

Institutionen för datavetenskap
Department of Computer and Information Science

Examensarbete

Evaluation of cross-platform development for mobile devices

av

Joy Friberg

LIU-IDA/LITH-EX-A--14/005--SE

2014-01-24



Linköpings universitet

Examensarbete

Evaluation of cross-platform development for mobile devices

av

Joy Friberg

LIU-IDA/LITH-EX-A—14/005--SE

2014-01-24

Handledare: Anders Fröberg

Examinator: Erik Berglund

Abstract

Developing an application for several platforms can be time consuming because each platform has its own operating system and different developing language. Cross-platform development makes it possible to develop an application that will work on several platforms. This report will evaluate this kind of development by doing a case study for the company CGI. The case study will evaluate which cross-platform methodology is the preferred choice for this specific vacation booking application I developed for CGI. The different methodologies I studied were web, hybrid, interpreted and cross-compiled. The preferred methodology for this vacation booking application I developed was in this case the hybrid alternative. When selecting this methodology I also chose two different tools and those two were Ice-nium and jQuery Mobile. The purpose of this report was to find out if cross-platform development can be a substitute to native programming and by evaluating and developing cross-platform I found out that it can be a substitute if the application is not too complex. In this specific case I also believe that hybrid development is a good substitute to native development for this kind of applications.

Contents

1	Introduction	1
1.1	Method	1
1.1.1	Case study	1
1.1.2	My approach	3
1.2	Purpose	3
1.3	Scope and limitations	3
1.4	Report structure	3
1.5	Sources	4
2	Application	5
2.1	Requirements	5
3	Background	6
3.1	Operating system	6
3.1.1	iOS	7
3.1.2	Android	7
3.1.3	Windows Phone	7
3.2	Web technologies	8
3.2.1	HTML5	8
3.2.2	CSS3	8
3.2.3	JavaScript	8
4	Cross-platform development	9
4.1	Web	9
4.2	Hybrid	10
4.3	Interpreted	11
4.4	Cross-compiled	12
4.5	Different tools	13
4.5.1	PhoneGap	13
4.5.2	Icenium	14
4.5.3	Titanium	15
4.5.4	Xamarin	15
4.6	UI-frameworks	16
4.6.1	jQuery Mobile	16

4.6.2	Sencha Touch	17
5	Results	18
5.1	Observations	18
5.1.1	PhoneGap	18
5.1.2	Icenium	19
5.1.3	Titanium	19
5.1.4	Xamarin	20
5.1.5	jQuery Mobile	20
5.1.6	Sencha Touch	21
5.2	Comparison	21
5.2.1	Web application	21
5.2.2	Hybrid application	21
5.2.3	Interpreted application	22
5.2.4	Cross-compiled application	22
5.3	Selection of tools	22
5.4	Application	23
6	Conclusion	25
7	Discussion	26
7.1	Future work	26

List of Figures

1.1	Web application	2
4.1	Web application	10
4.2	Hybrid application	11
4.3	Interpreted application	12
4.4	Cross compiled application	13
4.5	Overview of PhoneGap.	14
4.6	Overview of Xamarin.	16

Chapter 1

Introduction

CGI-group's offices in Linköping and Norrköping want their booking system for vacation, duty and when on parental leave to be accessed via mobile devices. The employees at these offices have mobile phones provided by the company and they can choose between three different platforms, iOS, Android and Windows Phone. The existing system is coded in C# with ASP.net using web forms and is reachable through CGI's intranet. There are different approaches to develop this booking system for a mobile device and one approach is to develop one application for each platform in the their respective native language. Then there are other approaches that are using cross-platform development and the developer can reuse code for each platform and that is what this report will evaluate.

1.1 Method

Since CGI wants a cross-platform application for the vacation system, I believe a case study would be the preferred methodology. Because the application would in that case be the specific case in the study. I could have done an evaluation in a more general perspective but hopefully I will reach a deeper understanding with this case study and not a too comprehensive scope. I need to evaluate the different approaches to develop this application in this specific case for CGI.

1.1.1 Case study

A case study's purpose is to give a deeper understanding of specific research and can either be an exploratory, descriptive, explanatory or improving analysis of an experience [1]. An *exploratory* analysis's purposes are to find out what is happening and giving new ideas for future work. A *descriptive* analysis is more of describing the current state of a specific occurrence. An *explanatory* analysis's purpose is looking for an explanation of a problem or

a situation and an *improving* analysis is more of trying to improve a specific part of the occurrence.

A case study is based on empirical research and could either be inductive or deductive. An *inductive* approach is influenced by observations and the researcher starts by doing observations and looking at patterns from these observations. From these identified patterns the researcher sets up hypotheses and relates them to already existing theories or explain a new one (figure 1.1 a)). In a *deductive* approach the researcher starts with a theory and comes with a hypothesis for the specific theory (figure 1.1 b)). An exploratory analysis has more characteristics from the inductive approach and an explanatory analysis has more characteristics from the deductive approach.

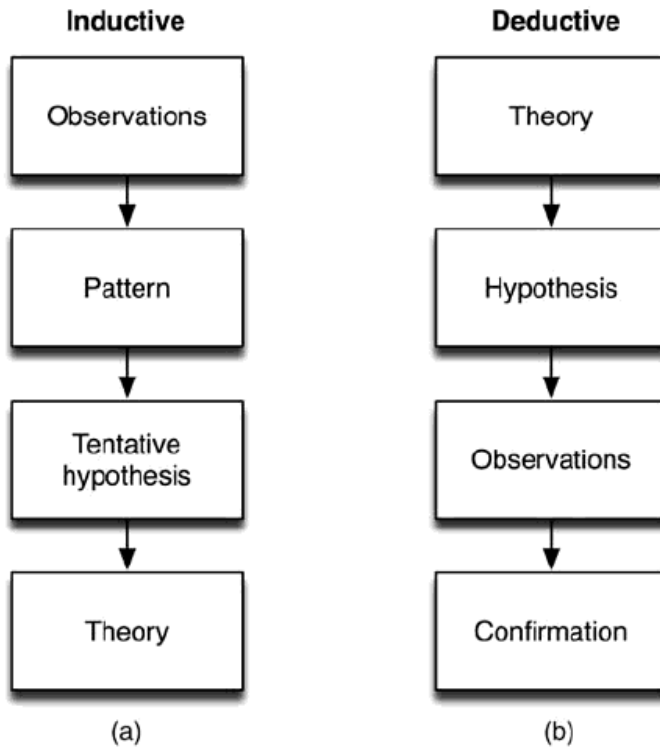


Figure 1.1: Inductive and Deductive approaches [1]

There are a lot of different approaches and guidelines of making a case study but here are some recommendations:

- A theoretical basis with defined question for the research.

