OULU UNIVERSITY OF
APPLIED SCIENCES

Jari Järvi

COMPARISON OF ANDROID 2, MAEMO 5 AND WINDOWS PHONE 7 FROM SOFTWARE DEVELOPER'S POINT OF VIEW

COMPARISON OF ANDROID 2, MAEMO 5 AND WINDOWS PHONE 7 FROM SOFTWARE DEVELOPER'S POINT OF VIEW

Jari Järvi
Bachelor's thesis
September 2011
Information Technology
and Telecommunications
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology and Telecommunications, Software Development

---

Author: Jari Järvi
Title of Bachelor's thesis: COMPARISON OF ANDROID 2, MAEMO 5 AND WINDOWS PHONE 7 FROM SOFTWARE DEVELOPER'S POINT OF VIEW
Supervisors: Veijo Väisänen (OUAS) and Aarne Taube (Nokia Corporation)
Term and year of completion: Autumn 2011
Number of pages: 47

ABSTRACT

This thesis was made for Nokia Corporation during summer 2011. The objective was to compare Android 2, Maemo 5 and Windows Phone 7 mobile platforms from a software developer's point of view.

Each one of these platforms has a good reason for being in this comparison. The market share of Android 2 devices has grown rapidly during the past few years and it has become a very popular mobile platform. Maemo 5 is somewhat old but it can take advantage of the newest features of the Qt library such as Qt Quick. Windows Phone 7 for its part is interesting because it will be the main platform of the Nokia smartphones in the future.

This thesis aims to give an overview of the platforms and the software development environments as well as present experiences from the process of developing a simple Flickr application for each platform.

Keywords: Android, Maemo, Windows Phone 7, Qt, QML, Silverlight, Java

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikka, ohjelmistojen kehitys

Tekijä: Jari Järvi
Opinnäytetyön nimi: COMPARISON OF ANDROID 2, MAEMO 5 AND WINDOWS PHONE 7 FROM SOFTWARE DEVELOPER'S POINT OF VIEW
Työn ohjaajat: Veijo Väisänen (OAMK) ja Aarne Taube (Nokia Oyj)
Työn valmistumislukukausi ja -vuosi: Syksy 2011
Sivumäärä: 47

TIIVISTELMÄ

Tämä opinnäytetyö on tehty Nokia Oyj:lle kesällä 2011. Tehtävänä oli vertailla Android 2, Maemo 5 ja Windows Phone 7 mobiilialustoja ohjelmistokehittäjän näkökulmasta.

Jokaisella näistä mobiilialustoista on hyvä syy olla mukana tässä vertailussa. Android-laitteiden määrä on lisääntynyt viime vuosina nopeasti ja se on saavuttanut suuren markkinaosuuden. Maemo 5 on vanhahko alusta, mutta sillä voidaan hyödyntää uusimpia Qt-kirjaston ominaisuuksia. Windows Phone 7 puolestaan tulee tulevaisuudessa olemaan Nokian älypuhelinten pääasiallinen käyttöjärjestelmä.

Tavoitteena on antaa yleiskuva alustoista ja ohjelmistokehitysympäristöistä sekä esittää kokemuksia yksinkertaisen Flickr-sovelluksen kehitysprosessista jokaiselle alustalle.

Asiasanat: Android, Maemo, Windows Phone 7, Qt, QML, Silverlight, Java

PREFACE

I would like to thank Aarne Taube from Nokia Corporation for this opportunity to study this interesting and current topic.

Oulu 7 September 2011

Jari Järvi

CONTENTS

SYMBOLS AND ABBREVIATIONS

ADT	Android Development Tools
A-GPS	Assisted Global Positioning System
AVD	Android Virtual Device
GIMP	GNU Image Manipulation Program
GNU GPL	GNU General Public License
GNU LGPL	GNU Lesser General Public License
GPS	Global Positioning System
GPU	Graphics Processing Unit
GTK+	GIMP Toolkit
HSPA	High-Speed Packet Access
IDE	Integrated Development Environment
JDK	Java Development Kit
NFC	Near Field Communications
SDK	Software Development Kit
SMS	Short Message Service
RAM	Random Access Memory
RISC	Reduced Instruction Set Computing
QML	Qt Meta-object Language
WLAN	Wireless Local Area Network
WVGA	Wide Video Graphics Array (800x480 pixels)
XAML	Extensible Application Markup Language

1 INTRODUCTION

Mobile operating systems and devices have evolved rapidly during the last decade. Large touch screen, fast internet-connectivity and multi-tasking operating system are now standard features on a mainstream smartphone. This has pushed mobile devices towards computers and changed the way people are using them.

The evolution of the devices has also affected software developers. The devices are capable of running more complex software, and digital distribution platforms such as Apple's App Store and Google's Android Market provide an easy way to get the software to the mass market. In addition, the users have adopted the digital distribution very quickly. The App Store was opened in July 2008, and in July 2011 over 15 billion applications had been downloaded (46; 47).

The objective of this thesis is to compare three modern mobile operating systems from the software developer's point of view by developing a native application for each. The target platforms are Android 2, Maemo 5 and Windows Phone 7.

2 PLATFORMS

The purpose of this chapter is to give an overview of the three platforms being compared in this thesis. The main topics include how software can be installed and distributed as well as how multitasking is handled and what kind of hardware is supported.

2.1 Android 2

Android is an open mobile platform built on top of Linux kernel 2.6. It includes an operating system, middleware and applications such as web browser and media player. (1; 2.)

Google is leading the Android Open Source Project which is responsible for the maintenance and development of Android. Currently versions 2 and 3 are developed simultaneously, version 2 for smartphones and version 3 for tablet computers. (1; 2; 3.) Version history of Android 2 is presented in table 1.

TABLE 1. Android version history (8; 9; 10; 11)

Version	Codename	Release date	New notable features
2.0	Eclair	October 2009	Browser supports HTML5 Bluetooth 2.1
2.0.1	Eclair	December 2009	
2.1	Eclair	January 2010	
2.2	Froyo	May 2010	Improved browser, Dalvik VM and kernel memory management performance
2.3	Gingerbread	December 2010	Improvements in power management, user interface and system performance
2.3.3	Gingerbread	February 2011	Improved NFC support
2.3.4	Gingerbread	May 2011	

2.1.1 Android Applications

Android applications are written in the Java programming language and executed by a Dalvik virtual machine. By default each application runs on its own process and has its own Dalvik virtual machine instance. This isolates applications from each other and enhances the security. (1; 15.)

Before an application can be installed to a device, it must be packed into an Android Package file using tools provided by the Android SDK. The Android Package is an archive file which has an .apk suffix. It contains all data the application requires and the executable of the application, which is compiled with the Java language compiler and then converted into Dalvik executable format. (15.)

All Android Packages must be signed with a digital certificate, which is typically self-signed (48). Therefore applications can be created, signed and distributed by anyone.

The Android operating system assigns a unique Linux user ID to each application. File permissions for all files of the application are set in the way that other applications cannot access them. However, it is possible to give two applications the same Linux user ID, and thus they can access each other's files. Applications having the same user ID can also share the same process and the Dalvik virtual machine instance. (15.)

2.1.2 Android Market

Android Market is an online software distribution service which has been designed to make finding and installing Android applications easy for the users of Android devices (50). It is developed by Google and it was opened in October 2008 (5).

Developers have to register before they can publish software in the market. The registration costs $25 but it is charged mainly to prevent the abuse of the mar-

ket and to encourage higher quality products. Once the application is uploaded to the service and published, it is immediately visible to the users. (5.)

The applications can be either free or priced. The developer gets 70% of the revenue from each purchase and the rest goes for carriers and billing. Google does not take any money from the purchase. (5.)

2.1.3 Android Multitasking

When the user leaves an Android application, the Android operating system does not end the process but keeps it in the background allowing it to continue working. If the system is running out of memory, the operating system decides which processes are currently important to the user and simply force-kills the least important ones. (16.)

