Bachelor's Thesis

Degree Programme in Information Technology

Specialization: Internet Technology

2014

Razaq Adeleke Shonubi

COMPARING APPLE'S IOS WITH SAMSUNG'S BADA MOBILE SOFTWARE DEVELOPMENT PLATFORMS

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR´S THESIS | ABSTRACT
TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Information Technology

2014 | 50 pages

Instructor: Ferm Tiina

Razaq Adeleke Shonubi

COMPARING APPLE'S IOS WITH SAMSUNG'S BADA MOBILE SOFTWARE DEVELOPMENT PLATFORMS

The providers of software development kits have made it possible for developers to be able to develop both personal and commercial applications. Due to this fact, applications are developed by third party developers and anyone interested in developing an application. In order to develop on any desired platform, it requires using a software development kits.

There are different operating systems used in the process of development but there are two very significant providers that are impacting the mobile application industry and both own major global market shares, Apple iOS and Samsung bada. The main purpose of this thesis is to compare iOS and bada SDKs, focusing on user friendliness, development process, and publishing.

The comparison was done by installing both platforms, developing an application with each of them and going through the publishing process involved in distributing each application. Every taken step is documented. The experiments of this thesis are based on the available documents provided by both Apple and Samsung in addition to materials related to mobile application development.

The aims and objectives of comparing two platforms were successful. Both platforms were installed and three applications were developed on each platform. Furthermore, application distribution process was achieved and well documented.

KEYWORDS: bada, iOS, mobile application, Objective-C, Software Development Kit

CONTENTS

ABBREVIATIONS.....7

1 INTRODUCTION.....8

2 APPLE OPERATING SYSTEM: iOS.....10

2.1 The iOS Architecture.....10

2.1.1 Cocoa Touch Layer.....11

2.1.2 Media Layer.....11

2.1.3 Core Service Layer.....11

2.1.4 Core OS Layer.....11

2.2 Overview of iOS software development kit.....11

2.2.1 Requirements for developing with iOS software development kit.....12

2.2.2 Installation of iOS platform.....12

2.2.3 Xcode.....16

2.2.4 Xcode IDE.....16

2.2.5 Simulator.....17

2.3 Application development using iOS IDE.....18

2.3.1 Programming HelloWorld for iOS device.....18

2.3.2 Developing a utility application for iOS: Calculator.....18

2.3.3 Geo-location via a web viewer for iOS.....19

2.4 Application icon and version number.....19

2.5 Deployment.....20

2.5.1. Create and download a Certificate Signing Request (CSR).....20

2.5.2 Create and download a provisioning profile.....22

2.5.3. Sign and build in Xcode.....24

2.5.4 Releasing the application into App Store.....24

3 SAMSUNG OPERATING SYSTEM: BADA.....25

3.1 Architecture of bada.....25

3.1.1 Kernel 25
3.1.2 Device 25
3.1.4 Service 26
3.1.4 Framework 26
3.2 Requirements for developing with bada software development kit 26
3.2.1 Installation of bada platform 26
3.2.2 bada IDE 30
3.2.3 Emulator 30
3.3 Application development using bada IDE 31
3.3.1 Programming HelloWorld for bada device 32
2.3.2 Developing a utility application for iOS: Calculator 32
2.3.3 Geo-location via a web viewer for bada 33
3.4 Application Icons and name 33
3.5 Deployment 33
3.5.1 Packaging an Application 34
3.5.2 Certifying and Publishing Applications 34
3.5.3 Releasing the application into Samsung App 35
4 Comparison features of the platforms 36
4.1 Installation Programming and Publishing on iOS 36
4.2 Installation Programming and Publishing on bada 36
4.3 Advantages and disadvantages of using iOS as a platform 37
4.4 Advantages and disadvantages of using bada as a platform 38
5 CONCLUSION 40
REFERENCES 41

FIGURES

Figure 1. Applications layered on top of iOS 9

Figure 2. Xcode 3 and iOS SDK 4.3.0 .dmg package 11

Figure 3. Xcode and iOS SDK packet 12

Figure 4. License Agreements for the Xcode. 12

Figure 5. Agreeing with the terms of the Xcode. 12

Figure 6. Choosing destination disks. 13

Figure 7. Selecting the components to be installed. 13

Figure 8. Xcode and iOS SDK installation in progress 14

Figure 9. Successful installation message 15

Figure 10. Xcode welcome screen 15

Figure 11. Xcode IDE 16

Figure 12. Simulator 17

Figure 13. HelloWorld running on the iOS simulator 18

Figure 14. Calculator on the iOS simulator 19

Figure 15. Geo-location running on the iOS simulator. 20

Figure 14. Setting version number 21

Figure 15. Creating a Certificate Signing Request 21

Figure 16. Certificate Signing Request created 22

Figure 17. Creating Provisioning Profile 22

Figure 18. Provisioning Profile Created 23

Figure 19. Architecture of bada 24

Figure 20. SDK download page 26

Figure 21. bada SDK installation software's welcome message 26

Figure 22. License Agreement for the bada 27

Figure 23. selecting the components to be installed 27

Figure 24. bada SDK and IDE installation in progress 28

Figure 25. bada SDK installation completed 29
Figure 26. Emulator..... 30
Figure 27. Welcome screen for development..... 31
Figure 28. HelloWorld running on the bada emulator..... 32
Figure 29. Calculator running on the bada emulator..... 33
Figure 30. Geo-location running on the bada emulator 34
Figure 31. Application names and version setting……..... 35
Figure 32. Generating the bada app package with Make Package 36
Figure 33. Privilege Check…..... 37
Figure 34. Welcome message from seller office 38

TABLES

Table 1. Comparison features 38

ABBREVIATIONS

API	Application Programming Interface
CA	Certificate Authority
CSS	Cascading Style Sheets
CSR	Certificate Signing Request
DVI	Digital Visual Interface
GB	Gigabyte
GHz	Gigahertz
GNU	General Public License
HTML	Hypertext Markup Language
ID	Identification
IDE	Integrated Development Environment
iOS	Apple’s mobile operating system
JPEG	Joint Photographic Experts Group
MB	Megabyte
OS	Operating System
PNG	Portable Network Graphics
REST	Representational State Transfer
SDK	Software Development Kit
UI	User Interface
USB	Universal Serial Bus
XML	Extensible Markup Language

1 INTRODUCTION

This thesis focuses on comparing Apple iOS with Samsung's bada by illustrating, analyzing, creating and publishing a mobile application on the two-mentioned platforms In addition it handles various aspects of iOS and bada application development, and it may be seen as a brief introductory guide to mobile application development on iOS and bada.

With an increase of mobile phone subscribers, several applications and software are in different style and design. Mobile phones are not solely improved in size and quality, but also embedded with a lot of functionality such that it can be used in place of desktops. Although, most people use their phones simply to make calls and send text messages, they also use their phones as a scanner, print via Wi-Fi, listen to MP3s, surf online, play games/ use it as a console and run several other kinds of applications. SDKs with accompanying simulators/emulators offer an easy path for engineers to understand how to utilize the software for the purpose of mobile application development no matter how novice they may be. Apple and Samsung have released their operating systems and subsequent development environments, allowing individual user and developers to create all kinds of applications on their platforms. This has led to the success of online application stores such as App Store for Apple users and Samsung's App.

The aim of this thesis is to compare installation process of Apple's iOS Software Development Kit and Samsung's bada SDK to create applications with it. Similarities and differences between the iOS and bada are discussed as well as the distribution of applications on both application stores.

Listed below is the framework of this thesis:

Chapter one deals with an overview of literature research.

Chapter two and three introduce the mobile operating system, SDK installation, and application development using Xcode IDE and bada IDE.

Chapter four discusses about the similarities, differences, advantages and disadvantage of iOS and bada software development platforms.

Chapter five provide a conclusion, summary, references and appendices of all taken processes.

The scope of this thesis work is listed but not limited to the following:

1. Installation of iOS SDK and bada SDK.

2. Developing three different applications namely, HelloWorld, Calculator and a WebView application on both platforms.

3. Publishing process an application into app store for both platforms.

The background information for this work is collected from Apple developer programs and Samsung developer program

The practical part of the thesis involves installation of both platforms and developing six applications in both platforms.

2 APPLE OPERATING SYSTEM: iOS

iOS is the operating system that runs on iPhone, iPod touch, and iPad devices. This operating system uses the device hardware and provides the technologies needed to implement an application. The operating system has embedded API apps with several system applications, such as Phone, Mail, and Safari, which provide standard system services to the user [1].

The iOS SDK comprises of tools required for application development whereby an application is built using the iOS system frameworks and programming in Objective-C language Applications are installed directly on a device and are always available to users after being downloaded via the app store. The application and any user data are synced to the user's computer through iTunes [1].