- A chain of evidence built on arguments and explanations.
- A fully documented research.

1.1.2 My approach

Based on the description above, I will do an exploratory case study with characteristics from both a deductive and an inductive approach. I will start with my purpose and questions I got for this research. Then I will explain the background needed for the later observations which will be based on my own thoughts and scientific data. From these observations I will also make a hypothesis of a preferred approach and then do an observation of this approach by doing an application. The confirmation part is then discussed in the last chapter.

1.2 Purpose

The purpose is to evaluate relevant approaches for cross-platform development and then select the best approach and tools for the application development for CGI. The development of the application will give me a deeper understanding about the chosen approach and tools. I want to find out if cross-platform development is a good substitute to native mobile programming.

1.3 Scope and limitations

- Testing will be performed on a Samsung Galaxy S3 and an Ipad Mini. No Windows Phone will be available.
- Time is limited a therefore not enough time to learn a lot of new languages and tools.
- I will not buy any licences for tools but instead use free licences or trial versions.

1.4 Report structure

Introduction: Background information about the thesis, method, purpose and how to approach the different tasks.

Application: Will present the application with its description and different requirements.

Background: Background information about the different platforms and other techniques and information that will be relevant for the study.

Cross-platform development: What cross-platform development is and what tools the report is going to evaluated.

Result: The data collected from me testing the different tools will be presented in this section together with the comparison of these tools based on scientific data. The selected tools are described with a motivation and are presented after the comparison. The data collected from the application development with these tools are also explained here.

Conclusion: My own opinion and hypothesis confirmed or denied about cross-platform development for the developed application together with an overall discussion about cross-platform development in general. I also write what I believe can be future work out of this report.

1.5 Sources

Since these tools are updated frequently to match the updates and changes for multiple platforms, some of the sources may be outdated. Blogs, and similar can be up to date but may lack of reliability or can be biased.

Chapter 2

Application

CGI has a vacation booking system today on their intranet but they would also like an application in their mobile devices so they can easily book a vacation through their device. This booking system will not only be able to book vacation but also parental leave and when they are on duty. A user will also be able to see all his or her bookings and also other employees' bookings. An employee is in different work group for different projects and through the application an employee will be able to sort by these groups to see only relevant employees. Since CGI employees only have Windows Phone, iOS and Android devices it is a requirement that it will work on these devices. It needs to be an application installed on the device and not a web page.

2.1 Requirements

All CGI employees in Linköping and Norrköping:

- will login with their CGI account
- can book vacation, duty or when on parental leave.
- can see when they are going on vacation, duty and when on parental leave.
- can be part of different groups.
- can see bookings from members in their own groups.

Chapter 3

Background

Before describing the different cross-platform approaches this chapter will explain different technologies that are relevant for cross-platform development.

3.1 Operating system

I will only describe three different platforms since CGI only have mobile devices for these three. The term platform can be explained as a set of hardware and software components and in the terms of mobile devices the components are the operating system and the hardware components. The operating system is managing the hardware on the device, it is providing a framework for implementing native applications and it provides different built in application such as the browser. Of the three operating systems, Globally, Android got the biggest market share, iOS is second and Windows Phone got the smallest market of these three (the third biggest on the market)[2].

Operating system	2013 Market share	2017 Expected market share
Android	75.3%	68.3%
iOS	16.9%	17.9%
Windows Phone	3.9%	10,2%

Operating system	Browser	Language	Official IDE
iOS	Safari	Objective C	Xcode
Android	Android Browser	Java	Eclipse
Windows Phone	Internet Explorer	C# .net	Visual Studio

3.1.1 iOS

iOS is the operating system developed by Apple and was first released in 2007 [3]. It is an OS based on UNIX for iPhones, iPads, iPod touch and Apple TV. The latest version today (January, 2014) is iOS7 which was released September 18th and some major changes were that the graphical user interface changed appearance and applications are now in full-screen. The official IDE for iOS development is Xcode and the programming language is Objective-C.

The standard browser on iOS devices is the Safari browser which is based on Webkit. iOS devices only have one home button and not a menu- or back-button like the Android devices. The iOS emulator is installed with Xcode tools and along with the iOS software development kit (SDK). The emulator can simulate different devices and iOS versions, some hardware can be simulated but as example the camera and accelerometer can't be simulated.

3.1.2 Android

Android 1.0 was release in 2008 by Google and the latest version today (January, 2014) is 4.4 (KitKat) [4]. Android is open source and built on the Linux kernel and it is used in many different devices such as smartphones, tablets, Google Glasses, smart watches, etc. Native Android applications are written in Java and the officially supported IDE is Eclipse using the Android Development Tools (ADT).

The most common browser on Andorid devices is the Android Browser and it is based on Webkit. Android devices usually have a back-button, menu-button and a home button. With the Android SDK you can create different Android emulators with all of the different Android versions. You can also set different properties such as different CPUs and also set values to acceleromter and screen size.

3.1.3 Windows Phone

Windows Phone was first released in 2010 by Microsoft [5]. It is a successor to Windows Mobile which was released in 2003. The most recent version of Windows Phone is Windows Phone 8 (January, 2014) and the major changes were that it supports new powerful hardware such as quad- and octo-core CPUs and higher resolutions.

The official IDE is Visual Studio and the programming languages are C# or Visual Basic. The user interface can be made by using XAML and the developer have two choices of development frameworks such as Silverlight

and XNA. Silverlight and XNA is on top of the .NET framework and Silverlight is more for business applications and simple 2D games when XNA is for creating more complex games [6].