Each time the user leaves an application, the current state of its activities is saved. This allows restoring the application's state when the user returns to it in case the kernel has killed it from the background. (16.) See chapters 3.1.3 and 3.1.4 for more information about activities and services in the Android system.

2.1.4 Hardware Requirements

The Android Compatibility Definition is specified separately for each platform version. Common requirements for platform versions 2.1, 2.2 and 2.3 are a touch screen, OpenGL 1.0 support, a USB connector and high speed wireless network connection. Additionally the device must provide Home, Menu and Back functions with either dedicated buttons or as software implementation. (55; 56; 57.)

A rear-facing camera, an accelerometer and a GPS receiver are required in versions 2.1 and 2.2 but not in 2.3. (55; 56; 57.)

2.2 Maemo 5

Maemo 5 is a GNU/Linux based mobile platform developed by the Maemo Community of which Nokia Corporation is a member and a sponsor of (13).

The Maemo platform has an open development model and it is developed in collaboration with many open source projects. It uses the Linux kernel, Debian package management and GTK+ widgets. (14.) The version history of Maemo 5 is presented in table 2.

TABLE 2. Maemo 5 version history (7)

Version	Codename	Release date	New notable features
PR 1.0	Fremantle	November 2009	
PR 1.1	Fremantle	January 2010	MS Exchange 2003 support
PR 1.1.1	Fremantle	February 2010	
PR 1.2	Fremantle	May 2010	Qt 4.6.2 New Maps application Skype and XMPP video calls
PR 1.3	Fremantle	October 2010	Qt Mobility API 1.0.2 Qt 4.7

2.2.1 Maemo 5 Applications

Maemo 5 applications can be created using various programming languages and frameworks. More information about software development is presented in chapter 3.2.

The preferred way of distributing Maemo 5 applications is Debian packages. However, the Maemo 5 operating system does not limit how the application is installed to the system. It can be done by copying all files to the right directories but a Debian package is certainly a faster and more convenient way. (49.)

The Package management system in Maemo 5 is able to download application packages the from package repositories which are FTP or HTTP sites such as Maemo.org. Maemo 5 software is also available in the Nokia Ovi Store.

2.2.2 Maemo.org Software Repository

As described in the previous chapter, the Package manager of Maemo 5 can download applications from package repositories. There are three main repositories that concern the developers of Maemo 5 applications: Extras, Extras-testing and Extras-devel.

Extras is the primary repository for stable Maemo 5 applications and it is enabled by default in a Maemo 5 device. Extras-devel is a repository for developers. Anyone can upload packages there by registering a Maemo Garage account for free. Once the application is in the Extras-devel, the developer can promote it to the Extras-testing. (51; 52; 53.)

In the Extras-testing the members of the Maemo community can give either positive or negative votes for the application. The votes increase or decrease the Karma points of the application. Once the application has been in the Extras-testing for at least ten days and it has the Karma of 10 or better, it is promoted to the Extras repository. The Application can be demoted back to the Extras-devel if after the ten days it has the Karma of -5 or worse and the maintainer of the application also votes down. (52.)

2.2.3 Ovi Store

The Ovi Store is a part of the Ovi by Nokia service which includes maps, music, applications and messaging (54). Developers can upload their applications to the Ovi Store after registering as a publisher and paying a 1€ registering fee. Before the application is published, Nokia conducts a QA (Quality Assurance) review for the content. After the QA review, the developer will receive an email

either confirming that the application has been accepted or giving information about the problems which must be fixed. (54.)

The applications in the Ovi Store can be either free or priced. The developers are paid 70% of the net revenue from purchases made with a credit card and 60% of the purchases made using the operator billing. (54.)

2.2.4 Maemo 5 Multitasking

The Maemo 5 has a real multitasking allowing multiple applications to run simultaneously. The running applications are presented on Maemo Dashboard, see figure 1. (13.)

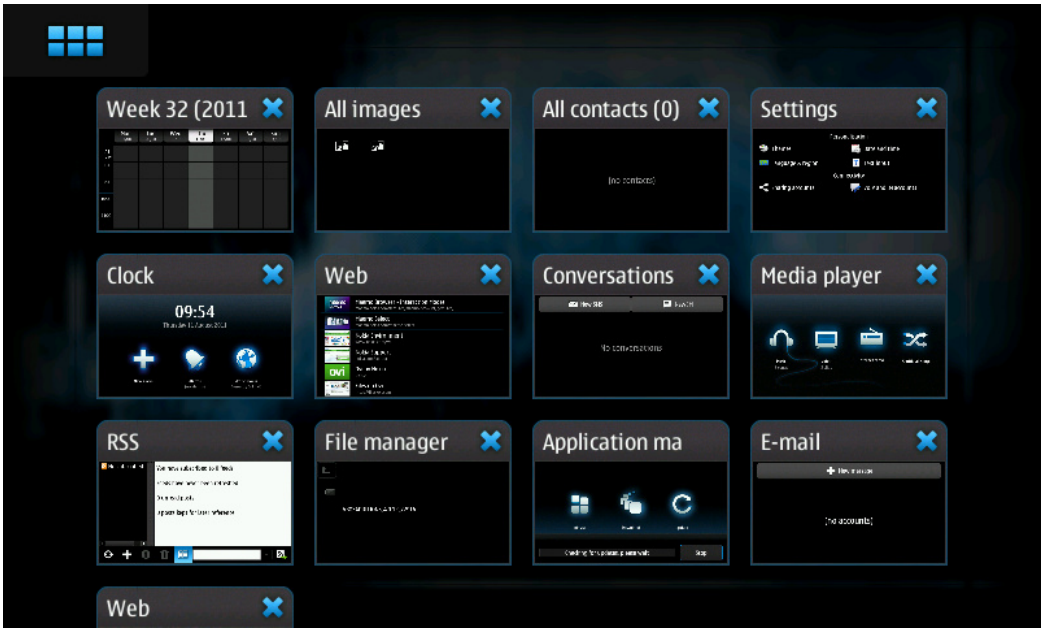

FIGURE 1. The Maemo Dashboard displays all running applications

2.2.5 Nokia N900

The Nokia N900 was released in August 2009 and it is currently the only device running Maemo 5. The key features of the N900 are QWERTY keyboard, WVGA touch screen, ARM Cortex-A8 processor, 5 Megapixel camera and fast

internet connectivity with HSPA and WLAN (12; 13). See figure 2 for a picture of the device.

FIGURE 2. Nokia N900

2.3 Windows Phone 7

Windows Phone 7 is a completely new mobile platform developed by Microsoft. It features a new user interface design and it is not compatible with Windows Mobile applications. (45.)

TABLE 3. Windows Phone 7 version history (39; 40.)

Version	Codename	Release date	New notable features
7.0.7004.0		October 2010	
7.0.7390.0	NoDo	March 2011	Copy & paste
7.1.7720.6	Mango	July 2011	Internet Explorer 9 Twitter and LinkedIn Conversation threads Multitasking

2.3.1 Windows Phone 7 Applications

Windows Phone 7 applications are written in either C# or Visual Basic .NET programming language. For more information about Windows Phone 7 software development, see chapter 3.3.

A Windows Phone 7 application is packed into a XAP file which is a ZIP archive that contains all the files the application requires. The XAP files can be deployed directly to the Windows Phone Emulator and a developer unlocked Windows Phone 7 device. (33, p. 20) See chapter 2.3.2 for more information about unlocking Windows Phones.

2.3.2 Windows Phone Marketplace

Applications can be installed to the Windows Phone 7 only from the Windows Phone 7 Marketplace. However developers can unlock their devices for development purposes and deploy applications directly from a Windows PC. (33, System Requirements.)

A Windows Phone 7 device can be unlocked by registering to the Microsoft App Hub. For the annual $99 USD subscription the developer can unlock the device and submit free or priced applications to the Windows Phone Marketplace. The same subscription also allows developing Xbox LIVE Indie Games. (6.) At the time of writing this thesis the Windows Phone Marketplace was not available in Finland.