2.1 The iOS Architecture

The iOS architecture (Figure 1.) is similar to the basic architecture found in Mac OS X, iOS acts as an intercessor between the hardware and embedded or developed applications that can be utilized for any purposes on the devices. An application communicates with the hardware through a series of well-defined system interfaces that keep the hardware unchanged. This model makes it straightforward to develop applications that operate constantly on the device's hardware [1].

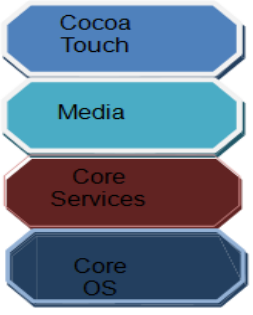

Figure 1. Applications layered.

2.1.1 Cocoa Touch Layer

The iOS application uses a framework known as UIKit for Cocoa Touch layer to implement features such as selecting a text, swiping, copy and paste, and graphical user program. The technology in this layer is based on the Objective-C programming language. This layer also offers push notification for alerting users of new information from an application even if the application is not in use [1].

2.1.2 Media Layer

The technologies in the media layer contain graphics, audio, and video technologies used in creating different multimedia experience available on a mobile device. Included features in media layer are the ability to support animation, 3D images, still images, etc. The technology in media layer is known as Core Animation and is based on Objective-C [1].

2.1.3 Core Service Layer

Core Service offers the frameworks that are used in all applications, most of which are database related. An example is Address Book framework. In addition, included in Core Service layer are critical foundation frameworks, such as the core definition of Apple's Object-oriented data types [1].

2.1.4 Core OS Layer

The Core OS layer contains the low level features that almost all other technologies are designed upon. Even though an application is not using these technologies directly, there is possibility that they are being used by other frameworks [1].

2.2 Overview of iOS software development kit

The iOS SDK was released on 6th of March, 2008, and developers had Apple's authorizations to develop "user-friendly applications" for the iPhone, iPod, and two years later for iPad. With the help of "iPhone simulator", developers can test run their application and get the full picture of how the application will function as well as its features. SDK has an environment known as Xcode where developers can write their codes in Objective-C with some parts written in C or C++. The SDK contains the commonly used frameworks and tools for iOS development. It has frameworks for interface design, network communication, database communication, audio and graphics.

2.2.1 Requirements for developing with iOS SDK

The requirements for Xcode are Mac OS X Snow Leopard version 10.6.7 or newer version with minimum of 10GB of free disk space. Snow Leopard Mac OS X 10.6.7 was the first OS to support Intel based products only, so essentially all Mac computers manufactured after the release of Snow Leopard on in 2009 should be able to run Xcode. However, there is an exception with OS X version 10.8.0 Lion, due to incompatible of Xcode 3 and Xcode 4. Therefore, it requires download via Mac App Store [2].

2.2.2 Installation of iOS platform

This chapter walks through the Mac OS X and the Xcode 3 installation process.Being able to download the iOS SDK and Xcode requires signing up as Apple Developer. The SDK is available in Apple's Developer site: http://developer.apple.com/xcode/. Included in the extracted package is an introductory pdf files about Xcode and iOS (Figure

2.). Installation was started by double clicking iOS SDK mpkg file (Figure 3.) as instructed in the pdf file.

Figure 2. Xcode 3 and iOS SDK 4.3.0 .dmg package.

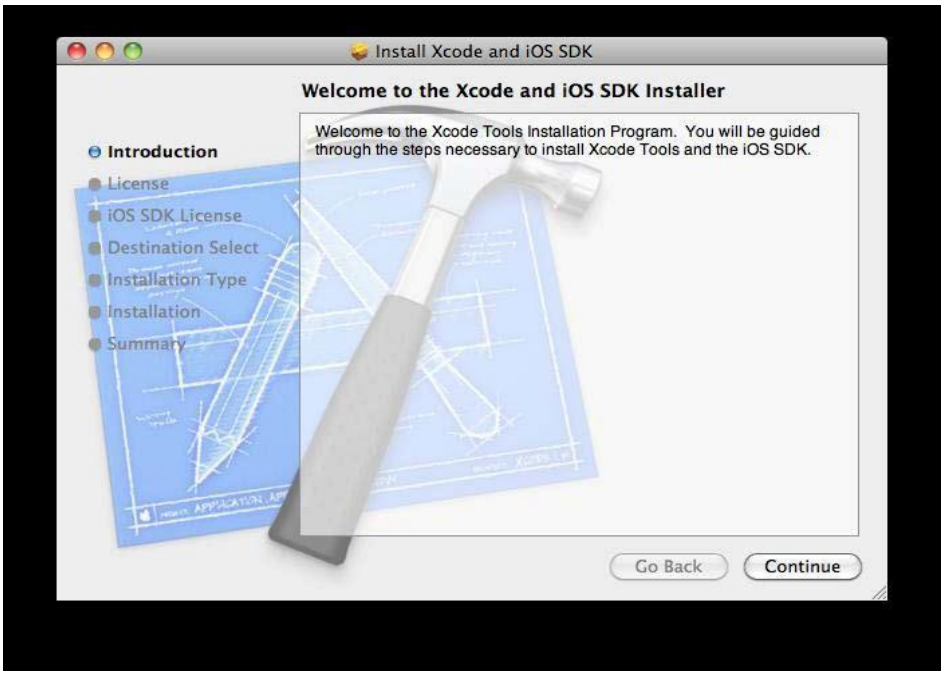

Figure 3. Xcode and iOS SDK packet.

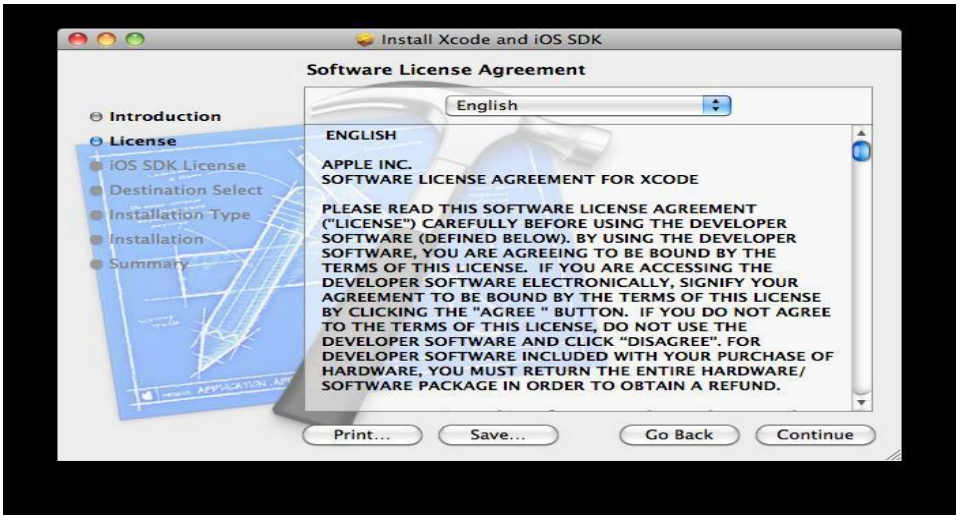

Figure 4. License Agreements for the Xcode.

Figure 5. Agreeing with the terms of the Xcode.

License agreement is required to be agreed upon before installation process can continue as shown in Figure 4. and Figure 5. The next step is to choose the install destination (Figure 6.) and next to continue installation.

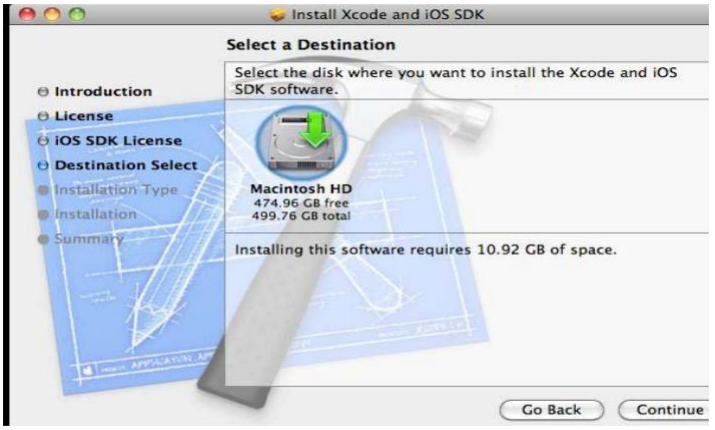

Figure 6. Choosing destination disks.

The next step shows all required packages which are: “Essentials, System Tools, UNIX Development and Documentation” Mac OS X 0.4 SDK was skipped in this thesis process because by default Mac OS X SDK and iOS SDK are installed, including iOS Simulator (See Figure 7. for details).