In Windows Phone 8 the default web browser is Internet Explorer 10 (IE10) and it is based on the Trident engine. Windows Phone devices usually have three buttons: back, start and search. The emulator simulates a real windows phone and is integrated in Visual studio however, the emulator can't test the hardware features such as GPS or the camera.

3.2 Web technologies

Some of the cross-platform tools requires web technologies to work and this section will describe these technologies.

3.2.1 HTML5

The fifth version of hypertext markup language (HTML) is called HTML5 and was released in 2004 [7]. Technologies that are associated with HTML5 are for example: Drag and drop, geolocation and web sockets. Webkit-based browsers and trident engine 7.0 which IE10 is based on support many HTML5 features but they differ at performance and some minor capabilities.

3.2.2 CSS3

Cascading style sheets 3 (CSS3) is a style sheet language for what the user interface will look like [7]. CSS3 is divided into different modules and there are over 50 of these modules. The most recommended (by CSS Working Group) can be divided into the following categories: Backgrounds/borders, color, media queries, multi-column layout and selectors. With CSS3 you can create rounded corners, shadow effects and with media queries you can detect which type of mobile device the user is using.

3.2.3 JavaScript

JavaScript is an interpreted programming language meaning that it executes instructions directly in a browser's JavaScript engine without compiling [8]. JavaScript is a prototype-based scripting language which is dynamic and like other scripting languages the type is not associated with the variable but with the value. JavaScript can create objects and those are associative arrays together with prototypes. These prototypes are a way to do behavior reuse (in many object oriented language it is done with inheritance) and it is done by cloning existing objects.

Chapter 4

Cross-platform development

There are a lot of tools available for developers and the ones presented in this section are the ones I consider most relevant based on popularity, maintenance (latest update, update interval), supported platforms and comprehensiveness. Appcelerator Titanium Mobile does not yet support Windows Phone but the reason I am still considering it is for two reasons: it is widely used in the cross-platform development community and it will support Windows Phone in a near future (see more at 4.5.3).

4.1 Web

Web applications are designed to run in the phone's browser and get the content from the internet (see fig 4.1)[9]. The mobile device will not need to store any content that are specific to the web application since the application is server driven. Some of the operations will be done on the client side depending on how the application is designed. For example you can do calculating and similar operations on the client and the operations that need to save to a database or similar need to connect to the server and perform the database update on the server.

Web applications are built on internet technologies which are common to many browser technologies such as HTML, CSS and JavaScript [18]. There are many libraries and frameworks for developing web applications for mobile devices, jQuery Mobile and Sencha Touch are two of them. These libraries try to create functionality and design similar to a native application.

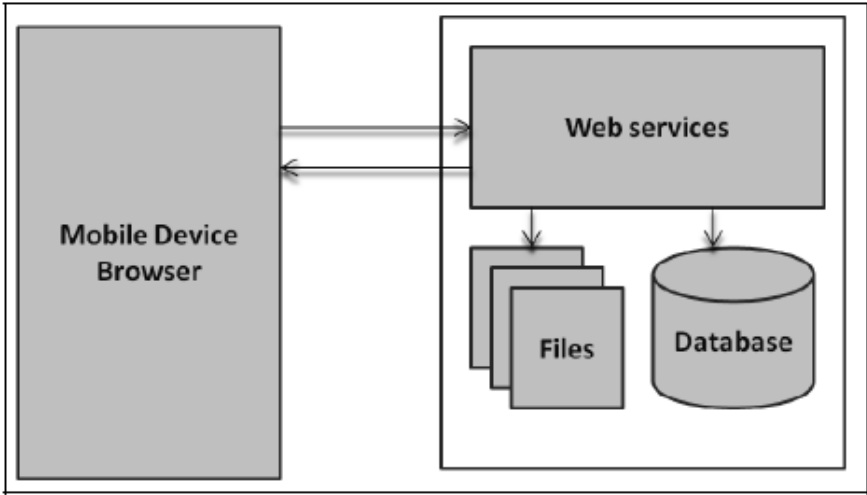


Figure 4.1: Web application [9]

4.2 Hybrid

Using internet technologies and reaching native application programming interfaces (APIs) to create an application is called a hybrid application [9]. Just like a web application, the application is run in the phone's browser and uses the browser's engine. The significant difference is that the application is installed on the device in a container and has access to the device's hardware.

The browser's engine will display the HTML and CSS in a full screen web view control. Through the JavaScript abstraction layer the application gets its access to hardware features. These features are accessible through JavaScript APIs (see fig 4.2). The Hybrid application can use most of the web application's advantage and also use the devices' features however, the application needs to be downloaded to the device unlike the web application. PhoneGap is a widely used tool for developing hybrid applications [18].

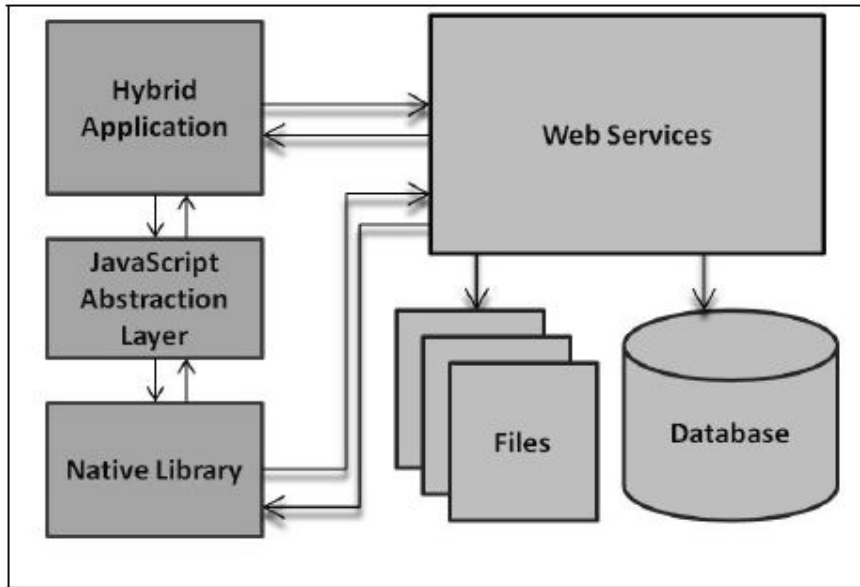


Figure 4.2: Hybrid application [9]

4.3 Interpreted

Interpreted applications are made with interpreted programming language such as JavaScript and Ruby [9]. The interpreter interprets the application code after the code is deployed to the device and it executes the code at runtime. Because of this interpretation across different platforms, there is a cross-platform support [18]. The logic for the application needs to be platform-independent and can for example be a set of commands in XML. A known approach for developing interpreted applications is Appcelerator's Titanium.

