

Android versus Windows Phone

Jesse Tuuri

Opinnäytetyö
Toukokuu 2012
Tietotekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikka
Ohjelmistotekniikka

Jesse Tuuri:
Android versus Windows Phone

Opinnäytetyö 38 sivua, josta liitteitä 0 sivua
Toukokuu 2012

Tämä opinnäytetyö käsittelee älypuhelinohjelmointia. Mahdollisista alustoista valittiin kaksi, tässä tapauksessa Android ja Windows Phone. Aluksi käsitellään lyhyesti älypuhelinohjelmointia yleisellä tasolla. Mitä ongelmia sirpaloituminen tuottaa ja mitä rajoituksia laitteet aiheuttavat.

Seuraavaksi käydään läpi molemmat alustat teknisten ominaisuuksien, ohjelmoinnin ja ohjelmointiympäristön puolesta. Sekä itse sovelluksen esittely. Miltä sovellus näyttää ja mitä tekniikoita sen tekemiseen käytettiin. Myös sovelluksen kehittäminen, toteutus, testaus ja rajoitukset käydään läpi.

Lopuksi sitten pohditaan näiden kahden alustan eroja niiden vahvuuksien, heikkouksien ja tulevaisuuden osalta. Kuinka helppoa on lähteä ohjelmoimaan kyseisellä alustalle tai miten helppoa sen markkinointi ja mahdollisesti rahan tekeminen sillä on.

ABSTRACT

Tampere University of Applied Sciences
ICT Engineering
Software Engineering

Jesse Tuuri:
Android versus Windows Phone

Bachelor's thesis 38 pages, appendices 0 pages
May 2012

This thesis handles mobile programming, more accurately programming for Android and Windows Phone. A RSS feed reader was made for both systems and through this we try to figure out the differences and problems in the systems.

First part of the thesis is mostly about technical information, programming and programming environment in Android and Windows Phone. After that program itself is covered. How does it work and how it looks like.

Chapter 5 is about development, implementation, testing and limitations of the program. The program wasn't marketed in Google Play or in Windows Phone Marketplace but they are still covered in this thesis.

In the last part of the thesis, we look closer in the differences between the two platforms. What are their strengths and weaknesses and in what ways they are similar. Also how does the future look like for the platforms.

Key words: Android, Windows Phone, programming, platform

SISÄLLYS

1	JOHDANTO	7
1.1	Älypuhelinohjelmiston kehitys	7
1.2	Tämä työ	7
2	ANDROID	9
2.1	Yleistä.....	9
2.2	Tekninen esitys	9
2.2.1	Activity.....	10
2.2.2	Intent	10
2.2.3	Services	11
2.3	Ohjelmointiympäristö.....	11
2.4	Arkkitehtuuri.....	14
2.5	Työkalut.....	15
2.6	Ohjelman asentaminen puhelimeen	18
3	WINDOWS PHONE	19
3.1	Yleistä.....	19
3.2	Tekninen esitys	19
3.2.1	Silverlight	19
3.2.2	XNA.....	20
3.3	Ohjelmointiympäristö.....	20
3.4	Arkkitehtuuri.....	22
3.5	Työkalut.....	23
3.6	Ohjelman asentaminen puhelimeen	24
4	SOVELLUKSEN ESITTELY	25
4.1	Yleisesti	25
4.2	Käyttöliittymä	25
4.2.1	Käyttöliittymä Androidissa	25
4.2.2	Käyttöliittymä Windows Phonessa.....	28
4.3	Tekniikat.....	30
4.3.1	Android	31
4.3.2	Windows Phone.....	31
5	SOVELLUKSEN KEHITYS.....	32
5.1	Kehitystyö.....	32
5.2	Toteutus	32
5.3	Testaus.....	32
5.4	Rajoitukset.....	32
5.5	Sovelluksen markkinointi.....	33

5.5.1	Google Play	33
5.5.2	Windows Marketplace	33
6	POHDINTA	34
6.1	Ohjelmointi	34
6.2	Markkinat	35
6.3	Yhteenveto	35
	LÄHTEET	38

LYHENTEET JA TERMIT

RSS	Rich Site Summary
XML	Extensible Markup Language
XAML	Extensible Application Markup Language
SQL	Structured Query Language
SDK	Software Development Kit
URL	Uniform Resource Locator
USB	Universal Serial Bus
XAP	Silverlight application package
Drag & drop	Valittu kohde vedetään ja tiputetaan haluttuun paikkaan

1 JOHDANTO

1.1 Älypuhelinohjelmiston kehitys

Ohjelmien kehittäminen puhelimelle tuottaa omat haasteensa. Ohjelmointikieleen puuttumatta pitää ottaa huomioon puhelimen näyttö, muistirajoitukset ja mitä kaikkia ominaisuuksia puhelimesta edes löytyy, joita pystytään hyödyntämään. Koska samalla käyttöjärjestelmällä voi olla useita erilaisia puhelinmalleja, näiden asioiden huomioon ottaminen voi olla hyvin hankalaa, ainakin jos haluaa että kehitetty ohjelma on yhteensopiva mahdollisimman monen puhelimen kanssa. Apple on ainoa, jota tämä nk. sirpaloituminen ei niinkään koske. Periaatteessa pitää vain päättää ohjelmoitko iPadille vai iPhoneille. Kaikista muista käyttöjärjestelmistä sitten sirpaloitumista löytyykin (Android, Symbian, Windows Phone, BlackBerry).

Sirpaloituminen tarkoittaa juuri sitä, kun puhelinmalleja on niin paljon erilaisia, että ohjelmistojen kehittäminen ja jopa itse käyttöjärjestelmän päivittäminen on haastavaa ellei mahdotonta saada onnistumaan jokaiselle puhelimelle, koska puhelimiensa näytöt, muistit ja muut ominaisuudet eroavat toisistaan.

Usein puhelinvalmistajat vielä lisäävät omia ominaisuuksiaan käyttöjärjestelmään, mikä taas tuottaa lisää ongelmia. Käyttöjärjestelmien päivityksien vastuu yleensä onkin puhelinvalmistajille itsellään, jotka päättävät pystytäänkö päivitetty käyttöjärjestelmä ottamaan käyttöön tietyissä puhelimissa. Tällä hetkellä ajankohtaisin esimerkki aiheesta on Androidin Ice Cream Sandwich -päivitys, joka ei ole saatavilla kovinkaan monelle vanhemman käyttöjärjestelmän omaavalle puhelimelle. Muun muassa Samsung on ilmoittanut, ettei uusin päivitys ole saatavilla heidän lippulaivapuhelimensa vanhemmalle mallille, Samsung Galaxy S:lle, mutta S2:lle päivitys tulee. Syy tälle on, ettei Galaxy S:n teho eikä muisti riitä käyttöjärjestelmän sopivaan pyörittämiseen.

1.2 Tämä työ

Vaikka älypuhelimille on paljon erilaisia käyttöjärjestelmiä, tässä työssä keskityn vertailemaan Windows Phonea ja Androidia. Mielestäni nämä kaksi ovat mielenkiintoisimmat vaihtoehdot, koska Android on noussut hyvin nopeasti

älypuhelinmarkkinoilla ja Windows Phone sen sijaan on Microsoftin yritys päästä kunnolla kiinni älypuhelinmarkkinoille ja se on järjestelmä, jonka Nokia päätyi valitsemaan uudeksi käyttöjärjestelmäkseen pystyäkseen kilpailemaan Androidia ja iOS:ää vastaan.

Esimerkki ohjelmana käytetään RSS-lukijaa. Periaatteessa ainoa vaatimus kännykälle on, että siitä löytyy internet-yhteys. Joten tässä työssä ei kauheasti tarvitse murehtia kännyköiden ominaisuuksista, näyttöjen koosta tai muistimääristä. Ohjelma on yksinkertainen, mutta tarkoitus on lähinnä keskittyä Windows Phonen ja Androidin eroihin ohjelmoinnissa, työkaluissa ja ohjelmien myynnissä. Tietysti pyrin tuomaan esiin eroavaisuuksia, joita löytyy mutta eivät tule ilmi tässä kyseisessä ohjelmassa, pääasiassa grafiikkaan liittyvät eroavaisuudet.

2 ANDROID

2.1 Yleistä

Android on Googlen mobiilikäyttöliittymä, jota käytetään pääasiassa kännyköissä ja tableteissa. Androidia kehitti alun perin Android Inc., jonka Google osti 2005 ja nykyään Androidia kehittää Open Handset Alliance. Open Handset Alliance on teknologia- ja mobiiliyritysten muodostama ryhmä. Useat ryhmän jäsenistä pyrkivät tekemään rahaa Androidilla, joko myymällä puhelimia, palveluja tai mobiilisovelluksia. Android on avoin ympäristö, joka pohjautuu Linuxiin. Androidia kehitetään aktiivisesti, mutta sitä jaetaan ilmaiseksi niille, jotka sitä haluavat käyttää laitteissaan.

Kuka tahansa voi ladata ohjelmointiympäristön Androidiin ja sitä kautta kehittää omia sovelluksia sille. Applikaatioiden ohjelmoimiseen käytetään Java-kieltä. Tämä ei kuitenkaan kuulu Oraclen Java-kieliin ja Oracle haastoikin Googlen oikeuteen Java-kieleen liittyvistä tekijänoikeuksista ja patenteista. Androidille on myös mahdollista ohjelmoida muilla kielillä, kuten C ja C++.

Android on tällä hetkellä johtava käyttöjärjestelmä älypuhelimissa. Maailmanlaajuisesti Androidin markkinaosuus 2011 vuoden lopussa oli 50,9% (Gartner). Useimmilta puhelinvalmistajista löytyy Android-käyttöjärjestelmän omaavia puhelimia, muun muassa Samsung, HTC, LG, Motorola, Sony, ZTE ja muutamia muita pienempiä valmistajia.

2.2 Tekninen esitys

Android-ohjelma voi koostua muutamista erilaisista komponenteista. Käsittelen lyhyesti Activyn, Intentin ja Servicesin tarkoituksen Android-ohjelmassa. Tässä työssä ei kuitenkaan käytetty näistä kolmesta kuin Activitya ja Intentiä. Vaikkakin Servicesiä olisi ollut mahdollista käyttää.