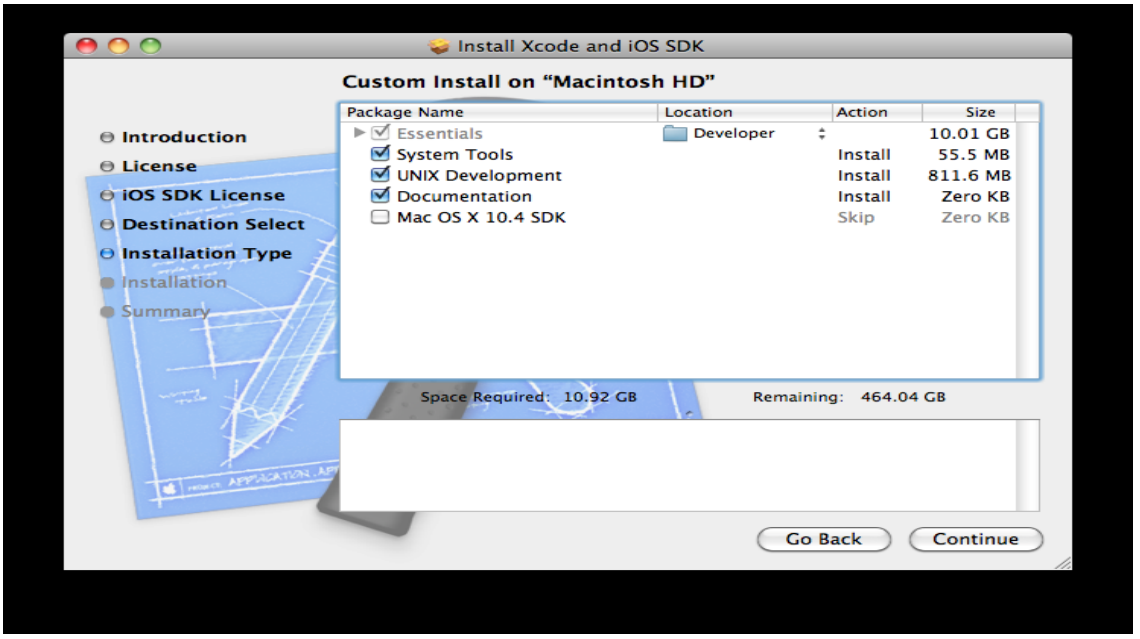

Figure 7. Selecting the components to be installed.

Having selected the required tools, the actual installation process starts (Figure 8.). The estimated time (51 minutes) it will take for the packages to be installed varies due to network.

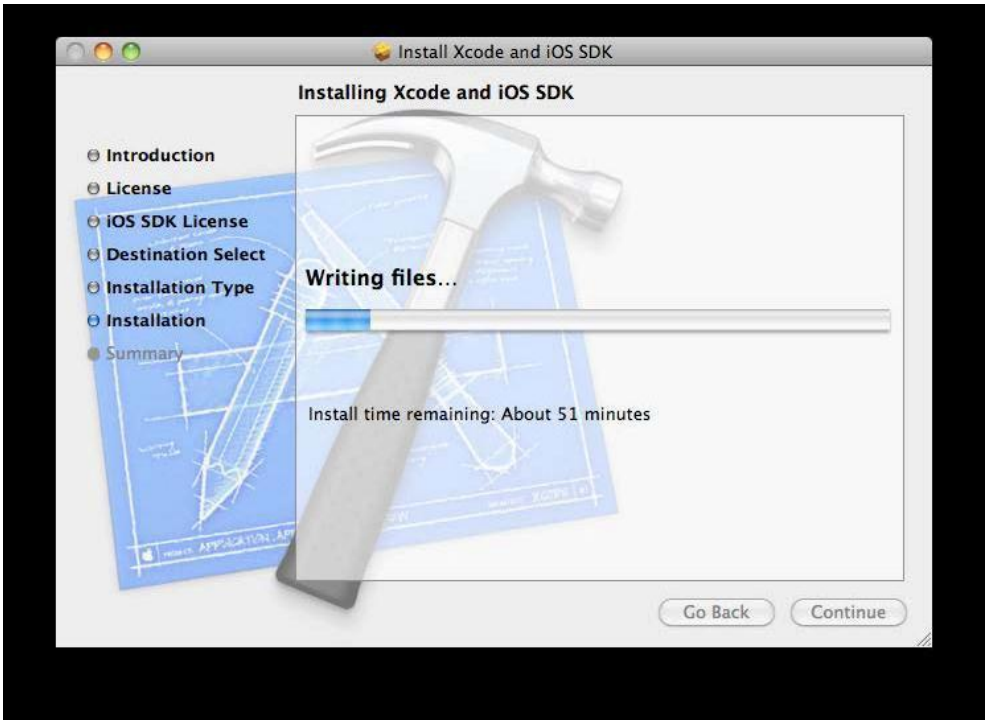

Figure 8. Xcode and iOS SDK installation in progress.

Approximately thirty-five minutes later, a successful software message was displayed by the installer (Figure 9.).

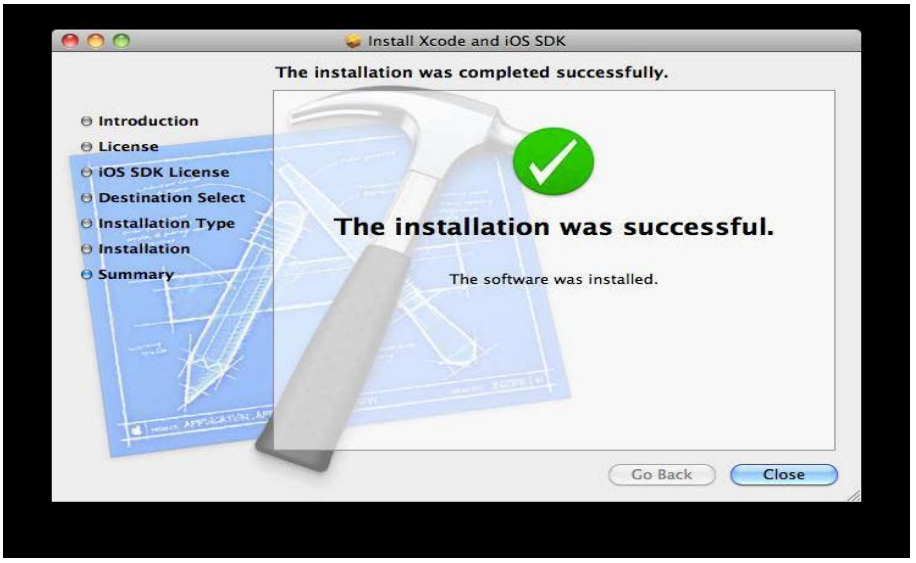

Figure 9. Successful installation messages.

2.2.3 Xcode

Xcode is the complete toolset for building OS X and iOS applications. Aside from developing applications for iOS devices with the Xcode, it can be used to develop applications for Mac, too. The Xcode toolset includes the Xcode IDE, with the Interface Builder design tool and Apple LLVM compiler fully integrated. The Instruments analysis tool is also included, along with some other supporting developer tools [3].

Figure 10. Xcode welcome screen.

2.2.4 Xcode IDE

Xcode integrates all the tools required for successful application development. The Xcode workspace is Xcode (Figure 10.). As the user types, Live Issues will immediately give alert to any coding mistakes that may lead to code error while running, displaying a message bubble beside the code for more details for correction. The Run button is used to launch the Mac application (app), and upload it to the test device for immediate debugging [3].

Figure 11. Xcode IDE.

2.2.5 Simulator

The iOS Simulator (Figure 12.) simulates the execution of the intended application in much the same way as working on the actual iOS device. It is also recommended to check every step of the development process by saving and running the code to see how the app works. Due to the fast mode loading and debugging, the simulator is perfect for testing an app in order to be sure that the UI works perfectly as intended [3].

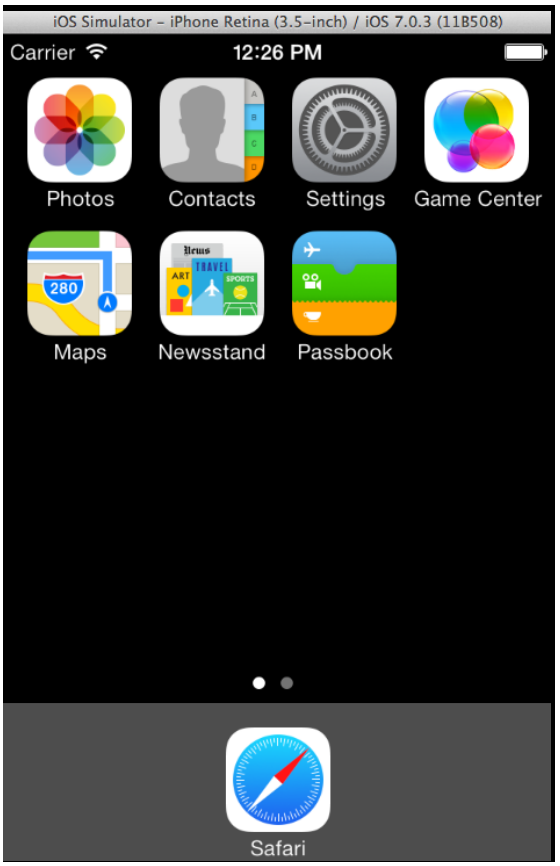

Figure 12. Simulator.