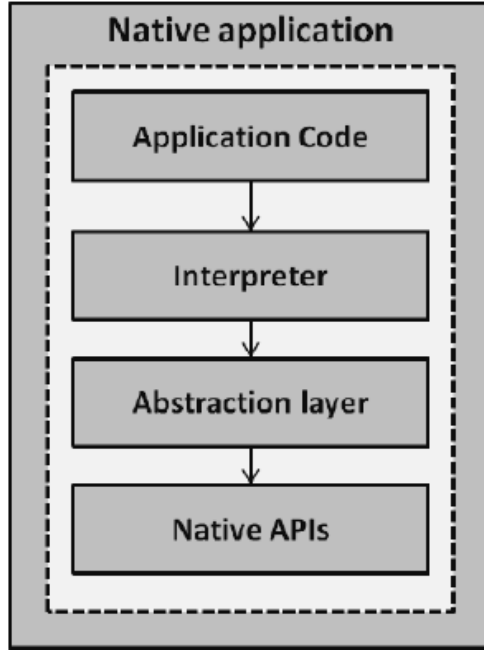


Figure 4.3: Interpreted application [9]

4.4 Cross-compiled

Cross compiled applications or generated applications have a cross compiler that turns the source code into native binaries [9] [18]. The developer creates the code in a programming language like C# and then the application code is cross compiled with a compiler for each specific platform to native code (see fig 4.4).

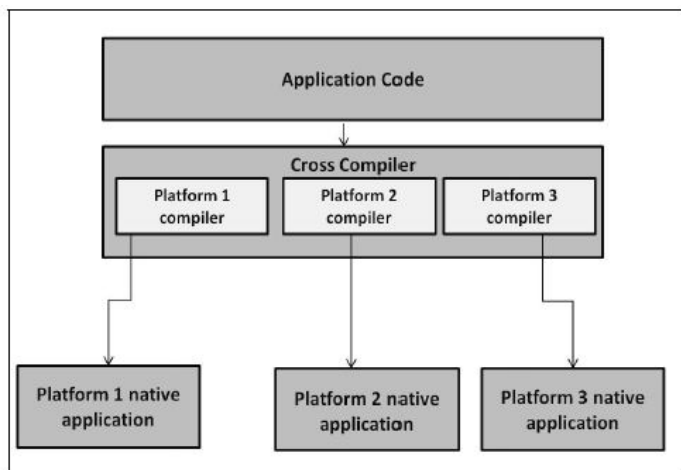


Figure 4.4: Cross compiled application [9]

4.5 Different tools

This section will describe six different tools for the different types of cross-platform development. The web applications tools jQuery Mobile and Sencha Touch is describe in the next section called UI-frameworks. Below is an overview of the different tools.

Tool	Type	Language	Android/iOS	Windows	MVC	Official IDE
jQuery Mobile	Web	HTML/JavaScript	Yes	Yes	No	-
Sencha Touch	Web	HTML/JavaScript	Yes	Yes	Yes	-
PhoneGap	Hybrid	HTML/JavaScript	Yes	Yes	Yes	No official
Iceium	Hybrid	HTML/JavaScript	Yes	No	No	Mist, Graphite
Titanium	Interpreted	JavaScript/TSS	Yes	No	Yes	Titanium Studio
Xamarin	Cross-compiled	C#	Yes	Yes	No	Visual Studio, Xamarin Studio

4.5.1 PhoneGap

PhoneGap is a framework for mobile development which is owned by Adobe Systems and founded by Nitobi [10]. It was developed in 2008 and the first stable release was in 2009 and version 1.0 was released in 2011. It is a distribution of Apache Cordova which Adobe/Nitobi donated to Apache Software Foundation as an open source version of PhoneGap. In PhoneGap you develop in HTML5 and JavaScript and with PhoneGap's APIs you can use the native function in respective platform. In other words, you develop a web application which can reach the native functions. In PhoneGap 3 the supported platforms are iOS, Android and Windows Phone but in 2.9 Blackberry, Symbian and WebOS are also supported.

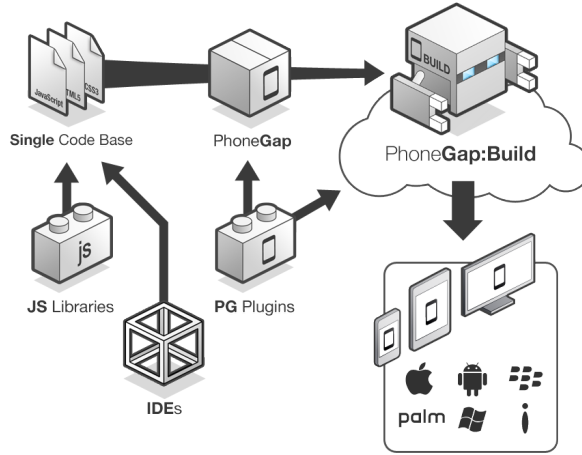


Figure 4.5: Overview of PhoneGap [11]

PhoneGap Build is PhoneGap’s cloud-based service where you upload your web assets and build your applications on each of the supported platforms simultaneously [12]. You can also attach your Git or SVN repository with the PhoneGap Build service. This means that you don’t need a specific IDE or install the various SDK to build your applications and you don’t even need an OSX-based computer to develop for iOS.

Running a PhoneGap application is done in three steps. The first step is that the native shell will be launched for respective platform and then in the second step uses the browser on that platform which later in step three runs your web application. How PhoneGap is using the native functions is done different on each specific platform. For example in Android, the native function can be called immediately hence the Java objects are marshalled into the web application. In iOS, the JavaScript in the web application dispatch a call which is received by the native code.

