ANALYSIS OF MULTI-PLATFORM MOBILE APPLICATION DEVELOPMENT

by

Courtney B. Thaden
Bachelor of Science, University of North Dakota, 2010

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota
May
2014

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

This thesis, submitted by Courtney B. Thaden in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done, and is hereby approved.

Naima Kaabouch

Naima Kaabouch, Chairperson

A. R. Miles

Arthur Miles

04/25/14

Saleh Farque

This thesis is being submitted by the appointed advisory committee as having met all of the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

Wayne S. Swisher

Wayne Swisher
Dean of the Graduate School

april 28, 2014

Date

Title	Analysis of Multi-Platform Mobile Application Development
Department	Electrical Engineering
Degree	Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in her absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

Courtney B. Thaden
April 25, 2014

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my original advisor, the late Dr. Richard Schultz, who through his vision and hard work sought to create a learning environment to inspire individuals to reach their potential. I wish to express my appreciation to my current advisor, Dr. Naima Kaabouch, for her assistance in helping me find a career I love. Thank you also to my committee members, Dr. Arthur Miles and Dr. Saleh Faruque for their knowledge, expertise, and friendship.

Thank you to my mother and siblings respectively, Betsy, Geoffrey, and Sarah for your understanding and patience throughout this process. I love all of you more than you will ever know. A special thank you to my mother and Sarah for their constant encouragement and superb editing skills.

I dedicate this paper to all the strong females in my life, past and present, especially my mother Betsy, my sister Sarah, and my late grandmother Winnie.

ABSTRACT

The variety of mobile devices and their operating platforms has rapidly increased. With this increase come separate standards, programming languages, and distribution markets. Typically developers want to deliver their products to a variety of users encompassing various platforms; however choosing to develop using a native program for a platform can delay the development and release on another platform. Multi-platform development applications were created in order to deploy applications to various platforms in a more timely and cost efficient manner by using a single code base.

The purpose of this study was to investigate the multi-platform development applications MoSync, Appcelerator, and PhoneGap, create a test application using each multi-platform development application to run on the Android emulator and iOS simulator to determine performance, and also determine which multi-platform application was best suited for allowing a developer to create a mobile application that could be utilized on a variety of platforms.

CHAPTER I

INTRODUCTION

Introduction

Mobile device application development has increased with the rising number of smartphones on the market (Boardman, 2012; Tech Terms, 2014). The variety of smartphone devices is ever expanding, as well as their powerful operating platforms (Charland & Leroux, 2011). Each platform involves separate standards, programming languages, and distribution markets (Corral, Sillitti, & Succi, 2012b). Typically developers want to deliver their products to a variety of users encompassing various platforms; however choosing to develop using a native program for a platform can delay the development and release on another platform. Multi-platform development applications were created in order to deploy applications to various platforms in a more timely and cost efficient manner, with the principle idea of "develop once, deploy everywhere" (Blom, Book, Gruhn, Hrushchak, & Köhler, 2008; Corral et al., 2012b).

Statement of the Problem

Studies have shown that many people are turning to multi-platform applications to develop a mobile application once which can then be deployed on multiple platforms, but what remains to be shown is which multi-platform development application would be best. The multi-platform applications were analyzed to determine ease of use and proper functionality on two target platforms. Determining the answers to these questions may lead to discovering new capabilities and functionalities that are needed within these applications and may also help developers identify the development application that could be most efficient to use in the creation of applications for multiple platforms.

Purpose of the Study

The purpose of this study was to investigate multi-platform development applications currently on the market used to develop mobile applications. Differences between mobile application platforms have been studied and documented. Each platform was unique and possessed different behaviors, capabilities, and features. What remained to be determined was which multi-platform application would be best suited for allowing a developer to create a mobile application that could be utilized on a variety of platforms.