2.3.3 Windows Phone 7 Multitasking

The Windows Phone version 7.0 does not have a real multitasking for all applications. Instead, when a user leaves an application by pressing the start button, the application is terminated but its current state is saved and the application is put in a so called tombstoned state. The application is terminated and its state is discarded when it is left by pressing the back button. When an application

that is in a tombstoned state is launched again its state is resumed and it appears to the user as if it was never terminated. (33, p. 121-122.)

Windows Phone 7.1 codenamed the Mango features a fast application switching which preserves the application in memory when the user navigates away from it. In same way as Android, Windows Phone 7.1 may terminate the applications from the background if memory is required for other applications. However, the state of the application is saved so it can be resumed. (4.)

In addition, Windows Phone 7.1 allows the applications to perform certain tasks such as playing audio, executing scheduled tasks and transferring files in the background (4).

2.3.4 Hardware Requirements

Microsoft has specified that all Windows Phone 7 devices must have six hardware buttons: power/sleep, volume up, volume down, camera, back, start and search. The back, start and search buttons can be implemented as capacitive touch buttons. (37.) The screen must have a resolution of 800x480 pixels and it must support a 4-point multi-touch. A data connectivity must be available through a cellular network and Wi-Fi. The device must also have at least 256 MB of RAM, at least 8 GB of flash storage, a GPU supporting DirectX 9 hardware acceleration, a digital camera and A-GPS support. (38.)

Additionally, all devices must have the following sensors: accelerometer, compass, light and proximity. However, the accelerometer is currently the only one of these that can be accessed by developers.

3 SOFTWARE DEVELOPMENT

3.1 Android Software Development

Currently, officially supported Android development tools are available for Windows XP (32-bit), Windows Vista, Windows 7, any modern Linux desktop distribution and Mac OS X 10.4.9 or later (18).

The Android SDK provides the required tools and libraries for creating applications. Additionally, an IDE or editor is required. The preferred development environment is Eclipse IDE with an ADT (Android Development Tools) plugin. The ADT plugin makes it possible to directly invoke all required development tools from Eclipse. (17.)

Typically Android applications are written in the Java programming language but the Native Development Kit (NDK) makes it possible to use the C or C++ code as well.

3.1.1 Android SDK

The Android SDK has a modular structure, containing Android platforms, add-ons, tools, samples and documentation. The installation of these components is done via the Android SDK and AVD Manager, which is included in the Android SDK starter package. See figure 2.

The Android emulator is also a part of the SDK. It makes it possible to run Android applications on the development computer without a physical device. More detailed information about the Android emulator and Android virtual machines is presented in chapter 3.1.5.

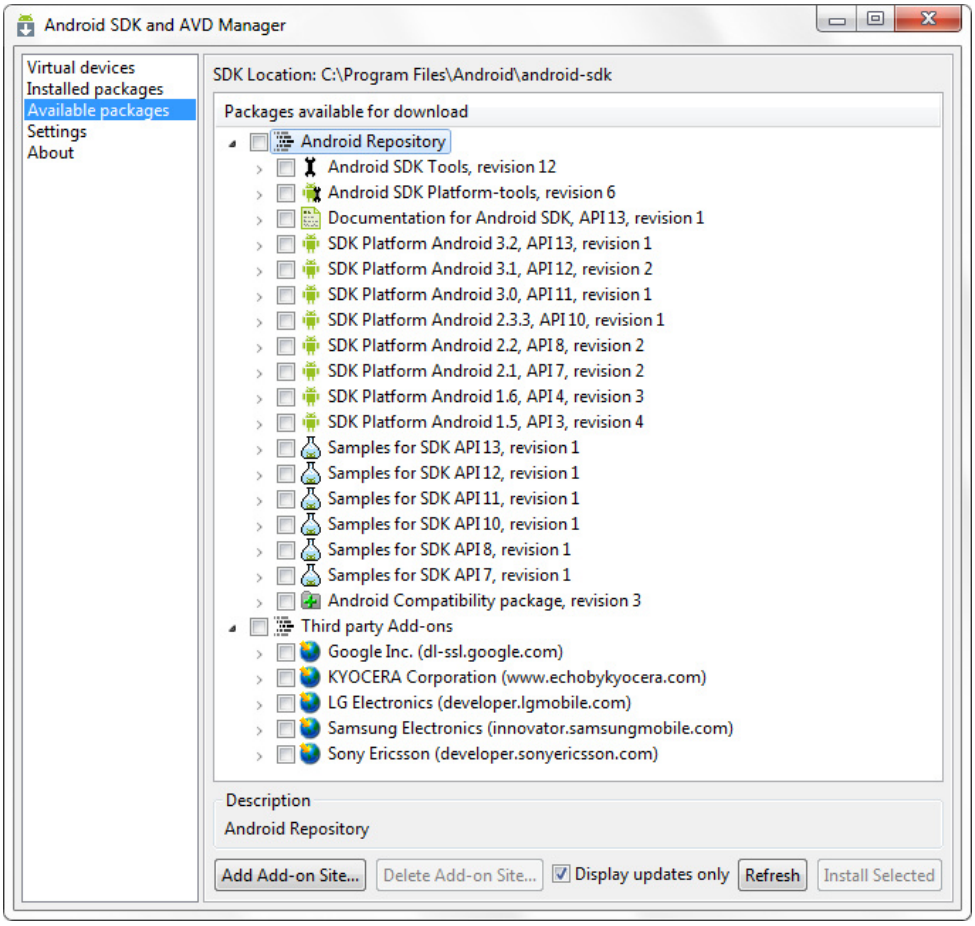

FIGURE 3. Android SDK and AVD Manager

3.1.2 Native Development Kit

The NDK provides tools for implementing parts of the application with native-code languages such as C and C++. This allows reusing the existing libraries and in some cases provides a better performance. (19.)

The system headers and libraries provided by the NDK are guaranteed to be supported by all later platform releases starting from 1.5 (19). See table 4 for the list of the libraries included.

TABLE 4. Set of system headers and libraries included in the Android NDK (19)

The C library (libc)
The Math library (libm)
JNI interface headers
Zlib compression (libz)
Android logging (liblog)
OpenGL ES 1.1
OpenGL ES 2.0
Pixel buffer access for Android 2.2 and above (libjnigraphics)
Headers for minimal C++ support
OpenSL ES native audio libraries
Android native application APIS

Using the NDK always increases the application complexity. In most applications it does not provide real benefits since using the C code instead of Java rarely results in large performance increases. Therefore, the NDK should only be used for reusing large amounts of the existing C or C++ code or in CPU-intensive calculations such as signal processing or physics simulations. (19.)

3.1.3 Activities, Intents and Tasks

An activity is a very central component of an Android application. Each activity provides a screen which can contain objects derived from the View-class such as button and text-field widgets. The Activity can fill the whole screen or it can be smaller and displayed on top of other activities. Typically, each application consists of multiple activities. For example, a contacts application could have one activity for displaying a list of contacts and another for viewing detailed information about one contact. (41.)

Activities inside other applications can be reused (42.) For example, the contacts application could provide an option to send an email to the contact. This feature could use the activities of an email application for writing and sending the email.

An activity can start other activities by defining an intent. The intents are messages that contain data and an action that the sender wants to be taken. If there are multiple activities for handling a particular intent, the system prompts the user to select which activity should be used. (42.)

A task is a collection of activities required for doing a certain job. A new task is created each time an application which is not in the background is started. The first activity of a task is called the root activity. The root activity and all other activities created for the task are pushed into a stack called the back stack in the order they are created. When the user presses the back button, the current activity is popped from the top of the stack and destroyed. After the root activity of a task is destroyed, the task no longer exists. (42.) The back stack is visualized in figure 5.