4.5.2 Icenium

Icenium is provided by the American company Telerik Inc. and enables the developers to use HTML5 and JavaScript for iOS and Android development [13]. Icenium is a set of tools for creating cross-platform mobile applications. The three major tools it uses are Cordova, jQuery Mobile (see) and Kendo UI. However you don’t need to use jQuery Mobile or Kendo UI, you can use what JavaScript library you want. Icenium provides two different types of IDEs. One is called *Mist* and it is an online-based IDE and the other is called *Graphite* and it is a Windows desktop IDE. Through Mist, the developer

can test in a web-based emulator and also build the application in the cloud. By simply scanning a QR-code, you can easily test the application directly on your device. Even iOS applications without being signed with a provision key can be tested on a device. This is possible through an iOS application called Icenium Ion and it is through this application the tested application you develop will be run.

4.5.3 Titanium

Appcelerator Titanium Mobile is developed by Appcelerator Inc and was founded in the United States during 2006 [14]. They started with mobile applications in 2008 when they release Appcelerator Titanium Mobile (sometimes only referred as: Appcelerator Titanium, Appcelerator or Titanium). The supported platforms are: iOS, Android, Blackberry and Tizen. Windows Phone is not supported but according to a blog, posted by an Appcelerator developer, it is planned that Appcelerator Titanium will support Windows Phone 8 in late 2013 [15].

The languages Titanium users develop in are JavaScript and TSS and the official IDE is Titanium Studio which is based on Eclipse. Alloy is the MVC-design Appcelerator supplies. The model is using the JavaScript framework backbone.js, the view is using XML and TSS and the controller is plain JavaScript.

4.5.4 Xamarin

Xamarin is a company founded in 2011 and got a set of tools for cross-platform development such as Xamarin.Android (previously Mono for Android) and Xamarin.iOS (Previously Mono). With the different tools, the developer can construct native applications for iOS, Android [16]. The development language is C# and you can either choose to develop in Xamarin's own IDE called Xamarin Studio or you can install add-ins to Visual Studio. Since it is already in C# the code can be reused with Silverlight to create an application for Windows Phone. Xamarin's products are not free. There is a Starter subscription which allows you to deploy to the devices, app stores and using Xamarin Studio. However, you have a limited size on your application and you can only write code in 32 kilobyte total. The cheapest alternative which allows you to write unlimited code is the Indie subscription which costs \$299 a year per platform, per developer.

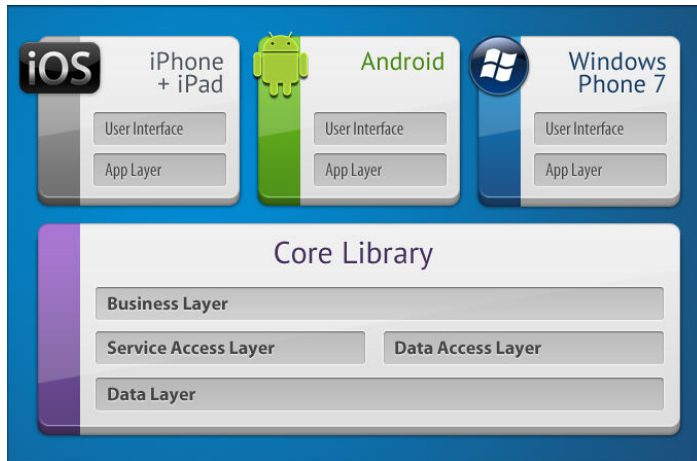


Figure 4.6: Overview of Xamarin [16]

4.6 UI-frameworks

User interface frameworks described in this section are the ones that will work with a hybrid development tool such as PhoneGap. These frameworks use JavaScript libraries and CSS to give applications a more detailed UI. Some frameworks can also provide a more native look for each platform.

4.6.1 jQuery Mobile

jQuery Mobile is a plugin to jQuery and is made for mobile devices and uses HTML5, CSS3 and JavaScript [7]. One huge difference between jQuery and jQuery Mobile is that jQuery uses the constructor:

```
$(document).ready(){
// Do something
}
```

and jQuery Mobile uses the constructor:

```
$(selector).live('pageinit', (function(event) {
//Do something
}));
```

The difference here is that jQuery sends an event when a webpage is loaded but jQuery Mobile sends an event when a page view has been initialized. The reason is that you can have multiple page views in one HTML5 web page. This is recommended hence there is no need for additional internet access when it is downloaded once. The downside is that the initial download will be slower.

4.6.2 Sencha Touch

Sencha Touch is a partially open source framework for mobile applications and was announced in June 2010 [?]. Applications are mostly written in JavaScript and CSS3 but also some HTML can occur and the supported platforms are Android, iOS, Windows Phone 8, BlackBerry, Kindle and Bada. Sencha Touch is a MVC-based framework and is built on the JavaScript library EXT.

The main elements of Sencha Touch are the components and these components allow nesting or docking with other components. The framework also offers data integration from different sources such as AJAX and JSON. Sencha have a lot of their widgets and features that will give a native feel to the application.

Chapter 5

Results

In this section I will first present the different tools and my own opinions on each tool. Then I will present a comparison based on scientific data. Last I will choose the set of tools for the creation of the CGI application.

5.1 Observations

The observations were that I developed a test application and tested the application on an iOS device and an Android device. I tested the different tools I described in chapter 4 and this section contains my own thoughts, feelings and opinions.

5.1.1 PhoneGap

With PhoneGap I used the Cordova plugin in Visual Studio and used the Windows Phone Emulator. The Windows Phone emulator is a bit faster than the other emulators and that is why the Windows emulator was chosen. From pressing run in Visual Studio until the application where installed on my Windows emulator and up and running was only 5,5 seconds. I also used PhoneGap build to test on real devices. For that I needed to pack my project in a ZIP-file, upload it to the cloud service, build it with provision keys, using a QR-code alternatively download it to my computer and send it via Bluetooth and then install it on my device. So each time I wanted to test on a real device I needed to do this procedure. The good thing is that it builds for all the platforms at the same time simultaneously. Thanks to the use of different IDE's for the development, PhoneGap can be used on different desktop operating system such as Mac, Linux and Microsoft Windows. Unfortunately, sometimes there are some exceptions because not all IDEs are compatible on all operating systems.