Significance of the Study

The analysis of multi-platform development applications could provide a developer a better understanding of the differences among multi-platform development applications and may lead to discovering new capabilities and functionalities that are needed within these applications. It also may assist developers in identifying the most efficient development application to use when creating applications for multiple platforms. Analysis and subsequent findings could possibly allow developers to have more time to focus on improving applications rather than spending their time on slow, individual platform development. Findings may also reveal areas where existing multi-platform development applications are lacking, thus allowing for improvements within multi-platform development applications to be created.

Definition of Terms

The following terms are defined to provide meaning and understanding in relation to this study:

Application (Apps): A software program that runs on a computer or mobile device and most commonly referred to as “apps” (Tech Terms, 2014).

Application Programming Interface (API): A set of commands, function, and protocols which programmers can use when building software for a specific operating system. An API allows programmers to use predefined functions to interact with the operating system instead of writing them from scratch (Tech Terms, 2014).

Closed System: Is licensed computer software carrying a copyright in which the source code is not made available to the general public. It is also known as proprietary software or closed source software (Wikipedia, 2014).

Debug: To eliminate software program errors commonly called “bugs” (Tech Terms, 2014).

Developer: A person or organization that designs and writes software and is often referred to as an application developer. The term generally refers to designers and programmers in the commercial software field (PCMag, n.d.).

Event Listener: An interface that is the primary method for handling events within computer software (W3C, 2003).

Extensible Markup Language (XML): A metalanguage that is used to create markup languages for specific applications and is used to define documents with a standard format that can be read by any XML-compatible application (Tech Terms, 2014).

Graphical User Interface (GUI): refers to the graphical interface of a computer that allows users to click and drag objects with a mouse instead of entering text at a command line (Tech Terms, 2014).

Hybrid App: An application in which some or all of your UI and business logic is written in HTML, CSS, and JavaScript running within a "native wrapper" such as a Titanium WebView or PhoneGap container. Hybrid apps have limited access to the device hardware, though such access varies by mobile operating system and development framework. Hybrid apps offer app store distribution and operation without a live network connection (Appcelerator, n.d.).

Interface: An interface can refer to either a hardware interface that connects two or more electronic devices together or the means in which a person controls a software application or hardware device (Tech Terms, 2014).

Internet: A communications network consisting of countless networks and computers that allow people to share information (Tech Terms, 2014).

Model-View-Controller (MVC): A software pattern that divides a given software application into three interconnected parts for implementing user interfaces (Wikipedia, 2014).

Multi-Platform Application: An application which is developed for multiple operating systems or platforms. Typically some or all of the user interface and logic is written in HTML, CSS, and JavaScript running within a “native wrapper.” These applications have limited access to the device hardware, though such access varies by mobile operating system and development framework. Sometimes multi-platform applications are also called hybrid applications (Appcelerator, n.d.; Tech Terms, 2014).

Native Application: An application that runs directly on a mobile device and has access to the hardware features of that device. Typically these applications can be run without a live network connection (Appcelerator, n.d.).

Open System: Licensed computer software in which the copyright holder makes the source code available to the public and provides the rights to study, change, and distribute the software to anyone and for any purpose. Also known as open software standard or open standard (Wikipedia, 2014).

Operating System (OS): An operating system "OS" is software that communicates with the hardware and allows other programs to run (Tech Terms, 2014).

Platform: A computer's operating system that allows the running of certain software. Platform examples include Windows and MacIntosh operating systems (Tech Terms, 2014).

Portable Operating System Interface (POSIX): This refers to a family of standards specified by the IEEE for maintaining compatibility between operating systems. POSIX defines the application programming interface (API), along with command line shells and utility interfaces, for software compatibility with variants of Unix and other operating systems (Wikipedia, 2014).

Smartphone: A smartphone is a mobile phone that includes advanced functionality beyond making phone calls and sending text messages and may be capable of running third party applications (Tech Terms, 2014).

SMS: "Short Message Service." SMS is used to send text messages, typically up to 160 characters in length, to mobile phones (Tech Terms, 2014).

Software Development Kit (SDK): A collection of software used for developing applications for a specific device or operating system (Tech Terms, 2014).