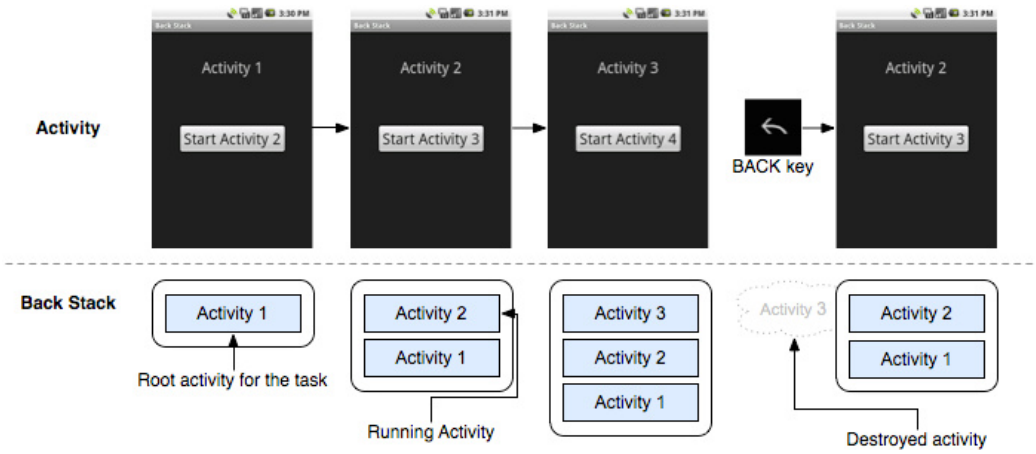

FIGURE 5. Android back stack (42)

3.1.4 Services

A service in the Android system is an application which does not have a user interface but performs a certain task in the background. For example, a music player can start a service for playing music and it continues playing even when the user starts to use another application. (43.)

3.1.5 Android Emulator

The Android emulator is based on the QEMU processor emulator. It provides a virtual ARM device which runs a full Android system and has all other software and hardware features of a typical Android device except of making actual phone calls. In addition, the emulator provides debugging features such as simulating arriving phone calls and a console for logging the kernel output. (20.) A screenshot of the Android Emulator is presented in figure 6.

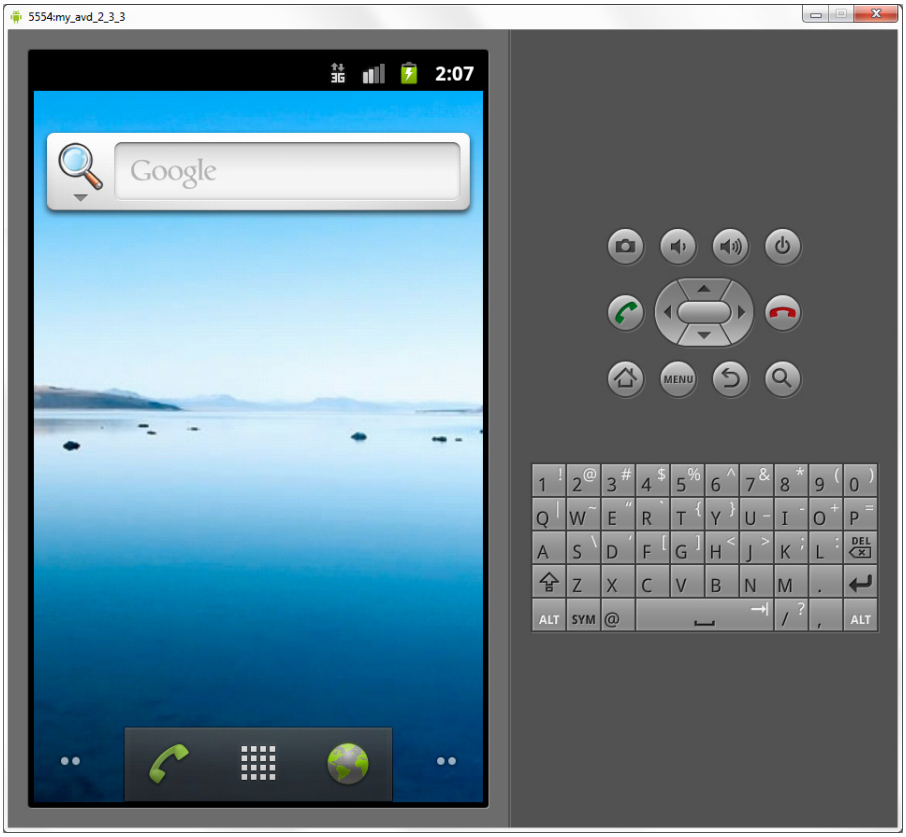

FIGURE 6. Emulator running Android 2.3.3

The hardware configuration and Android platform version is selected by creating an Android virtual device (AVD) configuration. Creating multiple AVD configurations makes it easy to test the application on different hardware and Android platform versions. (20.) See figure 7.

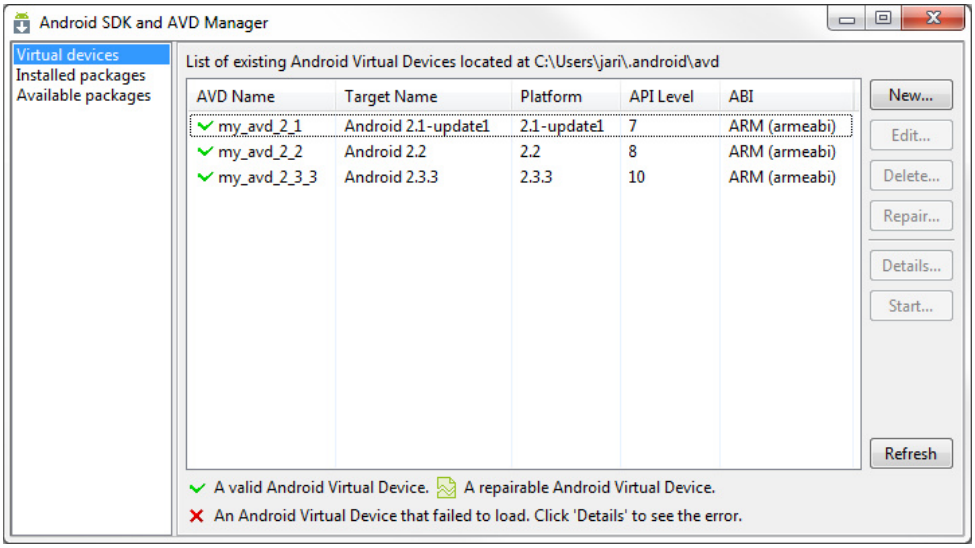

FIGURE 7. Android SDK and AVD Manager

3.2 Maemo 5 Software Development

Maemo 5 applications can be created with the GTK+ toolkit and the Qt 4.6. or 4.7 (29). Python support is also available through the PyMaemo project in the Extras repository (30).

The Maemo 5 SDK supports officially 32-bit Debian based GNU/Linux distributions. However, installation on other Linux distributions is possible. The SDK uses a sandbox environment which is built on top of the Scratchbox cross-compilation toolkit. This sandbox provides development environments for the x86 and Armel targets from which the x86 target is used for the active development and the Armel only for cross-compiling applications for the actual device. Developing in the x86 environment is preferred because the development tools can be run natively instead of using emulation. (21.)

Maemo 5 applications can be run on the development computer by using Xephyr X11 server with the Scratchbox. The Xephyr is used to display the Maemo application which is compiled for the x86 target and running inside the Scratchbox. Usually the Maemo 5 UI Framework is also started inside the Scratchbox to provide the same look and feel for the application as in a real device.

(21; 22; 23). Figure 8 visualizes the structure of the Maemo 5 SDK. The Xephyr displaying the Maemo 5 UI Framework is presented in figure 9.

