



LUNATECH
RESEARCH

Native Cross-platform Mobile Application Development

by
W. de Kraker (0815283)

CMI-Program *Informatics* – Rotterdam University

May 7, 2012

First supervisor *Dhr. Y. S. Tjang*
Second supervisor *Dhr. A. Chamani*

Abstract

Nowadays mobile devices are vastly integrated into modern society. They bring us one step closer to satisfy our ever growing need to have information available anytime, anywhere. To help gain access to information on mobile devices we use so called *apps*.

However, the fragmented nature of today's mobile ecosystem poses a challenge for developers to develop mobile applications which are suitable to run on all mobile devices (*cross-platform*, since there is no de facto standard).

Currently there are several cross-platform mobile application development frameworks which offer a solution to this problem.

Lunatech, having expressed its interest in mobile app development, would like to know which of these framework, *if any*, suits Lunatechs needs best. A study has been setup in order to resolve this question. The results of which are laid out in this thesis.

Credits

Contents

Abstract	ii
Credits	iii
Introduction	2
Problem statement	2
Research questions	2
Background	3
Lunatech Research B.V	3
Rotterdam University of Applied Sciences (Hogeschool Rotterdam)	3
Stager	3
Mobile platforms	5
Introduction	5
Marketshare and trend	6
Defining native	7
Introduction	7
Native mobile applications	7
Alternative mobile application types	7
Comparison	8
Exsisting solutions to Cross-platform Mobile Application Development	10
Introduction	10
Applicationcelerator Titanium	11
Rhodes	11
Worklight	12
MoSync	12
PhoneGap	12

Sencha Touch	12
jQTouch	12
Comparisson	12
Developing cross-platform native applications with Titanium	13
Inner workings	13
Case study	14
Stager app	14
Stager application requirements	14
Used techniques and methodologies	14
Conclusion and Recommendations	15
Project goals	15
Stager case study	15
Cross-platform Mobile Application Development using Titanium	15
Future work	15
Evaluatie	17

Introduction

Problem statement

Lunatech has demand for the development of cross-platform mobile applications. Currently¹ these applications are been developed using webtechnologies such as HTML5 and Javascript. A mobile application developed this way is refered to as webapp because it runs in a browserbased environment and is often hosted at a webserver rather than downloaded to the device itself.

The problem with webapps is that they lack in user experience. This is mainly due manner in which user interface components are build in HTML. Every platform has its own set of recognizable elements, but these cannot be accessed from within the browser environment. As a result of this the app will feel unearthly to the user because it's style doesn't match the rest of the platform. It tries to look and feels native, but never gets around the fact that it's a webapp.

The direct alternative to webapps are native apps, native are writting using technologies proprietary to each platform, hench the term 'native'. What these applications lose in terms of cross-platform support they make up in terms of user experience. A native app has acces to all the platforms proprietary libraries and can rely on the user interface elements provided through these libraries.

Lunatech would like to know how to make use of the look-and-feel from native apps with the cross-platform support of webapps.

Research questions

Main research question:

- *How to develop a cross-platform mobile application while retaining the native look-and-feel?*

Sub research questions:

- *How is the native look-and-feel defined?*
- *Which solutions to cross-platform mobile application development currently exist?*
- *Which of these solutions offer the defined native look-and-feel?*

¹Note: when mentioning the word 'current', it refers to the old situation as the process to get to the actual current situation is being illustrated

Background

Lunatech Research B.V

Lunatech provides application development services, completely based on open-source web and Java technologies and open standards. They are early adopters of new technology, and use cutting-edge frameworks and tools. To stay up-to-date, their developers have the opportunity to research, try new technologies and contribute to open-source projects. The company is dominated by software developers. Everyone (except the director) writes code, on top of which some staff have a secondary management role, and the staff who will deliver a project interact with the customer directly.

Rotterdam University of Applied Sciences (Hogeschool Rotterdam)

Rotterdam University is one of the major Universities of Applied Sciences in the Netherlands. Currently almost 30,000 students are working on their professional future at the university. The university is divided into eleven schools, offering more than 80 graduate and undergraduate programmes in seven fields: art, technology, media and information technology, health, behaviour and society, engineering, education, and of course, business.[2]

Stager

In 2011, live music venue WORM - Instituut voor Avantgardistische Recreatie hired Lunatech to build *Stager*, a modern web-based resource planning and ticketing application to help manage live music events. Lunatech took the opportunity to use the relatively new Play framework to build a web application with an HTML5 and Java architecture. Stager has broad requirements ranging from high performance and security for the public ticket sales component to high usability for the internal resource planning component that will be used for hours a day by employees and being open to enhancements in the future for new customers. [?]

