OULU UNIVERSITY OF
APPLIED SCIENCES

Ravi Thapa

HTML5 AS A CROSS PLATFORM TECHNOLOGY FOR THE MOBILE APPLICATION DEVELOPMENT

HTML5 AS A CROSS PLATFORM TECHNOLOGY FOR THE MOBILE APPLICATION DEVELOPMENT

Ravi Thapa

Bachelor's Thesis

Spring 2013

Oulu University of Applied Sciences

Degree Programme in Information Technology

PREFACE

The following thesis was commissioned by the Finland based company Comvise Oy. This thesis is an extensive research on the potential of HTML5's canvas and geo-location, for the cross-platform mobile application development with PhoneGap.

I would like to convey a special gratitude to my thesis supervisor, Mr. Veikko Tapaninen for this constructive criticisms, patience and support. I would like to thank Mr. Jarmo Karpelin, for the approval of thesis topic and Mrs. Kaija Posio, for the language inspection of this thesis. Another person I must mention is our Degree programme head, Mrs. Lea Hanilla, for her assertive guidance and support.

I would like to remember the help of my parents, friends studying at OAMK and company representative Mr. Binay Guragain. Last but not the least, I would like to thank Comvise Oy for providing this thesis topic.

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Degree Programme in Information Technology

Tekijä: Ravi Thapa
Opinnäytetyön nimi: HTML5 AS A CROSS-PLATFORM TECHNOLOGY FOR THE MOBILE APPLICATION DEVELOPMENT
Työn ohjaaja: Veikko Tapaninen
Työn valmistumislukukausi ja -vuosi: Kevät, 2013 Sivumäärä: 74

Matkapuhelinteollisuus on parhaimmillaan nykymaailmassa. iPhonen julkaisusta vuonna 2007 ja uusien käyttöjärjestelmien käyttöönotosta, maailma on nähnyt nopeaa kehitystä sekä uusia ulottuvuuksia tekniikassa. HTML5, alunperin rakennettu web-teknologia, on yksi niistä. HTML5:ssä ja sen lisäosissa on paljon potentiaalia muuhunkin kuin vain verkkosovelluksiin. Tämä työ keskittyy HTML5:n kehitysmahdollisuuksiin mobiilisovellusten tuottamiseen eri alustoille.

Työn aiheena on HTML5:n kaksi tärkeintä osa-aluetta, Canvas ja Geo-paikannus, ja se tuli suomalaiselta yritykseltä Comvise. Eri mobiilikäyttöjärjestelmien yhteensopivuuden ja mahdollisuuksien esittäminen HTML5:llä vaati laajaa tutkimustyötä.

Hybridisovellus kehitettiin HTML:ää, JavaScriptiä ja CSS:ää käyttämällä. JQuery mobilea käytettiin käyttöliittymän luomisessa, ja PhoneGapin PhoneGap Builderia lähdekoodin kääntämisessä. HTML5:n mobiilikäyttöjärjestelmien alustojen osoittamisen lisäksi tämä työ selittää lähdekoodin kääntömahdollisuuden ilman, että erillisiä SDK:ta ja IDE:tä tarvitsee asentaa käyttöjärjestelmään.

Kehitetty havainnollinen sovellus hyödyntää yhteensopivat ominaisuudet HTML5 kankaalle tehdäkseen animaatiomallin. HTML5 geo-paikantajaa on käytetty erottamaan maantieteelliset koordinaatit mobiililaitteen kautta PhoneGap API:n kanssa. Maantieteellisiä koordinaatteja käytetään näyttämään laitteen sijainti Googlen”road type”- kartalla. Ominaisuudet kuten Googlen satelliittikartta ja katunäkymä ovat käytössä sovelluksessa. Sovellus on rakennettu iOS:lle, Androidille, Windows Phonelle, BlackBerrylle, Symbianille ja WebOS:lle.

Asiasanat: HTML5, Cross-platform, Mobile, Operating-system, Hybrid, Canvas, Geo-location, PhoneGap

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology

Author: Ravi Thapa
Title of Bachelor's thesis: HTML5 AS A CROSS-PLATFORM TECHNOLOGY FOR THE MOBILE APPLICATION DEVELOPMENT
Supervisor: Veikko Tapaninen
Term and year of completion: Spring, 2013 Number of pages: 74

Mobile phone industry is at its best in today's world. Since the launch of iPhone in 2007 and introduction of various new operating systems, the world has seen a rapid development in new dimensions of technology.
HTML5 originally build for web technologies is one of them. HTML5 with its extended features over its previous version, projects its potential beyond just web applications. This thesis is focused on the potentiality of HTML5 to develop cross platform mobile application.

The research topic on the field of HTML5's two important elements Canvas and Geo-location was assigned to me by a Finland based company, Comvise. An extensive research was done to illustrate HTML5's capabilities and compatibilities across different mobile operating systems.

A hybrid application was developed using web technologies HTML, JavaScript and CSS. JQuery mobile was used for the user interface framework and PhoneGap's PhoneGap Builder for the compilation of source codes. In addition to verifying the HTML5's cross-platform nature on mobile operating systems, this thesis explained the ability to compile a source code without having to install separate SDK and IDE for each operating systems.

The developed demonstrative application utilizes the compatible features of HTML5 canvas to make animation rendering. HTML5 geo-location with PhoneGap API is used to extract geographical co-ordinates from the mobile device through an abstraction layer. The obtained geographical co-ordinates are used to show the device location in the Google "road type" map. Features like Google's satellite map and street-view are enabled within the application. The application was built across iOS, Android, Windows Phone, BlackBerry, Symbian and WebOS.

Keywords: HTML5, Cross-platform, Mobile, Operating-system, Hybrid, Canvas, Geo-location, PhoneGap

CONTENTS

TIIVISTELMÄ 4

ABSTRACT 5

1 ABBREVIATIONS AND SYMBOLS 9

2 INTRODUCTION 12

3 MOBILE PHONE INDUSTRY AND CROSS-PLATFORM DEVELOPMENT 13

4 HTML5 16

4.1 Evolution of HTML (Version 1 – 4) 16

4.2 Evolution of HTML5 17

4.3 Desktop Browser Support for HTML5 18

4.4 HTML5 in Mobile Devices 19

5 INTRODUCTION TO MOBILE OPERATING SYSTEMS 21

5.1 Apple iOS 21

5.2 Android OS 22

5.3 Windows Phone OS 23

5.4 BlackBerry OS 24

6 INTRODUCTION TO SCALABLE VECTOR GRAPHICS (SVG) 25

6.1 SVG on mobile devices 25

6.3 SVG compatibility and support on mobile OS 26

7 HTML5 CANVAS 27

7.1 Technical aspect of HTML5 <canvas> element 28

7.2 HTML5 canvas in Desktop Browsers 28

7.3 Comparison between Canvas and SVG 30

7.4 HTML5 canvas Specifications 33

7.5 Canvas declaration 33

7.6 Conformance requirements 34

7.7 Drawing on the Canvas 34

7.8 Transformations 36

7.9 Text 37

7.10 Images and Videos 37

7.11 Pixel manipulation 38

7.12 HTML5 Canvas Compatibility and Support on Mobile Devices 38

8.0 APPLICATION DEVELOPMENT WITH HTML5 CANVAS 40

8.2.1 Advantages of using HTML5 Canvas for Gaming 41

8.2.2 Draw backs of HTML5 canvas when used for gaming 42

8.3 Additional HTML5 frameworks 42

8.4 Codec issues of multimedia file formats used in canvas 43

9.0 HTML5 GEO-LOCATION 45

9.1 Methods in Geo-location API 46

9.1.1 Parameters used in Geo-location API 46

9.2 Error Handling 47

9.3 Security and Privacy in HTML5 Geo-location 48

9.4 Requirements for implementation of Geo-location API 49

9.5 Using HTML5 Geo-location API in an Application 50

9.6 Using mapping services with HTML5 51

9.7 Geo-location API support 52

10 HYBRID TECHNOLOGIES FOR CROSS PLATFORM DEVELOPMENT 53

10.1 PhoneGap 53

10.2 jQuery Mobile 55

11. PROOF OF CONCEPT 56
11.2 Application scope 57
11.3 Application testing and findings 59
11.3.1 Android OS 60
11.3.2 Windows Phone OS 62
11.4 Findings 62
FURTHER DEVELOPMENT 64
CONCLUSION 65
LIST OF REFERENCES 66
LIST OF FIGURES 74

1 ABBREVIATIONS AND SYMBOLS

Abbreviation	Detail
2D	Two Dimensional
3D	Three Dimensional
.apk	Android Application Package
API	Application Programming Interface
CDMA	Code Division Multiple Access
CPU	Central Processing Unit
CSS	Cascading Style Sheet
DB	Data Base
DOM	Document Object Model
DRM	Data Rights Management
GHz	Giga Hertz
GPU	Graphical Processing Unit
GPS	Global Positioning System
H.264	Standard for video compression
HTC	High Tech Computer Corporation
HTML	Hyper Text Markup Language
HTML2	Hyper Text Markup Language version 2.0
HTML3	Hyper Text Markup Language version 3.0
HTML3.2	Hyper Text Markup Language version 3.2
HTML4.1	Hyper Text Markup Language version 4.1
HTML5	Hyper Text Markup Language version 5
HTMLWG	Hyper Text Markup Language Working Group
IDE	Integrated Development Environment
IDL	Interactive Data Language
IDs	Identity
IE	Internet Explorer
iOS	i (iPhone) Operating System

JDK	Java Development Kit
JPEG	Joint Photographic Experts Group
JS	JavaScript
LCD	Liquid Crystal Display
MAC	Media Access Code
MATHML	Mathematical Markup Language
MB	Mega Byte
MD5	Message-digest algorithm
MIT	Massachusetts Institute of Technology
MP4	Media Player Version 4
MSDN	Microsoft Developer Network
MVC	Model View Controller
OGG	Free open container
OHA	Open Handset Alliance
OSX	Operating System X
PDA	Personal Digital Assistant
PC	Personal Computer
Q3	Third Quarter
QNX	UNIX like commercial operating system
RAM	Random Access Memory
RFID	Radio Frequency Identification RGB
RIM	Research in Motion
SDK	Software Development Kit
SGML	Standard Generalized Markup Language
SVG	Scalable Vector Graphics
SVGT	Scalable Vector Graphics Tiny
SVGB	Scalable Vector Graphics Basic
THEORA	Open video codec
UI	User Interface
URL	Uniform Resource Locator
VGA	Vector Graphics Array

VM	Virtual Machine
W3C	World Wide Web Consortium
WVGA	Wide Vector Graphics Array
WWW	World Wide Web
WebGL	Web Graphics Library
XAP	File format for package management system in Windows Phone
XHTML	Extensible Hyper Text Markup Language
XML	Extensible Markup Language
XNA	Microsoft provided set of tools with a managed runtime environment

2 INTRODUCTION

Comvise is Oulu and Finland - based Telecom Company, dedicated to provide optimized software and entrepreneurship. It is private company owned by its own employees. Employees having an average work experience of around 15 years, Comvise is always dedicated to new technologies and innovations.

HTML5 though still at a developing stage, is widely proclaimed as a potential technology for a cross-platform mobile application development. Comvise wanted to carry out research on the field of HTML5 canvas element and geo-location API to testify its cross platform nature. Furthermore, the company wanted to use PhoneGap to build an HTML5 application. As a student myself, learning different programming languages in school, I have always thought of the availability of *write once – use everywhere* coding system with one IDE to support all platform developments. I was granted the privilege to carry out this research by the company.

The first part of this thesis is dedicated to the research of HTML5's potentiality and its support across different mobile operating systems. The research is centered on a canvas element and geo-location API. As a proof of the concept, the second part of the thesis is dedicated to develop an illustrative application using web technologies HTML, JavaScript and CSS. PhoneGap framework and jQuery UI framework were used for the application development as instructed by the company.

Utilizing the canvas and geo-location functionalities, the built application renders an optimized animation using canvas properties, and Google map services using geo-location API. PhoneGap API was used to extract geo-location information from the device. Online 'PhoneGap build' was used as a cross platform compiler engine.

3 MOBILE PHONE INDUSTRY AND CROSS-PLATFORM DEVELOPMENT

Since the launch of the original iPhone in 2007 and, initial advancements shown by RIM's BlackBerry devices and Nokia's Symbian devices, the mobile phone industry has shown new dimensions of technology. Multi-touch, accelerometer, maturity on Internet and map-services are some major achievements, in the field. APIs are released by OS companies to let developers make the use of such features. Applications are widely developed for private and corporate customers. As a concrete history itself, due to the integration of these new dimensions of technologies in mobile devices, the market has shown a tremendous interest and growth in the Smartphone business. The market share of the current mobile operating systems and the projected market share are presented in *Table 1 and Figure 1*.