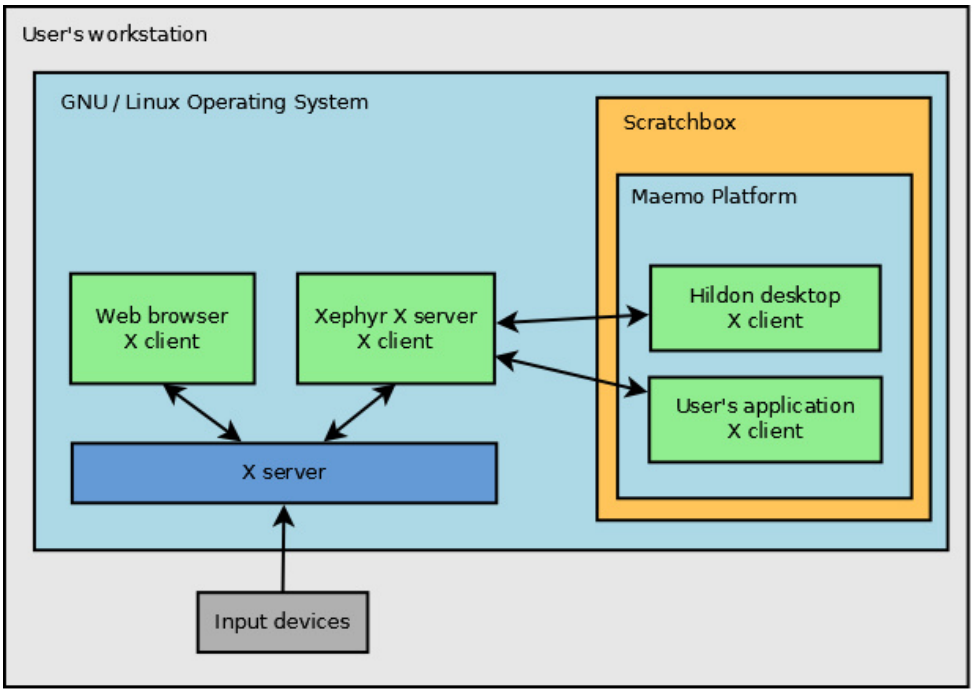

FIGURE 8. Using Xephyr to display an application running inside Scratchbox (21; 22; 23)

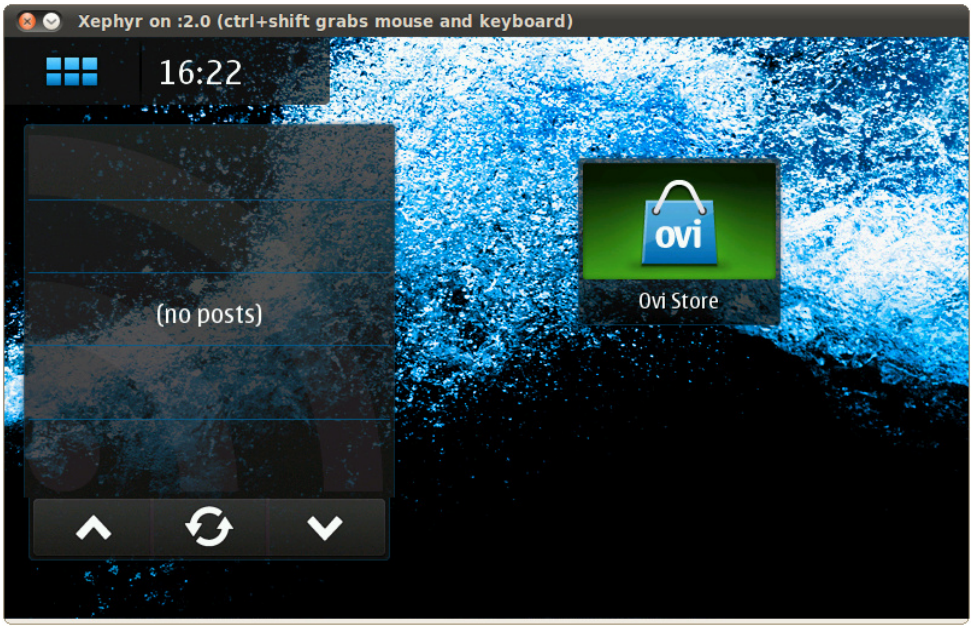

FIGURE 9. The Maemo 5 UI Framework on Xephyr

3.2.1 GTK+

GTK+ is a cross-platform UI library featuring support for many UNIX-like platforms as well as Windows and Mac OS X. It is one of the core components of Maemo 5 as the user interface is based on GTK+ widgets. (31.)

The architecture of GTK+ is object-oriented although it is written in C. Bindings for several other languages including C++, Objective-C and Python are also available. (31.) The object-oriented architecture with C language is achieved through use of the GObject system (32.)

3.2.2 Qt

Maemo 5 supports the latest stable release of Qt, the 4.7.

Qt is a cross-platform C++ application and UI framework. Supported platforms include GNU/Linux, Windows, Mac OS X, Maemo 5, Meego, Symbian and many others. One of the design goals of Qt is to decrease multi-platform application development time by making it possible to use a single source tree for all platforms. Qt is available as an open source edition as well as a commercial edition. The open source version is available under GNU LGPL 2.1 and GNU GPL 3.0 with Qt special exception. Qt has been used widely in both open source and commercial applications. It is the base of KDE project and it has been used by companies and organizations such as Adobe, Google, IBM and NASA. (26.)

Qt has been developed for over 15 years. On June 2008 Nokia completed its acquisition of Trolltech, the original developer of Qt. Currently it is being developed by the Qt Development Frameworks division of Nokia. (27; 28.)

The class library of Qt is extensive, featuring modules for example GUI widgets, network programming, OpenGL, WebKit integration, SQL database support, XML and unit testing (26).

Inter-object communication with signals and slots is one of the core features of Qt. It provides an easy to use and safe alternative for callback functions.

3.2.3 Qt Quick

Qt Quick framework is designed to make developing applications with rich user interfaces easier and faster. It consists of QML language, Qt Declarative module and tool support in Qt Creator. (24.) Qt Quick was introduced with Qt 4.7 (26).

QML is a declarative language based on JavaScript. It is designed for describing user interfaces and their behavior. The entire application can be implemented with QML, but more effective solution is to implement the application logic with C++ and only the user interface with QML. C++ objects and QML are able to communicate through the signal and slot system of Qt. (24; 25.)

3.3 Windows Phone 7 Software Development

Windows Phone 7 applications are written in .NET managed code using either C# or Visual Basic .NET programming language. Currently there are two frameworks to choose from, Silverlight and XNA. XNA is designed for games and Silverlight for other applications. However making games with Silverlight and using XNA in traditional applications is possible.

Starting from Windows Phone version 7.1 both Silverlight and XNA can be used in the same application. Earlier platform versions can only use some of each other's libraries (33, p. 2-3).

The Windows Phone 7 SDK supports Windows Vista Service Pack 2 and Windows 7. It includes Visual Studio 2010 Express for Windows Phone, Windows Phone Emulator, XNA Game Studio, Expression Blend for Windows Phone, samples and documentation. (34.)

3.3.1 Silverlight for Windows Phone

Silverlight for Windows Phone application framework is a subset of the normal Silverlight framework but it does provide some additional phone-specific features (45).

Silverlight applications consist of XAML markup and code. The XAML can be used to define the application's UI layout and handling the creation and initialization of the UI objects. Furthermore the XAML can also handle the binding of underlying data to UI controls and the connections between different UI controls so that their properties get updated without explicitly defining event handlers in the code. However all of this can be done also in the code. (33, p. 138.)

Methods and event handlers must be implemented in the code and the XAML is required for using page navigation and for customizing controls with templates. (33, p. 138-139.)

The XAML can also be generated using a design program such as Expression Blend, which is included in the Windows Phone 7 SDK.

3.3.2 XNA Game Studio 4.0

The XNA Game Studio extends Microsoft Visual Studio with the XNA Framework and tools. The XNA Framework is a managed-code class library targeted for Windows, Windows Phone 7 and Xbox 360 game development. (35.)

In the same way the Silverlight for Windows Phone is a subset of the normal Silverlight, the XNA implementation on Windows Phone is a subset of the full XNA framework. (45.)

3.3.3 Windows Phone Emulator

The Windows Phone emulator provides a virtualized environment for developing, debugging and testing applications. It supports images of the Windows Phone operating system 7.0 or higher. (36.)

The emulator has most of the features present on an actual device, including networking, location data, accelerometer data, persistent data storage and keyboard input. See figure 10 for a screenshot of the emulator.