5.1.2 Icenium

Icenium is as mentioned in 4.5.2 built on Cordova and while testing this tool I could take the exact same code from my PhoneGap workspace and add it to my Icenium workspace, it also works the other way around. While working in Mist I experienced developing and testing to be done very quickly. The reason was that I was enabled to test directly in a web based simulator and didn't have to wait for a slow emulator. The downside with the web based emulator was that I couldn't test the native functions or widgets at each specific platform and it was only good to test the responsiveness and some other design conditions such as positioning and size.

The great benefit with Icenium is that testing on a real device can be done very rapidly. The first time I want to test an application I need to scan a QR-code from the screen with my Android and iOS device. In the Android device I needed to change some developer setting that allowed me to install a not signed application. In the iOS device I needed an application named Icenium Ion that would run my unsigned application. With the QR-code I could download the application directly on my devices and with the previously described setting I could also install and run without any problems. I could also alter and update my code in the web-based IDE Mist, save my settings then without redoing the previous described procedure, I could just update my application and it would download the latest change to my already installed application.

While having the application installed on my devices I experienced the application to be a bit slow. Icenium doesn't today support Windows Phone and it isn't free, however \$19 a month also includes the otherwise not free library, Kendo UI Mobile. I used a 30 days trial version while testing this tool. Since it is built on PhoneGap and uses Kendo UI Mobile and jQuery Mobile there is a lot of documentation, tutorials and examples available on the internet.

5.1.3 Titanium

I developed in Appcelerator's own IDE, Titanium Studio. First time developing, I did it without using the MVC-based Alloy and I experienced it as the code got a bit disordered which was solved when I started developing in MVC-based Alloy. The programming techniques between these two approaches were also quite different. The first approach without using Alloy, I wrote all my code in a JavaScript file. With Alloy I got a much more structure on my code and I experienced it to be much easier to work with. TSS is very similar to CSS3 and wasn't too difficult to understand.

To use Titanium Studio you need to add the platforms' SDKs and it uses the SDKs' emulators. Developing for Android, Titanium creates an

own "device" to use and I find this emulator to be very slow. I did some tests with my application and it took between 2.5 to 3 minutes to run the application until it is installed on the emulator. I tried to make it faster but without any great success.

I also experienced the application to run smoothly on my devices without much loading time.

5.1.4 Xamarin

I developed with Xamarin.Android in Windows 7 and Visual Studio 2010, with Xamarin.iOS I used Xamarin Studios on a Macintosh. The installation of the plugin in Visual Studio was no problem and it worked well with the Andorid SDK. However I find the emulator being relative slow (approximate 17,5 seconds) and for me who like to test relatively often when I implement new functions found it irritating. Another downside with Xamarin's products is that a well done UI is a bit tricky and difficult to implement. One of the advantages with an application developed with Xamarin tools is that the application is running fast with not too long experienced waiting time.

I also experienced that the code size limited in the starter edition was reached, just programming my test application. This means that if Xamarin's benefits are greater than the other tools, I will have to pay for the Indie subscription \$299 a year times two (\$299 for each platform). Since the existing web-based application which will be programmed in a application is written in C#, it is possible I can reuse some of its code.

5.1.5 jQuery Mobile

While developing with jQuery Mobile I used it together with Icenium and since it was as an option while creating project, the installation was very easy. Developing hybrid applications with jQuery Mobile makes the applications on the different platform look exactly the same, which could be a good and a bad thing. It could be a good thing if the user uses multiple devices with different operating systems and a bad thing if the user appreciates an operating system familiar design. A downside with jQuery Mobile is that I experienced it slower than Sencha Touch. Another downside is that it is difficult to get a good structure of the code because of how jQuery Mobile handles pages. It is possible to store different pages in different HTML documents but then you have to load them one at a time. It doesn't have to be a problem to load them one at a time since they are stored on your device and not on a server.

5.1.6 Sencha Touch

I tried to develop Sencha Touch with both PhoneGap and Icenium and it was very difficult to make these tools work together and I never succeeded with the Icenium integration. It also has a steep learning curve since it is a little bit different to a usual web UI. Since you write most of the code in JavaScript and nearly nothing in HTML it makes it a little bit different. I like The MVC pattern hence it makes the code more structured. I also experienced Sencha Touch to be faster than jQuery Mobile. Another good thing about Sencha Touch is that it has platform specific design.

5.2 Comparison

The comparison is based on scientific data and is evaluated from a company usage view. The comparison is not taking a monetary view in consideration.

5.2.1 Web application

The advantages with a web application is that it does not require any installation in the mobile device. This is also a disadvantage since you will always need an internet connection to be able to use your application. More disadvantages with always needed to be online are that it might suffer from worse performance due to delays in the connection. Another advantages is that since it is not installed in your device, you don't need an application update and instead you can maintain your application by update the application on the server. The different browsers on the devices are properly standardized and therefore the user interface can be reused across different platforms. Since the application is web based there is of course no application on the device and the user must use an address to navigate to the application [9][18].

5.2.2 Hybrid application

A hybrid application is installed on the device and of the major advantage is that a hybrid application can perform native platform features. The application is powered by the device computing capabilities. However the execution is performed in the browser and therefore has worse performance than a native application. A hybrid application uses JavaScript hardware abstraction layers and therefore is an issue to communication vulnerabilities with the device's native library (see figure 4.2). To create a platform native look and feel with a hybrid application you might be require to style each platform independently. Hybrid application can be used with a UI-framework to get the native feel and look but it will be a worse performance.[9][18].

5.2.3 Interpreted application

An interpreted application does provide the native look and feel well and the business logic can also be reused across the different platforms. With specific framework API's the developer can use device hardware and platform specific features. When it comes to reuse the code to the user interface on the different platforms, it is depended on the interpreters.[9][18]

5.2.4 Cross-compiled application

Applications that are cross-compiled can deliver all the native features and also reach the devices' hardware, software and native user interface components. A cross-compiled application also have a better performance and is closer to a native application's performance.[9][18].