2.2.1 Activity

Activityt ovat periaatteessa Androidin eri näkymiä, tosin ne voivat olla myös pelkkiä dialogilaatikoita tai muuten huomaamattomia. Androidissa voi olla useampia Activityja käynnissä kerralla ja jopa yksittäinen ohjelma voi sisältää useita Activityja. RSS-lukijaohjelmassani on vain kaksi Activitya, päänäkö ja syötenäkö. Kaikki ohjelman Activityt pitää listata AndroidManifest.xml-tiedostoon.

Koska Activityja voi olla useampia käynnissä, löytyy niiltä erilaisia tiloja, joihin ne voivat mennä. Eli kun joku toinen Activity käynnistetään, aiempi Activity siirtyy taustalle, mutta se voidaan palauttaa sieltä vielä takaisin samassa tilassa, missä se oli ennen uuden käynnistämistä. Mutta koska lukemattomien Activityjen päällä pitäminen syö mahdottomasti muistia, Android voi itse tuhota taustalla olevia käyttämättömiä Activityja, muiden muistia tarvitsevien tieltä. Harvemmin tämä tuottaa mitään ongelmia, mutta suurempien ohjelmien kanssa pitää ottaa huomioon, että Android saattaa tuhota aiemmin käytetyn Activityn, mitä ei sitten voidakaan enää palauttaa samassa tilassa takaisin, missä se oli siitä lähdeettäessä.

2.2.2 Intent

Intentien avulla ohjelmoija voi kutsua muita osia ohjelmasta. Eli periaatteessa lähettää viestejä ohjelman muille osille. Tässä ohjelmassa Intentin avulla lähetetään päänäköstä tieto syötenäkymään, mitä syötettä halutaan tarkastella. Intentillä voi kutsua joko suoraan luokkaa oman ohjelman sisältä tai se voi lähettää hieman laajemman kutsun, minkä tyyppistä ohjelmaa se tarvitsee.

```
Intent viewIntent = new Intent("android.intent.action.VIEW",  
Uri.parse(linkList[position]));
```

Tässä ohjelmassa syötenäkö kutsuu tällä koodinpätkällä selainta, johon syöte avataan. AndroidManifest.xml-tiedostossa määritellään myös Intentit, eli millä tavoin kyseistä Activitya voidaan kutsua. Jos joku muu on määritellyt ohjelmassaan oleville Activityille Intentejä, pystyt myös kutsumaan jonkun toisen tekemää ohjelmanpätkää omasta ohjelmastasi.

2.2.3 Services

Services ovat taustalla pyöriviä tehtäviä, jotka eivät tarjoa käyttöliittymää. Tässä ohjelmassa tätä ominaisuutta ei käytetä. Periaatteessa tätä olisi voinut käyttää, laittamalla ohjelma aktiivisesti taustalla tarkastelemaan, koska syötteihin on tullut uutta luettavaa ja ilmoittanut niistä käyttäjälle. Riippuen käyttäjän lisäämien syötteiden määrästä, tämä ominaisuus saattaisi olla ongelmallinen, koska päivityksiä saattaisi jatkuvasti esiintyä jossakin tilatuista syötteistä.

2.3 Ohjelmointiympäristö

Android-ohjelman komponentit ja asetukset löytyvät AndroidManifest.xml-tiedostosta, joka luodaan automaattisesti, kun uusi projekti aloitetaan. Täällä on listattuna ohjelman Activityt, käytettävät Intentit, sekä tarvittavat luvat.

Esimerkiksi tässä ohjelmassa tarvitaan lupa käyttää internetiä, joten se tarvitsee ilmoittaa manifestissa. Käyttäjälle ilmoitetaan ohjelmaa asennettaessa, mitä kaikkia luvia ohjelma vaatii, tällöin laitteeseen asennettu ohjelma ei pysty salaa käyttäjän tietämättä käyttämään esimerkiksi internetiä.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fi.tamk.opnt"
    android:versionCode="1"
    android:versionName="1.0" >

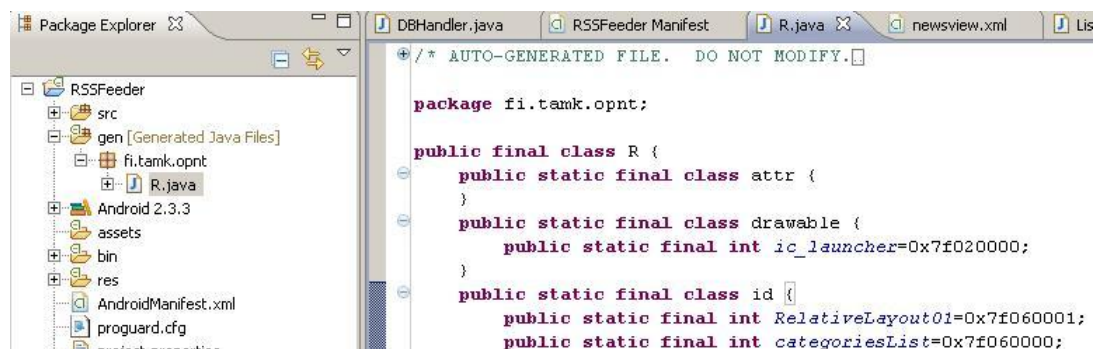
    <uses-sdk android:minSdkVersion="10" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application android:label="@string/mainTitle">
        <activity android:name=".RSSFeederActivity"
            android:label="@string/mainTitle">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <action android:name="android.intent.action.PICK" />
                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />
                <data android:scheme="http" />
                <data android:mimeType="application/rss+xml" />
            </intent-filter>
        </activity>
        <activity android:name=".NewsList"
            android:label="@string/listTitle" />
    </application>
</manifest>
```

Kuva 1: AndroidManifest.xml

Ohjelmasta löytyy myös automaattisesti luotu R.java-tiedosto. Tämä tiedosto sisältää referenssejä ohjelman eri resursseihin, kuten kuviin, ääniin, eri näkymiin jne. Käyttäjän ei itse tarvitse muokata tätä tiedostoa ikinä, eikä hän oikeastaan edes voi. Mikäli käyttäjä yrittää itse jotain kirjoittaa R.java-tiedostoon, se automaattisesti yliajetaan kun tiedostoa seuraavan kerran automaattisesti päivitetään.



```
/* AUTO-GENERATED FILE. DO NOT MODIFY.

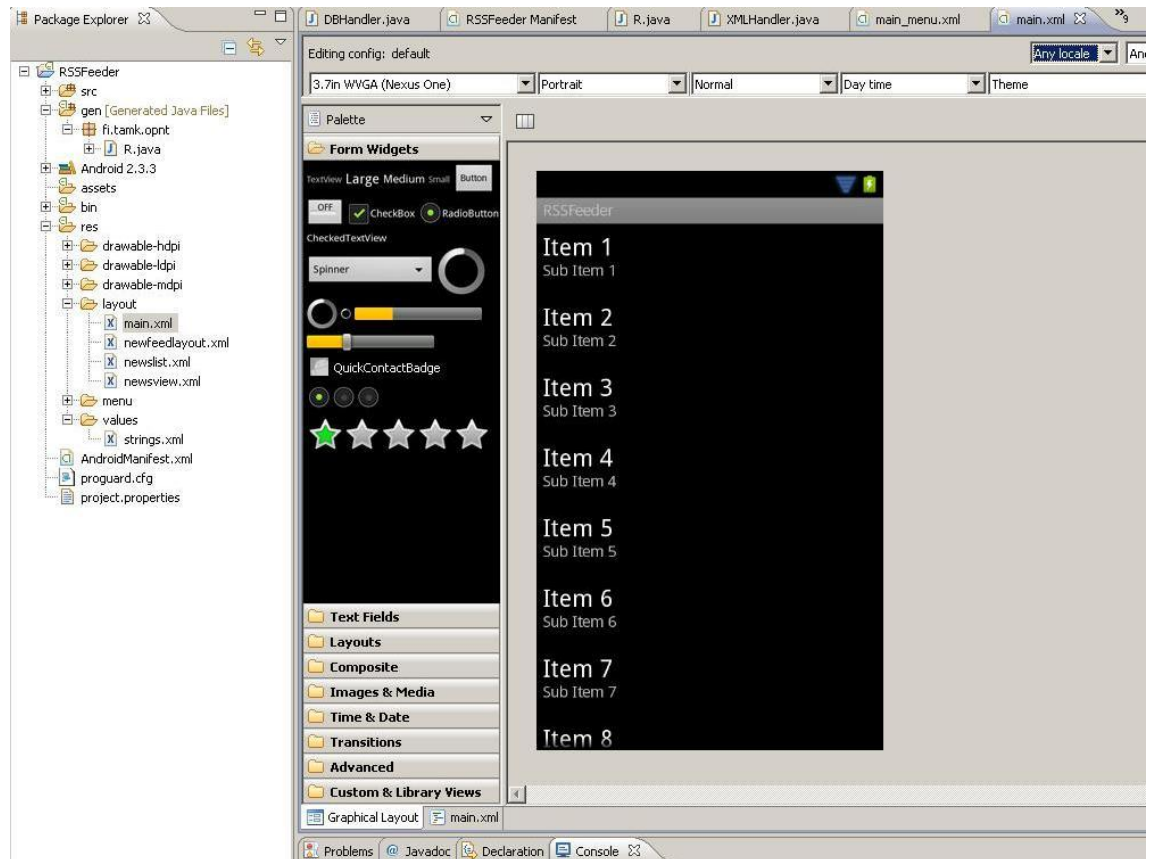
package fi.tamk.opnt;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class id {
        public static final int RelativeLayout01=0x7f060001;
        public static final int categoriesList=0x7f060000;
    }
}
```

Kuva 2: R.java

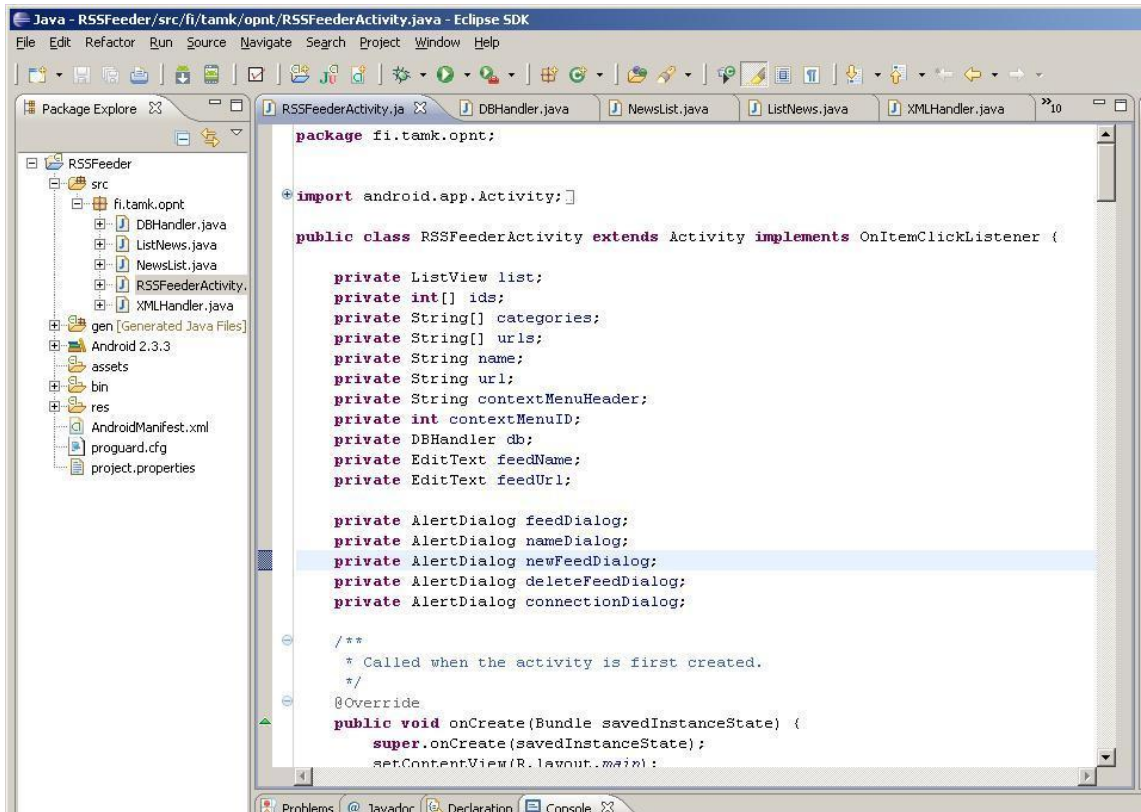
Graafiset ulkoasut löytyvät res->layout -polusta. Täällä ovat kaikille Activityille niiden graafiset näkymät. Tiedostot ovat XML:iä ja niitä voi muokata joko valmiin graafisen liittymän avulla, mistä voi drag&drop-tyyliin lisätä haluamiaan osia. Jokaiselle

tiedostolle löytyy myös XML-välilehti, missä voi suoraan XML-koodilla muokata ulkoasua. Erittäin kätevää on hyödyntää kumpaakin ominaisuutta haluamallaan tavalla.



Kuva 3: Näkymä main.xml

Src:stä löytyvät projektiin liittyvät java-tiedostot eli itse ohjelmointia vaativa osuus löytyy täältä. Luodun luokan voi tehdä Activityksi, laittamalla luokan perään ”extends Activity”.