2.3 Application development using iOS SDK

Objective-C is the programming language used in the iOS SDK with a file extension name .h for header files which contain class, type, function and constant declarations; .m files are source files which contain both Objective-C and C language and .hh files are source files, but contain C++ code in addition to Objective-C and C code. The application code is not complete without a nib file with an extension name .xib, which contains the application's user interface codes; this file is automatically created by the Xcode when a view-based project is created. Nib files cannot be read with a source - or a hex editor, since they are in a proprietary format. Instead, a separate application called Interface Builder is used for designing the UI [4].

2.3.1 Developing HelloWorld for iOS

HelloWorld is often the kickoff point for programming tasks. The conduct of this example is to show a similar programming of HelloWorld as an application, but an action will be required from users to tap a button whereby the application will

displayed the greeting message “Hello, User!”. After implementing and compiling the code without encountering errors, iPhone Simulator displays the application as it will work on a target device (Figure 13.).

Figure 13. HelloWorld running on the iOS simulator.

2.3.2 Developing a utility application for iOS: Calculator

The next implementation is to develop an application whereby the user will need to input some figures to be calculated and display the result. After completing the designing and programming of the application, it was tested on the simulator for several days. See (Figure 14.) for screen shot from the simulator.

Figure 14. Calculator.

2.3.3 Geo-location via a web viewer for iOS.

The third experiment is to use a browser directly from the simulator and view a map website. The reasons behind this third experiment are two. One, is to take note if the platforms view web pages in a different ways. Two, is to see if user exact location can be determined. Figure 15. showing current location at the time of experimenting.

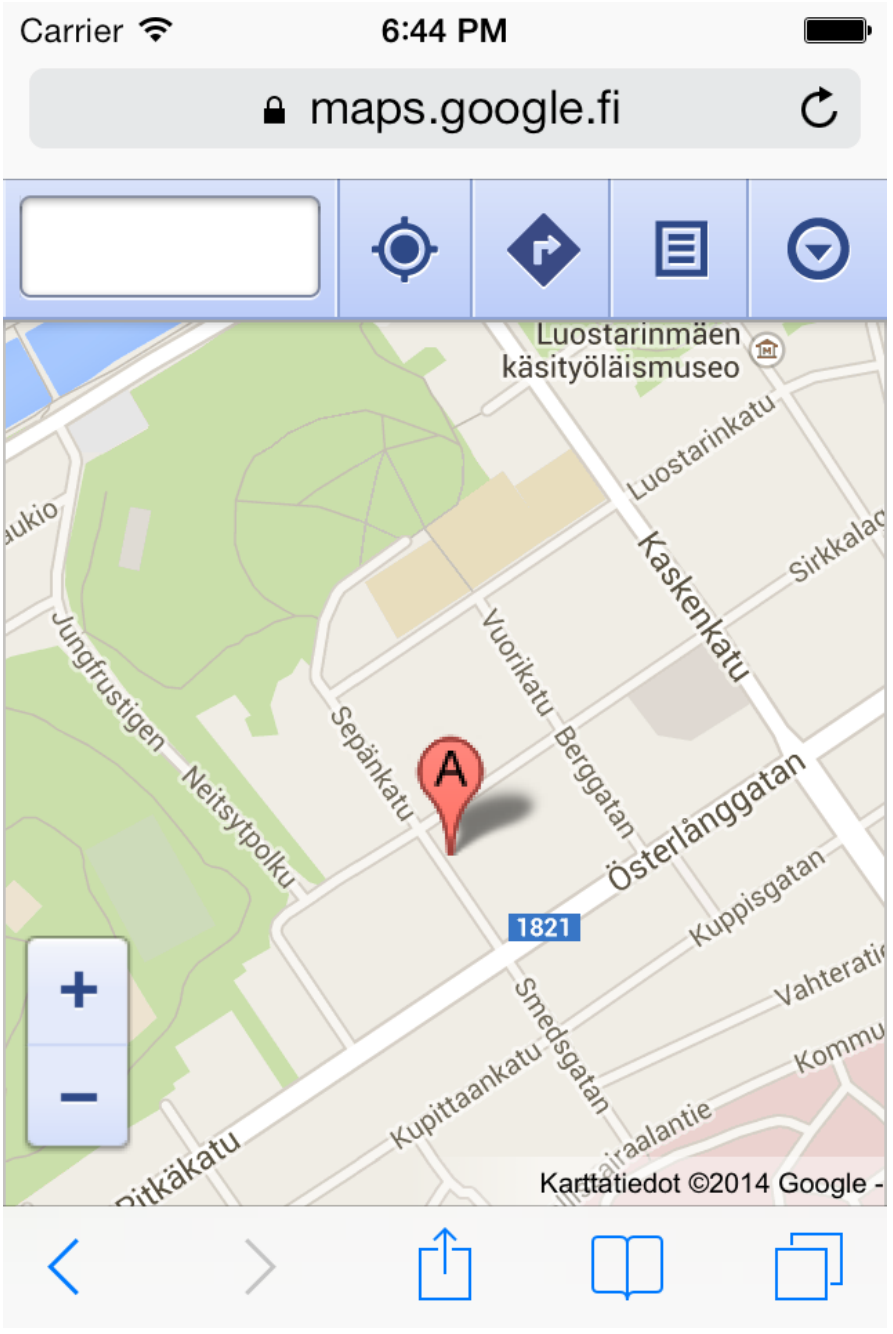

Figure 15. Geo-location.

2.4 Application icon and version number

The application icon is used to symbolize the application on the devices home screen. All icons have unique sizes, for instance, an iPhone application icon should be a 57 x 57 pixel in either a JPEG or PNG file. Adding the icon to a project is done by dragging and dropping the image file into the “Resources” folder. Moreover, it can be added by selecting “Resources” and choosing

Project > Add to Project, and browsing the file from the system.

There are different ways of setting the project version number via Xcode one example is to navigate from “Build setting” > “Versioning” and modify the default “Current Project Version” see Figure 14. for details. Note that; the value must be an integer or floating point number, such as 2012 or 24.7.

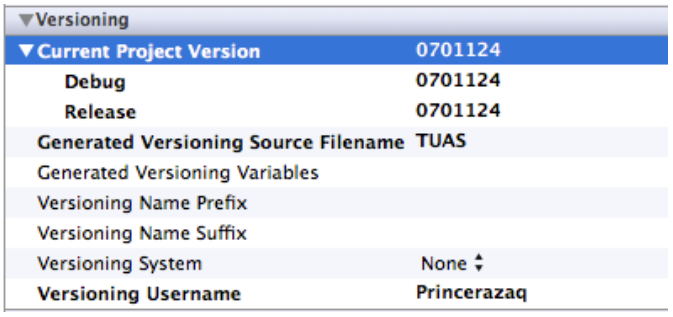

Figure 14. Setting version number.

2.5 Deployment

After the completion of coding and successful building there are some steps that need to be taken for distribution of your app via App Store. To prepare an app for distribution, it is necessary to obtain an enterprise distribution certificate from Apple and sign the code in Xcode [4]. To begin the process, the app needs to be certified and provisioned within the iOS Developer, e.g., Enterprise Program, and then signed and built in Xcode.

2.5.1. Create and download a Certificate Signing Request (CSR)

To distribute an iOS app, the designated Agent in your Developer Program membership will need to create a distribution certificate, by locating the Keychain Access Utility via the application folder on the Mac machine used to create the application. Further, the CSR should be submitted into iOS Provisioning Portal on the member center of Apple Developer page, Figure 15. And Figure 16. show the process taken and the obtained certificate.

