Smartphone Application Development using HTML5-based Cross-Platform Framework

Si-Ho Cha¹ and Yeomun Yun²,*

¹ Dept. of Multimedia Science, Chungwoon University
36-8, Dowha-dong, Nam-gu, Inchon, 121-812, South Korea
shcha@chungwoon.ac.kr

² Dept. of Applied Music, Chungwoon University
San 29, Namjang-ri, Hongseong, Chungnam, 350-701, South Korea
hippie740@chungwoon.ac.kr

Abstract. The mobile operating systems (OS) used by modern smartphones are too diverse such as Google's Android, Apple's iOS, Microsoft's Windows Phone, and so on. Smartphone application development is done using native platform such as iPhone using Objective-C, Android using Java, Windows Mobile using C# and so on. Therefore, a cross-platform framework which supports 'Write once and deploy everywhere' is required to support the development of smartphone applications. This paper presents the HTML5-based cross platform framework which uses PhoneGap and Webkit to support the development of smartphone applications that are written as Web applications.

Keywords: Smartphone Application, Cross-Platform, HTML5, PhoneGap, Web Application.

1 Introduction

The mobile operating systems (OS) used by modern smartphones include Google's Android, Apple's iOS, Nokia's Symbian, RIM's BlackBerry OS, Samsung's bada, Microsoft's Windows Phone, Hewlett-Packard's webOS, and embedded Linux. Such operating systems can be installed on many different phone models, and typically each device can receive multiple OS software updates over its lifetime [1]. In general, there are three different solutions for smartphone development: 1) native, 2) Web, and 3) hybrid. Native application is the application that works natively. Native application code is written specifically for a particular phone’s operating system. In native development, smartphone application development is basically done using native platform say iPhone using Objective-C, Android using Java, Windows Mobile using C# and so on [2]. Web application is the application that renders via a Web browser using Web application solutions including HTML, CSS, and JavaScript. There are three specific techniques in Web application development that are borrowed for these non-Web frameworks: 1) layout with mark-up (HTML5/CSS); 2) using URLs to
identify screen layouts and visual state; and 3) incorporating dynamic languages, such as Javascript and Ruby [3]. Hybrid application is the combination of a native application and a Web application. Smartphone frameworks are influenced by the rapid application development techniques we are seeing in Web development today [4].

Different Smartphone operating systems come up with their own App store for distributing native application. There are some issues associated with the native application development such as, Out-of-sync data, No two mobile platforms share a mobile application, and there are too many mobile operating systems exist in the market such as iPhone, Android, Blackberry, Symbian etc. Same application needs to be developed for different platform differently. This increases development cost [2]. The best approach to create truly cross-platform app is to use HTML5 [5] and JavaScript based on cross-platform frameworks. One interesting framework for creating HTML5/JavaScript based apps is PhoneGap [6] and Webkit [7]. It has quite many API's available and wide support for different platforms.

2 Related Works

There are constraints to be considered in selecting a native platform compared to the cross-platform. Table 1 shows the advantages and disadvantages of using native platform versus cross-platform environments [3].

<table>
<thead>
<tr>
<th></th>
<th>Native</th>
<th>Cross-Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>• Library update</td>
<td>• Write once, run a lot of places</td>
</tr>
<tr>
<td></td>
<td>• Direct technical support</td>
<td>• Open Source solution</td>
</tr>
<tr>
<td></td>
<td>• Stable</td>
<td>• One programming language family for all</td>
</tr>
<tr>
<td></td>
<td>• App store and device portal solution</td>
<td>• Fast development</td>
</tr>
<tr>
<td></td>
<td>• Better UI design result, can take full advantage of display</td>
<td>• Reduced long term maintenance cost</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td>• Not all have Open Source solution</td>
<td>• Limited direct technical support</td>
</tr>
<tr>
<td></td>
<td>• Different programming language</td>
<td>• Unstable</td>
</tr>
<tr>
<td></td>
<td>• Slow development time</td>
<td>• UI design depends on the platform and is limited</td>
</tr>
<tr>
<td></td>
<td>• Requires many budget and experience</td>
<td></td>
</tr>
</tbody>
</table>

3 Cross-Platform Framework

We introduce the cross-platform for smartphone applications as showed in Fig. 1. Our cross-platform framework uses HTML5, PhoneGap, Webkit, and so on. PhoneGap [6] is an open-source tool that lets you use JavaScript, HTML, and CSS to code an application once, then deploy it to the iPhone, Android, and BlackBerry. It is support to give Web developers JavaScript access to popular mobile device features, like the
Smartphone Application Development using HTML5-based Cross-Platform Framework

camera, GPS, the accelerometer, local SQLite databases and more, without having to write full applications.

![Diagram of Cross-Platform Framework]

**Fig. 1.** Our cross-platform for smartphone application development

jQTouch [9] is a jQuery [10] plug-in for mobile Web development with native animations, automatic navigation, and themes for mobile WebKit browsers on the iPhone, Adnroid, iPod Touch, and other forward-thinking devices. Webkit [7] is an open source library that renders HTML on the page. The iPhone and Google Android both use Webkit directly. Ajax (Asynchronous JavaScript and XML) [11] is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. JSON (JavaScript Object Notation) [13] is a text-based open standard designed for human-readable data interchange.

**References**

11. JSON. http://www.json.org/.