WORM

WORM is een instituut voor avantgardistische recreatie te Rotterdam, bestaande uit een kunstenaarscollectief, een podium met winkel en een Parallele Universiteit (DIY-werkplaatsen voor film, muziek en media). Geboren onder de sterren van punk, dada, fluxus, situationisme en futurisme is WORM uitgegroeid tot een eigengereide organisatie die de 'Do-It-Yourself' mentaliteit van hun

voorouders combineert met ultra-pragmatisme, liefde voor techniek(en) en goede boekhouding. De output van WORM is film, radio, concerten, cursussen, partys, publicaties, performances, webprojecten, installaties, workshops en een opeenhoping van tactiele media en internet. WORM focust zich (blijmoedig en toch serieus) op avantgarde, middelenschaarste en opensource. #todo: translate [?]

Mobile platforms

Introduction

The following chapter presents a concise overview of current mobile operating systems for mobile platforms, specifically smartphones and tablets.

A smartphone can be defined as a smart phone is a next-generation, multifunctional cell phone that provides voice communication and text-messaging capabilities and facilitates data processing as well as enhanced wireless connectivity.[5]

Apple iOS

iOS is a proprietary mobile operating system, developed by Apple Inc. It was originally released in 2007 for the iPhone and iPod Touch. iOS also became the main operating system of the iPad and Apple TV.

Google Android

Android is a opensource mobile operating system, developed by the Open Handset Alliance, led by Google and other companies.[6]

BlackBerry OS

BlackBerry OS is a proprietary mobile operating system, developed by RIM(*Research In Motion*) for its line of BlackBerry mobile devices.

Windows Phone 7

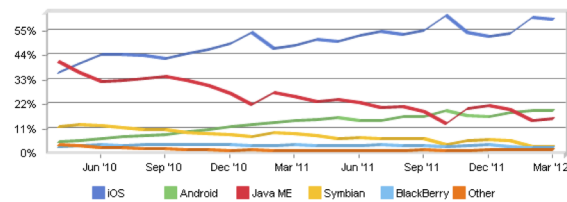
Windows Phone 7 is a mobile operating system, developed by Microsoft as a succesor to its Windows Mobile platform.

Other platforms

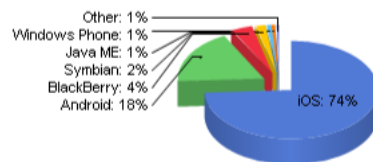
Java ME

Symbian

Marketshare and trend



World wide mobile OS Marketshare trends, April 2010 up to may 2012



Operating System	Total % Market Share
iOS	74.04
Android	18.36
BlackBerry	3.84
Symbian	1.75
Java ME	0.83
Windows Phone	0.68
Bada	0.29
Windows Mobile	0.14
Kindle	0.05
Samsung	0.03
LG	0.01
ZTE	0.00
Palm	0.00

Table 1: Marketshare in the european continent as of march 2012[4]

Defining native

Introduction

This chapter will define the paradigm of *native look-and-feel*.

Native mobile applications

A native application is an application inherent to the platform for which it was built using techniques proprietary to the platform. For example, an iOS application is native when written in Objective-C and an Android application is written in Java. Native applications are typically fast and can access the device's native API's.

The native look-and-feel

When written in the native framework for a platform a mobile application receives access to the available public libraries of the platform. These libraries include the UIKit(*on iOS*) which provides the developer with a pre-fabricated set of user interface components. These can be seen as the building blocks for the graphical user interface on that platform. When used, the general style of the mobile application gains consistency to the overall user interface design of the platform's operating system. This gives an application its native look, which in turn participates to the *native feel*.

The *native feel* of a mobile application can be defined as the speed in which the user interface elements, the responsiveness of user interface elements to touch events, and smoothness of the animation in which the user interface elements are moved. A native mobile application has the advantage to hardware acceleration. This means its code has been precompiled and directly executed by the device CPU, rather than having to be interpreted by the device's browser. As a result of this the user interface elements are rendered faster and it *feels* smooth.

Alternative mobile application types

Web applications

A mobile web application is an application developed with web technologies as JavaScript and HTML5 with CSS3. It is in fact nothing more than a website designed to fit on mobile devices, often they resemble the style of a native application rather than a traditional website. These applications are built with a JavaScript library to add support for scrolling and handling touch events. Touch

events are handled via user interface elements provided by the library. Examples of these libraries include jQtouch, SenchaTouch.

Hybrid applications

A hybrid application in mobile development refers to an application which use a native shell to wrapped around web app. There are generally two forms of native shells, the first is a *webview* and the second a native framework which exposes a javascript API to provide the web application access to otherwise native API's.

Webview-based hybrid applications

A webview-based hybrid application is a webbased mobile application wrapped in a webview. A webview is a view or element which acts like a browser would, e.g. it is able to render HTML and run javascript. It is readily available in the native libraries. The advantage of a webview-based hybrid application over an normal web application is that it can be published via the devices native application publishing platforms. e.g. a webview-based hybrid application targetted for the iPhone can be placed in the Apple appstore.

Worklight is an example of a framework which can be used to develop webviewbased hybrid applications.

Framework hybrid applications

A Framework based hybrid application is a webviewbased application build upon a framework which provides a Javascript API to allow the application access to otherwise native API's. The framework is written in the platforms native programming language making it possible to access the native API, such as reading contact list, composing of SMSes, full access to the location API, etc.