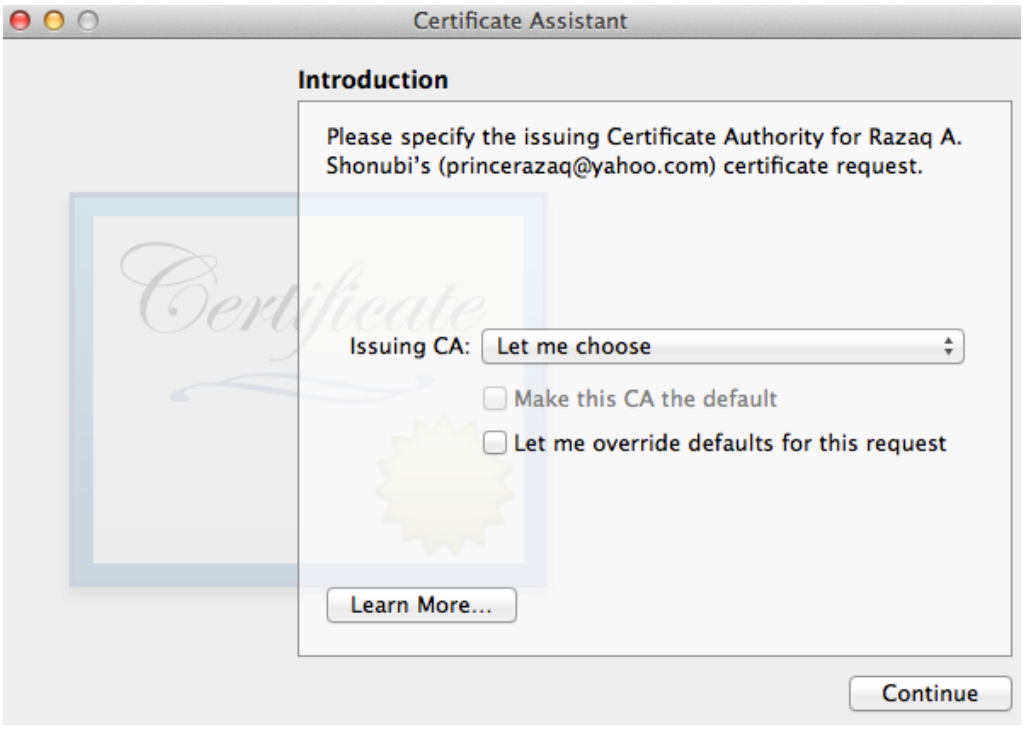

Figure 15. Creating a Certificate Signing Request (CRS).

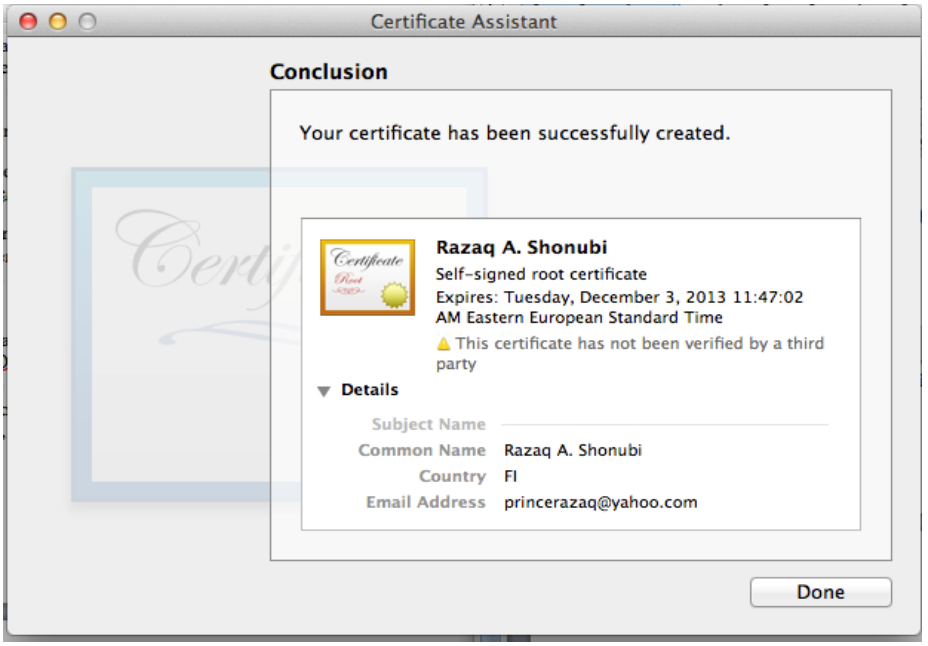

Figure 16. Certificate Signing Request created (CRS).

2.5.2. Create and download a provisioning profile.

After the creation of CRS, the next step is to create a Provisioning Profile for the app. This can be done from the Provisioning section of iOS Provisioning Portal by selecting “New Profile”, entering a name for the profile, selecting the certificate, App ID, device, and submit the form. The profile will show immediately on the Provisioning page for download (Figure 17. And Figure 18.).

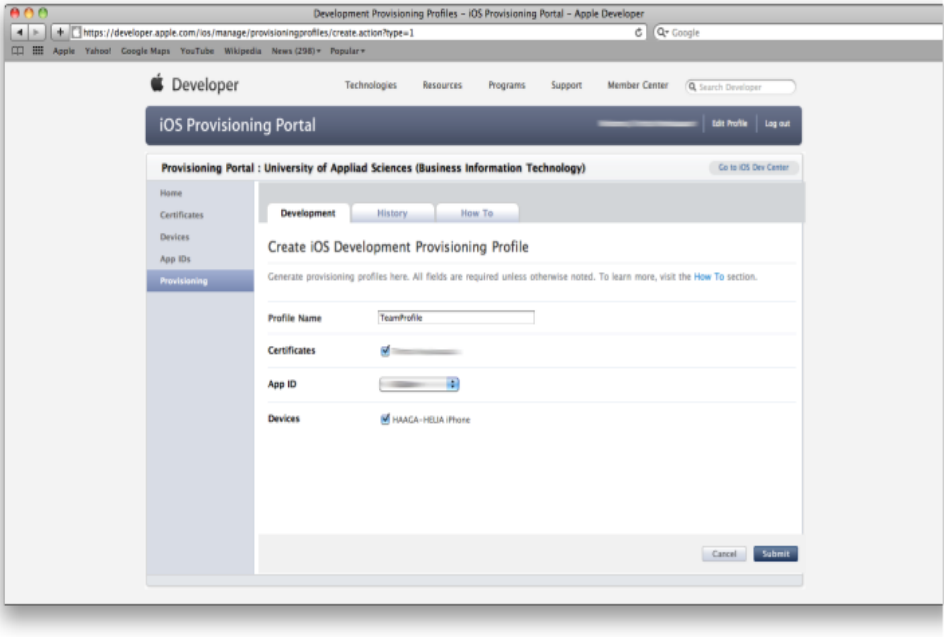

Figure 17. Creating Provisioning Profile.

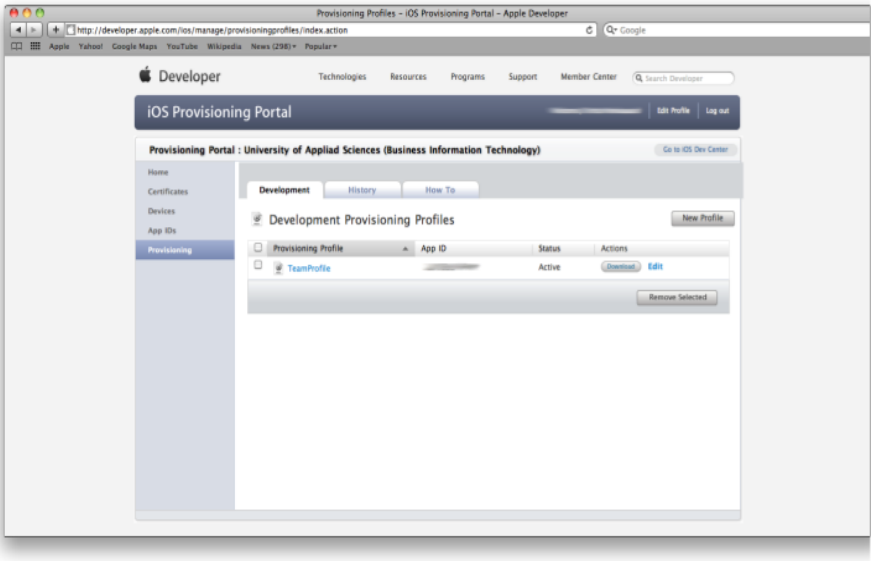

Figure 18. Provisioning Profile Created.

2.5.3. Sign and build in Xcode.

Once the distribution certificate and provisioning profile are installed, the code needs to be signed in Xcode. When the app is signed, Xcode packages it for enterprise distribution with a single export process. Xcode Organizer is used for sharing and selecting the options for the newly added project in the archive. This process automatically packages the app, the provisioning profile, and other elements needed for wireless distribution [5].

2.5.4 Releasing the application into App Store

Releasing an app requires being a member of iOS Developer Program and the registration fee in 2013 is $99. After deployment, the final stage is launching the app by using an iTunes Connect account to validate the application, in order to add the app information before launching it to the store. The iTunes account is the same as the iOS Developer Account. Note that the Team Admins have enough privileges to perform this operation [5].