```

package fi.tamk.opnt;

import android.app.Activity;

public class RSSFeederActivity extends Activity implements OnItemClickListener {

    private ListView list;
    private int[] ids;
    private String[] categories;
    private String[] urls;
    private String name;
    private String url;
    private String contextMenuHeader;
    private int contextMenuID;
    private DBHelper db;
    private EditText feedName;
    private EditText feedUrl;

    private AlertDialog feedDialog;
    private AlertDialog nameDialog;
    private AlertDialog newFeedDialog;
    private AlertDialog deleteFeedDialog;
    private AlertDialog connectionDialog;

    /**
     * Called when the activity is first created.
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}

```

Kuva 4: RSSFeederActivity.java

2.4 Arkkitehtuuri



Kuva 5: Androidin arkkitehtuuri. (<http://developer.android.com/guide/basics/what-is-android.html> 14.04.2012)

Kuvassa 5 näkyy kaikki Androidin arkkitehtuurin isoimmat komponentit.

Applications:

Kaikki korkeimman tason komponentit ovat täällä. Puhelin, selain, viestit jne.

Application Framework:

Tältä tasolta löytyy näkymät, taustalla pyörivät ohjelmat, hälytysten asettaminen ja paljon muuta. Kehittäjä pystyy hyödyntämään paljon valmiita komponentteja jotka sijaitsevat tällä tasolla ja myös antaa muiden hyödyntää itse tekemiään osia. Tämä antaa Android-ympäristölle paljon joustavuutta ja vapautta, kun kaikkea ei tarvitse aina tehdä itse alusta, vaan voi mahdollisuuksien mukaan käyttää hyväksi jo olemassa olevia ominaisuuksia.

Libraries:

Täällä on useita C/C++-kirjastoja, joita käyttäjät pystyvät hyödyntämään Application Frameworkin kautta. Muun muassa grafiikkamoottoreita ja SQL-kantaa.

Android Runtime:

Täällä sijaitsee Java-ohjelmointikielen peruskirjastot ja Dalvik Virtual Machine (DVM). Jokaisella Android-aplikaatiolla on oma DVM. DVM on suunniteltu mobiilialustoja ajatellen. Se pystyy pyörittämään useita instansseja ilman merkittävää muistin viemistä, mikä on erittäin kriittistä ottaen huomioon mobiilialustojen muistikapasiteetin ja mahdollisten instanssien määrän, mitä DVM:n pitää pystyä pyörittämään.

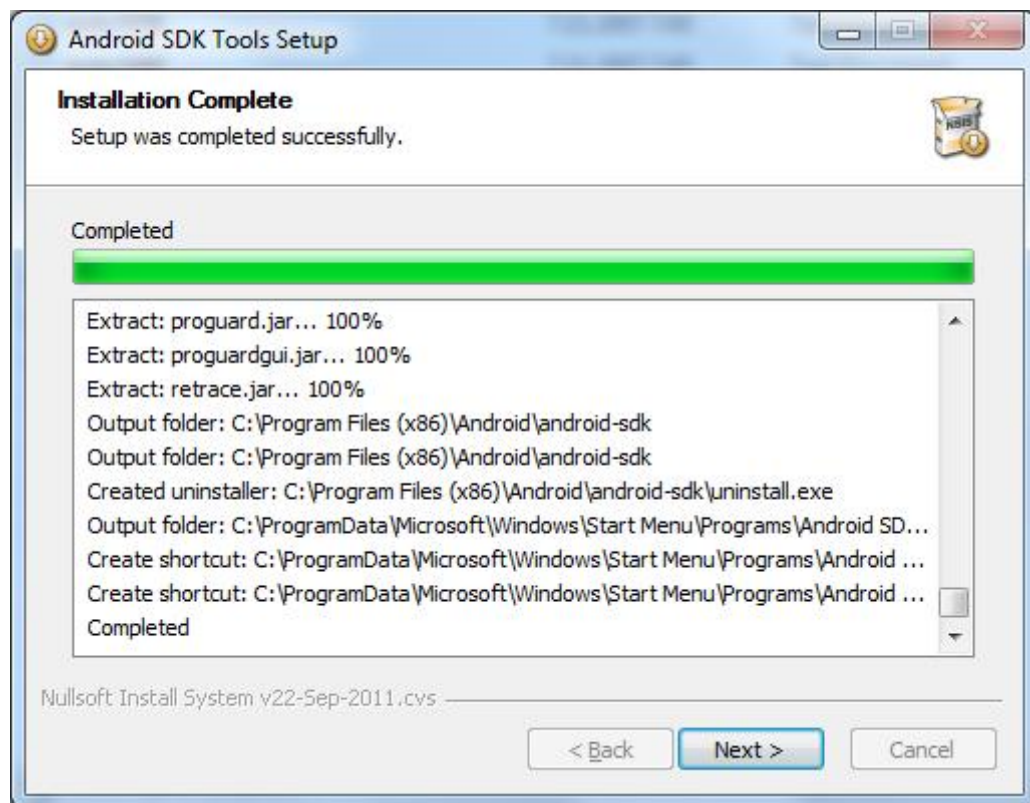
Linux Kernel:

Kernel toimii abstraktina tasona hardwaren ja ohjelmien välillä. Androidissa hyödynnetään Linuxin 2.6 versiota turvallisuudessa, muistin hallinnassa ja prosessien hallinnassa.

2.5 Työkalut

Androidille ohjelmoiminen onnistuu Eclipsen ja Android SDK:n yhdistelmällä, näitä voi käyttää sekä Windowsissa, Linuxissa että MacOS:ssa, joten käyttöjärjestelmä ei ole rajoite Androidille ohjelmoitaessa. Ympäristön asentaminen vaatii hieman temppuilua, mutta internetistä löytyy hyvin kattavat ohjeet, miten kullekin käyttöjärjestelmälle voi ympäristön asentaa. Tämän takia en yksityiskohtaisesti aio selittää jokaista vaihetta ympäristön asentamisessa, mutta käyn pintapuolisesti läpi mitä kaikkea vaaditaan. Selitän vain yleisimmän ja ei-niin-kokeineille Android-ohjelmoijille suositellun ympäristön asentamisen.

Ensimmäiseksi kannattaa varmistaa että koneelle on asennettu Java Development Kit (JDK). Niiden, jotka aloittelevat vasta Androidille ohjelmoimista kannattaa asentaa Eclipse. Kun JDK ja Eclipse on asennettuna koneelle, pitää ladata Android SDK. Tämä vaihe ei juuri Next-napin painamista enempää vaadi.

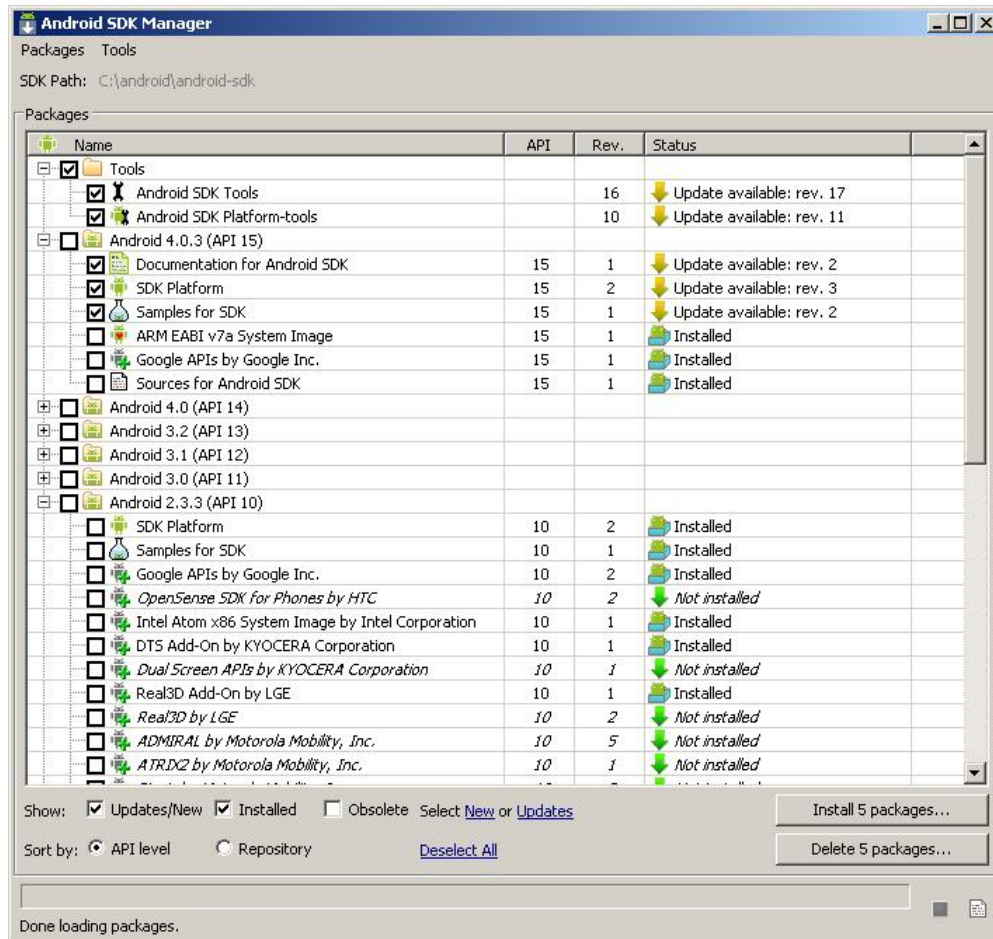


Kuva 6: Android SDK:n asennus.

Kun SDK on asennettu, käynnistetään Android SDK Manager, josta valitaan mitä kaikkia alustoja ja ominaisuuksia halutaan asentaa käytettäväksi ohjelmointiympäristöön. SDK Managerin voi käynnistää, joko polusta mihin sen asensi tai sitten Eclipsen kautta. Tarvittaessa SDK Managerista voi käydä asentamassa tai päivittämässä paketteja, jos haluaa. Joidekin pakettien asentaminen saattaa vaatia

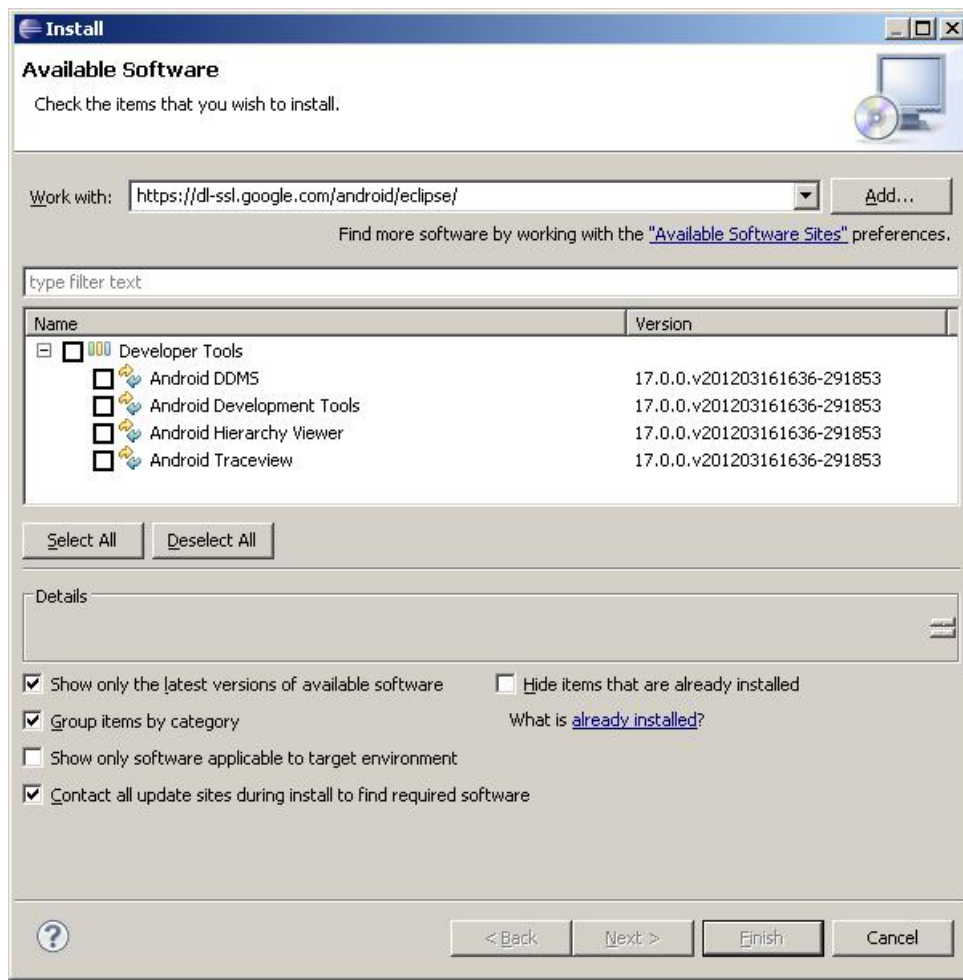
rekisteröitymistä sivustoille. Omassa tapauksessani näin oli ainakin Motorolaan liittyvien pakettien asentaminen, mutta jos ei erityistä tarvetta paketeille ole, ne voi jättää asentamatta.

Paketit mitkä ovat suositeltua asentaa ovat SDK Tools, SDK Platform-tools, SDK platform, Documentation, Samples ja USB Driver.



Kuva 7: Android SDK Manager