TABLE 1. Sales report of mobile operating systems (IDC Worldwide Mobile Phone Tracker 2013, Date of Retrieval 20.4.2013)

Operating System	**Q3 2012 Shipment Volumes**	**Q3 2012 Market Share**	**Q3 2011 Shipmnet volumes**	**Q3 2011 Market Share**	**Year-over-year Change**
Android	136,0	75,0%	71,0	57,5%	91,5%
iOS	26,9	14,9%	17,1	13,8%	57,3%
BlackBerry	7,7	4,3%	11,8	9,5%	-34,7%
Symbian	4,1	2,3%	18,1	14,6%	-77,3%
Windows Phone7/ Windows Mobile	3,6	2,0%	1,5	1,2%	140%
Linux	2,8	1,5%	4,1	3,3%	-31,7%
Others	0,0	0,0%	0,1	0,1%	-100,0%
Totals:	**181**	**100%**	**123.7**	**100%**	**46.4%**

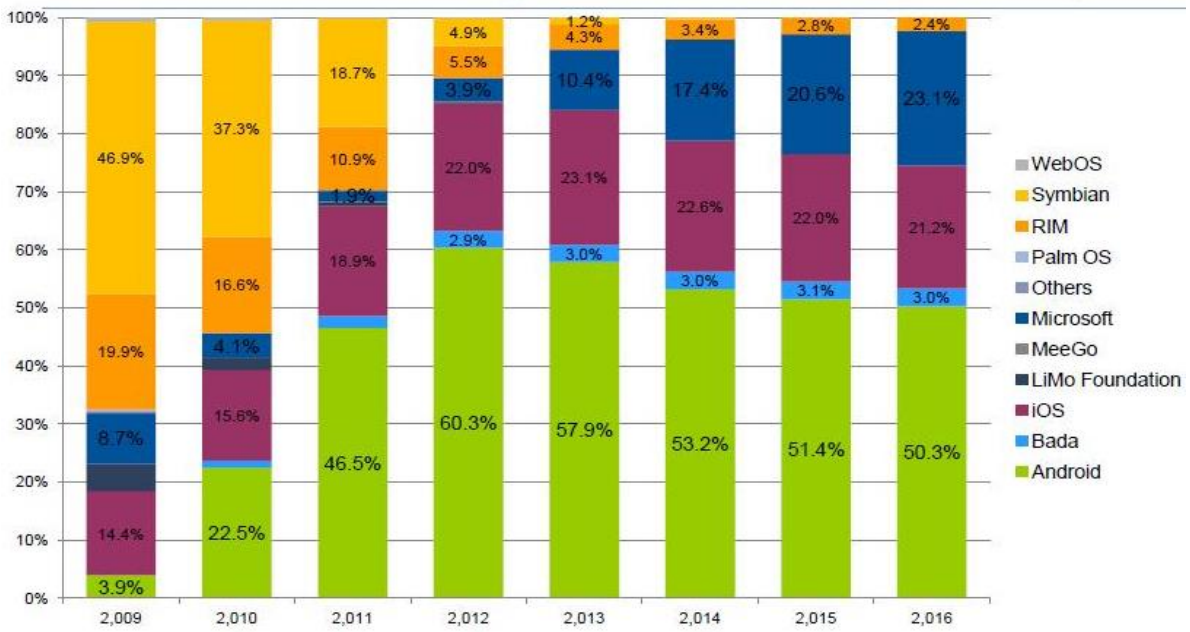

FIGURE 1. Screenshot of mobile operating systems' market share and future prediction (PhoneGap 2012a; Forbes.com LLC 2013, Date of Retrieval 20.4.2013)

While the mobile industry is at its best, a large number of users are migrating from Personal computing devices (desktop computers) to Smartphones and Tablet PCs to carry out their daily tasks. Corporate organizations and educational institutions are actively integrating their services and resources into mobile devices. The best possible software-applications are being developed for all types of customers. Companies dedicated to application and software development are being successfully established. A multiple variety of operating system platforms are being provided to customers by companies like Apple, Google, RIM (Research in Motion) and Microsoft. These companies play a key role on offering a variety of choices for the customers, with their respective mobile operating systems and devices. Customers from both common world and the corporate world are making their choices based on the features and eco-system these devices can offer. (PhoneGap 2012a; Forbes.com LLC 2013, Date of Retrieval 20.4.2013.)

The Presence of multiple operating systems from different companies helps to fragile any possible monopoly on both technological and economical aspect. For any application development company, profit-percentage from the sold application, a publishing mechanism of the application and technical support/resources for the application development, impact their entire business module. Hence, presence of multiple operating systems and evenly distributed market share is both beneficial and necessary. On the contradiction however, it is also evident for software and application development

companies, the necessity to construct applications not only for a single operating system, but for all. It is inevitable to ignore the importance of software development cycle in any of these operating system platforms. Moreover, it is necessary to develop and update applications, to stand out and stay current on this competitive industry. (PhoneGap 2012a; Forbes.com LLC 2013, Date of Retrieval 20.4.2013.)

Developing exactly the same application on different platforms these days requires a large amount of human and technical resources. Hence, also economic resources. It is a tedious task for the developers and software companies to develop and enroll updates to the applications. In addition, there is a great amount of risk involved in developing applications for a new operating system, which comes along in the market. Therefore, there is a need for a “write once, use everywhere” cross-platform application development language. HTML5 is emerging as one. (PhoneGap 2012a; Forbes.com LLC 2013, Date of Retrieval 20.4.2013.)

4 HTML5

HTML5 is the fifth revision of HTML (Hyper Text Markup Language). It is centered to work on developing technologies with backward compatibility, for existing browsers, as of June of 2004. HTML5 is an improvement over HTML4.1 and XHTML2.0, therefore it brings new features and corrections of syntax. HTML5 also provides a common ground to write either an HTML or XHTML syntax in one markup language. This was not possible in the previous versions of both HTML and XHTML. HTML5 focuses on extending, improving and simplifying currently available markup for the document. Therefore, it introduces markups and APIs for complex web applications. (Adobe Systems Inc. 2012a; Mozilla Developer Network 2005-2012; Wikipedia 2012a, Date of Retrieval 10.11.2012.)

4.1 Evolution of HTML (Version 1 – 4)

Tim Berners Lee purposed the concept of the Internet based hypertext system that will be capable of cross linking the text pages remotely. Thus introducing a hyperlink, SGML was made into HTML. It was created using the same markup language as ENQUIRE. This later became the foundation of WWW we know today. In order to verify his idea, Lee made software (now known as a browser), to interpret the text, and a server to store the file(s). Later Dan Connolly in his leadership accumulated and drafted the HTML tags. He published a document type definition for HTML2. (Addison Wesley Longman 1998; Wikipedia 2012a, Date of retrieval 15.11.2012.)

The third version of HTML, HTML3 was elaborated and published as an Internet draft in 1995. It was not labelled as a standard, due to large numbers of different proposals with new ideas. In 1996 W3C's HTMLWG published the HTML 3.2 as a standard. This was a stripped down version of HTML3 draft and some mutations were made. In this version Mathematical formula Markup was totally removed. Later, it was standardized as MATHML. In December 1997, W3C published the fourth version of HTML, HTML 4. It was released in three variants as Strict, Transitional and Frameset. (Addison Wesley Longman 1998; Wikipedia 2012a, Date of retrieval 15.11.2012.)

After publishing HTML 4.1, W3C decided to use XHTML (an XML based markup language) over the HTML. XHTML 1.0 was published in January 2000. This was more of an XML version of the HTML4 documentation. In May, 2001 W3C released XHTML 1.1. XHTML had implementation, compatibility and interoperability issues. However W3C decided to invest in it as the future. (Addison Wesley Longman 1998; Wikipedia 2012a, Date of retrieval 15.11.2012.)

4.2 Evolution of HTML5

HTML 4.1 and XHTML 1.1 had complex issues like implementation, compatibility and interoperability. Despite new proposals from Apple, Opera and Mozilla Foundation on HTML and XHTML, W3C declined and continued its development on XHTML. The dissatisfied team members with W3C, formed the WHATWG. This was the community to work on HTML development formed by Apple Inc., Mozilla Foundation and Opera Software. They jointly started developing the HTML5 after the HTML 4.1's last update in 2000. HTML5 was initially was conceived as Web Form 2.0 and Web Apps 1.0, which were later merged as a unit HTML5. Following the new proposals from WHATWG, W3C has been working with WHATWG on the new standard HTML5 since 2007. In their joint effort WHATWG dedicates to create specifications for HTML5, while HTMLWG (part of W3C) publishes them as an HTML standard. W3C has volunteered to expire XHTML 2.0 in 2009. (Wikipedia 2012a, Date of retrieval 15.11.2012.)

HTML5's first Public working draft was published by the WHATWG on 22 January 2008. HTML5 was partially implemented in browsers without reaching the final recommendation status. In May 2011, the working group advanced a "Last Call" invitation to communities, working both internally and externally to reach the technical soundness of the specification. On July 2012, WHATWG and W3C declared to work on different branches. W3C has started to work on a comprehensive test suite to achieve the broad interoperability for the full specification by 2014. The team was focused on a single definitive standard "Snapshot". WHATWG initiated working on HTML5 as a 'living Standard', i.e., it will continuously be improved and updated. (Adobe Systems Inc. 2012a; Mozilla Developer Network 2005-2012; Wikipedia 2012a, Date of Retrieval 10.11.2012.)

4.3 Desktop Browser Support for HTML5

HTML5, the markup language to construct and present the World Wide Web, is not a separate software needed to be installed in the system, but an update to the HTML language. Users only need to allow updates to their Web-browsers to support HTML5 features. Currently, the devices, which customers purchase with the support for the latest and greatest browsers, generally support HTML5 out of the box without having to install any additional software. All major desktop browsers support HTML5 as a whole with few exceptions. Major browsers are considered to be: IE (Internet Explorer), Firefox, Chrome, Safari and Opera. (Adobe Systems Inc. 2012a; Mozilla Developer Network 2005-2012; Wikipedia 2012a, Date of Retrieval 10.11.2012.)

HTML5 with its specification offers new APIs that can be used with a JavaScript scripting language. Its taxonomy and status is presented in *Figure 2*. Major desktop browsers fully support a DOM accessing method "getElementByClassName", a Hashchange" event, a "contenteditable" attribute and "New semantic" elements. The attribute for an external scripting"defer", Canvas, "Inline SVG", and audio and video elements are also fully supported. Features such as "Drag and Drop" which was not previously supported by Opera is now fully supported. "Offline web application" to view content even in case of no internet access is fully supported. However, "Session history management" is currently supported by FireFox, Chrome and Opera only. A script to download content without pausing the parser "async", a security mechanism to check untested codes from a third party "sandbox" and "WebGL" API to render 3D and 2D graphics are partially supported. The over-all browser support for HTML5 attributes by the latest version of Internet Explorer(10.0), Firefox(20.0), Chrome(26.0), Safari(6.0) and Opera (12.1) are 69%, 74%, 92%, 73% and 73% respectively. (@Fyrd 2012a, Date of retrieval 20.4.2013.)

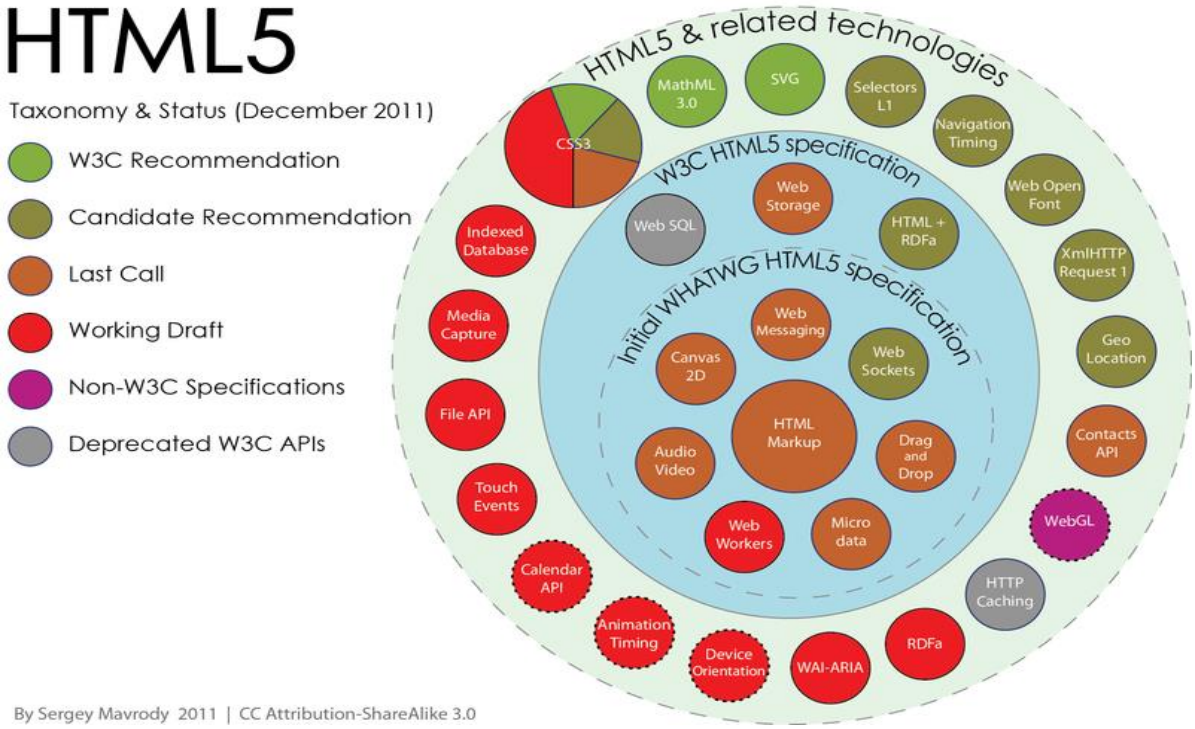

FIGURE 2. Screenshot of taxonomy and status of HTML5 elements (Sergey Mavrody 2011, Date of retrieval 20.5.2012.)