Tablet: A portable computer that uses a touchscreen as its primary input device (Tech Terms, 2014).

Web App: A mobile-ready web page accessed from a desktop or mobile browser, and typically formatted specifically to address the screen sizes and user interaction expectations of a mobile device. Web apps excel at platform reach, a "no-download" installation process, and instant application updates for all users. Web apps typically require a constant network connection (Appcelerator, n.d.).

World Wide Web Consortium (W3C): An international community that develops open standards to ensure the long-term growth of the Web (W3C, 2012).

Organization of the Study

This study has been organized in five chapters. Chapter I provides an introduction to the study, statement of the problem, the purpose of the study, significance of the study, and definitions of terms. Chapter II provides a literature review regarding the development of mobile applications, mobile development platforms, mobile platform languages, and multi-platform application development. Chapter III provides the methodology and design of the study. Chapter IV provides the results of this study, while Chapter V provides a conclusion and discussion.

CHAPTER II

REVIEW OF THE LITERATURE

Introduction

Mobile devices use a variety of powerful operating systems or platforms, each of which involves separate standards, programming languages, and distribution markets (Corral et al., 2012b). Typically developers want to deliver their products to a variety of users using various platforms, but choosing to develop for one platform can delay development and release on another platform (Corral, Janes, & Remencius, 2012a). It can also be very expensive to develop native applications for each platform as there are numerous platforms (Corral et al., 2012a). Developers are tasked with having to make the tough decision of which platform to develop for first, on their list of targeted platforms (Corral et al., 2012b). These problems, when developing mobile applications, have led to the growth in creation of multi-platform applications (Holzer & Ondrus, 2011).

Developing Mobile Applications

Mobile application development has become very popular among people of varying programming skills (Boardman, 2012). This could be due to the relatively low cost and short time commitment an application takes to cultivate from the conception of an idea to readying it for distribution (Boardman, 2012). Novice developers have many useful resources readily available which allow them to learn the necessary skills while attempting to develop applications. Some of these resources include: online tutorials, developer forums, and books (Boardman, 2012). Online tutorials and books are offered for various experience levels ranging from amateur to advanced. Although operating platforms change quite frequently making it difficult to find a current book containing the latest version, the changes typically are not drastic enough to make the book obsolete (Boardman, 2012). Also developer forums should not be overlooked as many questions that a developer might ask are typically answered on some forum (Boardman, 2012).

According to Computerworld magazine's editor in chief, Scot Finnie (2013), the following are five tips for developing successful mobile applications that developers should keep in mind:

1. In order to succeed, a mobile application must solve a problem.
2. Focus on one thing and do it well.
3. If you build it…nope, they probably won't come.
4. Applications need optional user notifications.
5. Don't force users to run your application instead of visiting the corporate website. (p. 40).

In regards to tip number one, a mobile application must offer a useful benefit to the user or people will not use it (Finnie, 2013; Wong, Khong, & Chu, 2012). The mobile application could be designed to solve a variety of problems, including saving time or money, entertaining or enlightening, delivering important functionality, or offering a novel service (Finnie, 2013). Tip number two, Finnie (2013) believes to be the most important recommendation. It is better to do one thing very well, than to do multiple things mediocre because going feature crazy could wind up derailing a project (Finnie, 2013). Tip number three simply means that although a designer may have his or her application in a store available for download, the store is not a direct channel to everyone and does not guarantee that people will want it, need it, or use it (Finnie, 2013). Tip number four reminds developers that notifications are not appropriate for all applications, so use them only as needed (Finnie, 2013). The final tip, number five, expresses the idea that a mobile application should concentrate on improving the user experience and utility of the mobile version of the website rather than replace the corporate website altogether (Finnie, 2013).

Mobile Development Platforms

Apple's iOS and Google's Android™ mobile platforms have been the two front runners in the mobile market in the past few years, but the two are very different platforms (Emmanouilidis, Koutsiamanis, & Tasidou, 2013; Sharma, 2011).