Suosittelavaa on myös asentaa Android Development Tools (ADT), jotka lisäävät helpottavia ominaisuuksia Android-ohjelmointiin. Esimerkiksi voit luoda paketin, millä voit asentaa ohjelman Android-laitteeseesi. ADT:n voi asentaa Eclipsen kautta suoraan menemällä Help -> Install New Software ja syöttämällä URL, mistä ADT löytyy.



Kuva 8: Android Development Toolsin asentaminen.

2.6 Ohjelman asentaminen puhelimeen

Käännetyt Android-ohjelmat ovat .apk-muodossa. Näitä tiedostoja voi asentaa puhelimeensa kahdella tavalla. Voit joko ladata Google Playsta ohjelman, joka löytää muistikortillasi olevat .apk-tiedostot ja pystyy asentamaan ne. Tai voit asentaa ohjelman SDK:n kautta.

Android SDK:n lisäksi sinulla pitää olla asennettuna Android USB –ajurit, jotka mahdollistavat SDK:n yhdistämisen puhelimeen USB:n kautta. Tämän jälkeen puhelimen asetuksista pitää sallia muiden kuin myytävien sovellusten asentaminen puhelimeen.

Kun tarvittavat ohjelmat on asennettuna ja asetukset kunnossa, tarvitsee vain avata komentoriivi ja kirjoittaa: *adb install tiedostopolku/tiedostonimi.apk*

Tämän jälkeen ohjelma on asennettuna puhelimeen.

3 WINDOWS PHONE

3.1 Yleistä

Windows Phone on Microsoftin käyttöjärjestelmä älypuhelimille. Se saapui markkinoille 2010 vuoden lopulla ja on seuraaja Windows Mobile –järjestelmälle, mikäli oli Microsoftin aiempi käyttöjärjestelmä puhelimille. Koska Windows Phone kehitettiin nopeasti, eivät Windows Mobilen ohjelmat toimi sillä. Windows Phonen käyttämä ohjelmointikieli on Visual Basic tai C#.

Windows Phonella on vielä suhteellisen pienet markkinat, osittain johtuen siitä että Microsoftin suurin kumppani Nokia on nyt vasta alkanut kunnolla saamaan puhelimiaan ulos, joten nousua todennäköisesti on luvassa. Microsoftin markkinaosuus älypuhelimissa maailmanlaajuisesti 2011 vuoden lopussa oli 1,9% eli merkittävästi Androidia pienempi. Älypuhelinikäyttöjärjestelmissä katsottuna Microsoft sijoittuu kuudenneksi.

Vaikka Nokia onkin Microsoftin suurin kumppani Windows Phonen levittämisessä, myös muut valmistajat jakavat Windows Phone –käyttöjärjestelmän omaavia puhelimia muun muassa Samsung, LG, HTC ja Dell.

3.2 Tekninen esitys

Aiemmin ohjelmoidessa Windows Phonelle, käyttäjän piti päättää kumpaa runkoa käyttää ohjelmassaan, Silverlightia vai XNA:ta. Molemmilta löytyy yhteisiä luokkia mitä pystyy käyttämään, mutta periaatteessa ero tulee visuaalisella puolella. Windows Phone 7.5:stä eteenpäin on ollut mahdollista käyttää molempia runkoja samassa ohjelmassa parin uuden luokan avulla, SharedGraphicsDeviceManager ja UIElementRenderer. Työssä tekemässäni ohjelmassa käytän vain Silverlightia.

3.2.1 Silverlight

Microsoftin Silverlight on ollut jo aiemmin käytössä Web-ohjelmoinnissa, mutta nyt se on tuotu myös mobiilialustalle. Silverlight onkin enemmän applikaatioiden ja muiden

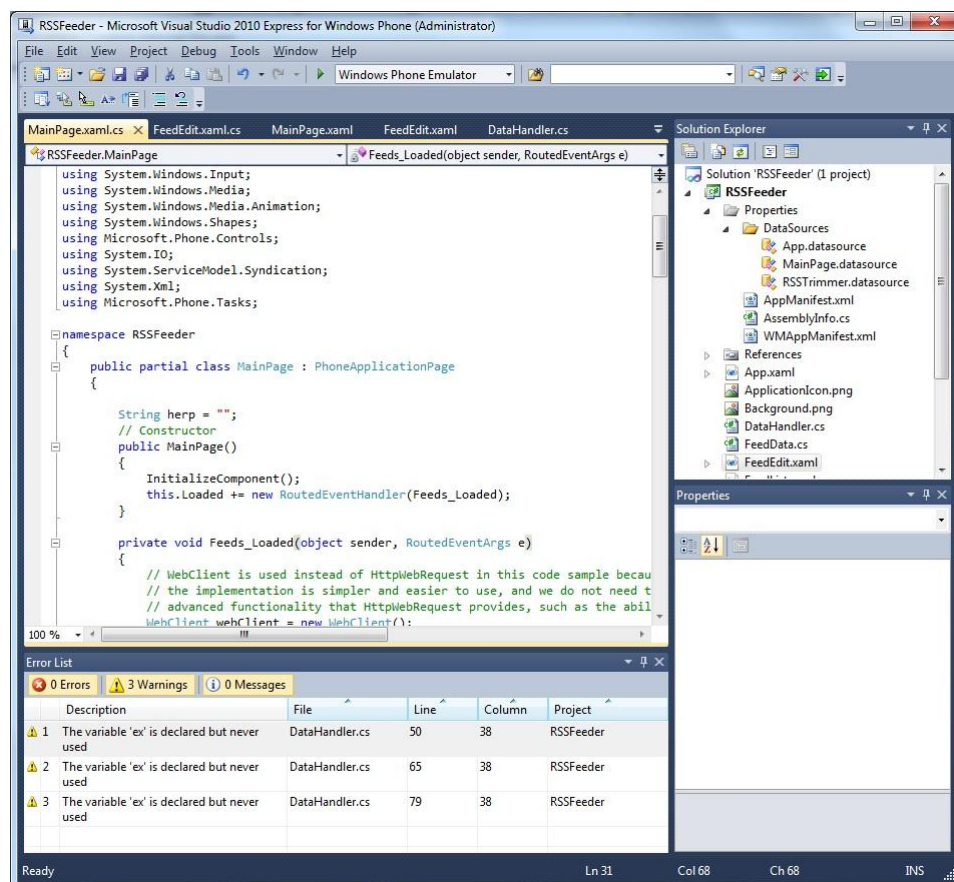
hyötyohjelmien ohjelmoimiseen suunnattu. Visuaaliselta osalta Silverlight hyödyntää XAML:ia, millä määritellään näkymän ohjaimet ja paneelit. Vaikka koodi tiedostojen takana voi toteuttaa logiikkaa, se pääasiassa käsittelee ohjaimista lähetettyjä tapahtumia.

3.2.2 XNA

XNA on suunniteltu lähinnä pelien kehittämisen kannalta. Se sisältää pääsyt 2D ja 3D – grafiikoihin, ääniin ja muuhun mediaan. Alunperin XNA:ta käytettiin Microsoftin Xboxille ohjelmointiin ja sillä voi myös suoraan PC:lle ohjelmoida.

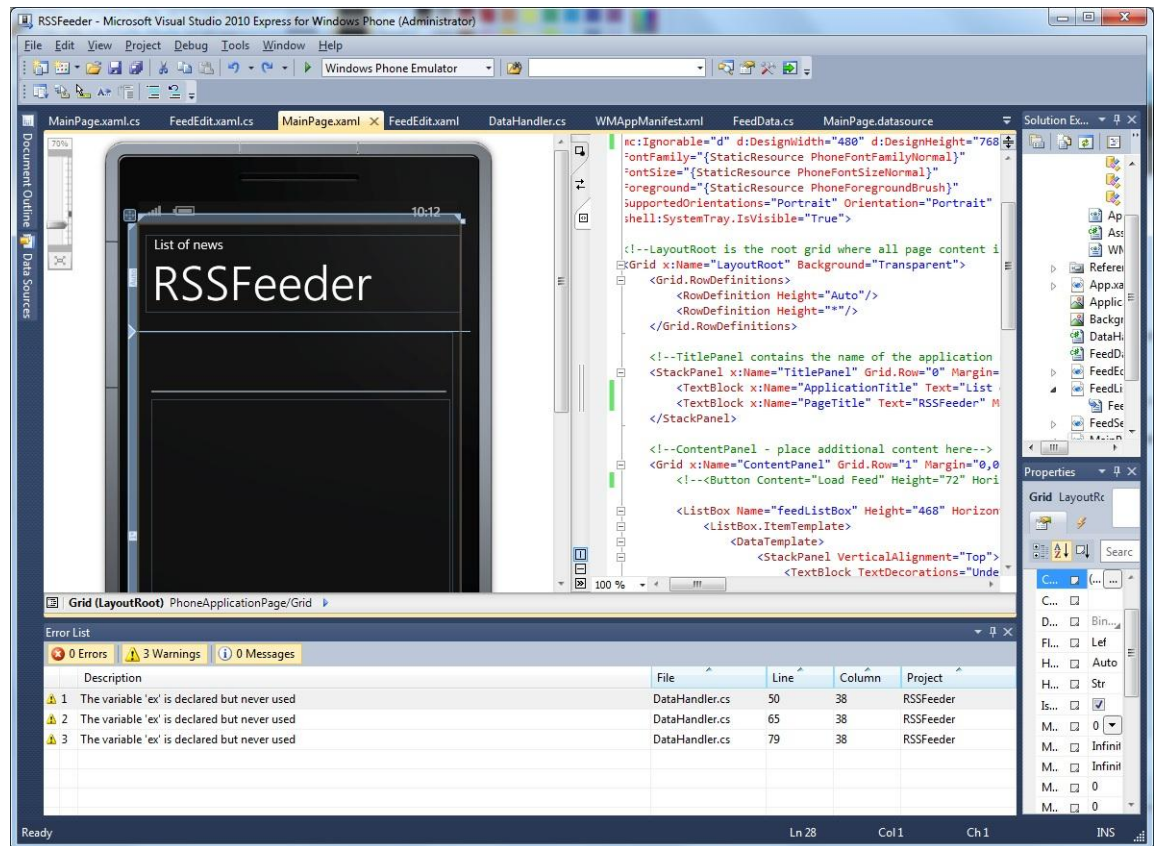
3.3 Ohjelmointiympäristö

Windows Phonelle ohjelmoitaessa, Androidin java-tiedostoja vastaavat .cs-tiedostot. Täällä toteutetaan ohjelman koodilogiikka. Jokaiselle näkymälle automaattisesti luodaan myös cs-tiedosto, johon kirjoitetaan kyseiseen näkymään liittyvä koodi.



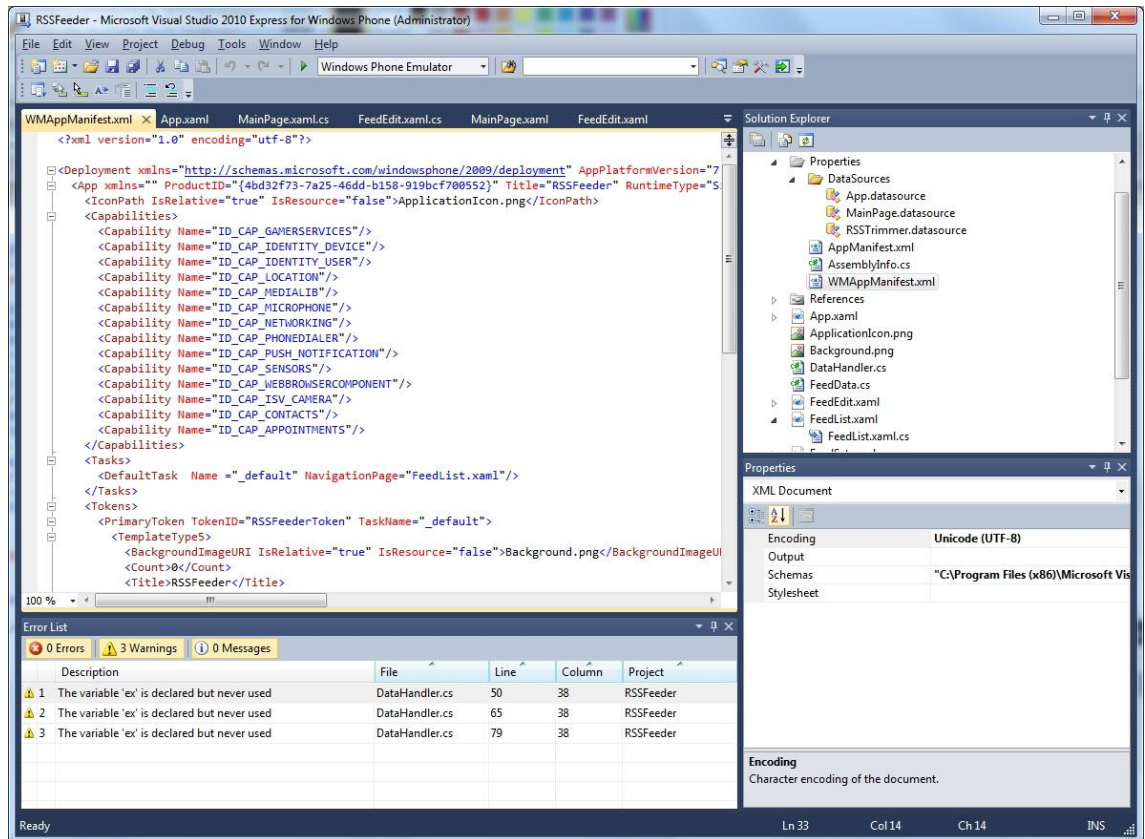
Kuva 9: MainPage.xaml.cs

Näkymiä voi muokata, joko graafisesti drag & drop –tekniikalla, tai ohjelmoimalla suoraan XAML:ää, joka on periaatteessa sama kuin XML, mutta ei yhtä laaja.



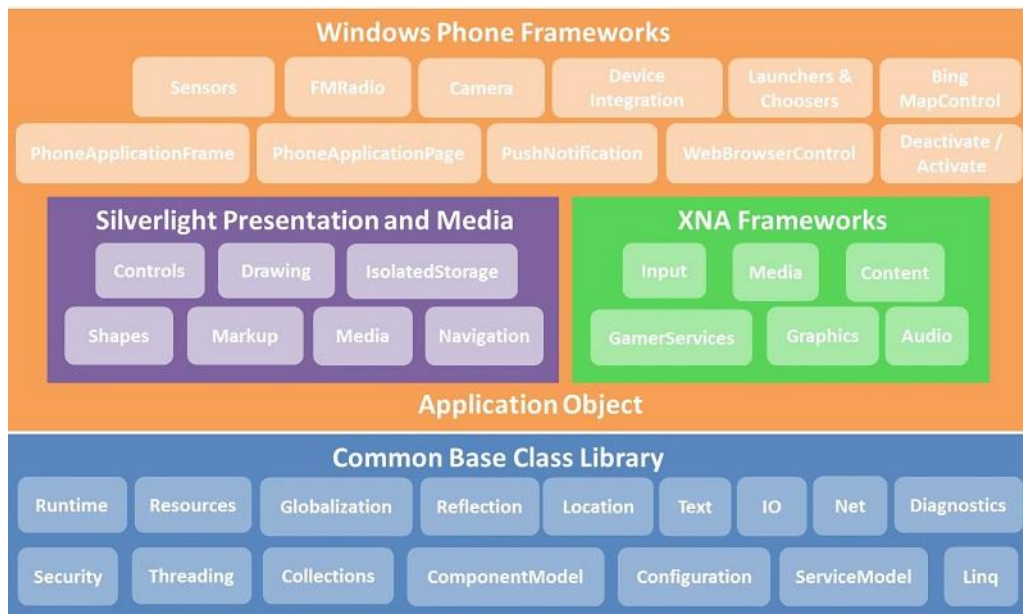
Kuva 10: Näkymä MainPage.xaml

Windows Phonesta löytyy myös WAppManifest, mikä on täysin vastaava kuin AndroidManifest. Siellä määritellään ohjelman ID ja mitä puhelimen ominaisuuksia ohjelma käyttää. Muun muassa media ja verkkoyhteys.



Kuva 11: WMAAppManifest.xml

3.4 Arkkitehtuuri

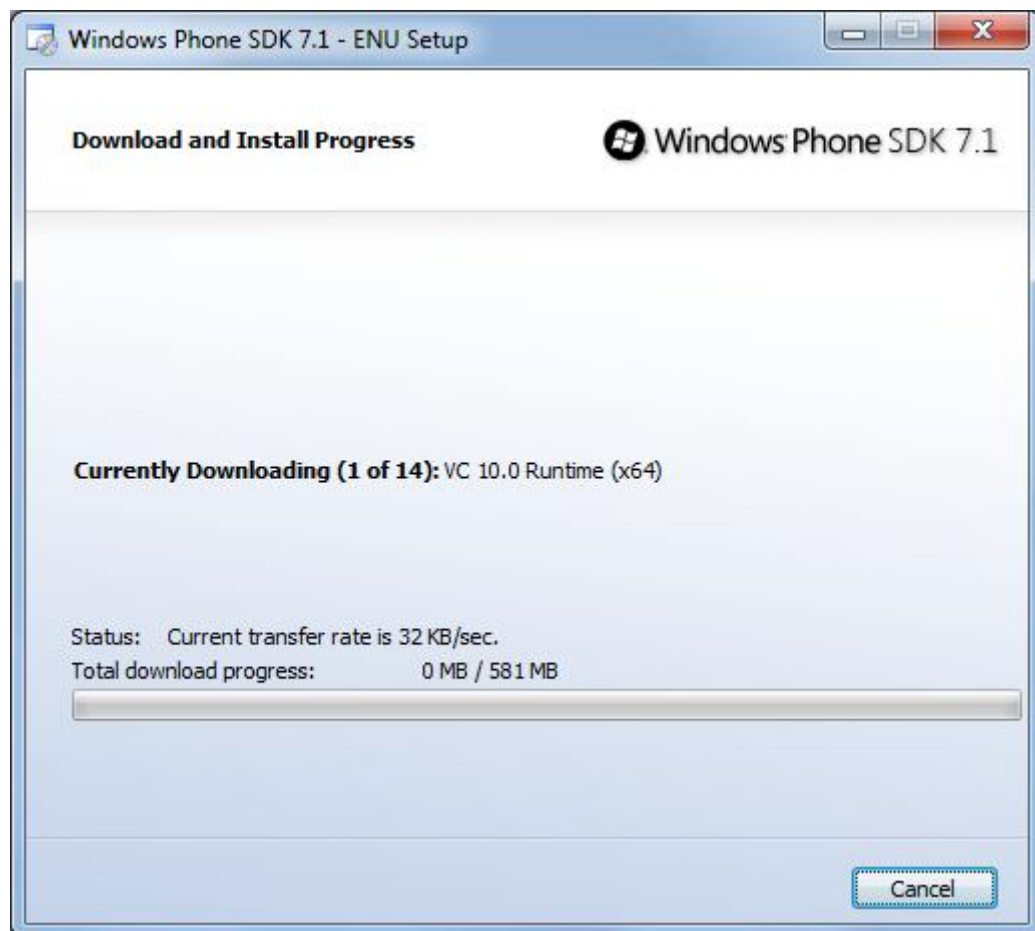


Kuva 12: Windows Phonien arkkitehtuuri. (<http://msdn.microsoft.com/en-us/library/ff402531%28v=vs.92%29.aspx> 05.05.2012)