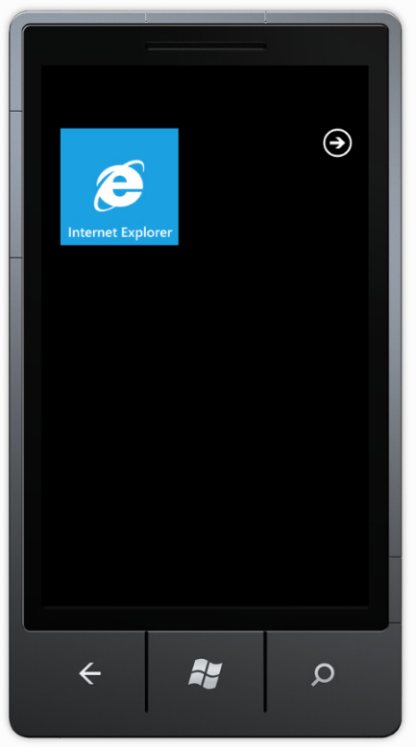

FIGURE 10. Windows Phone 7 emulator

4 DEVELOPING THE FLICKR APPLICATION

This chapter presents the implementation of the Flickr application for the three target platforms. The functional requirements for the application are that it must be able to download a list of images from the Flickr public RSS feed and it must provide an option to view the images in full screen.

The applications have small differences between each other both in the UI and at the architectural level. This is simply due to the fact that the platforms and programming languages are different. However, all three applications have the same functionality.

4.1 Android

The Android version of the Flickr application was implemented in Java using the Android API. See figure 11 for a screenshot of the application.

During the development of this application it turned out that the Android SDK provided a lower level API than the Qt Quick and the Silverlight. The main difference was that asynchronous downloading of the images had to be implemented in Java whereas the other platforms provided an image component which took care of it. Mainly because of this, the Android version of this application has about 100 lines more code than the others. However, this does not mean that the Android API would be more difficult to use than the other two.

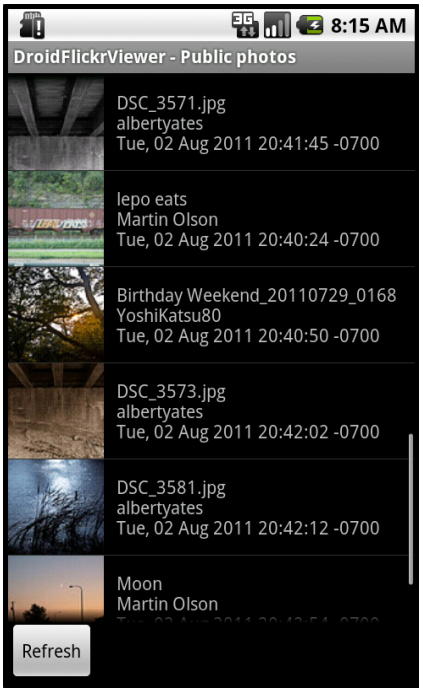

FIGURE 11. The Flickr application on Android 2.1

4.1.1 Installing the Android SDK

The SDK was installed on two computers, one with a 64-bit Windows 7 Professional and another with Ubuntu 10.04 to see if there were any differences, but the installation and overall behavior of the SDK was quite similar on both.

The SDK installation required JDK 6, Android SDK Starter Package, Eclipse 3.7 and the ADT Plugin for Eclipse. After downloading and installing all components, the last step was to download the target platforms and create an Android Virtual Device configuration with the SDK and AVD manager. See figures 3 and 7 for screenshots of the SDK and AVD manager.

4.1.2 Architecture

The Android version of the Flickr application consists of four main components: an activity for displaying the image list, an activity for displaying the full screen image, an ImageDownloader class and a FlickrRssFeed class.

The FlickrRssFeed handles the downloading and parsing of the RSS and it uses the ImageDownloader to download the thumbnails and the full screen images when required. Both the ImageDownloader and the FlickrRssFeed use the abstract AsyncTask class provided by the Android API for performing asynchronous operations outside the UI thread.

4.1.3 Creating a User Interface with Android Layout Files

The Flickr application has two Activities, one for displaying a list of images and another for viewing them in full screen. See figure 11 for a screenshot of the image list activity.

The user interface layout of an activity can be defined with an XML file. The layout is based on hierarchy of ViewGroup and View objects. The ViewGroup can contain other ViewGroup and View objects. The View class is the base of UI elements such as buttons and text fields. (44.) See example 1 for a definition of a row in the image list of the Flickr application.

EXAMPLE 1. Android layout definition for a row in the image list

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    android:orientation="horizontal"

    <ImageView android:id="@+id/thumbnail"
        android:scaleType="fitCenter"
        android:layout_height="75dip"
        android:layout_width="75dip" />

    <TextView android:id="@+id/text"
        android:layout_gravity="center_vertical"
```

```
        android:layout_width="0dip"
        android:layout_weight="1.0"
        android:layout_height="wrap_content"
        android:paddingLeft="10dip"/>
</LinearLayout>
```

In example 1 the base ViewGroup object is a LinearLayout. It contains two view objects, an ImageView and a TextView. Figure 11 presents a screenshot of the Flickr application's image list where each row is defined by this layout.

4.2 Maemo 5

The Flickr application for Maemo 5 was implemented using the Qt Quick. See figure 12 for a screenshot of the application.

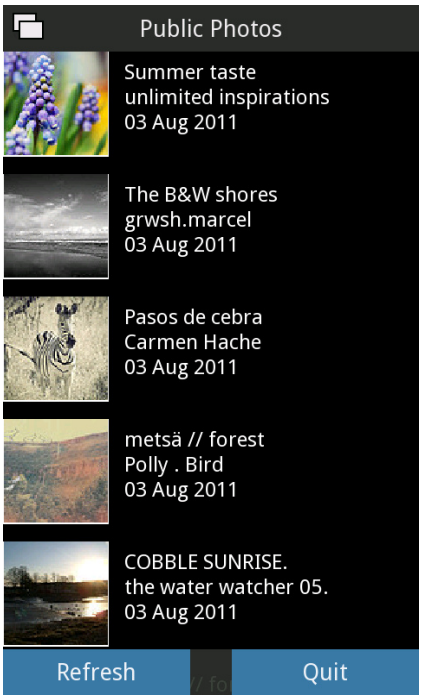

FIGURE 12. Flickr Application on Maemo 5

4.2.1 Installing the Maemo SDK

Installing the Maemo SDK was easy using the official GUI installer but the version of Xephyr server in Ubuntu 10.04 repositories did not work with the SDK at first.

The SDK installer creates a start_xephyr.sh shell script for starting The Xephyr and the Maemo 5 UI Framework. However, the Xephyr version in the Ubuntu 10.04 repositories crashed with a segmentation fault when it was started with the parameters given by the start_xephyr.sh. The solution was to remove the -kb parameter.

4.2.2 Architecture of the Flickr application for Maemo 5

The architecture of this application is very simple. The UI is implemented with QML and it communicates with a Controller C++ component. The Controller handles downloading of the RSS and it uses a FlickrRssParser class for parsing the RSS data and creating a list of FlickrImgInfo objects. The FlickrImgInfo class represents a single Flickr image. It contains the image title, author, date and addresses of the thumbnail and the original image.

The downloading of the images is not implemented in C++ since the QML Image element downloads images simply by setting a HTTP address as the image source.

4.2.3 Creating a User Interface with QML

A QML document consists of a tree of QML elements. In example 2 the ImageListDelegate.qml presents a simple QML document with two elements, an Item and an Image.

The user interface of the Flickr application requires two different views, an image list and a full screen view. The Flipable element in QML was used as a con-

tainer for these views. The full screen view has only an Image element and a progress bar showing the download progress of the image. The implementation of the image list with the ListView element is discussed in the next chapter.

4.2.4 QML ListView with C++ Data Model

The Maemo 5 version of the Flickr application uses a QML ListView element to display a list of images. The ListView element has a model which is the data it displays and a delegate which describes how the data should be displayed in the list. The delegate is implemented in QML and the model in C++. The Model is a QList which contains QObject pointers to FlickrImgInfo objects.

As described earlier in chapter 4.2.2, the FlickrRssParser creates a list of FlickrImgInfo objects, each of which represent a Flickr image. The image title, author and other information is declared using the Q_PROPERTY macro and therefore it is available through the property system of Qt and it can be accessed from QML. See example 2 for code example of the C++ data model in the QML ListView.

EXAMPLE 2. Simplified example of code used to create the image list in the Maemo 5 version of the Flickr application