PhoneGap is an example of a framework which can be used to develop mixed hybrid applications.

Comparison

Web applications are quick and cheap to develop. Written entirely in HTML5, CSS and JavaScript. Executed by the mobile browser and therefore cross - platform by default, but less powerful than native apps.

Hybrid Applications (Web), the app's source code consists of web code executed within a native wrapper that is provided by a framework.

Hybrid Applications (Mix), the developer augments the web code with a Javascript API to create unique features and access native APIs that are not yet available via the browser, such as AR, NFC and others.

Native Application are platform-specific. Requires unique expertise and knowledge. Pricey and time consuming to develop but delivers the highest user experience of all approaches.



Different types of mobile applications[3]

Existing solutions to Cross-platform Mobile Application Development

Introduction

In today's industry there exist several cross-platform mobile application development frameworks which offer a solution to cross-platform problem. All of these frameworks provide a custom solution for crossing the bridge between platforms. In order to determine which one should be adopted by Lunatech for mobile development the following criteria have been determined for comparison:

1. *Platform support*
Which platforms and their versions are supported by the framework.
2. *Native UI support*
Whether or not native user interface elements are supported for each supported platform.
3. *Programming language*
Which programming language is used to develop using the framework.
4. *IDE (Integrated Development Environment)*
Which IDE can be used to develop using with the framework.
5. *License type*
Which license types are available.
6. *Application type*
Which type of mobile application is produced using this framework.

The cross-platform criterion is based on Lunatech's requirement to build mobile applications for the operating systems have at least a 10 percent marketshare in the European continent. Second comes the support for native user interface elements. Together these criteria form the essence of the main research question: "*How to develop a cross-platform mobile application while retaining the native look-and-feel?*" The remaining criteria are of secondary importance, they will provide more detailed means to compare the frameworks which offer native user interface support.

The following solutions have been chosen for review: *Titanium, Rhodes, Worklight and MoSync*. These are derived from the list Existing solutions²

²see attachment: *Existing solutions*

Appcelerator Titanium

Appcelerator Titanium is an commercially supported opensource platform for developing cross-platform mobile applications. It was introduced by Appcelerator Inc in December 2008. Built upon the Eclipse IDE Titanium offers a Javascript API to native mapper classes which allow the developer to generate truly native cross-platform mobile applications.

Platform support and native capability

As of May 2012 Titanium supports iOS and Android. Next to building a native application for these platforms Titanium offers the option to generate a web application. Support for Research In Motion (BlackBerry) is in active (however closed from public) development. May first 2012 Appcelerator announced that it is extending its core value of cross-platform native application development beyond iOS and Android, on to RIM's BlackBerry devices.[1]

Techniques and tools

TitaniumStudio is an Eclipse based IDE with integration the propriatory mobile SDKs and simulators. Native applications built with Titanium are written using JavaScript.

Application type

Applications produced by

Rhodes

Rhodes is an open source Ruby-based framework to build native applications for all major smart-phone operating systems (iPhone, Android, RIM, Windows Mobile and Windows Phone 7). These are true native device applications (not mobile web applications) which work with synchronized local data and take advantage of device capabilities such as GPS, PIM contacts and calendar and the camera.

Platform support and native capability

iOS, Android, BlackBerry, Symbian, Windows Mobile

Techniques and tools

Eclipse based studio, Ruby & HTML

Application type

Worklight

Worklight Studio is an eclipse based IDE for the cross-platform development of mobile applications. Worklight Studio was introduced in 200x by Worklight Inc. In early 2012 Worklight Inc. became an IBM company. Worklight Studio offers mobile development through the use of webtechnologies such as HTML5, and Javascript.

Platform support and native capability

iOS, Android, BlackBerry, Windows Mobile

Techniques and tools

Eclipse plugin, HTML5, CSS, Javascript

Application type

Hybrid web

MoSync

The MoSync mobile SDK offers cross-platform development through the use of webtechnologie or C/C++.

Platform support and native capability

Techniques and tools

Application type

PhoneGap

Sencha Touch

jQTouch

Comparisson

Developing cross-platform native applications with Titanium

Inner workings

Case study

Stager app

Stager application requirements

Events

Notifications

Tickets

Mobile payment

Used techniques and methodologies

Javascript

RequireJS

JSON

Playframework

Java

Conclusion and Recommendations

Project goals

Stager case study

Cross-platform Mobile Application Development using Titanium

Evaluation of Titanium

Limitations of Titanium

Future work

Bibliography

- [1] Jill Asher. Appcelerator Expand Native Mobile Application Support on RIM's BlackBerry Devices, 2012.
- [2] Hogeschool Rotterdam. Rotterdam University, 2012.
- [3] IBM-Worklight. Mobile application types, 2012.
- [4] Netmarketshare. Mobile Top Operating System Share Trend, 2012.
- [5] L.M. Ni. Spotlight: The Rise of the Smart Phone. *IEEE Distributed Systems Online*, 7(3):3–3, March 2006.
- [6] Open Handset Alliance. Android Philosophy and Goals, 2012.

Evaluatie