4.4 HTML5 in Mobile Devices

One of the HTML5's advancement over Adobe Flash, is its smooth performance even in low hardware capable devices such as smartphones and tablet computers (Steve Jobs 2010, Date of retrieval 12.2.2012). HTML5 stand on different spectrum than the resource heavy Adobe Flash. Flash was originally created for the devices which used mouse as a pointing device. Hence, it is not optimized for the low battery powered and touch based devices. Furthermore, Flash-Lite's run time engine optimized for mobile devices has performance issues of vector graphics, due to complex processing. Due to this reason, as of November 2011, Adobe announced to stop its support for a mobile flash and started investing in HTML5. Adobe purchased PhoneGap (Apache Cordova), an open source mobile development framework which was originally produced by Nitobi. This framework is used to make a hybrid application for different mobile platforms using HTML, JavaScript and CSS, basically HTML5.

(Adobe Systems Inc. 2012a; Mozilla Developer Network 2005-2012; Wikipedia 2012a, Date of Retrieval 10.11.2012.)

Essentially, HTML5's capabilities extend over the fences of Flash-player, as it offers some unique capabilities like an offline storage of web contents such as a list of URLs, HTML, CSS, JavaScript and resources up to 5MB size as cache. This means that previously visited web-applications and sites with a network connectivity, can be accessed in the same way even without network connectivity. With its efficient nature on CPU/GPU in comparison with flash and its availability of a local storage, HTML5 is a step closer to be a cross-platform mobile application development environment. (Adobe Systems Inc. 2012a; Mozilla Developer Network 2005-2012; Wikipedia 2012a, Date of Retrieval 10.11.2012.)

A research firm Strategy Analytics Forecast has estimated the business volume of HTML5 compatible phones to exceed 1 billion units by year 2013 AD (CBS Interactive 2012, Date of retrieval 10.10.2012). The native application that are developed for a particular operating system, not only lag behind on the aspect of not being able to work on another operating system but also on different version of the same operating system. In such scenario, HTML5's potentiality which uses HTML, CSS3 and JavaScript programming language, stands strong. However, the business module of the operating system companies would not allow the web application to be installed due to revenue issues. This means HTML5 can only be used to develop a hybrid application that behaves as native application. (Kravchick 2011, date of retrieval 16.5.2012.)

Since, for the execution of an HTML code, the application needs to trigger the browser engine within the application itself, this effects on the GPU and CPU performance adversely and can be significant factor in relatively low powered smartphones. Considering the size of cookie files is only 5KB, the unique feature of HTMl5 to have an offline content storage up to 5MB, can also be a bigger security threat in itself. (Kravchick 2011, date of retrieval 16.5.2012.)

5 INTRODUCTION TO MOBILE OPERATING SYSTEMS

Four major mobile operating system are included in this thesis as an area of research. Their operating system architecture, history and market share are highlighted in this section.

5.1 Apple iOS

iOS, previously known as iPhone-OS, is the operating system for mobile devices originally launched in 2007 by Apple Inc. iOS was the first operating system to provide a user interface based on direct manipulation using multi-touch gestures. Other Innovative features like multi-touch gestures, swipe, a pinch to zoom and a rubber band scrolling were introduced at its launch. The original iPhone also had features like a device orientation, a proximity sensor and a full matured version of web browser Safari. (Wikipedia 2012c, Date of Retrieval 5.11.2012; MacWorld 2007, Date of Retrieval 14.11.2012.)

iOS is a UNIX operating system with the Darwin foundation. It is based on four layers; abstraction, Core Services, Media and Cocoa Touch for the user Interface. The Objective C programming language is used for the native application development environment. (Wikipedia 2012c, Date of Retrieval 5.11.2012; MacWorld 2007, Date of Retrieval 14.11.2012.)

The latest version of iOS as of October 2012 is 6.0. Apple does not issue a license for use of iOS to any other hardware manufacturers. Apple Inc. following their core motto "*companies who love their software should make their own hardware*", builds its own hardware. This is in the same way, as it is in their desktop Operating system OS X. Since the launch of iOS, it has been bundled with the Apple's smartphone device iPhone. Today iOS runs on a music player and tablet devices such as iPod touch and iPad. (Wikipedia 2012c, Date of Retrieval 5.11.2012; MacWorld 2007, Date of Retrieval 14.11.2012.)

Apple Inc. has sold over 410 million iOS powered devices, and plans to sell 1 billion units by 2012. (Third Door Media 2012; Apple Insider 2012, Date of retrieval 25.10.2012). Apple Inc. allows developer to build applications using the APIs they have released for iOS devices. Apple is the market leader for application eco-system support. Thus, its application purchasing platform "App Store" has over 700,000 iOS

applications available as of September, 2012. (Wikipedia 2012c, Date of Retrieval 5.11.2012; MacWorld 2007, Date of Retrieval 14.11.2012.)

5.2 Android OS

Android is a mobile device operating system originally developed and established by Android Inc. in 2003. After Google Inc. purchased the company in 2005, Google Inc. has been developing Android with the adjoining support for the Open handset Alliance. (Open Handset Alliance 2012, Date of retrieval 20.20.2012.)

"The Open Handset Alliance (OHA) is a consortium of 84 firms to develop open standards for mobile devices. Member firms include Google, HTC, Sony, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung Electronics, LG Electronics, T-Mobile, Sprint Nextel, NVidia, and Wind River Systems." (Wikipedia 2012e, Date of retrieval 20.10.2012.*)*

Android is built using open Linux Kernel from the ground up. It is based on the ARM architecture hardware platform. Even though Android runs the Java source code, it does not use Java virtual machine. Instead it uses a Dalvik virtual machine. This is due to both business and technical reasons. Dalvik VM is an open source while Java VM is licensed under Oracle. However, the main reason is the performance issues. Java VM is used for a wide variety of devices hence optimization for mobile devices is minimum. Dalvik VM is optimized for battery consumption and low CPU powered mobile devices, as it is register based opposed to a stack based Java VM. Dalvik has a different set of Java libraries than JDK. Dalvik translates the Java-byte code Dex-code. This Dex-code combined with resources and native libraries are used to make Android "..apk" (Executable) files. Hence, Android SDK can be used with the Java Programming language to build native android applications. (Marko Gargenta 2010, date of retrieval 20.10.2012.)

Android is an industry leader in terms of the market share of mobile operating systems. Over 75 percent of the total smartphones sold in the third quarter of 2012 were Android powered phones, totaling to 136 million units (Android Central 2012, Date of retrieval 11.11.2012). Android has its own native browser

known as Android browser. The latest version of the Android is Android 4.2, also known as Jelly Bean. Google's Nexus 7 was the first device to ship Google-Chrome 18.0 as a standard and native browser for any Android powered devices. (Vox Media 2012, Date of retrieval 11.11.2012.)

5.3 Windows Phone OS

Windows phone OS is a mobile operating system developed by Microsoft Inc. It is a successor of Windows mobile, which was released in the year 2000. The windows phone OS was announced in October, 2010. Windows phone's updated version, Windows phones 8 was released on 29 Oct, 2012. Windows phone's application development is based on XNA and Microsoft Silverlight. Windows phone OS has natively bundled – Internet Explorer Mobile. It has a rendering engine based on Internet Explorer 10. (Wikipedia 2012e, date of retrieval 11.11.2012.)

Unlike Android Operating system, Microsoft demands, and has standardized minimum standards for the hardware features for symmetric experience. The hardware requirements enforced by Microsoft, since the launch of Windows phone 7 are: A Qualcomm Snapdragon processor, a DirectX graphics hardware support with a hardware acceleration for Direct3D using programmable GPU, the minimum of 256MB of RAM, the minimum of 8GB flash memory, 802.11 b/g wireless (802.11n is optional), Bluetooth, an FM Radio, a four-point multi-touch capacitive touch screen, GPS, an accelerometer, a magnetometer (compass), a proximity sensor, a light sensor, 800x480, a WVGA display resolution with the minimum 16 bits of color per pixel, the minimum of a 5-megapixel camera, the minimum of a VGA resolution video capture, a vibration motor, a micro-USB 2.0 and a 3.5mm stereo headset jack with a three-button detection support, buttons - start, back, search Power, volume Up, volume Down and a two-stage camera. (Pearson Education, Que Publishing 2012, Date of retrieval 18.5.2013.)

Windows phone has over 120,000 applications in its application purchasing console – "Windows phone Store". Applications developed for Windows phone 7 are being constantly updated to work with the latest Windows phone 8 version. However, only 13,000 applications have been verified till today. (TNW 2012, Date of retrieval 05.11.2012).

5.4 BlackBerry OS

BlackBerry smart phone devices are developed by RIM. BlackBerry devices run on the BlackBerry operating system. Like Apple Inc. RIM does not allow the licensing of BlackBerry OS to any hardware manufacturers, therefore they make their own hardware and software. BlackBerry was originally launched in the market as a smart-phone that had a physical keyboard on the device itself. BlackBerry devices were very popular with customers for their rich support for wireless networks. Its features like web browser, e-mail services and BlackBerry messenger are widely popular among customers. BlackBerry OS has its own native browser, a blackberry mobile browser. BlackBerry has sold over 200 million devices, as per September 2012. (Wikipedia 2012f, date of retrieval 05.11.2012.)

After the launch of iOS, the popularity gained by Android devices and entering of Windows Phone in the smartphone market with features like multi-touch, accelerometer, gyroscope, etc. BlackBerry decided to invest in touch based handheld devices to stay competitive in the mobile device market. BlackBerry 10 OS based on the QNX operating system was launched in first quarter of 2013 globally (Research in Motion Limited 2012; Wikipedia 2012f, date of Retrieval 06.06.2013)

6 INTRODUCTION TO SCALABLE VECTOR GRAPHICS (SVG)

The SVG specification is an open standard that has been under development by W3C since 1999. SVG has a file-name extension as .svg or .svgz. SVG is branch of specification for the standardized encoded XML file-format to define two-dimensional (2D) vector graphics. It can be interactive or dynamic. SVG as an XML file, has the support for searching & indexing, scripting and compressing. SVG can be edited in any text editors. Vector Markup languages like VML has a support for separation of orders, in drawing and documenting, in the case of over-lapping objects. This feature does not come in direct effect with SVG, as it does not have outright support for z.-indices. (W3C 2010a; Refsnes Data 1999-2012a; Wikipedia 2012g, Date of retrieval 2.5.2012.)

Scalable Vector graphic extends three types of graphical modules like vector shapes, raster graphics and texts. The grouping, styling and saving the previous state are possible. Enabled SVG functionalities according to W3C are “Paths”, “Basic Shapes”, “Text”, “Painting”, “Color”, “Gradients and Patterns”, “Clipping”, “masking and composing”, “Interactivity”, “Scripting”, “Animation”, “Fonts and Metadata”. (Wikipedia 2012g; W3C 2011b, Date of retrieval 2.5.2012.)

6.1 SVG on mobile devices

SVG has seen two modernization versions since year 2001, version 1.1 and version 1.2. The version 1.2 is still at a drafting stage, therefore the version 1.1 is currently the recommended version. SVG mobile group introduced a simpler version targeting at low powered mobile devices, SVG Tiny (SVGT) and SVG Basic (SVGB). SVG Tiny was specified for feature and hardware capability restricted mobile devices, like cell phones or feature phones. SVG Basic was developed for namely richer in feature and higher hardware processing powered equipped smart-phones and PDAs. An additional version SVG Print is a set of guidelines for the printable SVG 1.2 and Tiny 1.2 documents. SVG Print is still at the drafting stage. (W3C 2010c, Date of retrieval 3.5.2012.)

6.3 SVG compatibility and support on mobile OS

SVG features are widely supported in the native web browsers of major mobile operating systems Android (Native android and Chrome for Android), iOS Safari, BlackBerry Browser and IE (Internet Explorer for Windows Phone).

TABLE 2. Support for SVG methods on mobile devices (@Fyrd 2012b, Date of retrieval 20.4.2013.)