Kuva 12 pääpiirteisesti esittää Windows Phonen rungon arkkitehtuurin. Common Base Class Library sisältää perusluokkia joita sekä Silverlight, että XNA voi hyödyntää. Ja sitten on Silverlight violetilla pohjalla ja XNA vihreällä pohjalla, joista löytyvät niiden hyödyntämät luokat.

3.5 Työkalut

Kehitysympäristön Windows Phonelle pystyy lataamaan suoraan Microsoftin sivuilta. Windows Phone SDK on huomattavasti yksinkertaisempi asentaa kuin Androidin ympäristö, periaatteessa riittää että lataa Visual Studio 2010 Express For Windows Phone -paketin ja painelee Next-näppäintä kunnes asennuspaketti ilmoittaa ympäristön asentuneen.



Kuva 13: Windows Phone SDK:n asennus.

Tämän jälkeen voit alkaa ohjelmoimaan Windows Phonelle. Huono puoli Windows Phone SDK:ssa on se, että se tarvitsee Windows Vistan tai Windows 7:n toimiakseen,

eikä se myöskään tue virtuaalikoneita. Joten tämä on hyvin rajoittava tekijä WP:lle ohjelmoitaessa.

3.6 Ohjelman asentaminen puhelimeen

Windows Phonen ohjelmien asennustiedostot ovat .xap-tiedostoja. Jotta ohjelman voi asentaa puhelimeen, pitää puhelin olla rekisteröity ja tähän vaaditaan App Hub –tunnus. Jokaisella App Hub –tunnuksella voi olla kolme rekisteröityä laitetta. App Hub –tunnuksen lisäksi tarvitaan Windows Live ID ja Zune-ohjelma.

Rekisteröidäksesi laitteen, sinun tarvitsee käynnistää se, liittää USB:lla tietokoneeseen ja käynnistää Zune. Tämän jälkeen pitää käynnistää Windows Phone SDK ja valita Windows Phone Developer Registration. Tämä avaa ikkunan, johon käyttäjän pitää syöttää Windows Live ID:nsä, mikä on yhdistetty App Hub –tunnukseen. Jos tunnus oli kelvollinen, laite on tämän jälkeen rekisteröity.

Rekisteröinnin jälkeen käyttäjä pystyy asentamaan ja ajamaan ohjelmia puhelimessaan. Tämä toimii periaatteessa samalla tavalla kuin laitteen rekisteröiminen. Erona, että käyttäjän pitää SDK:sta käynnistää Windows Developer Tools, valita Application Deployment ja tämän jälkeen aukeaa ikkuna, jossa kysytään asennetaanko ohjelma emulaattoriin vai laitteeseen ja XAP-tiedoston polku. Sitten vain painetaan Deploy ja ohjelman pitäisi asentua puhelimeen.

4 SOVELLUKSEN ESITTELY

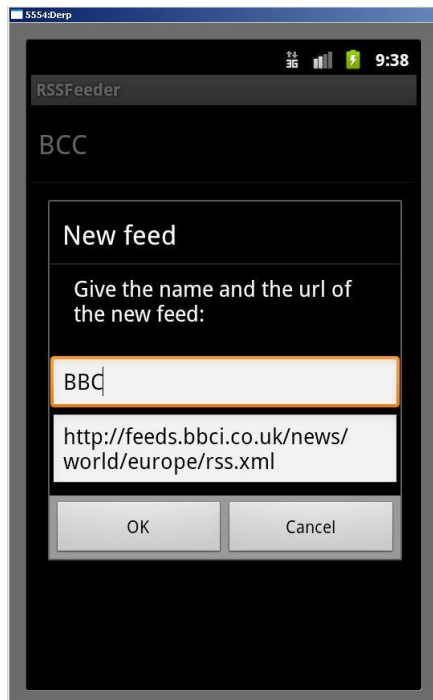
4.1 Yleisesti

Jotta pystyisin vertailemaan kunnolla Androidin ja Windows Phonen eroja, ohjelmoin kummallekin alustalle pienen ohjelman. Tässä tapauksessa RSS-lukijan. Ideana on, että käyttäjä pystyy lisäämään osoitteita, mistä ohjelma sitten hakee syötteen. Ensimmäisessä näkymässä näkyy käyttäjän lisäämät syötteen ja syötteen valitsemalla käyttäjä näkee syötteen listauksen.

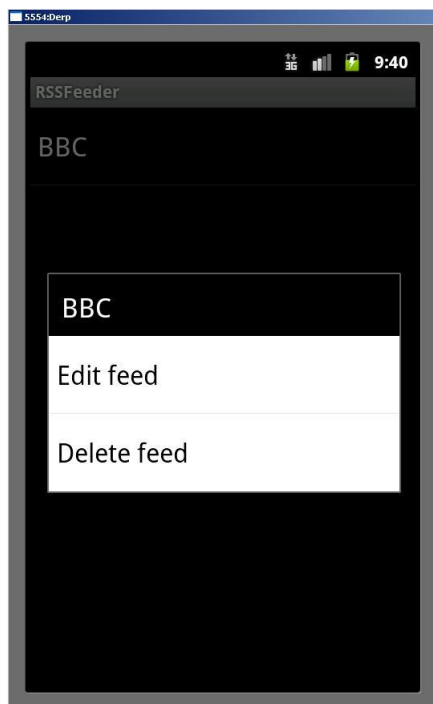
4.2 Käyttöliittymä

Käyttöliittymä on yksinkertaisin mahdollinen, ihan käyttömukavuudenkin takia. Painamalla nappia voit lisätä uusia syötteitä listaan. Kun listattua syötettä painaa kerran, ohjelma vie uuteen listaan, joka listaa kyseisestä syötteestä löytyvien uutisten otsikot. Painamalla back-nappia, pääsee palaamaan takasin syötelistaan. Painamalla uutista ohjelma avaa selaimen ja sitä kautta näyttää uutisen sisällön. Syötteitä pystyy myös poistamaan ja muokkaamaan tarvittaessa. Ohjelma palauttaa virheen, mikäli internet-yhteyttä ei pystytä muodostamaan.

4.2.1 Käyttöliittymä Androidissa



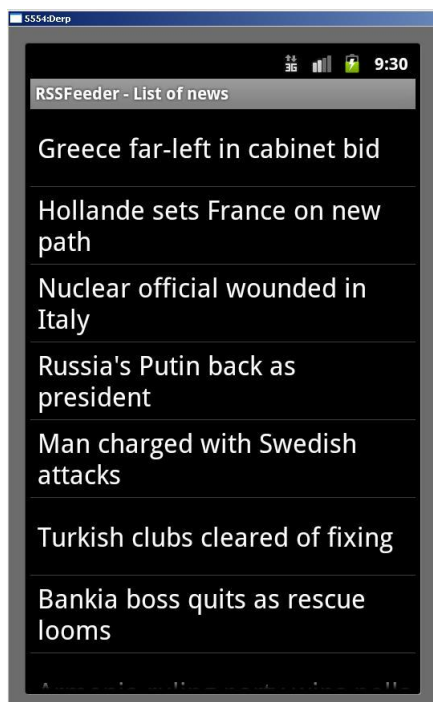
Kuva 14: Uuden syötteen lisääminen.



Kuva 15: Syötteen muokkaaminen ja poistaminen.



Kuva 16: Syötteen muokkaaminen.



Kuva 17: Syötteen sisältö.



Kuva 18: Utinen selaimessa.

4.2.2 Käyttöliittymä Windows Phinessa



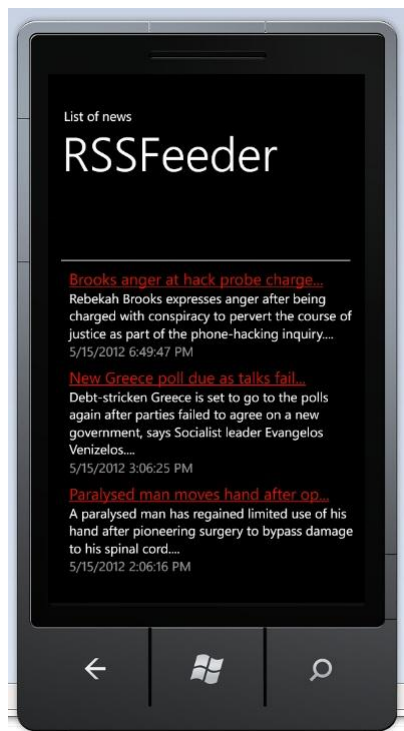
Kuva 19: Syötenäkymä.



Kuva 20: Uuden syötteen lisääminen.



Kuva 21: Syötteen muokkaaminen ja poistaminen.



Kuva 22: Syötteen sisältö.



Kuva 23: Uutinen selaimessa.

4.3 Tekniikat

4.3.1 Android

Androidissa käytin kahta Activitya, RSSFeederActivity, joka on ohjelman päänäköymä, mihin listataan käyttäjän lisäämät syötteen ja NewsListiä, joka listaa syötteen sisällön.

Erikseen löytyy luokka, johon kerätään taulukoihin syötteen sisältämät tiedot: otsikko, luokka ja päivämäärä. Sekä luokka, joka parsii XML-koodista nämä tiedot ulos ja luokka, jolla kontrolloidaan tietokantaa, johon syötteen on tallennettu. Tietokantana Android käyttää SQLiteä. NewsList-näkymässä linkkiä painettaessa, Android lähettää signaalin, jolla etsitään selainta, mihin linkki avataan.