Android

Android, an Apache-free software platform for mobile devices based on Linux, was launched by Google in 2007 to advance open standards for mobile devices (Gavalas & Economou, 2011). The openness of Android allows the analysis and understanding of code which can lead to better feature comprehension, bug fixes, further improvements regarding new functionalities, and the ability to port to new hardware (Gandhewar & Sheikh, 2011). An open source software allows for customization to suit specific needs in different ways, but also allows for collaboration between developers (Proffitt, 2011).

The Android software stack includes an operating system, middleware, and key applications. To break it down even further, the Android Architecture, shown in Figure 1, contains four distinct layers; Linux Kernel, Libraries, Application Framework, and Applications (Gandhewar & Sheikh, 2011).

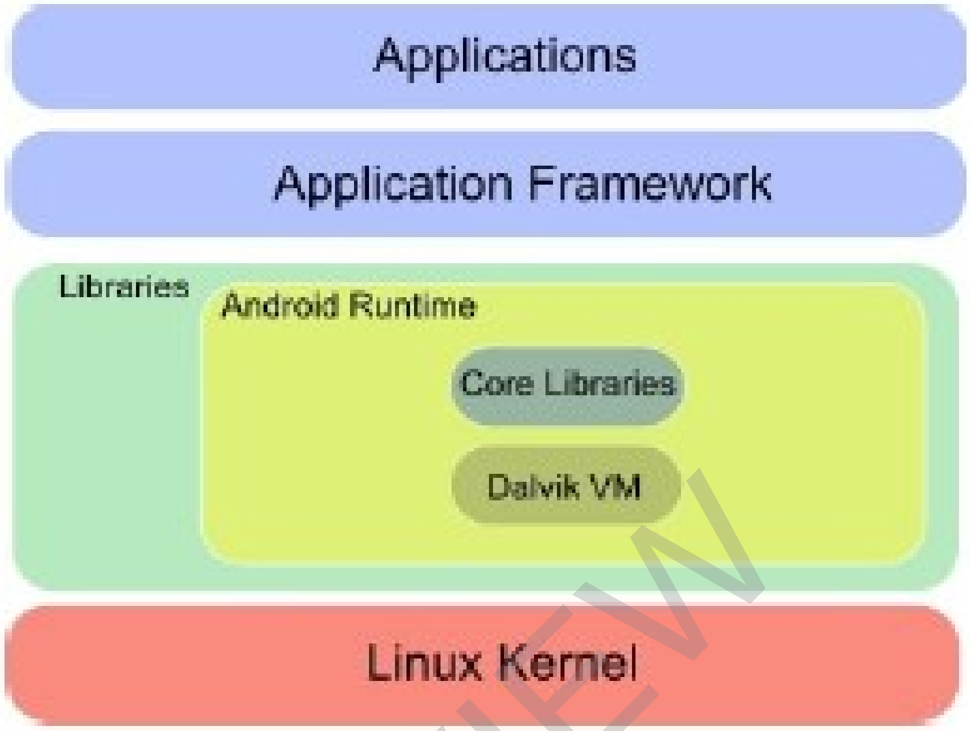

Figure 1. Android Architecture. (Gandhewar & Sheikh, 2011, p. 13, reprinted with permission).

The Linux Kernel, which was built with Linux version 2.6 operating system (OS), that Android relies on for core system services such as security, memory management, process management, network stack, and driver model acts as an abstraction layer between the hardware and the rest of the software stack (Gandhewar & Sheikh, 2011). For example, the camera driver is found in the Linux Kernel and allows the user to send commands to the camera hardware (Sharma, 2011).

The layer above the Linux Kernel is the Libraries (Gandhewar & Sheikh, 2011). The Libraries layer entails two parts; C/C++ libraries and the Android Runtime (Gandhewar & Sheikh, 2011). The C/C++ libraries are all written in C and C++ and get called up through a Java interface (Gandhewar & Sheikh, 2011). Examples of C/C++