SVG Methods	**iOS Safari**	**Android Browser**	**IE (Internet Explorer)**	**Chrome for Android**	**BlackBerry Browser**
#SVG (Basic support)DZ # Inline SVG in HTML5 # SVG in CSS backgrounds	Version 6.0	Version 9.0 and above	Version 9.0 and above	Version 25.0	Not supported
#SVG filters #SVG effects for HTML	Full support	Not supported	Version 10.0	Full support	Full support
SVG SMIL animation #SVG fonts	Full support	Version 4.0 and above	No support	Full support	Full support
#SVG fragment identifiers	No support	No support	Version 10.0	No support	Version 10.0

7 HTML5 CANVAS

HTML5 canvas element is used for drawing and dynamic rendering of 2D shapes using CSS3 and JavaScript on the first hand. As HTML5 Canvas is widely supported by the desktop browser and follow-up support being implemented for the respective mobile browsers by mobile OS companies, it currently is the best candidate to be the cross platform for web animations and gaming console. (Google Developer I/O 2011a, Date of retrieval 20.12.2011.)

TABLE 3. Programming language support of different mobile OS (Google Developer I/O 2011a, Date of retrieval 2011)

Android OS	Dalvik and **HTML5**
iOS	objective C and **HTML5**
Windows Phone 7 / Windows Phone 8	XNA/Silverlight and **HTML5**
BlackBerry 10 OS	Java and **HTML5**

The complete break-down of support for a canvas element across different versions of the mobile operating platform, are illustrated in 'canvas-compatibility' chapter 7.12 (7.12 HTML5 Canvas Compatibility and Support on Mobile Devices) of this thesis.

HTML5 have been under development for a number of years, Apple Inc. initiated its contribution by introducing canvas to a dashboard widget in Mac OSX and then to a web-kit. When the Canvas element was first introduced by Apple in their proprietary browser safari, it was welcomed by a mixed reaction and treated as Apple Inc.'s proprietary technology. Later it was implemented in Safari browser and Google Chrome. Gecko 1.8-based browser like Firefox version 1.5 started their support for the canvas element in 2005. Opera browser support came in year 2006. In 2007, Apple syncronized the technology with W3C and made it open, providing the mechanism to be used by all. Very well documented APIs have been maintained since then, and almost all mainstream web browsers support most of their features. (Ethertrank 2011; WhatWG 2012a; Wikipedia 2012a, Date of retrieval 12.2.2012.)

7.1 Technical aspect of HTML5 <canvas> element

A canvas element written as <canvas> is library for graphics. It consists of an HTML code to allocate the draw-able region with height and width attributes. The <canvas> element with a scripting language like JavaScript allows a functionality to access and use common 2D APIs, used for drawing dynamic graphics using scriptable rendering of 2D shapes and bitmap images. The execution is based on updating bitmap. As it is a low level procedural model, and does not have a built in scene graph, scripting is used for drawing, not the canvas itself. The <canvas> is wisely used to build graphs, games, animation and also for image composition. (Ethertrank 2011; WhatWG 2012a; Wikipedia 2012a, Date of retrieval 12.2.2012.)

There are no support for the 3d context using HTML5 in current stage and only Blackberry 10 has announced to support it with its release. Even so, for its performance on the 2D context and promising future development, HTML5 is an appealing technology for developers itself. For a similar reason giant companies like Facebook Inc. and Google Inc. have started investing in HTMl5 canvas based App-store. In fact the app store for Facebook was recently launched on 14 May 2012. Due to the same reasons Google has already launched its "Web-store" for a Google Chrome browser, and it is only expanding day by day. Due to these scopes and reasons, HTML5 canvas is a promising technology to be a cross platform gaming and animation console. (Google Developer I/O 2011a; @Aaron Brandy 2012; AOL Tech 2013; Date of retrieval 18.5.2013.)

7.2 HTML5 canvas in Desktop Browsers

The web-browsers are expanding their support for HTML5 in each of their newer versions. As all computers are using the same major desktop browsers it is in the direction of being more and more hardware independent in terms of compatibility. However, performance is hardware depended as a web browser uses hardware accelerations of the local device. Microsoft's proprietary browser Internet explorer which did not have any good support for HTML5 till Internet explorer version 8(IE8), has launched its full support since the release of IE9 and IE10. On the bonus aspect, HTML5 canvas is not a mutually exclusive technology. It can run and be developed alongside with Microsoft Silverlight, SVG

and even Adobe flash. Therefore, it is providing a high possibility to be an ultimate animation and gaming console, choosing the best possible combination among these for any particular application, as the developer's delight. (John Bristowe 2011; @Fyrd 2012a, 20.4.2013.)

HTML5 canvas is not only promising to be a cross platform mobile application console for gaming and animation, but for the entire computer ecosystem that supports the latest and greatest browsers. That means devices having HTML5 supported browser irrespective of its hardware nature, can utilize the same HTML5 base code for animations. In this thesis though, the scope of HTML5 canvas will be narrowed down to a cross-platform mobile application development focusing on technical and business perspectives from different out-sourcing companies and OS vendors.

TABLE 4. Comprehensive desktop browser support for basic HTML5 canvas element and their market share (@Fyrd 2012a, Date of retrieval 20.4.2013)

Browser	**HTML5 canvass fully supported version(s):**	**Market Share**	**Remarks**
Internet Explorer	9.0, 10.0 (Near future)	26.43%	Not supported in version 8 or lower
Safari	5.0, 5.1, 5.2	14.09%	
Firefox	3.6, 9.0, 10.0, 11.0, 12.0, 13.0 (Near, future), 14.0 (further future)	21.79%	
Google Chrome	17.0, 18.0, 19.0 (Near future), 20.0 (further future)	24.99%	
Opera	11.6, 12.0	5.04%	

7.3 Comparison between Canvas and SVG

One of the major advancement in HTML5 canvas element is the ability to use SVG in-line HTML code. This has eliminated the mutually exclusive nature of canvas and SVG prevailed until the previous HTML version of HTML4.0.1 and XML version of XML1.x. (Rob Hawkes 2011a; John Bristowe 2011, Date of retrieval 12.2.2012.)

HTML canvas is an immediate graphic mode, which is a bitmap drawing with rectangular pixels in it. Unlike SVG, canvas is a non-retain mode. When any graphics are drawn using canvas on a web-browser, the browser has no knowledge of it. Pixels are manipulated while drawing any graphics and the colors are changed, just like in the simple Microsoft Window's application MS-Paint (Microsoft paint). Previously drawn images are not in a browser's memory and hence they are not layered. (Rob Hawkes 2011a; John Bristowe 2011, Date of retrieval 12.2.2012.)

Drawing the instruction of pixels on the screen are the consequences of following the direction via Java-Script. This cannot be queried from DOM apart from the element itself and neither can be manipulated via CSS itself. This means that when doing the animation, games and painting, all the processes and heavy lifting has to be repeated by the developer. Therefore, from this perspective the use of canvas is difficult in itself. When implemented on mobile devices, HTML5 canvas stands as smart choice, as device's memory/RAM is not occupied by the trace of previously drawn graphics. This is good to limit phone's power consumption and excessive use of hardware resources. (Rob Hawkes 2011a; John Bristowe 2011, Date of retrieval 12.2.2012.). This was the main reason for Apple Inc. to announce its end of support for Adobe Flash and beginning of support for HTML5. They also announced that their mobile devices running iOS (operating system for Apple Inc.'s mobile devices like iPhone, iPad and iPod) will never run Adobe Flash. This was major technology news and hence fluked battle among Apple Inc. and Adobe at the time of announcement (Steve Jobs 2010, Date of retrieval 12.2.2012). Later however, even though Adobe had a major version of flash running on Android powered devices, it announced its end of support for mobile flash (Richard Lawer 2011, Date of retrieval 12.2.2012). As-well began its investment on HTML5 development.

The contrast technology, SVG can be styled with CSS and manipulated by the DOM and also preserves the sketches of the graphics. If attributes of an SVG object are changed, the browser can automatically re-render the shape. However canvas and SVG are not mutually exclusive, they can both be used in concert. Canvas comes in handy in cases like dynamic graphics, instead of drawing hundreds and thousands of images at different level, a single pixel or a set of pixel can be manipulated. Canvas haa an advantage over SVG, as it doesn't retain the previously drawn graphics or screen in browsers cache and memory, the animation performance is very fast, and hence it is preferred for the dynamic graphics over SVG. (John Bristowe 2011; Mihai Sucan 2010, Date of retrieval 6.2.2012.)

TABLE 5. Comparison of Canvas and SVG (Refsnes Data 1999-2012a, Date of retrieval 20.2.2012)

	Canvas	**SVG**
Abstraction	Pixel based (dynamic bitmap)	Shape based
Elements	Single HTML element	Multiple graphical elements which become part of the Document Object Model (DOM)
Driver	Modified through Script only	Modified through Script and CSS
Event Model	User Interaction is granular (x, y)	User Interaction is abstracted (rect, path)
Performance	Performance is better with smaller surface and /or larger number of objects	Performance is better with a smaller number of objects and/or larger surface.

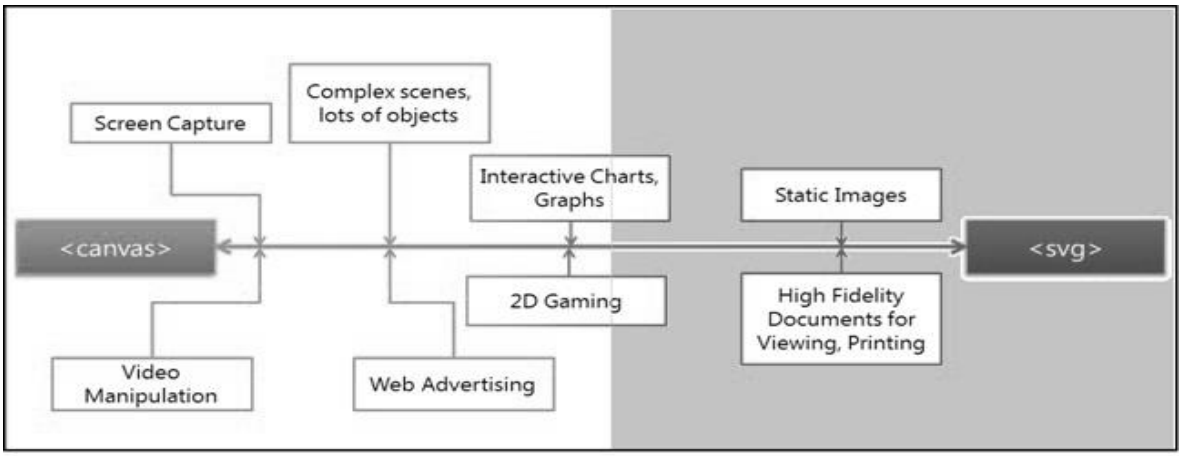

FIGURE 3. Best use case scenario for SVG and Canvas (MSDN Microsoft 2011b, date of retrieval 15.5.2012.)

TABLE 6. Major technical differences between Canvas and SVG (Refsnes Data 1999-2012a, Date of retrieval 20.2.2012)

Canvas	**SVG**
- *Resolution dependent* - *No support for event handlers* - *Poor text rendering capabilities* - *You can save the resulting image as .png or .jpg* - *Well suited for graphic-intensive games*	*- Resolution independent* *- Support for event handlers* *- Best suited for applications with large rendering areas (Google Maps)* *- Slow rendering if complex (anything that uses the DOM a lot will be slow)* *- Not suited for game applications*

Since HTML5 is still under development and at the drafting stage, currently canvas has a support only for the 2D graphics. In case of 3D graphics, they are to be rendered using alternative specifications like WebGL. (Wikipedia 2012e, Date of retrieval 15.5.2012.)

7.4 HTML5 canvas Specifications

HTML5 canvas specifications have been under development for a number of years. The projected completion date for the first full specification of HTML5, is by late 2014. At present, the following HTML5 <canvas> specifications are available for the uses:

1. State
2. Composition
3. Colors and styling
4. Line styles
5. Shadows
6. Simple and Complex shapes
7. Transformations
8. Focus management
9. Text
10. Images – also includes movie that plays on H.264 standards
11. Pixel manipulation – using direct RGB value for each pixel. (Refsnes Data 1999-2012b, Date of retrieval 23.5.2012.)

7.5 Canvas declaration

Canvas declaration in HTML5 is made using a *canvas id* tag. Considering the top-left corner of the screen as origin, with the value of *x,y* co-ordinates as (0,0), CSS is used to draw simple shapes on the screen. In order to access a 2D rendering content, JavaScript is utilized. A Canvas element acts as the window for the JavaScript manipulation.

```
<canvas id = "my Canvas" width ="499" height= "499"> <! – Content here -->
</canvas>
```

(Html5marketplace 2010-2011, Date of retrieval 2.7.2012.)