Kun syötettä painaa päänäköymässä pidempään, aukeaa valintalista sille, haluaako poistaa tai muokata syötettä, kuva 15. Poistossa syöte poistetaan myös lopullisesti kannasta. Syötteestä pystyy muokkaamaan nimeä ja osoitetta, kuva 16.

4.3.2 Windows Phone

Koska kyseessä ei ole mitenkään grafiikkaa hyödyntävä sovellus, käytin Windows Phonessa pelkästään Silverlightia. Eroavaisuuksien takia, käyttöliittymässä joutui tekemään pieniä eroja Androidiin verrattessa, pohjimmiltaan idea on kuitenkin sama.

Windows Phonen erona tekniikassa oli, että menu-näppäimen puuttuessa, käytin ApplicationBaria, johon laitoin napit uuden syötteen lisäämiseen ja vanhan muokkaamiseen.

En onnistunut löytämään tekniikkaa, jolla olisin saanut avattua mahdollisuuden muokkaamiseen ja poistamiseen painamalla syötettä pidempään. Joten kun käyttäjä valitsee syötteen muokkaamisen ja tämän jälkeen painaa jotain syötteistä, aukeaa syötteen muokkausnäköymä, kuva 21. Täältä voi myös poistaa valitun syötteen.

Syötteiden tallentamiseen käytin IsolatedStorageSettingsiä. Periaatteessa tallensin listan syötteistä ohjelmalle valittuun alueeseen puhelimen muistista, mistä listan pystyi halutessa myöhemmin hakemaan.

Myös Windows Phonessa linkki avataan erilliseen selaimen.

5 SOVELLUKSEN KEHITYS

5.1 Kehitystyö

Tämä osio ei paljoa vaatinut. Ideana oli yksinkertainen lista halutuista syötteistä. Joten näkymät oli kaksi erilaista listaa, joissa toisessa on syötteet ja toisessa syötteen sisältö. Eniten päänvaivaa aiheutti tilattujen tietojen tallentaminen puhelimeen, ilman että tiedot katosivat ohjelman sulkeuduttua. Tämä ratkaistiin lopulta käyttämällä Androidissa SQLiteä.

Jos ohjelmia haluaisi välttämättä jatkokehittää, voisi työkseen ottaa sen, että ohjelma muistaa avatut syötteet ja joskus tallennettuina olleet syötteet ja ne voisi lisätä tilaukseen nappia painamalla. Myös olisi mahdollista laittaa ohjelma tarkastelemaan tietyin väliajoin, koska syötteeseen on tullut jotain uutta. Tämä ominaisuus tosin saattaisi käydä nopeasti ikäväksi, jos käyttäjällä on useampia syötteitä seurattavana.

5.2 Toteutus

Toteutus lähti sillä, että ensin luotiin näkymät syötteille ja niiden sisällölle. Tämän jälkeen luotiin painikkeet, joilla syötteitä voi lisätä ja että sisällön pystyy avaamaan selaimeen. Aluksi syötteet tallennettiin suoraan muistiin väliaikaiseen muistiin.

Kun tämä saatiin toimimaan, kehitettiin tapa tallentaa syötteet pysyvästi puhelimeen.

5.3 Testaus

Ohjelmat testattiin niin, että kaikki luodut painikkeet kokeiltiin, että ne tekivät mitä halutaan. Tallennettujen syötteiden piti löytyä vielä sen jälkeen, kun ohjelma tai koko emulaattori käynnistettiin uudestaan ja poistamisen jälkeen syötettä ei saanut enää löytyä. Myös että syötteiden sisältö avautui selaimeen niitä painettaessa ja että ohjelma herjasi, jos internet-yhteyttä ei löydy.

Paljoa mahdollisia virhetilanteita ei tässä ohjelmassa pääse syntymään.

5.4 Rajoitukset

Mitään erikoisempia rajoituksia ei ohjelma aiheuta. Android-versio on suunniteltu 2.3.3:lle, mutta pitäisi toimia muillakin alustoilla. Ja Windows Phone –versio on suunniteltu 7.1:lle. Niin kauan kun puhelimesta löytyy yhteys internetiin ja selain, ohjelman pitäisi toimia moitteettomasti. Tietty pieninäyttöisillä puhelimilla lukeminen voi olla erittäin epämiellyttävää.

5.5 Sovelluksen markkinointi

5.5.1 Google Play

Googella on Google Play, entinen Android Market, mihin kuka tahansa pystyy oman Android-sovelluksensa laittamaan jakoon, ilmaiseksi tai haluttua maksua vastaan. Mikäli sovellus laitetaan maksua vastaan jakoon, myyjä saa siitä 70% ja 30% menee sovelluksen tapahtumakuluihin. Jotta tuotteita pääsee edes myymään Google Playhin, tarvitsee suorittaa kertaluonteinen \$25 maksu rekisteröityessä.

5.5.2 Windows Marketplace

Microsoftilla on Windows Marketplace, mikä on vastaavanlainen paikka, mihin kehittäjät voivat laittaa ohjelmiaan jakoon. Provision jako on samanlainen, jos haluaa myydä ohjelmaansa Windows Marketplaceissa. 70% tulee kehittäjälle itselleen ja 30% menee Microsoftille. Rekisteröityminen maksaa \$99 vuodessa, mutta opiskelijat voivat rekisteröityä ilmaiseksi Dreamsparkin kautta. Windows Marketplaceissa on rajoitus, että vain 100 ilmaista ohjelmaa pystyy jakamaan vuodessa, jokaista yli 100 menevää ohjelmaa kohden, kehittäjän pitää maksaa \$20.