3 SAMSUNG OPERATING SYSTEM: bada

Samsung bada is a mobile operating system for smartphones and it is the OS running on wave series devices, presented in 2010. The word bada in Korean means ocean and seashore. bada can accommodate countless applications developed by its users and it offers a fascinating space that provides exceptional satisfaction to developers.

3.1 Architecture of bada

Figure 19. shows the breakdown of architecture in bada

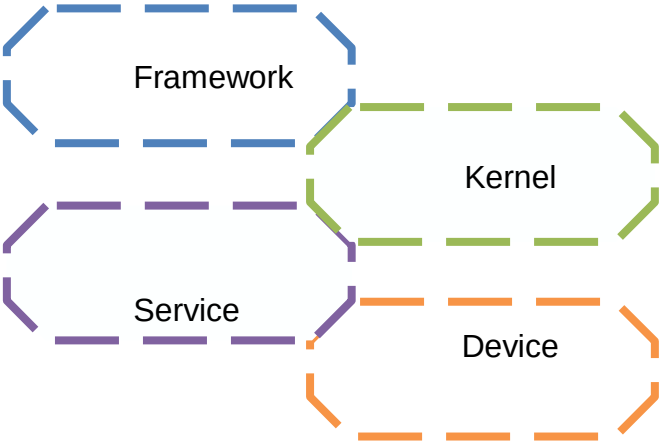

Figure 19. Architecture of bada.

3.1.1 Kernel

This layer contains either the real time operating system or the UNIX kernel, but has to do with the hardware configuration of a device in use of the application and it is known as the OS kernel.

3.1.2 Device

Here is the layer that holds altogether the core functions of the mobile device platform that are needed by the device OS, for example, the multimedia and graphics functionality; In addition, in this layer are the communication

components essential for the application. The following are the included further instructions window management, telephony, graphics and security.

3.1.3 Service

On this layer, there are services concerned with functions that are provided by an application component and server assisted elements. An application component provided by the service layer includes messaging instruments and contact. The server assisted features are provided by **Representational State Transfer** (**REST**), Web-service components that interconnect with the service components. Access to these features is possible via APIs (application program interfaces) on the Framework layer.

3.1.4 Framework

This layer contains C++ and Web frameworks of bada. The C++ framework consists of an application framework, interfaces and classes that offer access to the functionality of the underlying layers. The interfaces provided by an open API framework embody some basic interfaces that are necessary for all applications in handling basic device features, utilities, and for creating the user interface. The Web framework provides well-established standards and features, such as WAC 2.0, HTML, CSS, and JavaScript, as well as JavaScript-based cross-platform APIs for UI controls and events [6].

3.2 Requirements for developing with bada software development kit

To develop a mobile device application on bada platform requires having a computer running Windows XP 32-bit or later and at least 2 GB of RAM memory and the screen resolution of the system must be greater than the resolution of the Emulator. Additionally, required is a Samsung account, for downloading the SDK from the bada developer site.

3.2.1 Installation of bada platform

This chapter deals with the bada SDK and IDE installation process. The SDK/IDE is available on the bada developer site: http://developer.bada.com/devtools/sdk#. After downloading the installer, bada_SDK_<version>.exe, package, the file is shown on the desktop and double click on the folder to extract the files.

The SDK set up wizard starts and we selected the entire required platform binary, followed by Next and click on Install. Few minutes later, the SDK set-up wizard notified that the installation is complete. Included in the extracted package is a getting started with bada introductory pdf files.

Figure 21. bada SDK installation software's welcome message.

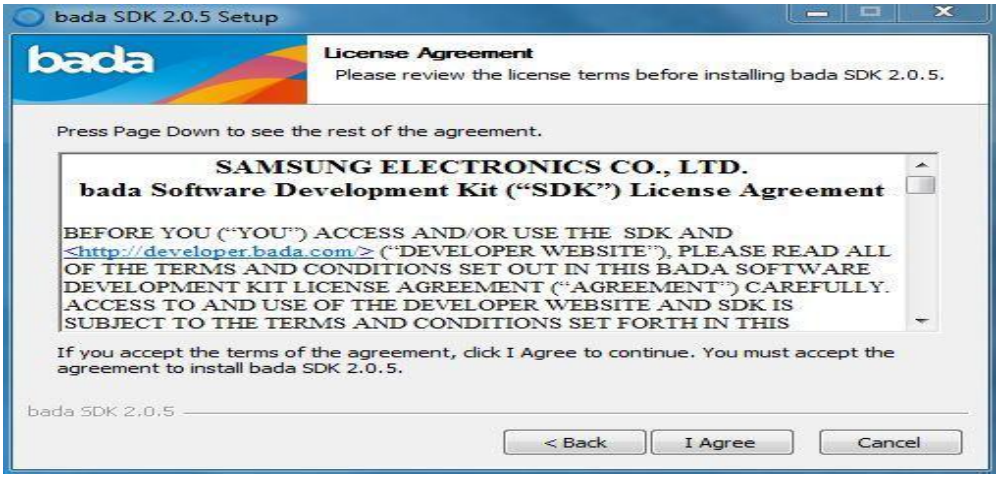

Figure 22. License Agreement for the bada.

The license agreement is required to be agreed upon before the installation process can continue (Figure 22.). The next step shows the required components that maybe needed for a target development which are: Platform Binary, WVGA, HVGA, and WQVGA. See Figure 23. for details.

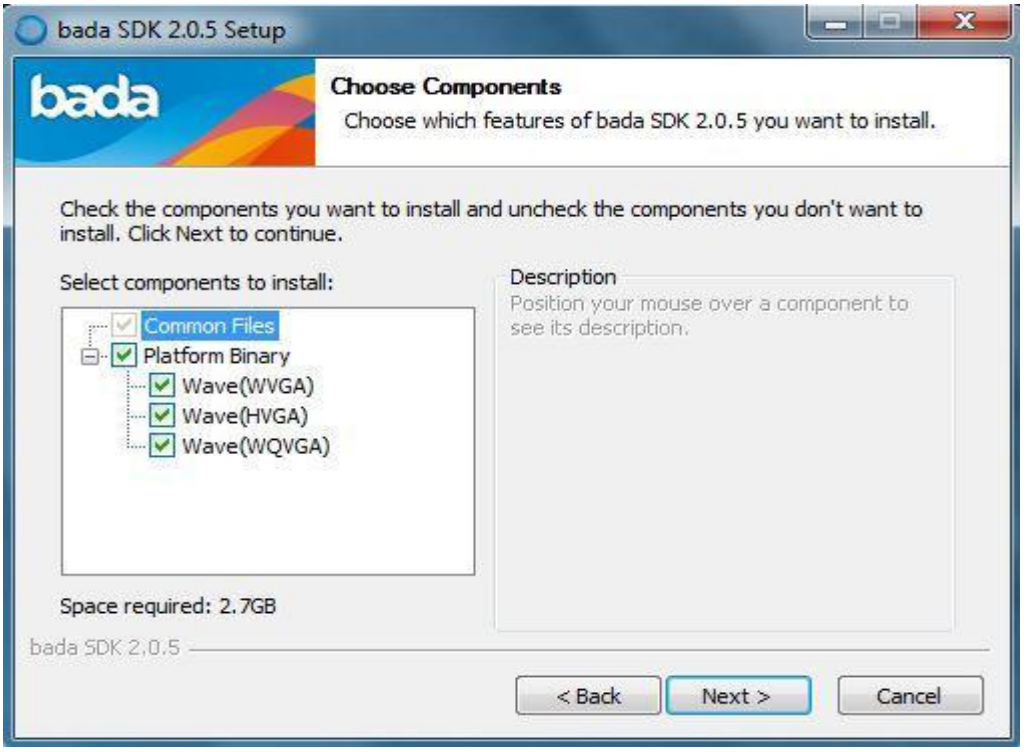

Figure 23. selecting the components to be installed.

Having selected the required components, the actual installation process started (Figure 24.) and it took about 45 minutes for the components to be

installed but this may vary depending on network packages.

Figure 24. bada SDK and IDE installation in progress.

Four-five minutes later, a completed software message was displayed by the installer; (See Figure 25. for details).

Figure 25. bada SDK installation completed.

3.2.2 bada IDE

The bada IDE is an environment that offers a set of tools for application development, for example, UI Builder, debugger and C++ editor. The IDE is associated with an Eclipse development tools. These tools are supported with CDT features, like content assistance, code folding, syntax highlighting, and tabbed documents. Hence, bada IDE supports in developing applications that are linked to the UI and its debugger helps in any programming mistake [7].