7.6 Conformance requirements

The conformance requirement is a HTML specification. It deals with the confirmation for all requirements, classes, definitions, dependencies, terminology and typography conventions, which are declared in the core level of the HTML5 specification. Interfaces in HTML are coincided as in Web IDL. Web IDL is a programming language used for data analysis on the web. The conference declares the 2D type and implement in the interface using CanvasRenderingContext2D. This is achieved by using a canvas element method called *getmethod()*. When the method is used to return a new object for the contested 2D (two dimensional), only return *CanvasRenderingContext2D* is accepted. 2D context is a representation of the web page or an area of the website. The surface is allocated in any position using the coordinate system of *x* and *y*. The position shifts to the right as *the x value* increases and moves down while incrementing the *y* value. Since the origin (0,0) is on the top left corner of the screen, no negative values can be assigned to the *x* and *y* coordinates. (WhatWG 2012b; Apple Inc. 2010, Date of retrieval 2.7.2012.)

7.7 Drawing on the Canvas

HTML5 canvas provides a specification to draw different shapes, use different color styles and gradient backgrounds. The Canvas specification also provides the functionality to save the state of the previous canvas rendition.

Colors and style

Colors and style uses attribute *fillStyle()* to draw a simple geometric shape and fill its entire body with the provided 'rgb' value for the color. *strokeStyle()* is used to draw the shape with only outline of the shape being colored, with the provided 'rgb' color value. (Tutorials point 2012a; Refsnes Data 1999-2012b; W3C 2012d, Date of retrieval 10.7.2012.)

Line Styles

Line Styles provides functionality to draw the different style of line's ends such as round, square, bevel, butt and mietel. This attribute is also used to move the line in different location within the canvas and to

determine its width and length. (Refsnes Data 1999-2012b; W3C 2012d; Tutorials point 2012a, Date of retrieval 23.7.2012.)

Composition

Composition uses global attributes *globalCompositeOperation* and *globalAlpha* that are responsible for all the graphical drawing operations. They are used to create source-over, source-in, source-out, source-atop, destination-over, destination-in, darker, copy and xor, between any two shapes. (Refsnes Data 1999-2012b; W3C 2012d, Date of retrieval 4.7.2012.)

Simple& Complex shapes

Two significant attributes *ctx.beginPath ()* and *ctx.arc()* are used to draw the simple circle on the canvas (Refsnes Data 1999-2012; W3C 2012c, Date of retrieval 27.7.2012). In order to draw the complex shape like quadratic curves in the declared canvas context, attributes *ctx.linewidth* and *ctx.strokestyle* are defined for width property and color respectively (Refsnes Data 1999-2012b; W3C 2012d; Tutorials point 2012a, Date of retrieval 23.7.2012).

Bezier is a random, asymmetrical and interesting curve. Its shape can be determined by changing the values of various nodes in the curve path. In order to draw the Bezier curve, *bezierCurveTo* attribute is used to assign a new point in the currently drawn curve. The new point is not connected to the previous path by the cubic brazier. Clipping in canvas, is basically clipping or highlighting the section of graph. It is created with a simple attribute *ctx.clip(),* used alongside the attribute(s) to create a graph in the first place. (Refsnes Data 1999-2012b; W3C 2012d, Date of retrieval 27.7.2012.)

Shadows

Visual enhancement effect like a shadow can be implemented to the shapes in canvas. The Shadow specification uses attributes *shadowOffsetX, shadowOffsetY, shadowBlur* and *shadowColor* for the different types of shadow shapes, density and color, upon the gives coordinates of the shapes. (Refsnes Data 1999-2012b; W3C 2012d; Tutorials point 2012a, Date of retrieval 24.7.2012.)

Gradients and Patterns

This *createLinearGradient()* attribute is used to define slope-lines or the gradual increasing value, using the initial and final coordinates on canvas (Refsnes Data 1999-2012b; W3C 2012d, Date of retrieval

13.7.2012). The *(*=Pattern)* attribute can be used for patterns of both geometrical shapes and image files. The number of times for an object to be re-drawn and the location path can be manipulated. In both CanvasGradient and CanvasPattern attributes, a string value can be applied. The provided string value must parse and be synthesized with the CSS value (CSS <color> value) and the applied value. If neither of these values are provided as required, they are ignored and the attribute needs to retain its previous value. However, if the fill and stroke event takes place and if the provided color value is non-serialized, this might return opaque objects. (Refsnes Data 1999-2012b; W3C 2012d, Date of retrieval 17.7.2012.)

Canvas State

The canvas element's drawing state is a carbon copy of transformations and styles in the stack. When a saving method *ctx.Save()* is called, it pushes the state in stock. The *Save* state values are retained by using a restore method *ctx.Restores*. (Refsnes Data 1999-2012b; WhatWG 2012b; W3C 2012d, Date of retrieval 2.7.2012.)

7.8 Transformations

Transformation of shapes and paths in HTML5 is carried out using a Transformation matrix. The *transformation* includes translate, rotate and scale. It is applied to the coordinates while shapes and paths are drawn in canvas. All objects implementing a Canvas Transformation attribute or interface, have a current state of transformation matrix. Transformation matrix is initialized to the identity transformations of any created objects under Canvas Transformation interface. It may then be adjusted using the transformation methods such as **scales, rotate, translate** and **transform**. The transformations must be performed in reverse order. (Refsnes Data 1999-2012b; W3C 2012d, Date of retrieval 29.7.2012.)

In order to transform the transformation matrix by a defined scale, '*context. scale(x, y)'* align the *width(x)* and *height(y)* of canvas is used. The rotation transformation of the transformation metric *context.rotate(angle),* is applied in the provided diagnostics. The angle value is always defined in radians. Translating in canvas refers to the re-sizing of the dimension by the developer assigned alteration value. The Attribute *context.translate(x, y)* is used to change the transformation matrix to apply a translation

transformation in the provided diagnostics. The transformation matrix can be transformed by the argument values. The *context.transform(a, b, c, d, e, f)* method replaces the current transformation matrix, with the result of multiplying the current transformation matrix. The *setTransform(a,b,c,d)* resets the applied transformation to the identity matrix, and then the *transform (a, b, c, d, e, f)* method with the same arguments. (W3C 2012d, Date of retrieval 29.7.2012.)

7.9 Text

HTML5 canvas element using the 'Text' specification, has provided the capabilities for the customization and implementation of font style and arrangements. Their properties manipulates various aspects such as alignment, size, position on canvas and font coloring styles. Color styles includes both *fills* and *stroke.* Text specification has also provided the metrics - *context.measureText().* This is used for identification and display of the objects, it is implemented with a width property. The *context.measureText()* method returns metric objects depending on the current local font settings for the provided string texts. Attribute *metrics.width()* extracts the width value of the text provided in *context.measureText().* Obtained width values are represented as a pixel value. (W3C 2012d, Date of retrieval 29.7.2012.)

7.10 Images and Videos

The importing of external audio and video files is enabled in canvas. The imported video file format needs to be supported by an H.264 standard, to be used or played. Different attribute properties are utilized for the images and video files to be rendered or drawn in the canvas. The JPEG file "imagename.jpg" contained in the same directory as the main HTML or JavaScript file can be imported and drawn into the canvas as:

var img = new Image(); img.src = '/pic/imagename.jpg'; context.drawImage(image, dx, dy); context.drawImage(image, dx, dy, dw, dh); context.drawImage (image, sx, sy, sw, sh, dx, dy, dw, dh);
(W3C 2012d, Date of retrieval 29.7.2012).

If the imported file to the first argument in the attribute *context.drawImage(),* is not the image, canvas or video element, a *TYPE_MISMATCH_ERR* exception error occurs. An *INVALID_STATE_ERR* exception exists when the first argument has no image data. A *SYNTAX_ERR* exception means that the second arguments are beyond the allowed values. Image files must be fully decoded to be drawn in the canvas. (W3C 2012d, Date of retrieval 29.7.2012.)

7.11 Pixel manipulation

Pixels in the canvas can be manipulated using the JavaScript and the alterable RGB values in hexadecimal. Alpha values are used to determine the static behaviors. Pixels of anything drawn to the canvas including simple geometrical shapes and images can be manipulated for purposes such as cropping, resizing, drawing and changing the color of each pixel. The number of images pixel or a set of pixels can also be determined. Functions are carried out using the attributes proved with assigned vertices of the canvas in the arguments. The *golbalAlpha, globalCompositeOperation* and *shadow* attributes are ignored during the *context.putImageData* method, for the new drawing. Entire pixels in the canvas are replaced ignoring the previous composition, shadows and alpha blend. (Refsnes Data 1999-2012b; W3C 2012d, Date of retrieval 17.7.2012.)

7.12 HTML5 Canvas Compatibility and Support on Mobile Devices

The support for the HTML5 features on any mobile operating system is determined by the support of their native web browsers. The respective native browsers for Windows Phone Apple iOS, Android and BlackBerry are iOS Safari, Android Browser or Chrome for Android depending upon the version, and BlackBerry Browser.

TABLE 7. HTML5 Canvas support across native web-browsers of mobile operating systems (@Fyrd 2012c, Date of retrieval 11.11.2012)

HTML5 Canvas features	**Internet Explorer**	**iOS Safari**	**Android Browser**	**Chrome for Android**	**BlackBerry Browser**
JavaScript based dynamic drawing	9.0 – 11.0	3.2 – 6.0	2.1 – 4.1 (except Clipping)	18.0	7.0, 10.0
Method for displaying Text on screen	9.0 – 11.0	3.2 – 6.0	2.1 – 4.1	18.0	7.0, 10.0
CSS Canvas Drawing HTML5 for background image drawing	8.0-10.0 (not supported) 11.0 (support unknown)	3.2 – 6-0 (webkit)	2.1 – 4.1 (webkit)	18.0 (webkit)	7.0, 10.0 (webkit)
Full Screen API	Not supported	Not supported	Not supported	Not supported	10.0 (partial support)
WebGL 3D canvas graphics	8.0-10.0 (not supported) 11.0 (support unknown)	Not supported	Not supported	Not supported	10.0

Features like Full Screen API and WebGL for 3D animation is not supported by native mobile browsers. Although Full Screen API is not supported, building the hybrid application with PhoneGap will eliminate the need to operate an application in the full screen mode. The hybrid application built already runs on full screen mode by default. 3D animation is not possible in mobile devices. However, dynamic drawing, CSS and text are supported. Therefore, the HTML5 canvas element meets the need to be a cross platform 2D animation technology to some extent.

8.0 APPLICATION DEVELOPMENT WITH HTML5 CANVAS

8.1 Animation with HTML5 canvas

Animation in the canvas element is based on the concept of **Translation,** or setting the **set interval** and **set timeout.** JavaScript is used for both drawing and redrawing of graphics on the screen. Animation with the **set interval** and **set timeout** is a result of drawing, clearing and re-drawing of shapes in a different position, kept in a loop.

setInterval(callback, time); repeatedly executes the supplied code, with a time interval assigned between them in milliseconds.

setTimeout(callback, time); executes the supplied code only once, after a given time in milliseconds. (Rob Hawkes 2011; Date of retrieval 14.2.2012.)

Canvas needs to be drawn periodically by the developer himself. It is the developer's responsibility to allocate the function to draw it again, and again. This is a heavy lift from the developer's perspective initially. However, the advantage of this setting is evident, as the developer is incharge of everything. There is not any notion on the browser side what is being drawn in the context. There is no history. This allows an extremely fast root for the browsers to render new graphics, as there is no data to be cleared or to written on top of it. This is due to the absence of layers of images. Therefore, HTML5 canvas is light weight. An excellent performance can be achieved and is a concrete choice for gaming or animation, as the technology is entirely based upon pixel-manipulation. (Rob Hawkes 2011b; John Bristowe 2011, Date of retrieval 16.2.2012.)

Normally, the minimum time interval for updating the screen is kept at 16-17ms (mille-seconds), as most of the LCD screen shipped with mobile devices are at 60 Hz motion flow. Decreasing the update interval for animation and rendering of graphics will increase the processing cycle of a processor of devices. Saving the number of execution of processing is essential for both limitation the over use of hardware resources, and conserving the battery life in mobile devices. Normally, a 33 milliseconds gap is optimum for both smooth animation and limiting of excessive use of hardware resources. (Rob Hawkes 2011b; John Bristowe 2011, Date of retrieval 22.2.2012.)

8.2 Game development with HTML5 canvas

Gaming for HTML5 is gaining popularity every day, as it uses technologies that are normally used to make the web with. Big companies have been significantly interested in the funding of HTML5 game companies, e.g.: Acquisition of gaming engines, like Aves by Zynga, and rocket Engine by Disney. Also, a company like Facebook is getting involved in HTML5 gaming for its performance.

One of the advantages of HTML5 is the achievement of "write once use everywhere". The developer does not need to write code for each platform all the time. However, this is only 100% true in the case of browsers for particular devices. Few changes and tweaks are necessary as per the OS demands and screen resolution. (Rob Hawkes 2011b; John Bristowe 2011, Date of retrieval 27.2.2012.)