Ohjelmat myös tarkastetaan aina ennen kuin ne lisätään jaettavaksi. Näin varmistetaan, ettei jakoon pääse epäilyttäviä tai muuten laitteita vahingoittavia ohjelmia. Mikäli ohjelmaa ei hyväksytä jakoon, saa kehittäjä tästä ilmoituksen.

6 POHDINTA

Osiossa pohditaan molempien alustojen hyviä ja huonoja puolia sekä suurimpia eroavaisuuksia. Mikä on kummankin alustan vahvuus ja miltä tulevaisuus näyttää. Ja kummalla alustalla on paremmat markkinointimahdollisuudet.

6.1 Ohjelmointi

Jos lähdetään ihan ohjelmointiympäristöstä liikenteeseen, Windows Phone vie helposti voiton ympäristön asentamisen helppoudella. Ei tarvitse latailla erikseen ympäristöä, SDK:ta ja sitten vielä säätää emulaattori toimimaan, vaan kaikki hoituu yhdellä tiedostolla ja itse ei tarvi juuri mitään muuta tehdä kuin määritellä polku mihin ympäristön haluaa asentaa.

Mutta vaikka ympäristön asentaminen on huomattavasti kätevämpää Windows Phonella, häviää se siinä, että käyttäjällä tarvitsee olla Windows Vista tai Windows 7. Monelta varmasti löytyy kyseinen käyttöjärjestelmä valmiiksi, mutta paljon on myös ihmisiä, jotka ovat vielä pysytelleet Windows XP:ssä tai suosivat Linuxia tai OSX:ää. Tässä Android vie ehdottomasti voiton. Ohjelmointiympäristö toimii kaikissa yleisimmissä käyttöjärjestelmissä, joten se ei rajoita ihmisiä tekemästä ohjelmia Android-laitteilleen.

Ohjelmointitekniikka on kohtuullisen samanlaista molemmilla alustoilla. Graafinen puoli hoidetaan XML:llä tai Windows Phonen tapauksessa XAML:llä. Tämän lisäksi molemmilta löytyy manifesti, mihin määritellään ohjelman ominaisuuksia ja sitten itse koodit sisältävät tiedostot. Mitään merkittäviä eroavaisuuksia ei tältä osa-alueelta tunnu löytyvän.

Mitä ohjelmointikielen tulee, Android on pääosin pelkkää Javaa, mikä varmasti suurelle osalle ohjelmoijista on enemmän tai vähemmän tuttua. Java kun tuntuu vielä olevan hyvin hallitsevassa asemassa.

Mutta Windows Phone antaa vapauden valita Visual Basicin ja C#:n väliltä, mikä on varsin mukava asia. Jos jostain syystä toinen kieli tuntuu sekavalta tai haastavalta, voi ohjelmoida toisella.

Visuaalisen puolen molemmat alustat hoitavat hyvin samanlaisella tavalla. Drag & drop –tekniikalla, mutta ulkonäköä on mahdollista muuttaa myös XML-koodilla.

6.2 Markkinat

TAULUKKO 1: Markettien vertailu (14.05.2012).

	Google Play	Windows Phone Marketplace
Rekisteröityminen	\$25 kerran	\$99 / vuosi
Applikaatioita marketissa	440 645	82 626
Ilmaisia applikaatioita	72,4%	68,5%
Kehittäjän osuus hinnasta	70%	70%

Ohjelman markkinointi on huomattavasti miellyttävämpää Androidilla kuin Windows Phonella, tosin Androidillakin on omat ongelmansa. Android päihittää Windows Phonen helposti rekisteröitymisessä. Siinä missä Androidin kehittäjäksi rekisteröityminen on kertamaksullinen \$25, maksaa Windows Phonelle rekisteröityminen \$99 vuodessa.

Windows Phone Marketplacessa on myös rajoitus rekisteröityneillekin käyttäjille, että he saavat lähettää 100 ilmaista applikaatiota vuoden aikana. Yli sadan menevät ilmaiset applikaatiot maksavat kehittäjälle \$20 kappale. Tosin Windows Phone Marketplacen eri maksut selittyvät osittain sillä, että jokainen ohjelma tarkastetaan, ennen sen lisäämistä Marketplaceen. Google Play sen sijaan ei tarkista ohjelmia mitenkään, vaan on käyttäjien omalla vastuulla mitä he lataavat ja asentavat puhelimiinsa.

Ohjelmien tarkastaminen ja kohtuullisen suuri rekisteröitymismaksu, antavat vähän enemmän takuuta Windows Phonen applikaatioiden laadukkuudesta tai ainakin tiputtavat haitallisimmat ohjelmat ja jonkin verran luokattomien ohjelmien kehittäjiä pois.

Kummatkin marketit ottavat saman osuuden maksullisista ohjelmista. Jos kehittäjä jakaa ohjelmaansa maksua vastaan, saa hän myynnistä 70% ja 30% menee marketille.

6.3 Yhteenveto

Android tuntuu olevan enemmän mobiiliohjelmoijien leikkikenttä. Vaikka ympäristön asentaminen pientä säätöä vaatiikin, on ohjelmointi hyvin avointa ja oman ohjelman

siirtäminen puhelimeen tai jakoon erittäin vaivatonta. Tämä leikkikenttämäisyys tosin aiheuttaa sen, että Androidille pääsee paljon kelvottomia ohjelmia, joko tahallisesti vai vahingossa tehtyinä.

Windows Phone on hieman Androidia valvotumpi ympäristö. Puhelimeen ei saa tuosta vaan omia ohjelmiaan asennella ja kaiken pitää olla rekisteröitynyttä. Marketin ohjelmia valvotaan ja hinnat ovat sen mukaiset.