```
// C++
QList<QObject *> model;
model.append(new FlickrImgInfo(thumbnailUrl, title));
QDeclarativeContext c = view->rootContext();
c->setContextProperty("imageListModel", QVariant::fromValue(model));

// FlickrImgInfo.h
class FlickrImgInfo : public QObject
{
    Q_OBJECT
    Q_PROPERTY(QString thumbnailUrl READ thumbnailUrl)
    Q_PROPERTY(QString title READ title)
```

```
public:
    QString thumbnailUrl() { return m_thumbnailUrl; }
    QString title() { return m_title; }.
};

// main.qml
ListView {
    model: imageListModel
    delegate: ImageListDelegate {
        thumbnailUrl: model.modelData.thumbnailUrl
        title: model.modelData.title
    }
}

// ImageListDelegate.qml
Item {
    property string thumbnailUrl
    property string title
    width: 480;
    height: 120

    Text {
        x: 140;
        y: 10
        text: title
    }

    Image {
        x: 0;
        y: 0
        source: thumbnailUrl
        width: parent.height;
        height: parent.height
}
```

4.3 Windows Phone 7

The Windows Phone 7 version of the Flickr application was created using C# and the Silverlight for Windows Phone. See figure 13 for a screenshot of the application.

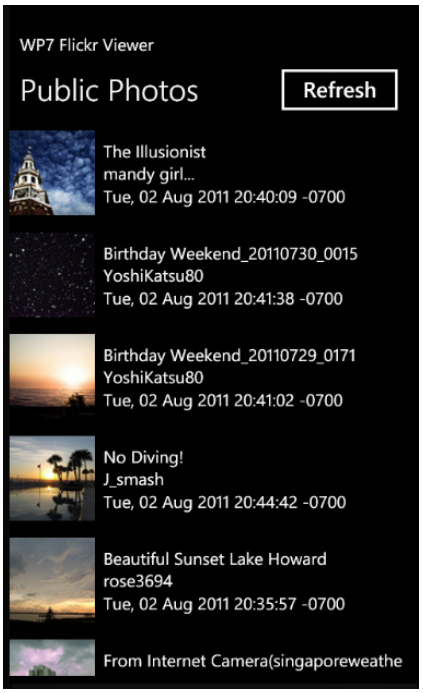

FIGURE 13. The Flickr application on Windows Phone 7

4.3.1 Installing the Windows Phone 7 SDK

The Windows Phone 7 SDK was installed on a PC running a 64-bit Windows 7 Professional. The SDK installer from the Microsoft Developer Network handled the installation automatically and everything was ready to be used once the installer had finished.

4.3.2 Architecture of the Flickr application for Windows Phone 7

The user interface was defined with XAML and the downloading and parsing of the RSS was implemented with C#. More detailed information about defining the user interface in XAML is presented in chapter 4.3.3.

The asynchronous downloading and parsing of the RSS is handled by a FlickrRssFeed class. It creates ImageListItem objects which contain the image details and addresses for the thumbnail and the full screen image. The ImageListItem objects are stored in an ObservableCollection which is bound to a ListBox, defined in the user interface XML.

4.3.3 Creating a User Interface with XAML

Example 3 presents an XAML definition of a ListBox element used in the image list of the Windows Phone 7 Flickr application. The ListBox definition uses an ItemTemplate to define how the data is displayed in the list.

EXAMPLE 3. XAML definition of the image list

```
<ListBox Name="ImageListBox" Margin="0,5,0,0" Height="650">
    <ListBox.ItemTemplate>
        <DataTemplate>
            <Grid Height="100">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="100" />
                    <ColumnDefinition MaxWidth="370" />
                </Grid.ColumnDefinitions>
                <Image
                    Grid.Column="0"
                    Source="{Binding ThumbnailUri}"
                    Width="100"
                    Height="100" />
                <TextBlock
```