8.2.1 Advantages of using HTML5 Canvas for Gaming

Game development and the gaming experience with the use of HTML and its canvas element has various benefits due to its wide range of support for the gaming related technologies. HTML has the support for the game data to be stored remotely in the server. Mongo DB and Redis are the popular and commonly used servers for such a purpose. However, Web-Sockets are also present in the HTML5, therefore data need not be queried and downloaded from a server every time the user wants to play the game. This enhances the gaming performance. A local storage is also present for this purpose. The user can pick up from the point where it was left previously, as cache game data is possible up to 5MB. Powerful hardware acceleration is enabled on mobile devices for HTML5 and also the touch inputs for game control on the screen are supported. This comes in addition to the already existing common controls such as keyboard, mouse and D pads on mobile devices. WebGL is 3D but currently uses the 2D canvas, as it has not yet matured. It is still in its version 1.0 (as of 15th –May-2012) in its specification. Though WebGL is not supported in mobile devices, the performance is regarded to be very good on desktop browsers, as it can also utilize the power of hardware acceleration. (Rob Hawkes 2011b; John Bristowe 2011, Date of retrieval 5.5.2012.)

8.2.2 Draw backs of HTML5 canvas when used for gaming

Not all browsers support everything of HTML5 yet (September 2012). For example WebGL has no support on Internet explorer, Safari browser and Opera. Also WebGL isn't supported on any native mobile web browsers. No DRM (Digital Rights Management) is present in the web application, this means anyone can see one's source code. This can lead to modification of the codes while playing multi-player games and hence manipulate the effect adversely by the user. Hybrid application(s) is not effected with DRM issues though, as the development is through the same compilation process as for the native applications. Canvas can make use multimedia file such as audio, but it lacks in performance when kept in loop for gaming. There is always a gap when in the start of the new loop. (John Bristowe 2011; Rob Hawkes 2011a, Date of retrieval 5.5.2012.)

8.3 Additional HTML5 frameworks

HTML5 canvas element, with some considerations can make use of additional features that are available to help the efficient use and development of applications. During the process of animation or game development redrawing, sprites and sounds are the major part of the heavy lifting task that is needed to be carried out by the developer. JavaScript frameworks such as *Easel JS* and *Raphael JS*, which use the JavaScript libraries are available to be used for majority of such heavy lifting tasks. A framework consists of the layer of interaction for canvas APIs and JavaScript, which a code developer needs to write. In SVG it is possible to use Hover events. It is best to use canvas for dynamic graphics and SVG for Static graphics (because it is based on a vector, it will scale in and out). The best practice for using SVG is displaying the online maps. To incorporate the SVG in canvas, a framework such as Raphael JS can be used. Since these frameworks are available in a wide variety for different purposes, they can be selected according to the animation requirements of the application and support/compatibility.

Keeping the logs of changes of each browsers, their version number and specification is a tedious task during the application development process, as they are changed rapidly and are being constantly updated. Therefore, in case of development of the web-applications, it is best to use a feature detection technology than a browser sniffing. This can also be enabled whilst building the hybrid application for

mobile devices. A framework such as *modernizr.com* can be used for such purpose. The *modernisr.com* framework is rich in features of canvas and other CSS3 engines, too. For drawing features like text shadow, both shadow and border radius, YepNote framework has the best available support.
(Rob Hawkes 2011b; John Bristowe 2011, Date of retrieval 20.8.2012.)

8.4 Codec issues of multimedia file formats used in canvas

Multimedia files can be used in concert with canvas when developing an application such as animation and gaming. Sound effects and video clips are often used to enhance the application. Considering the user type and target devices, the issue of codec needs to be addressed while developing an application. Still under development, HTML5 has standardized the way of embedding the content via its new elements <video> and <audio> in canvas, which have been absent in the web technology until now. The <video> in ideal case is capable of an extremely high compression, with the minimal use of a decode processor. This element also tries to solve the issues like the need to install plug-ins or players on the "web view" of a browser. The obligation to constantly update browser plug-ins, and security issues that come with the installation of such plug-ins are not completely hassle free either. While using a video or audio content, the issue of codec needs to be addressed. Different audio and video formats were made by different companies in the timeline of history. Therefore, licensing issues need to be addressed. Some formats are free on user's side for content streaming but the patent holder companies charge server hosts for hosting or streaming such media files. There is no existence of one format that is supported by all browsers. Therefore the recommended codecs that are comparatively widely supported and are cost free to use, are MP4 or H.264, ogg/Theora and WebM. (Jennifer Kyrnin 2013: W3schools 1999-2013, Date of retrieval 26.4.2013.)

Video Formats and Browser Support

Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg:

Browser	MP4	WebM	Ogg
Internet Explorer 9+	YES	NO	NO
Chrome 6+	YES	YES	YES
Firefox 3.6+	NO	YES	YES
Safari 5+	YES	NO	NO
Opera 10.6+	NO	YES	YES

- MP4 = MPEG 4 files with H264 video codec and AAC audio codec
- WebM = WebM files with VP8 video codec and Vorbis audio codec
- Ogg = Ogg files with Theora video codec and Vorbis audio codec

FIGURE 4: Browser support for video formats (W3schools 1999-2013, Date of retrieval 26.4.2013)

9.0 HTML5 GEO-LOCATION

W3C recommended Geo-location API for HTML5 is a high level interface that is used to share a user or device location with websites, servers and other devices. It is being developed by *Geolocation Working Group*. The scripting language JavaScript is used to capture the latitude and longitude of the user, based on the device used. Latitude and longitude are geographical coordinates on the Earth's surface. Latitude is a north-south coordinate and longitude is an east-west coordinate. These captured latitude and longitude values are sent to backend web server(s), which are used to find the location of user's device. These geographical co-ordinates can be decoded to find the address, with services such as 'Geocoding' from the Google. These latitude and longitude values are also the core of 'run-location' based applications, such as showing the location on a map or navigation. Location information is commonly gained from a device using GPS (Global Positioning System). It can also be deduced from various network signals such as IP address, RFID (Radio Frequency Identification), Wi-Fi (technology that allows to exchange data wirelessly) and Bluetooth MAC address. Cellular networks such as GSM/CDMA cell IDs can also be used for the purpose, as well as manipulative user inputs. However, the accuracy cannot be guaranteed due to the possibility of signal modification or signal loss by various physical reasons such as refraction and interference. Normally it is difficult to get an accurate location identification when devices are performing geo-location activities indoors or on low data network coverage areas. (Wikipedia 2012h; W3C 2012e, Date of retrieval 24.5.2012.)

Geolocation - Candidate Recommendation

Method of informing a website of the user's geographical location

*Usage stats:	Global
Support:	80.46%
Partial support:	0.03%
Total:	80.49%

Show all versions	IE	iOS Safari	Android Browser	Blackberry Browser	Chrome for Android
			2.1		
		3.2	2.2		
		4.0-4.1	2.3		
	8.0	4.2-4.3	3.0		
	9.0	5.0-5.1	4.0		
Current	10.0	6.0	4.1	7.0	18.0
Near future	11.0			10.0	

FIGURE 5. Screenshot of support table for HTML5 Geo-location API on mobile OS (@Fyrd 2012d, Date of retrieval 11.11.2012)

9.1 Methods in Geo-location API

There are namely 3 different Methods in Geo-location in HTML5

1. getCurrentPosition()
2. watchPosition()
3. clearWatch()

A functionality such as getting the current location of the device is used in a service like navigation. In HTML5 with the ***getCurrentPostion()*** geo-location method, geographical co-ordinates bundled with the information of direction and ground speed can be obtained. The speed is calculated in terms of co-ordinates of change of device location from previous to present position, distance between them and time taken for the co-ordinates to change. The geo-location information is updated periodically with the ***watchPosition()*** method. A unique transaction ID obtained by using this method is used to cancel the repetitive location update call, or to completely stop the process. In order to stop this cycle, the user end device executing geo-location methods for the location updates, ***clearWatch()*** is used. This will not only halt the process in the server end, but also on the user's device. This is a significant factor for the limited resources based mobile devices, as such to save the memory consumption and excessive uses of CPU. Solely for the navigation services purpose, a relatively new property of "Global Navigator" is implemented via a Geo-location object. Binding-specific casting methods with [NAVIGATOR] (the navigator interface), which is always used with location objects, captures the instance of Navigator. (MSDN Microsoft 2012c; Refsnes Data 1999-2012c; W3C 2012e, Date of retrieval 21.8.2012.)

9.1.1 Parameters used in Geo-location API

The ***getCurrentPostion()*** and ***watchPostion()*** consist of the following parameters to carry out the geo-location services:

showLocation is used to specify the callback method that returns the location information. It is executed asynchronously with the object corresponding to the Position object, where location information is stored. (W3C 2012e, Date of retrieval 24.5.2012.)

ErrorHandler is used to specify the callback method, which in the occurrence of error during the asynchronous processing is cited. It is called with the *PositionError* object that stores the returned error information. (W3C 2012e, Date of retrieval 24.5.2012.)

Options is the optional parameter that is used to hold a set of options to retrieve location information. The accuracy of returned location info, timeout for receiving location info and use of cached location information, can be specified. (W3C 2012e, Date of retrieval 24.5.2012.)

TABLE 8. Arguments of "options" used for retrieving geographical location (Refsnes Data 1999-2012c, Date of retrieval 23.8.2012)

Property	**Type**	**Description**
Accuracy	Boolean	False by default. This specifies if weather widget is seeking most accurate location estimation.
Time out	Number	This frames the amount of time (milliseconds) web- application is willing to wait for position information.
Maximum Age	Number	Specifies the expiry time in milliseconds for cached geo-location information.

9.2 Error Handling

Geo-location is a complex functionality. It is often prone to errors when the wireless mobile devices are used indoors, with interference and refraction of the signals, or simply due to connection and wireless coverage issues. Such errors need to be caught and handled properly. Error handling callback methods with *getCurrentPosition()* and *watchPosition()* returns *PositionError* object as the representation of such errors.

TABLE 9. PositionError object properties (Tutorials Point 2012b, Date of retrieval 27.8.2012)

Property	Type	Description
code	Number	Has numeric code as error.
message	String	Has a human-readable string for the description of the error.

TABLE 10. Typical error codes returned in PositionError object (Refsnes Data 1999-2012c; Tutorials Point 2012b, Date of retrieval 28.8.2012.)

Code	Constant	Description
0	UNKNOWN_ERROR	When method fails to get location from device due to unknown reason(s)
1	PERMISSION_DENIED	When method fails to get location from device due to permission denial from the device
2	POSITION_UNAVAILABLE	When device could not be located
3	TIMEOUT	When method fails to get location from device within the specified maximum timeout interval

An example of a code snippet:

function errorHandler (err){ if (err.code ==1){ //access denied

} …}

(Refsnes Data 1999-2012c; Tutorials Point 2012b, Date of retrieval 28.8.2012.)

9.3 Security and Privacy in HTML5 Geo-location

Geo-location API is used to reveal the location of the hosting device and the user. Cautious actions are essential to protect the user's privacy. A mechanism for the conformation of use of the user's device, needs to be implemented to protect the user's privacy. Therefore, location API should be executed in a device only after the user's approval. While implementing the geo-location API, agent must get the

permission from the user via User interface. User interface consists of a host document and URL components. For every permission exceeding beyond the current browsing session, user interface for the permission is to be revoked again. Recipients are only required to request for the user location when needed and should be used only for the task it was granted for. Location information is to be dumped once the task is complete in order to avoid unauthorized uses. Re-transmission of location data should be carried out in an encrypted form. If the permission was granted from the user to retain the location information, the user should have a full control to update or delete that information anytime. It is the recipient's task to keep the user in full awareness to dismiss the permission in the cases of change in the content holder's URL and accidental permission granted by the user. (W3C 2012e, Date of retrieval 24.5.2012.)

9.4 Requirements for implementation of Geo-location API

Geo-location API is used for various use cases while implementing to build mobile applications. It is used for wide functionalities such as refining search result, point of interest and news based on the user's proximity of location. Geo-tagging is used to bundle location information with a content like pictures, videos, audios, blogs etc. It is also useful for map services for the user's awareness on position on earth's crust and for services like navigation.

Geo-location API needs to meet the following requirements to implement the service on any mobile application or map services:

- Location data is provided as a pair of latitude and longitude coordinates.
- Accuracy of the retrieved location data should be provided and application should be allowed to specify the desired accuracy level.
- "One-shot" position updates should be supported.
- Updates should be enabled for the application to receive updates, if the location of the device is changed, or for those fixed location information which might have disrupted during the previous call.
- Access and request for cached location values are to be enabled for specified age or time.
- API must remain nescient to the underlying sources of location information.

