viitattu 25.1.2012. http://www.tietokone.fi/uutiset/android_ohittaa_ios_n_kehittajien_tarkeimpana_alustana

Savov, V. 2.4.2011. Android Market gets a web store with OTA installations, in-app purchases coming soon. Engadget. Viitattu 9.11.2011. <http://www.engadget.com/2011/02/02/android-market-gets-a-web-store/>

Sisättö, S. 2004. Mobiiliin sähköiseen yhteiskuntaan - internet taskussa. Tampere: Tammer-Paino.

Smart phone definition. N.d. Phone Scoop. Viitattu 1.11.2011. <http://www.phonescoop.com/glossary/term.php?gid=131>

Surur. 4.6.2011. MoVend offers In-App purchase SDK for Windows Phone 7 developers. WMPoweruser - Windows Phone Tech Site. Viitattu 8.1.2011. <http://wmpoweruser.com/movend-offers-in-app-purchase-sdk-for-windows-phone-7-developers/>

Taimio, H., Salmi, M-M. 2005. Opinnäytetyöllä onneen. Helian julkaisusarja A:22. Helsinki: Edita.

Windows Phone 7. 2011. Microsoft Windows Phone 7. Viitattu 8.11.2011. <http://www.microsoft.com/windowsphone/fi-FI/default.aspx>