```
                    Grid.Column="2"
                    VerticalAlignment="Center"
                    Margin="10"
                    Text="{Binding Details}" />
            </Grid>
        </DataTemplate>
    </ListBox.ItemTemplate>
    <ListBox.ItemContainerStyle>
        <Style TargetType="ListBoxItem">
            <Setter Property="Height" Value="120" />
        </Style>
    </ListBox.ItemContainerStyle>
</ListBox>
```

5 CONCLUSION AND DISCUSSION

Developing the Flickr application for Windows Phone 7 was slightly faster than for the other platforms. It has a high level API and the SDK provides good tools for developers. However the most important factor was that as a developer I felt very productive. I did not encounter any real problems, and as setting up the development environment was fast, I was able to focus on the important phase – developing the application.

At the same time I was worried about the fact that although the SDK is free of charge, developing for this platform is possible only on Windows and the devices have to be unlocked before any applications can be installed for testing.

Android 2 and the Maemo 5 are in turn more flexible and open platforms. They both allow installing applications from any sources, and in my opinion they are very developer-friendly.

As the Maemo 5 SDK works in a GNU/Linux environment, there is a wide range of powerful development tools available. Personally I prefer working with command line tools. However, the Qt Creator is a good alternative for developer's who like to use tools with a graphical user interface.

My initial impressions of the Qt Creator were very positive. It might have fewer features than Eclipse or Visual Studio but it did everything I needed.

The problem with Maemo 5 and its successor MeeGo is that the user base is very small. Android does not have this problem and it is difficult to find anything negative to say about the Android SDK. However, the fragmentation of the Android devices might be a problem since there are many different screen sizes and performance of the hardware varies. I did not have resources to study the real effects of the fragmentation but in theory it is a factor which developers must take into account.

Choosing between these platforms is purely a matter of taste. Each one has its strengths and weaknesses but they all are certainly capable of competing against each other.

LIST OF REFERENCES

1. Android Developers. What is Android? Available at: http://developer.android.com/guide/basics/what-is-android.html. Date of data acquisition 19 July 2011.
2. Android Open Source Project. Welcome to Android. Available at: http://source.android.com. Date of data acquisition 19 July 2011.
3. Android Developers. Android 3.0 Platform Highlights. Available at: http://developer.android.com/sdk/android-3.0-highlights.html. Date of data acquisition 19 July 2011.
4. Microsoft Developer Network. Multitasking for Windows Phone. Available at: http://msdn.microsoft.com/en-us/library/hh202866(v=vs.92).aspx. Date of data acquisition 12 August 2011.
5. Android Developers. Android Market: Now available for users. Available at: http://android-developers.blogspot.com/2008/10/android-market-now-available-for-users.html. Date of data acquisition 10 August 2011.
6. App Hub. Membership. Available at: http://create.msdn.com/en-us/home/membership. Date of data acquisition 12 August 2011.
7. Maemo.org. Open Development / Maemo Roadmap / Fremantle. Available at: http://wiki.maemo.org/Maemo_5. Date of data acquisition 11 August 2011.
8. Android Developers. Android 2.0, Release 1. Available at: http://developer.android.com/sdk/android-2.0.html. Date of data acquisition 20 July 2011.
9. Android Developers. Android 2.1 Platform. Available at: http://developer.android.com/sdk/android-2.1.html. Date of data acquisition 20 July 2011.
10. Android Developers. Android 2.2 Platform. Available at: http://developer.android.com/sdk/android-2.2.html. Date of data acquisition 20 July 2011.
11. Android Developers. Android 2.3 Platform. Available at: http://developer.android.com/sdk/android-2.3.html. Date of data acquisition 20 July 2011.

12. Nokia.com. Maemo 5 injects speed and power into mobile computing. Available at: http://press.nokia.com/2009/08/27/maemo-5-injects-speed-and-power-into-mobile-computing/. Date of data acquisition 9 August 2011.
13. Maemo.org. Community. Available at: http://maemo.org/intro/community/. Date of data acquisition 9 August 2011.
14. Maemo.org. Software Platform. Available at: http://maemo.org/intro/platform/. Date of data acquisition 9 August 2011.
15. Android Developers. Application Fundamentals. Available at: http://developer.android.com/guide/topics/fundamentals.html. Date of data acquisition 30 July 2011.
16. Android Developers. Multitasking the Android Way. Available at: http://developer.android.com/resources/articles/multitasking-android-way.html. Date of data acquisition 31 July 2011.
17. Android Developers. Introduction. Available at: http://developer.android.com/guide/developing/index.html. Date of data acquisition 23 July 2011.
18. Android Developers. System Requirements. Available at: http://developer.android.com/sdk/requirements.html. Date of data acquisition 23 July 2011.
19. Android Developers. What is the NDK? Available at: http://developer.android.com/sdk/ndk/overview.html. Date of data acquisition 23 July 2011.
20. Android Developers. Using the Android Emulator. Available at: http://developer.android.com/guide/developing/devices/emulator.html. Date of data acquisition 23 July 2011.
21. Maemo.org. Maemo SDK. Available at: http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Development_Environment/Maemo_SDK. Date of data acquisition 24 July 2011.
22. Maemo.org. Maemo 5 Final SDK Installation. Available at: http://wiki.maemo.org/Documentation/Maemo_5_Final_SDK_Installation. Date of data acquisition 24 July 2011.

23. Nokia Developer. Get Started with Maemo 5. Available at: http://www.developer.nokia.com/document/Get_Started_with_Maemo_5/. Date of data acquisition 24 July 2011.
24. Nokia Developer. Qt Quick. Available at: http://get.qt.nokia.com/education/CourseMaterials/L7-Qt_Quick/L7-pdf.zip. Date of data acquisition 25 July 2011.
25. Qt Reference Documentation. Qt Quick. Available at: http://doc.qt.nokia.com/latest/qtquick.html. Date of data acquisition 25 July 2011.
26. Qt Developer Network. Qt Whitepaper. Available at: http://developer.qt.nokia.com/wiki/QtWhitepaper. Date of data acquisition 27 July 2011.
27. Nokia.com. Nokia Completes Trolltech Acquisition. Available at: http://press.nokia.com/2008/06/17/nokia-completes-trolltech-acquisition/. Date of data acquisition 27 July 2011.
28. Qt. About Us. Available at: http://qt.nokia.com/about. Date of data acquisition 27 July 2011.
29. Nokia Developer. Maemo 5 SDK. Available at: http://www.developer.nokia.com/Resources/Tools_and_downloads/Other/Maemo/. Date of data acquisition 28 July 2011.
30. Maemo.org. PyMaemo. Available at: http://wiki.maemo.org/PyMaemo. Date of data acquisition 28 July 2011.
31. GTK+ Reference Manual. Part I. GTK+ Overview. Available at: http://maemo.org/api_refs/5.0/5.0-final/gtk/gtk.html. Date of data acquisition 28 July 2011.
32. Gnome Dev Center. GObject Reference Manual. Available at: http://developer.gnome.org/gobject/stable/chapter-intro.html. Date of data acquisition 28 July 2011.
33. Petzold, C. 2010. Programming Windows Phone 7. Redmond, Washington: Microsoft Press. Available at: http://download.microsoft.com/download/5/0/A/50A39509-D015-410F-A8F2-

A5511E5A988D/Microsoft_Press_ebook_Programming_Windows_Phone_7_PDF.pdf. Date of data acquisition 28 July 2011.

34. Microsoft Developer Network. Installing the Windows Phone SDK. Available at: http://msdn.microsoft.com/en-us/library/ff402530(v=VS.92).aspx. Date of data acquisition 28 July 2011.
35. Microsoft Developer Network. Getting Started with XNA Game Studio Development. Available at: http://msdn.microsoft.com/en-us/library/bb203894.aspx. Date of data acquisition 29 July 2011.
36. Microsoft Developer Network. Windows Phone Emulator. Available at: http://msdn.microsoft.com/en-us/library/ff402563(v=vs.92).aspx. Date of data acquisition 29 July 2011.
37. Microsoft Developer Network. First Look at Windows Phone. Available at: http://msdn.microsoft.com/en-us/library/hh202905(v=VS.92).aspx. Date of data acquisition 30 July 2011.
38. Microsoft Developer Network Hardware Specifications for Windows Phone Available at: http://msdn.microsoft.com/en-us/library/ff637514(v=VS.92).aspx. Date of data acquisition 30 July 2011.
39. Windows Phone. Windows Phone Update History. Available at: http://www.microsoft.com/windowsphone/en-us/howto/wp7/basics/update-history.aspx. Date of data acquisition 11 August 2011.
40. Wikipedia.org. Windows Phone version history. Available at: http://en.wikipedia.org/wiki/Windows_Phone_version_history. Date of data acquisition 11 August 2011.
41. Android Developers. Activities. Available at: http://developer.android.com/guide/topics/fundamentals/activities.html. Date of data acquisition 31 July 2011.
42. Android Developers. Tasks and Back Stack. Available at: http://developer.android.com/guide/topics/fundamentals/tasks-and-back-stack.html. Date of data acquisition 31 July 2011.
43. Android Developers. Services. Available at: http://developer.android.com/guide/topics/fundamentals/services.html. Date of data acquisition 31 July 2011.

44. Android Developers. User Interface. Available at: http://developer.android.com/guide/topics/ui/index.html. Date of data acquisition 4 August 2011.
45. Phone Development. Chapter 1 – Introducing Windows Phone 7. Available at: http://msdn.microsoft.com/en-us/library/gg490768.aspx. Date of data acquisition 4 August 2011.
46. Apple Press Info. iPhone 3G on Sale Tomorrow. Available at: http://www.apple.com/pr/library/2008/07/10iPhone-3G-on-Sale-Tomorrow.html. Date of data acquisition 8 August 2011.
47. Apple Press Info. Apple's App Store Downloads Top 15 Billion. Available at: http://www.apple.com/pr/library/2011/07/07Apples-App-Store-Downloads-Top-15-Billion.html. Date of data acquisition 8 August 2011.
48. Android Developers. Signing Your Applications. Available at: http://developer.android.com/guide/publishing/app-signing.html. Date of data acquisition 9 August 2011.
49. Maemo.org. Packaging, Deploying and Distributing. Available at: http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Packaging%2C_Deploying_and_Distributing. Date of data acquisition 9 August 2011.
50. Android Developers. Publishing on Android Market. Available at: http://developer.android.com/guide/publishing/publishing.html. Date of data acquisition 9 August 2011.
51. Maemo.org Extras. Available at: http://wiki.maemo.org/Extras. Date of data acquisition 10 August 2011.
52. Maemo.org Extras-Testing. Available at: http://wiki.maemo.org/Extras-testing. Date of data acquisition 10 August 2011.
53. Maemo.org Extras-Devel. Available at: http://wiki.maemo.org/Extras-devel. Date of data acquisition 10 August 2011.
54. Nokia Developer. Nokia Publisher Guide. Available at: https://p.d.ovi.com/p/g/ovistore_static/docs/Publisher_Guide.pdf. Date of data acquisition 10 August 2011.

55. Android Compatibility Program. Android 2.1 Compatibility Definition Document. Available at: http://source.android.com/compatibility/2.1/android-2.1-cdd.pdf. Date of data acquisition 1 August 2011.
56. Android Compatibility Program. Android 2.2 Compatibility Definition. Available at: http://source.android.com/compatibility/android-2.2-cdd.pdf. Date of data acquisition 1 August 2011.
57. Android Compatibility Program. Android 2.3 Compatibility Definition. Available at: http://source.android.com/compatibility/android-2.3-cdd.pdf. Date of data acquisition 1 August 2011.