(W3C 2012e, Date of retrieval 24.5.2012)

9.5 Using HTML5 Geo-location API in an Application

Geo-location APIs are used to gather the location information of device in terms of latitude and longitude geographical co-ordinates. The sole reason for acquisition of these location information lies on implementing them to build various location based services and applications, such as web-mapping applications. The HTML5 Geo-location API can be paired with different map services with respective APIs provided. (Google Developer, 2012b; Date of retrieval 25.5.2012)

In order to check the support for the Geo-location API by the application itself, the following PhoneGap supported code snippet is executed to test the support

```
if (navigator.geolocation) {
                    alert("Supported.");
          } else {
          alert("not supported");
}
</script>
```

The support for geo-location can also be tested with a feature sniffing library such as modernizr. The advantage of using modernizr is the provision for fallback, if the browser or the version of browser does not support the required feature. The following is the code snippet is executed for the test:

```
 function get_location() {
                      if (Modernizr.geolocation) {
                        navigator.geolocation.getCurrentPosition(show_map);
  } else {
                    // no native support; maybe try a fallback?
          }
}
```

On the Geolocation API support approved web browser(s), device's current location information is read as,

```
navigator.geolocation.getCurrentPosition(function(position) {
          var latitudeVariable = position.coords.latitude;
          var longitudeVariable = position.coords.longitude;  }
```

The acquired current position of the device as Latitude and Longitude is stored in designated variables. These values can be used with desired Map service(s). (Refsnes Data 1999-2012c, Date of retrieval 23.8.2012.)

9.6 Using mapping services with HTML5

On date, in order to use embed Google Maps in the application, Geolocation API is used with **Google Maps Java-script API V3**. In order to render Google map on the application, a element is used, which specifies the container to hold map and option - set. JavaScript code is used to add the Google Maps library as:

```
<script  src= http://maps.googleapis.com/maps/api/js?sensor=false
type="text/javascript"></script>
```

The sensor parameter needs to be defined. If GPS is required, the sensor information to the Google Maps on location, then the sensor value is set to 'true'. In any alternate case it is set to default 'false', as in above code snippet.

Retrieved location information (latitude and longitude) is used in Google maps as;

```
var coords = new google.maps.LatLng(latitude, longitude);
```

The rendered map can be customized as per the application need. In order to add map options *var mapOptions = {} is used.* The Zoom level of the rendered map is defined for the default case, simply as *zoom: 15.* These values can be altered according to the application need. In order to set the center of the displayed map, *center: coords* is used. Location's latitude and longitude co-ordinates are supplied in the *coords.* Similarly "mapTypeControl" and "navigationControlOptions", are used to allow the map control and navigation control respectively. (Google Developer, 2012b; Date of retrieval 25.5.2012.)

As per the native need of an application, 'Google Maps' provide various maps types. Four basic type of maps, and their allocation mechanism are:

"MapTypeId.ROADMAP displays the default road map view
MapTypeId.SATELLITE displays Google Earth satellite images
MapTypeId.HYBRID displays a mixture of normal and satellite views
MapTypeId.TERRAIN displays a physical map based on terrain information."

(Google Developer, 2012b; Date of retrieval 25.5.2012.)

In order to use the direction service from Google, a route function of *DirectionsService* is called. If the permission is granted, response can be used as desired, as represented below:

```
directionsService.route(request, function(response, status) {
   if (status == google.maps.DirectionsStatus.OK) {
      distance = response.routes[0].legs[0].distance.value;
   } else { // Exceptions are handled here     } });
```

For the other web-mapping services like Bing Map / Yahoo GeoPlanet and Nokia OVI map, **Bing Maps AJAX control API** and **OVI Map API** are respectively available. Both of these map services are enriched with the features similar to Google Maps such as initialization, navigation, markers, zoom bar, zoom scale, map controls and default allocation of map center. However, they lack in providing a variety of map view type. OVI map is regarded as the best services available for the navigation purpose. Navteq map data is the part of Nokia OVI map itself, which is also used by Yahoo, Bing and Garmin Navigation. (Google Developer 2012c; Nokia 2013, Date of retrieval 24.4.2013.)

9.7 Geo-location API support

Geo-location API is fully supported by all native web browsers of major mobile operating systems. Hence it meets the requirement for a cross-platform compatibility.

Geolocation - Candidate Recommendation

Method of informing a website of the user's geographical location

*Usage stats:	Global
Support:	83.39%
Partial support:	0.03%
Total:	83.42%

Show all versions	IE	iOS Safari	Android Browser	Blackberry Browser	Chrome for Android
	8.0		2.3		
	9.0		4.0		
Current	10.0	6.0	4.2	10.0	25.0

FIGURE 6. Screenshot of support for HTML5 Geo-location API across mobile operating systems (@Fyrd 2012e, Date of retrieval 24.4.2012)

10 HYBRID TECHNOLOGIES FOR CROSS PLATFORM DEVELOPMENT

10.1 PhoneGap

Initially owned by Nitobi and later purchased by Adobe, PhoneGap is an open-source mobile development framework which is used to make hybrid application for different operating systems using web technologies. PhoneGap's core software is Apache Cordova. Mobile devices come with a web browsers that support HTML5 (HTML, JavaScript and CSS). This supporting layer of web browsers is known as the abstraction layer or "Web View". PhoneGap is a bridge that enables JavaScript APIs through the abstraction layer of mobile operating systems, to access native hardware capabilities like GPS, accelerometer, camera, geo-location, compass etc., to be implemented in an application. These applications developed using web technologies behave as a native application on a device, and can be purchased from the application store of the respective operating systems. PhoneGap uses the native component of every operating system "Web View", to render these applications, and hence is a hybrid application. As such, in order to develop an application for any operating system it is no longer necessary to write the code on native programming language(s). (PhoneGap, Date of retrieval 18.5.2013.)

A different set of new JavaScript APIs, are not needed to be written, to access the hardware functionalities across different operating systems by the developer. PhoneGap has built-in functionality to access those capabilities. PhoneGap implements the API recommended by W3C, and fills the gap for needed APIs by itself, if absent. PhoneGap also provides plugins in special cases such as to support and access hardware features, which might not be available on other operating systems. (PhoneGap, Date of retrieval 18.5.2013.)

Using PhoneGap for application development does not expel the native user interface. PhoneGap is not a UI framework or MVC framework. It is bridge to use web technologies and present them as a native application on the devices. PhoneGap uses UI frameworks such as Sencha Touch, jQuery Mobile or Dojo for the look and feel of the application. (PhoneGap, Date of retrieval 18.5.2013.)

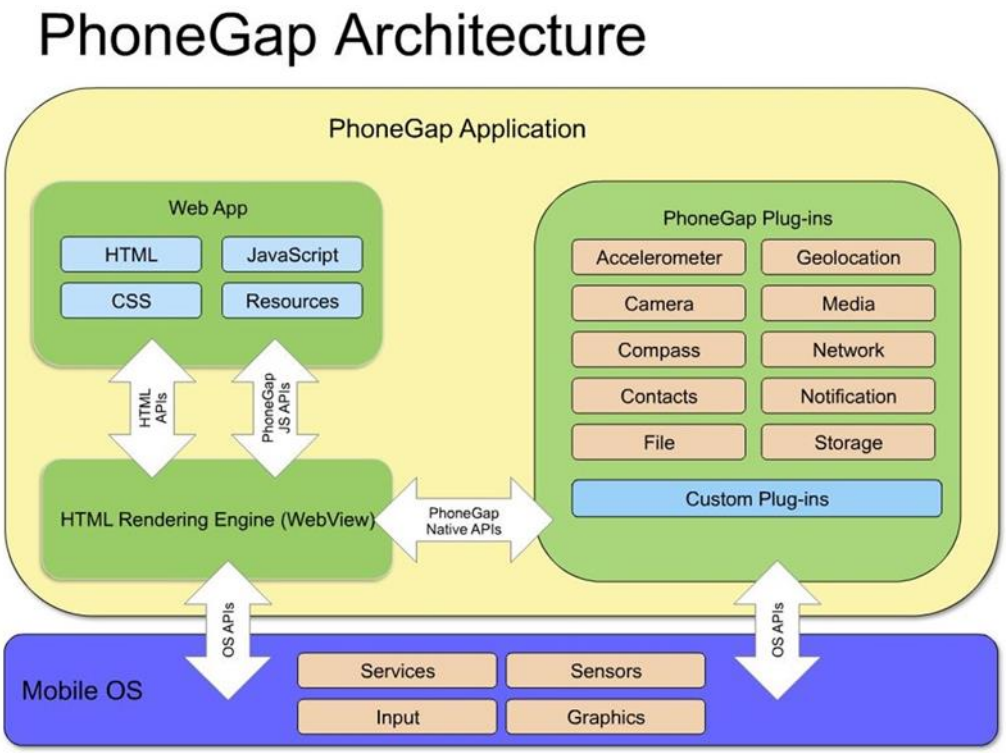

Figure 7. Screenshot of PhoneGap Architecture (IBM 2011, date of retrieval 18.5.2013)

In the native application development environment in order to compile, the application developer needs to install the SDK for each platform and on the local machine. Furthermore, for the compilation process, only specific IDE can be used for any particular native application development. This means, if the developer needs to develop an application for four or five different operating systems, four or five different SDK and IDE need to be installed. This is a huge burden for the local machine, which is used for the developmental process. PhoneGap solves this issue by compiling the codes with its online engine and generates binary files according to operating systems. These binary files are used for the installation of the application. Installation files are retrieved back by the developer, by manually downloading them from PhoneGap-build or by scanning generated QR code. (PhoneGap, Date of retrieval 18.5.2013.)

10.2 jQuery Mobile

jQuery mobile is a MIT licensed mobile UI application framework developed by jQuery team. It covers both the spectrum of native look and feel of different operating systems and has an additional custom interface. It is built on jQuery and jQuery UI foundation and is optimized for touch events in mobile devices and tablet computers. It has a wide compatibility across all major mobile operating systems including Android, iOS, BlackBerry, Windows Phone and beyond. jQuery was selected as a UI framework for the “demo-application” due to its wide compatibility and ability to work on top of PhoneGap. (Wikipedia 2012i; The jQuery foundation 2013, Date of retrieval 18.5.2013.)

11. PROOF OF CONCEPT

Proof of concept is an implementation of the research and finding to build a demonstrative cross platform application, which uses the same HTML5 code. The application is built upon the potentiality and compatibility findings of HTML5 canvas element and HTML Geo-location API for different mobile operating systems. The PhoneGap mobile development framework and the JQuery user interface framework, were used for the application development.1.1 Application development environment setup

A phoneGap application can be developed by two different methods. One of them is by downloading the PhoneGap SDK and placing the PhoneGap JavaScript file in the same folder as the HTML (Index.html) and other JavaScript files. Then the individual environment SDK and IDE and the setup to compile HTML5 codes need to be installed for one particular operating system at a time. This is a tedious and resource demanding approach. The second process of making applications for different mobile operating systems is made by using PhoneGap's online service "PhoneGap build". The HTML, JavaScript and CSS code can be written in any IDE. Entire written codes (files) are kept in a folder, zipped and uploaded to a PhoneGap build server. The PhoneGap build server produces different binary files for linked or uploaded HTML5 codes according to the operating system, as PhoneGap has a built-in SDK for individual operating systems. These separate files are used for installation of the application across all mobile OSs. Developer certificate and signing keys are assigned to a PhoneGap build server, for the redistribution of application. (PhoneGap, Date of retrieval 18.5.2013.)

PhoneGap Build comes with features such as debugger, which can be used to modify the application online. The previously installed application on mobile devices will be automatically updated. The second feature is "hydration", which allows to keep log of the previously built applications. The third and last feature is the ability to share an application with other collaborators for a developing and testing purpose. A built application can be either be kept private or shared with a public as an open source code. (PhoneGap, Date of retrieval 18.5.2013.)

With the optimum motto of *write once use everywhere*, and one *source code, one setup* and *one compilation process,* the application is build using online "PhoneGap Build". The mechanism of

PhoneGap build server is presented in *FIGURE 8*.This addresses both issues of a cross-platform programming language and one IDE for a compilation process.

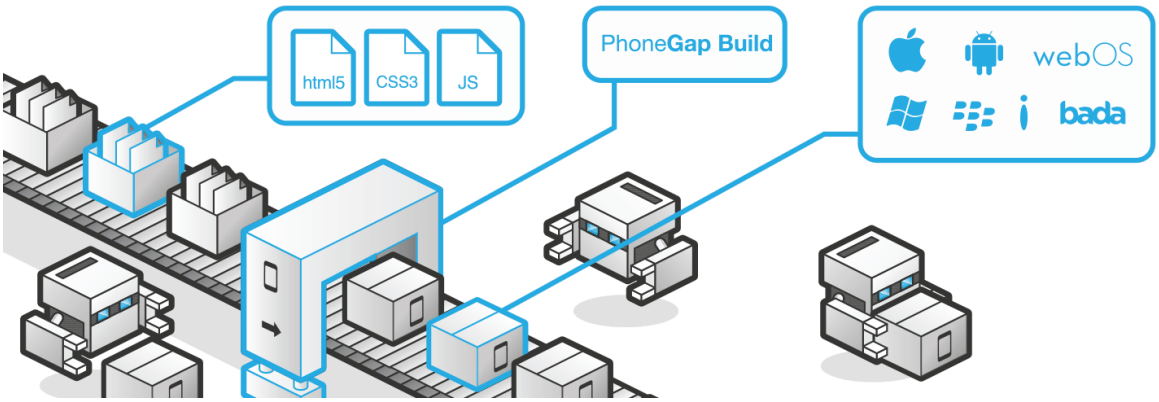

Figure 8. Screenshot of PhoneGap Build mechanism (Adobe 2013b, Date of retrieval 18.5.2013)

11.2 Application scope

The demonstrative application was built using Komodo edit IDE to write the HTML5 codes, and PhoneGap Build for the compilation. The application's scope is to utilize the canvas and geo-location properties that are compatible with the different mobile operating systems, as illustrated in the chapter 7.12 and chapter 9.7 of this thesis.