5.3 Selection of tools

According to Xanthopoulos's and Xinogalos's analysis [18], there is no best solution for cross-platform development overall. To choose a cross-platform tool or methodology you need to have several factors in mind. In their conclusion of their comparative analysis they believe that the cross-compiled application methodology is the most promising approach because most of the performance is close to a native application but the biggest disadvantage is that they could not find any cross-compiled supporting non-commercial development environments. Another disadvantage is the design is more difficult for a cross-compiled application. If native feel is necessary they believe that interpreted applications are the best choice and says that interpreted and hybrid applications overall performance is medium.

Hetikötter et al. [19] compares PhoneGap with Appcelerator's Titanium and they believe that PhoneGap is to be preferred if native look isn't a big requirement. Designing user interface in Titanium Studio can take a while. They don't offer a what-you-see-is-what-you-get editor like others. Titanium also requires a bit of titanium knowledge since it is a bit different from web development like PhoneGap. The authors also experienced that the performance of both applications in Titanium and PhoneGap where about the same. PhoneGap is the better choice if you need support for Windows Phone, the developer can use already known web development skills and native look and feel is not a requirement.

According to Rahul and Seshu at Siemens [9] an interpreted/crossed compiled application is not preferred and the hybrid application is preferred and web application is the perfect methodology. As a server driven application they mean that that the business logic is mostly on the server-side and that the mobile device has user interaction and provides information. If

however the developer needs to use any of the features on the device, such as notifications or calendar the hybrid would be the appropriated choice. Overall these authors says that if you want a server and using the device's features, choose hybrid. If you don't need the devices features, choose web and if you don't need a server and want to create a standalone application, the interpreted/cross-compiled alternative is the best.

Choosing a hybrid methodology I still need a web mobile framework to easier create a more compelling user interface. Dalmasso et al. [20] have compared PhoneGap without any additional framework with PhoneGap and jQuery Mobile, PhoneGap and Sencha Touch 2.0. The authors have compared the memory usage of these three alternatives and they have looked at proportional set size (PSS) and unique set size (USS). PSS is the amount of memory that is shared with other processes and is the memory that the application would not release if it was released. USS is the amount of memory that would be release if the application was terminated. As displayed in the table, adding UI elements take more of the memory and jQuery Mobile uses less memory than Sencha Touch.

Set of tools	PSS	USS
Just PhoneGap	12091	6036
PhoneGap with jQuery	14730	9424
PhoneGap with Sencha Touch	24526	20164

The hybrid tool I choose is Icenium. It is just like PhoneGap since they are both based on Apache Cordova. The reason I choose Icenium over PhoneGap is because of its support on testing on device fast and simple without rebuilding and reinstallations. Icenium is not free but I could use a 30 days free trial. I will still need to use PhoneGap hence Icenium doesn't support Windows Phone. When the application is finished I will upload my code to PhoneGap build to get my Windows Phone application.

Selection of the UI-framework will be jQuery Mobile. I liked Snecha Touch that it got platform specific design and I experienced it faster than jQuery Mobile but jQuery Mobile uses less PSS and USS than Sencha Touch and hopefully the performance can be improved. But Sencha Touch was also very difficult to install together with Icenium and that is a deal breaker.

5.4 Application

The application was as mentioned developed with Icenium and jQuery Mobile. I used the cloud based IDE called Mist. The jQuery features that I used were its navigation bar and how it handles pages and buttons. The first page is the page where you made your booking. The user would choose

a start date, an end date with datepickers and which type of vacation it would be, which was represented by radio buttons.

For the datepickers I first tried to use the native datepickers by simply choose the input type to date. There was a problem on Android and it didn't work to have two datepickers in an Android activity or a jQuery Mobile page in this case. Therefore I needed to use a different datepicker to make it work since I didn't want two different pages.

The second page is where a user can see and manage its own bookings. The bookings are represented by buttons and pressing a button makes a menu pop up. In this menu the user can choose to alter the booking, remove it or shut down the menu.

The third and last page is where a user can see its colleague's bookings. One of the biggest issues was how to graphical present the booked dates from different colleagues in a user friendly way. I used a third part library called fullcalendar which basically is a calendar and when pressing a day, the information about the booking for wanted colleagues for that specific day will be presented.

One other problem was to make the application communicate to CGI's servers but it is not relevant for this report. As I developed with Icenium I had no support for Windows Phone which was one of the requirements. Icenium is as mentioned in 4.5.2 based on Cordova just like PhoneGap which got support for Windows Phone. To create a Windows Phone application I could just copy the code and upload it on PhoneGap build and build it there.

Chapter 6

Conclusion

In my opinion, cross-platform development is good enough for this kind of application. I agree with Xanthopoulos and Xinogalos about that there is no general best solution for cross-platform development. Depending on what you develop, each scenario has a preferred methodology.

I believe that a hybrid application could be a substitute to native application. If fancy user interface with sliding effects and resemble, then the hybrid application's performance can be increased. The performance of the hybrid application is the biggest issue in my opinion and cross compiled applications are a lot better. Like I wrote in "Selection of tools" to get a symmetric user interface can be time-consuming and applications like the tools from Xamarin cost a lot of money. Since I also developed an application which basically could be a webpage, developing a hybrid application is a better choice because it is also using web technologies in the development.

A web application is the best choice if there is no intentions of using any of the device's features. However it was an application that would be installed on the device I needed to develop and therefore a web application was never a choice. After developing the application I still feel that Cordova-based Icenium together with jQuery mobile was the best choice. To be able to test on the device fast and simple was the feature which made me choose Icenium over PhoneGap. Icenium also support Kendo UI Mobile which offers a native look and feel and could be a substitution to jQuery Mobile, however I couldn't find any scientific data about this framework and therefore I couldn't exanimate it any further.

Chapter 7

Discussion

I believe that cross-platform development is a good complement to native development. The biggest issue in my opinion is the performance with the cross-platform applications are inferior compare to the native application. I think that many developers that weight differences between cross-platform development and native development choose the native alternative because of lack of performance. Cross-compiled applications got a relative good performance but lack in the design of the user interface. Adding frameworks to ease the design or making it look native takes memory like we saw when comparing PhoneGap with and without jQuery Mobile and Sencha Touch.