Mutta onko se sitten etu vai haitta, että ohjelmien jakamisesta on tehty niin hankalaa? Sanoisin että ei. Se, että vaaditaan rekisteröityminen, tarkistetaan jokainen jakoon laitettu ohjelma ja lyödään suhteellisen iso vuosimaksu päälle, varmistaa vain sen, että kelvottomimmat ohjelmat pysyvät poissa marketista. He, jotka oikeasti tekevät laadukkaan ohjelman, maksavat kyllä saadakseen ohjelmansa jakoon. Oli se sitten \$25 kerta tai \$99 vuodessa.

Kun taulukkoa katsoo, niin ilmaisten applikaatioiden suhteellinen määrä marketissa on lähes sama molemmissa marketeissa, joten Windows Phone Marketplacen huomattavampi rekisteröitymismaksu, ei näyttäisi vaikuttavan tavalla, että suurin osa applikaatioista olisi maksullisia. Myös koska Windows Phone Marketplacessa on tällä hetkellä huomattavasti vähemmän applikaatioita kuin Google Playssa, on mahdollisesti paljon helpompaa saada oma applikaationsa näkyviin, varsinkin jos se on laadukas. Pitää kuitenkin ottaa huomioon sekin, että potentiaalisten asiakkaiden määrä on huomattavasti suurempi myös Androidilla kuin Windows Phonella.

Viime aikoina on jonkin verran ollut uutisia myös Androidin tietoturvaongelmista, jotka ovat tietenkin ihan selvä juttu ottaen huomioon ympäristön avoimuuden. Tämä pistää miettimään, että yritykset ei välttämättä haluaisi jakaa työntekijöilleen kovin tietoturvatonta puhelimia, joten Windows Phone saattaisi olla vahvoilla yrityspuhelimenä.

Muutenkin yrityksissä paljon käytetään Microsoftin ohjelmistoja, joten jos Microsoft viimein saa mobiilialustansa lyömään enemmän läpi, olisi se looginen ratkaisu yrityksiltä ottaa Windows Phone –puhelimia työntekijöidensä käyttöön. Kokisin todennäköisimpänä, että Windows Phone alkaisi yleistymään enemmän yrityspuolella ja siitä ehkä sitten hitaasti myös yksityiseen käyttöön.

Suomalaisesta näkökulmasta helposti katsoo että Windows Phonella menee surkeasti, koska Nokia on laskenut pitkään alamäkeä, kun se aloitti yhteistyön Microsoftin kanssa.

Mutta jos unohdetaan Nokia pois kuvioista, niin Windows Phonella tuntuisi menevän varsin hyvin. Jos mietitään Windows Mobilea, niin melko harva valmistaja teki näitä puhelimia ja vielä vähemmän ainakaan itse törmäsin niihin kaupassa. Nykyään taas Windows Phone on paljon otsikoissa ja kaupoistakin eri valmistajien puhelimia kyseisellä käyttöjärjestelmällä löytyy. Pientä nousua on siis näkyvissä tällä saralla.

Paljon Windows Phonen menestykseen varmasti vaikuttaa se, miten paljon Microsoft hyödyntää laajaa ohjelmistotarjontaansa tällä alustalla. Ja tuoko se esimerkiksi jotain ylimääräistä hyötyä Xbox-pelikonsolin omistajille, mitä Android ei voisi tarjota.

On kuitenkin vaikea arvioida nouseeko Windows Phone täysin kilpailemaan Androidin kanssa. Tämä riippuu paljon siitä kuinka hyvin Windows Phone lopulta lyö läpi käyttäjille ja mitä tekee Android-leiri. Älypuhelin-alustat alkavat hyvin paljon muistuttamaan toisiaan, joten kilpailua tullaan todennäköisesti käymään enemmän tarjolla olevien isojen ohjelmistojen ja puhelimien fyysisten ominaisuuksien avulla.

LÄHTEET

AppBrain. Number of available Android applications. Luettu 14.05.2012
<http://www.appbrain.com/stats/number-of-android-apps>

Windows Phone AppList. Windows Phone Marketplace Statistics. Luettu 14.05.2012
<http://www.windowsphoneapplist.com/en/stats/>

MSDN. Use the Application Deployment Tool for Windows Phone. Luettu 14.05.2012
<http://msdn.microsoft.com/en-us/library/ff769512%28v=vs.92%29.aspx>

MSDN. Register Your Phone for Development. Luettu 14.05.2012
<http://msdn.microsoft.com/en-us/library/ff769508%28v=vs.92%29.aspx>

Karch, Marziah. What Is Google Android? Luettu 21.03.2012
http://google.about.com/od/socialtoolsfromgoogle/p/android_what_is.htm

Open Handset Alliance. FAQ. Luettu 21.03.2012
http://www.openhandsetalliance.com/oha_faq.html

Android developers. What is Android? Luettu 03.04.2012
<http://developer.android.com/guide/basics/what-is-android.html>

MSDN, Application Platform Overview for Windows Phone. Luettu 18.04.2012
<http://msdn.microsoft.com/en-us/library/ff402531%28v=vs.92%29.aspx>

Vogel, Lars 2012. Android Development Tutorial. Luettu 16.04.2012
<http://www.vogella.de/articles/Android/article.html>

Gartner 2012. Luettu 14.05.2012
<http://www.gartner.com/it/page.jsp?id=1924314>

Petzold, Charles 2010. Programming Windows 7. Washington: Microsoft Press.