3.2.3 Emulator

The emulator enables developers to use a target device-optimized environment to debug and test an application on a local machine before deploying it to the real target device. It uses native debuggers that enable developers to control the application's execution. Figure 26. illustrates the emulator control keys and menu which lets users easily control the emulator environment [7].

Figure 26. Emulator.

3.3 Application development using bada IDE

The programming language required in developing an application with bada IDE is C++ and it has different file extensions, such as .cpp, .hxx header file, .xml extensible markup language file. The most important of all the files is the manifest.xml because it contains all details needed to develop an application. The manifest file is generated for download after the creation of application information.

Figure 27. Welcome screen for development.

3.3.1 Programming HelloWorld for bada device

The development started similarly to the startup programming tasks HelloWorld. The process taken to achieve these requires firstly giving the application details on the bada website in order to generate a manifest file. This is followed by opening the manifest file using bada IDE and lastly, developing the application. After executing and compiling the code, the application was displayed on bada Emulator. See Figure 28. for screen shot.

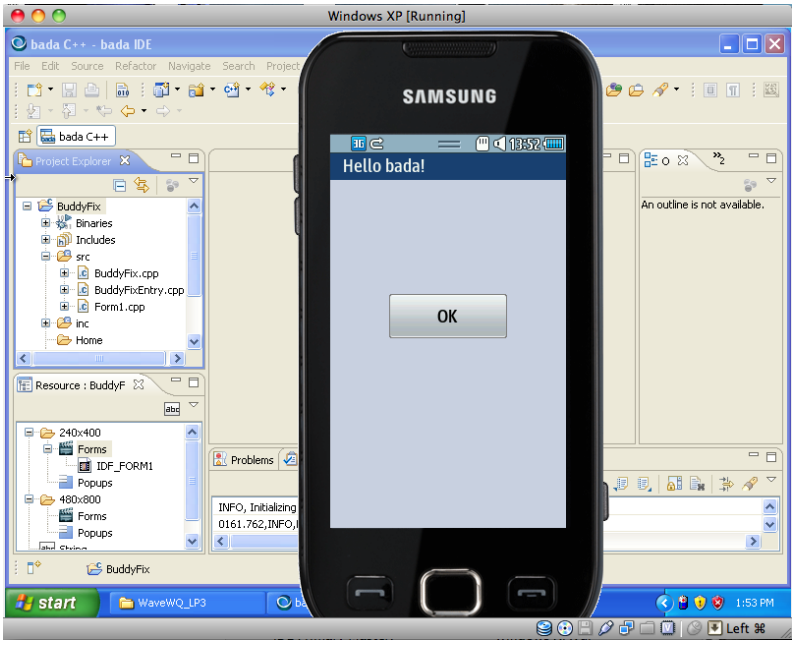

Figure 28. HelloWorld running on the bada emulator.

3.3.2 Developing a utility application for bada: Calculator

The next implementation is to develop an application whereby the user will need to input some figures to be calculated and display the result. Figure 29 shows the screen shot of the application during testing on emulator.

Figure 29. A calculator.

3.3.3 Geo-location via a web viewer for bada

Lastly, in this experiment it shows the accurate location during experimenting by navigating through a map web site and clicking on location icon. This experiment is to develop a web viewer application and browse the different web sites. See Figure 30 for details.

Figure 30. Geo-location.

3.4 Application Icons and name

There are two mandatory icons in bada; the first one is intended for the main menu and the other icon is intended for launching the application whereby a splash image is displayed during application initialization. Both icons could be in JPG file or PNG file and the icons are added to an application by defining them via the icon panel.

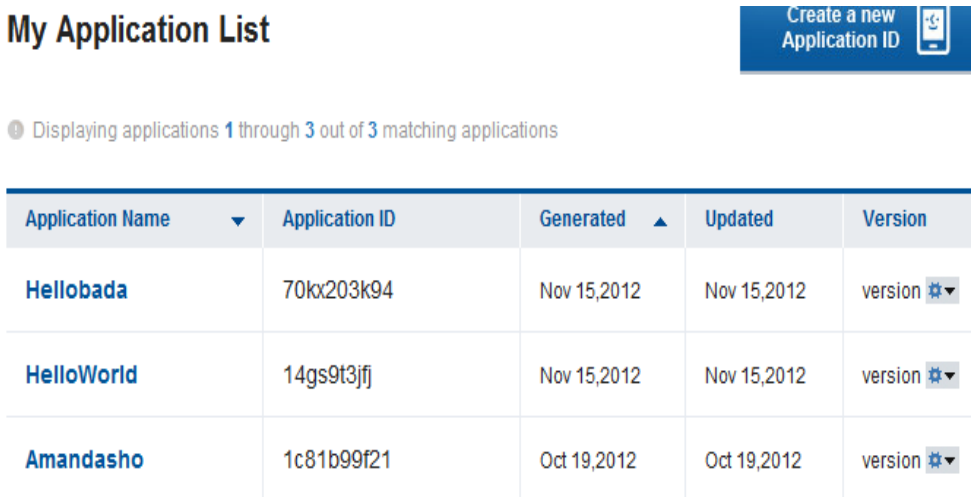

Figure 29. Application names and version setting.

3.5 Deployment

After successfully testing the application on the Emulator and the target device, the next step is packaging the application. There are two forms of packaging an application either by using bada IDE, or by using a command line command to build and package it at the same time. The command line command is only available for C++ and flash applications [7].

3.5.1 Packaging an application

In bada, application packaging is done by generating a package file to be distributed to Samsung Apps. A package is generated in the form of a zip format, which can be registered to Samsung Apps. Note that, the account created on the bada developer site can be used as a login account in Samsung Apps. Figure 30. shows the details of making the package.

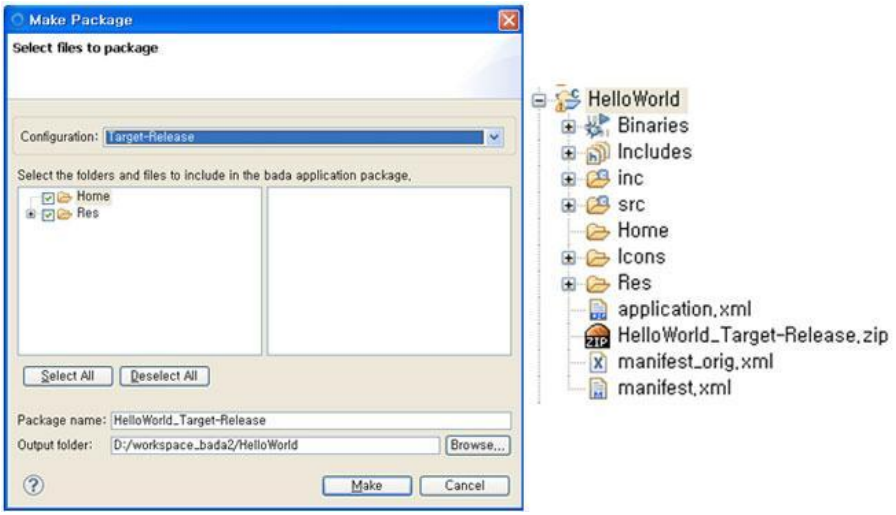

Figure 30. Generating the bada app package with Make Package.

3.5.2 Certifying and Publishing Applications

The final process is to check a privileged API group configuration of the application. The manifest.xml file contains all needed information about the application including privilege API of an application. After checking the privilege API and taking note that the app information is accurate, it is necessary to re-download the manifest.xml file and build the application again before distributing the application. Figure 31. shows the privilege checking.

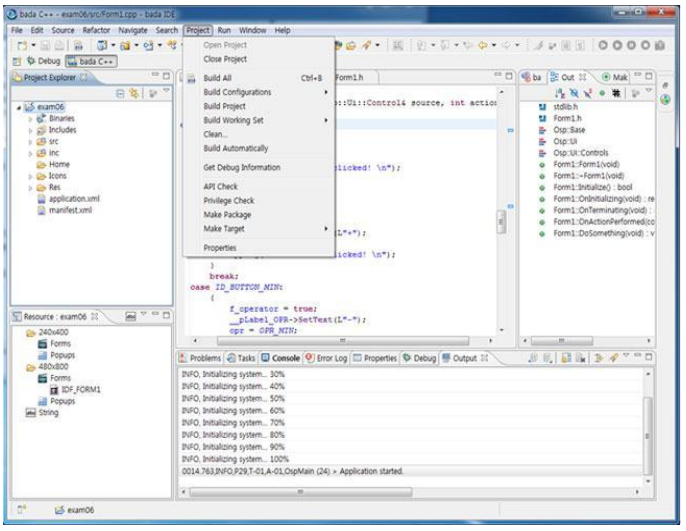

Figure 31. Privilege Check.