As I wrote in section 3.1 about the mobile operating systems' market share, experts believe that the Windows Phone market will increase to 10.2% of the market in 2017. A lot of the cross-platform development tools doesn't support Windows Phone but I think s Windows Phone continue to grow on the market, we will see more tools supporting this operating system.

I also believe that PhoneGap/Cordova, Titanium and Xamarin tools will continue to be amongst the popular tools on the market. These tools have almost been along since the three operating system were founded. A lot of cross-platform tools have been founded since then but these three have survived. These tools continue to release new updates and are not far behind of maintaining the tools when the different platforms release a new version. More tools like Icenium which help and make the development more efficient can be interesting in the future and maybe become bigger than any of these tools.

7.1 Future work

I would like an investigation that tries to make the cross-platform applications experience better performance. As I described in the discussion

that the performance is the biggest issue and solving this issue would make cross-platform development an easier choice when comparing cross-platform development versus native development. For future work I would also like to see a more comparable study when developing the same applications with tools from all of the different approaches.

Bibliography

- [1] Hst Martin Rainer, Austen Runeson Per, *Case Study Research in Software Engineering : Guidelines and Examples*. John Wiley Sons, Inc., Hoboken, New Jersey, 2012.
- [2] International Data Corporation, *Worldwide Mobile Phone Market Forecast to Grow 7.3% in 2013 Driven by 1 Billion Smartphone Shipments, According to IDC*. <http://www.idc.com/getdoc.jsp?containerId=prUS24302813>, Accessed: 2014-01-12.
- [3] Apple Inc., *Develop Apps for iOS*. <https://developer.apple.com/technologies/ios/>, Accessed: 2014-01-12.
- [4] Google Inc., *Android Develop*. <http://developer.android.com/develop/index.html>, Accessed: 2014-01-12.
- [5] Microsoft Corporation, *Windows Phone*. <http://www.windowsphone.com/en-us>, Accessed: 2014-01-12.
- [6] Ramzi N. Sansour, Nidal Kafri, Muath N. Sabha, *A survey on mobile multimedia application development frameworks*. Al-Quds University, Palestine Technical University-Kadoori, Arab American University, 2013
- [7] Oswald Campesato, *jQuery, CSS3, and HTML5 for Mobile and Desktop Devices*. MERCURY LEARNING AND INFORMATION, 2013.
- [8] Ecma International, *ECMAScript Language Specification*. Ecma International, 2011.
- [9] Rahul Raj C.P, Seshu Babu Toletym *A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach*. Siemens Technology and Services Pvt Ltd, 2012.
- [10] Mahesh Babu R, M. Balaji Kumar, Rakesh Manoharan M. Soma-sundaram, S.P.Karthikeyan, *PORTABILITY OF MOBILE APPLICATIONS USING PHONEGAP : A CASE STUDY*. R.M.K. Engineering College, Kavaraipettai, India, 2012.

- [11] PhoneGap, *PhoneGap*. <http://www.phonegap.com>, Accessed:2014-01-12.
- [12] Manuel Palmieri, Inderjeet Singh, Antonio Cicchetti, *Comparison of Cross-Platform Mobile Development Tools*. Mölardalen University, 2011.
- [13] Telerik Inc., *Icenium*. <http://www.icenium.com>, Accessed:2014-01-12.
- [14] Appcelerator Inc., *Appcelerator Titanium*. <http://www.appcelerator.com/titanium/> , Accessed:2014-01-12.
- [15] Appcelerator Inc., *Appcelerator Blog*. developer.appcelerator.com/blog/2013/01/titanium-support-plans-for-windows-8.html, Accessed:2014-01-12.
- [16] Xamarin., *Xamarin.Android*, *Xamarin.iOS*. <http://www.xamarin.com/> , Accessed:2014-01-12.
- [17] Henning Heitkötter, Tim A. Majchrzak, Benjamin Ruland and Till Weber, *Evaluating Frameworks for Creating Mobile Web Apps*. Department of Information Systems, University of Munster, Münster, Germany, 2013.
- [18] Spyros Xanthopoulos, Stelios Xinogalos, *A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications*. Directorate of Technical Services and Computerization Aristotle University of Thessaloniki, Department of Applied Informatics University of Macedonia, 2013.
- [19] Henning Heitkötter, Sebastian Hanschke, and Tim A. Majchrzak, *Evaluating Cross-Platform Development Approaches for Mobile Applications*. Department of Information Systems, University of Munster, Münster, Germany, 2013.
- [20] Isabelle Dalmasso, Soumya Kanti Datta, Christian Bonnet, Navid Nikaein, *Survey, Comparison and Evaluation of Cross Platform Mobile Application Development Tools*. Mobile Communication Department, EURECOM Sophia Antipolis, France, 2013.

 Avdelning, Institution Division, Department HCS, Dept. of Computer and Information Science 581 83 Linköping		Datum Date 2014-01-24
Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____	ISBN 00 <hr/> ISRN LiU-Tek-Lic-2014:2014 <hr/> Serietitel och serienummer ISSN Title of series, numbering -
URL för elektronisk version http://XXX		Linköping Studies in Science and Technology Thesis No. 005
Titel Title Evaluation of cross-platform development for mobile devices		
Författare Author Joy Friberg		
Sammanfattning Abstract <p>Developing an application for several platforms can be time consuming because each platform has its own operating system and different developing language. Cross-platform development makes it possible to develop an application that will work on several platforms. This report will evaluate this kind of development by doing a case study for the company CGI. The case study will evaluate which cross-platform methodology is the preferred choice for this specific vacation booking application I developed for CGI. The different methodologies I studied were web, hybrid, interpreted and cross-compiled. The preferred methodology for this vacation booking application I developed was in this case the hybrid alternative. When selecting this methodology I also chose two different tools and those two were Icenium and jQuery Mobile. The purpose of this report was to find out if cross-platform development can be a substitute to native programming and by evaluating and developing cross-platform I found out that it can be a substitute if the application is not too complex. In this specific case I also believe that hybrid development is a good substitute to native development for this kind of applications.</p>		
Nyckelord Keywords Cross-platform, mobile, Sencha Touch, jQuery Mobile, PhoneGap, Icenium, Appcelerator, Xamarin, hybrid, interpreted, cross-compiled, application		