LUNATECH
RESEARCH

Native Cross-platform Mobile Application Development

by
W. de Kraker (0815283)

CMI-Program *Informatics* – Rotterdam University

May 1, 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 mobile developers to develop apps 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 research has been setup in order to resolve this question, its result is layed out in this thesis.

Credits

Contents

Introduction

Problem statement

Lunatech has demand for the development of cross-platform mobile applications. Currently[1] 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 arround 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 propietary 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?*

[1] Note: 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 give themselves the advantage in software development. 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 our university. The university is divided into eleven schools, offering more than 80 graduate and undergrad- uate programmes in seven fields: art, technology, media and information technology, health, behaviour and society, engineering, education, and of course, business.[1]

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, 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

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.[3]

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.[4]

BlackBerry OS

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

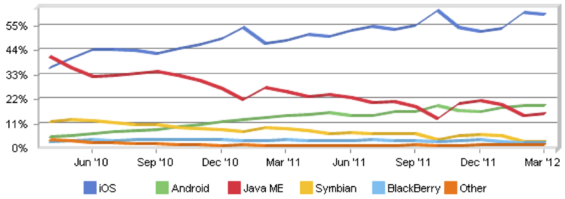
Windows Phone 7

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

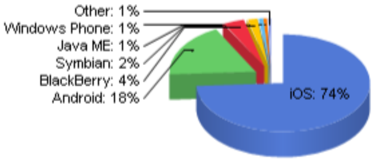
Java ME

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[2]

Defining native

Intoduction

The following chapter will define the *native look-and-feel*.

Native mobile applications

A native application is by definition an application inherent to the platform it was build for using techniques proprietary to the platform. For example, an iOS application is native when written in Objective-C and an Android app is written in Java. Native apps are typically fast can can acces the devices n

The native look-and-feel

When written in the native framework for a platform an mobile application receives acces 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 buildingblocks for the graphical userinterface on that platform. When used, the general style of the mobile application gains homogeneity to the overal user interface design of the platforms 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 in the speed which the userinterface elements, the responsiveness of userinterface elements to touch events, and finally smoothness of the animation in which the userinterface 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. As a result of this the userinterface 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 app rather than a traditional website. Often these application are build with a Javascript library to add support for scrolling and handling touch events. These touch events are handled via widgets, userinterface elements which provide the user with components composed similair to the native components. 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 wrap a web app. There are generally two forms of native shells, the first is webview and second a native framework which exposes a javascript API to provide the web app access to otherwise native API's.

Webviewbased hybrid applications

A webviewbased hybrid app is nothing more as a webbased mobile application wrapped in a webview. A webview is a view or element which is 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 webviewbased hybrid app over an normal web app is that it can be published via the devices native app publishing platforms. e.g. a webviewbased hybrid app 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.

Mixed hybrid applications

A Mixed based hybrid app is a webviewbased app build upon a framework which provides a Javascript API to allow the app access to otherwise native API's. The framework is written in the platforms native programming language, this provides the possibility 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.

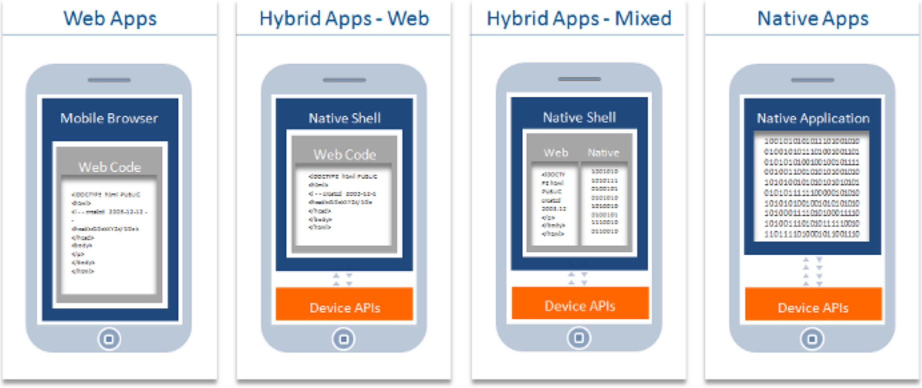
Comparison

Web apps - 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 Apps (Web) - The app's source code consists of web code executed within a native wrapper that is provided by a framework.

Hybrid Apps (Mix) - The developer augments the web code with native language to create unique features and access native APIs that are not yet available via JavaScript, such as AR, NFC and others.

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


Web Apps
Mobile Browser
Web Code
Hybrid Apps - Web
Native Shell
Web Code
Device APIs
Hybrid Apps - Mixed
Native Shell
Web
Native
Device APIs
Native Apps
Native Application
Device APIs

label x

Exsisting solutions to Cross-platform Mobile Application Development

Developing cross-platform native applications with Titanium

Inner workings

Case study

Stager app

Stager app requirements

Events

Notifications

Tickets

Mobile payment

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] Hogeschool Rotterdam. Rotterdam University, 2012.

[2] Netmarketshare. Mobile Top Operating System Share Trend, 2012.

[3] L.M. Ni. Spotlight: The Rise of the Smart Phone. *IEEE Distributed Systems Online*, 7(3):3–3, March 2006.

[4] Open Handset Alliance. Android Philosophy and Goals, 2012.

Evaluatie