3.5.3 Releasing the application into Samsung App

Finally, publishing the app requires login to Samsung App Seller. There are two options of selling an application in the store namely, private seller or Cooperate seller. Private sellers are for individuals who want to sell under their personal information such as name, etc. while the cooperate sellers are companies and any member of the company that can sell the application under the company name with a unique ID.

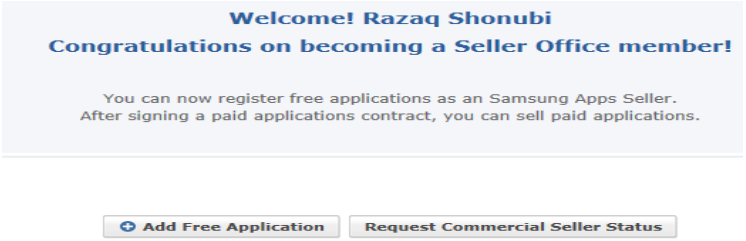

Figure 32. Welcome message from seller office.

4 COMPARISON FEATURES OF THE PLATFORMS

4. 1 Installation Programming and Publishing on iOS

The iOS platform is direct; installation file includes all the necessary components and steps by step guides from start to finish. There are two acts required by developers in regards Xcode. Firstly, the developers have to decide if they need a Mac OS X SDK package to be included and select a destination where the packages should be installed. However, by choosing a default setting all that is required is to keep clicking "next" until the end of the installation process.

Another feature a beginner will appreciate is the Xcode quick help, which may be opened within the code editor using alt key and double clicking an icon within the editor window. In addition, it provides an access to the reference documentation related to the needed help. The linking of the toolsets was clear and easy to understand unlike the corresponding tool within the bada IDE. After compiling codes, the simulator displays the app immediately for testing the app.

On the other hand, During the cause of the experiment, it was discovered that Apple do not have incremental update system for their software Xcode meaning that installed Xcode 3 cannot be upgraded to Xcode 4.0 or later version. The choice is to either keep using the older version or uninstall the older version and download the newer version.

Publishing in App Store is complex and it requires an annual fee to be able to publish in the App store. This thesis was able to give the steps and certification in publishing. Firstly, creating and downloading a Certificate Signing Request (CSR) followed by creating and downloading a provisioning profile. And lastly, the codes need to be signed and built m in Xcode.

4.2 Installation Programming and Publishing on bada

bada installation seems complex, after downloading the zip file of the software and extracting the files. After installing, the IDE should be copied from the SDK folder to a different folder because it would not start from the SDK folder. On the bada documentation, there was no instruction stating the need to copy the IDE from one folder to a new location.

However, after considering all processes involved in installation of bada SDK, it was not more problematic than the installation of some other similar software. Being on a safer side, it is best to install by default. What is more, it is possible to customize the installation if a developer needs some specific tools for application development.

The bada installation contains some very important files such as platform binaries and libraries, Header files, Emulator, Documents and Sample application. The bada IDE is an Eclipse-based platform and includes File explorer, UI builder, and Application wizard as additional plug-ins.

The programming language use in developing a bada application is C++ which enables easy startup of an app development immediately after installing the IDE. However, bada is restricted to very little documentation which made it very difficult to understand the techniques in bada and it requires reading and repetition in compiling.

The bada publishing process is straightforward and it is free to distribute application in the Samsung apps. All I did was to register as a private seller and I gave the details of my application and it was done. The approval took five working days and it is possible to monitored it via MyApplication within the Application tab. normally, the approval takes seven working days and a game application may takes up to two weeks.

Table1. Compare features.

	Apple	Samsung
Platform	iOS	bada
Development tool	Xcode IDE	bada IDE
Programming	Objecive-C	C++
App distribution	App Store	Samsung Apps
Operating system	OS X	Windows
Usage size	4.74 GB	641 MB
Development fee	$99	Free

4.3 Advantages and disadvantages of using iOS as a platform

Apple's policies in terms of third-party software uses have different rules and regulations and developers have to comply with Apple's software policies. Moreover, iOS installed package includes all the required tools needed to develop an application for any of their devices but the fault is that there is no availability of incremental upgrade system for their software. This was noted after Xcode 3 was installed and there was a later version of it, It was not possible to upgrade to a newer version

However, Apple has good documentation on their developer site, and there are several articles about iOS and object- oriented programming, syntax guides, example codes and information in regard with iOS platform in general. In addition, there are numerous guides and information from the start-up of an application development to its release to App Store. Whereas in bada most of documentation comes from Samsung's developer site and it is certain that the information is authentic.

An additional problem a developer of iOS applications may encounter is that, Apple owns the complete right and control of all the codes submitted by developers. Apple decide on which application will be published or not published in App Store with developers having no say in this decision. Furthermore, there is only one channel for apps distribution worldwide.

4.4 Advantages and disadvantages of using bada as a platform

bada is a growing platform in terms of popularity, efficiency and its development never stops as it has happened with iOS that have been in the industry for years. The bada official SDK platform is Windows making it easier for accessibility compared to Mac OX. Moreover, few people are aware of its existence due to the fact that Samsung released bada OS for only its Wave series mobile phones.

However, bada has a low number of available applications in its Samsung's App store and this has been due to a lack of a powerful core in its OS, and it has resulted in limited application development, but Samsung claims to be working on this problem. bada devices are cheaper compared to iOS devices this may be one of the reason bada has a large market share.

One problem encountered during the experiment on bada was that there is limited literature available from third- parties/ authors compared to iOS which has numerous journals, reports, etc. Nevertheless, the bada developer site could a source of information for application development. What is more, bada is free of charge, has a free SDK and developers can freely publish on the Samsung Apps. bada is worth considering as a platform and there is nothing to lose for trying it out.

5 CONCLUSION

The goals and objectives of the experiment work were to compare Apple's iOS with Samsung's bada software development platforms, to install both software development kits, check their functionality, developed an application on both environments and carried out a proper publishing process of both platforms.

It was noted that all the above mentioned experiment were performed. Both SDKs were installed; some applications were develop on the platforms, test and document the publishing process of the software's. The best part of the experiment was that it gives new and priceless information, such as, choosing the right SDKs, software requirements, publishing and mobile applications in general. Further, I found a lot of differences and beneficial uses of both platforms.

For instance, it was noted that there was little differences in the manner the platforms displayed a web page. Although the difference may not be noted if not compared. By taking a proper look at Figure

15. and Figure 30, it will be noted that in the web viewer in bada, clicking the location icon, it displayed the street, postal code and city name but in iOS it displayed only the geo-location. However, application developments in both platforms are the same.

This thesis as imparted a lot of knowledge of different kinds, ranging from software distribution via app store to third- parties development, using two different SDKs and finding the differences between the SDKs. Hopefully people will find the outcome of the experiment helpful. The thesis contains all the required documentation of the process taken during the experiment, and it can be used as a guideline for, mobile application development, platform consideration and for educational purposes.

In conclusion, both SDKs used in the experiment have their strengths and weaknesses, they are both suitable for the same project with different processes. Anyone developing application on the iOS platform needs to be familiar with the Apple computers and mobile devices because the iOS SDK requires having a Mac machine to use the software. Likewise bada SDK requires having a Windows operating system.

On the other hand, the iOS installation is just a step-by-step click to the end while the installation process in bada is more complicated. Nevertheless, if taking publishing application into consideration, bada will be on top due to the fact that its publishing mode is straightforward, but iOS requires plenty of digital certification to publish. Apparently, neither Apple's iOS software development platform nor Samsung's bada software development platform is better.

REFERENCES

[1] Apple: iOS Technology Overview: Cocoa Touch Layer Apple Developer

http://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOerview/iPhoneOSTechnologies/iPhone_OSTechnologies.html (Accessed March 21 2011)

[2] Runrev: How do become an iOS developer, Available at:

http://lessons.runrev.com/s/lessons/m/4069/l/23275-how-do-i-become-an-ios-developer (Accessed March 21 2011).

[3] Apple: Developer Tools, Available at:

https://developer.apple.com/technologies/tools/ (Accessed March 21, 2011)

[4] Allan, A. 2010. Learning iPhone Programming: From Xcode to App Store. Sebastopol, CA: O'Reilly Media, Inc.

[5] Hegarty P. (2010) Developing Apps for iOS. Stanford University Podcast, Available at : http://itunes.apple.com/us/itunes-u/developing-apps-for-ios-hd/id395605774

[6] bada: What is bada. Available at: http://www.bada.com/whatisbada/index.html

http://jyrom.tistory.com/category/Bada (Accessed January 29 2012). [7]

bada SDK installation. Available at:

http://www.badaforums.net/bada-developers/how-to-use-bada-ide-complete-guide-to-bada-developer/ (Accessed June 3 2012)