The application development and its requirements are listed below:

- The PhoneGap built application needs to behave as native application on devices, and the installation process same as of native application.

- Canvas drawing methods **Colors and styles**, **Composition**, **Simple & Complex shapes**, **Gradients** & **Patters**, **Canvas State**, **Text** and **Transformation** need to produce an animation rendering effect. Normally the animation in an HTML5 canvas is drawing the shapes on the canvas, erasing the entire canvas and redrawing in a different co-ordinate with in the canvas, within the time frame that is unnoticeable bye a human eye. Though this approach is efficient for

memory uses, this approach eliminates the possibility to test the **Transformation** functionality of the HTML5 canvas. Therefore, instead of using the *setTimeout()* method, the *setInterval* method is used for the demonstration purpose. Canvas is cleared with the *clearRect* method at intervals to reduce the memory uses by canvas.

- In order to test the lag and performance issues, an ”Add animation” button using jQuery Mobile is assigned. Its function is to multiply the animated shapes and gradients. The ”demo application” should be able to render these canvas functionalities. Performance issues were noted.

- The Geo-location part of the applicaiton renders the Google maps *(ROADMAP)* in the application with the ability to switch to Google ”*SATELLITE-MAP*” and Google ”Street-view”. PhoneGap geo-location API is used to receive the latitude and longitude co-ordinates from the device itself. The PhoneGap SDK was downloaded from the server, and an apache cordova file ” cordova-2.5.0.js” was included in the same folder as an ”index.html” file, to access the PhoneGap Geo-ocation API. The extraced latitude and longitude values are not that of ISP, but of the actual device itself.

- Google map is centered on the screen with the actual location of the device, and a marker is used to pin point the location. Google map is rendered using the vector graphics SVG. Google Street view, zoom function and map controls are enabled. The ”demo application” needs to get the geographical co-ordinates of the device accessing its hardware. Based on the extracted co-ordinates, the Google Maps functionalites need to be working. The permission from the user to access of device’s geo-location should be generated on UI.

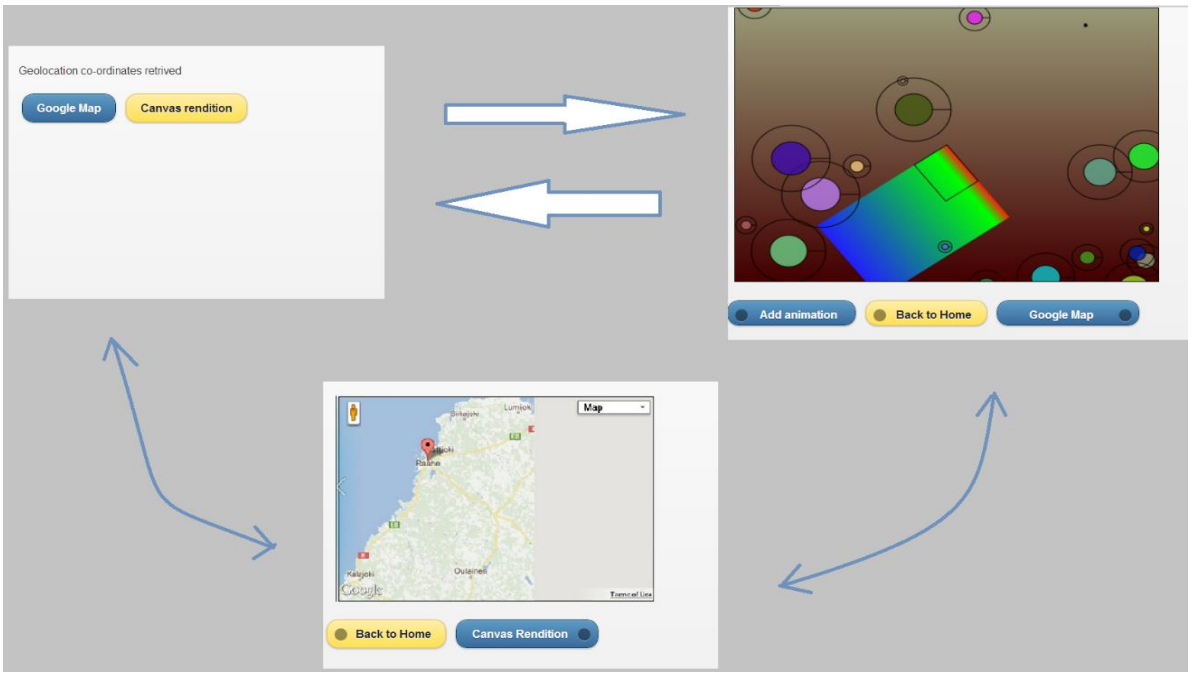

FIGURE 9. User Interface of the application utilizing canvas and geolocating

11.3 Application testing and findings

The source code of the application was zipped and uploaded to the PhoneGap Builder, which was then received as individual binary files for each operating systems. The screenshot of downloadable PhoneGap Build applications is presented in *FIGURE 10*. The application installer binary files were then scanned with a QR code reader or downloaded to be tested on separate platforms.

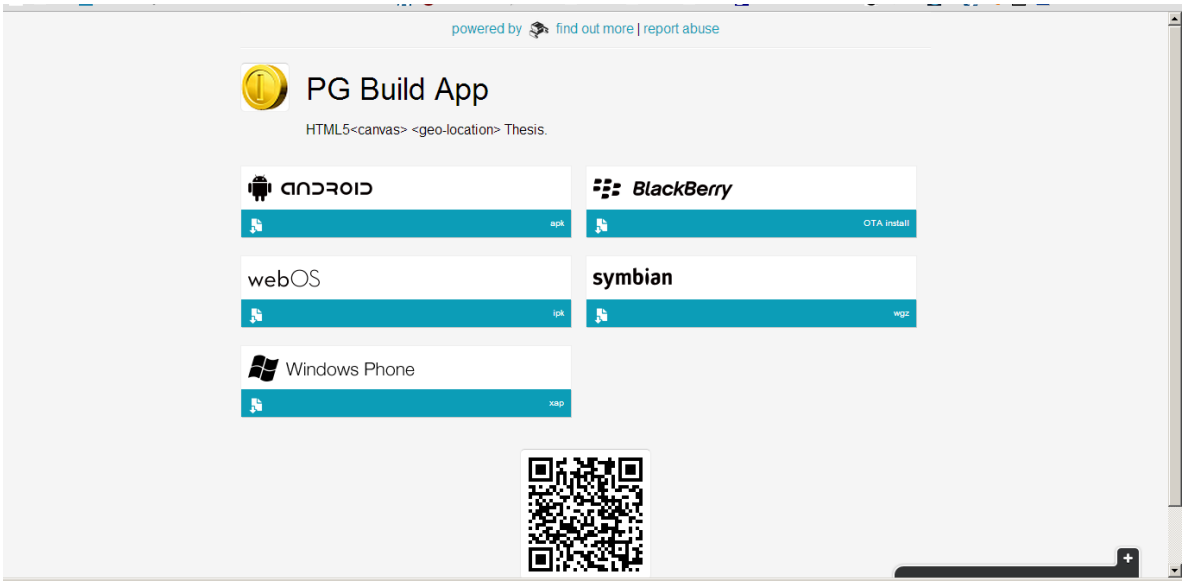

FIGURE 10. Screenshot of downloadable PhoneGap build application with QR code. (PhoneGap 2013b, Date of retrieval 21.5.2013)

The application installer files generated from PhoneGap Build are for iOS, Android, Windows Phone, BlackBerry, Symbian and WebOS. However, based on the availability of test device(s) and need of *developer's license*, the following operating systems were target for the testing accordingly to availability:

11.3.1 Android OS

A compiled *.apk* file was downloaded from PhoneGap Build on two separate devices, using the QR code scanning application. The devices Samsung Galaxy S (gt-I9000) with 1 GHz Cortex-A8 CPU running Android 2.3.3 and Samsung Galaxy S3 (I9300) with Quad-core 1.4 GHz Cortex-A9 CPU running android 4.2.0, were used for the testing purpose. The application running on GT-I9000 is presented in *Figure 11.* During the installation of the application, the permission to use geo-location was asked by the application. The permission is auto generated by a PhoneGap plug-in without having to do any modification to the Android manifest file. The permission UI generated by PhoneGap before installing the application is presented in *FIGURE 12.* Animation and Google Map view both were instantaneously loaded on I9300. On gt-I9000, the animation seemed cracked and the rendering of the animation was not smooth. This is due to the difference in hardware capabilities. Geo-location feature on both devices were accurate and

were instantly loaded. The application was successful to perform every functionality it was intended to build for.

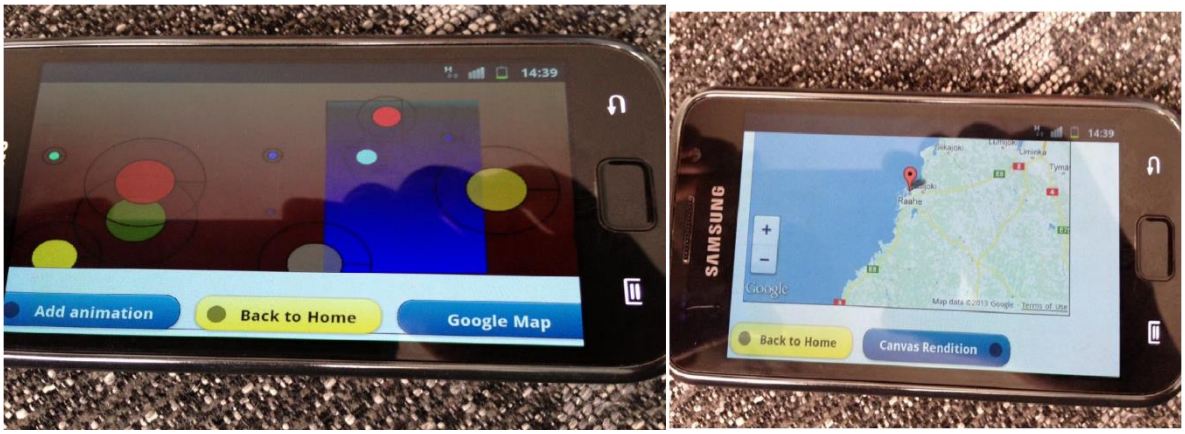

FIGURE 11. Snapshot of an application running on Samsung gt-i9000

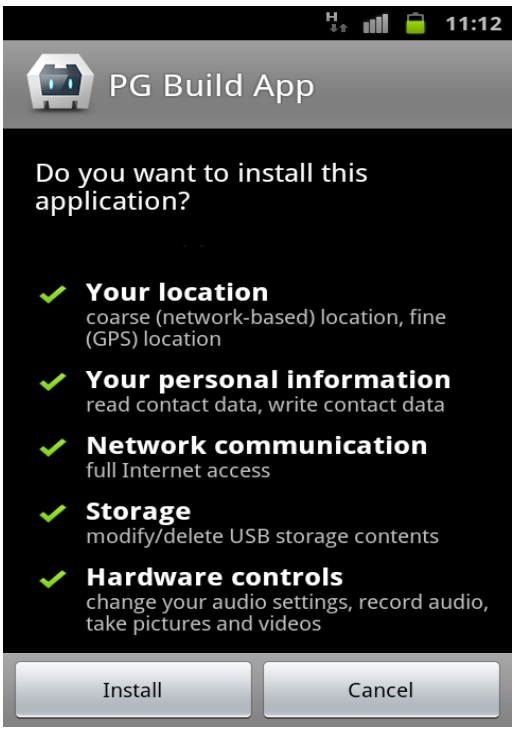

FIGURE 12. Screenshot of an application asking for permission before installation on mobile device Samsung gt-I9000

11.3.2 Windows Phone OS

The application was tested in a Windows phone 7.1 emulator due to the unavailability of the hardware. The screenshot of the Windows phone emulator is presented in *FIGURE 13.* An XAP file was downloaded from the PhoneGap Build and deployed to the emulator. The Canvas rendition worked as it was intended to. Even though the emulator could be hard coded to test the user assigned latitude and longitude value, the geo-location feature could not be tested to its full extend due to PhoneGap's ability to extract co-ordinates only from the device itself.

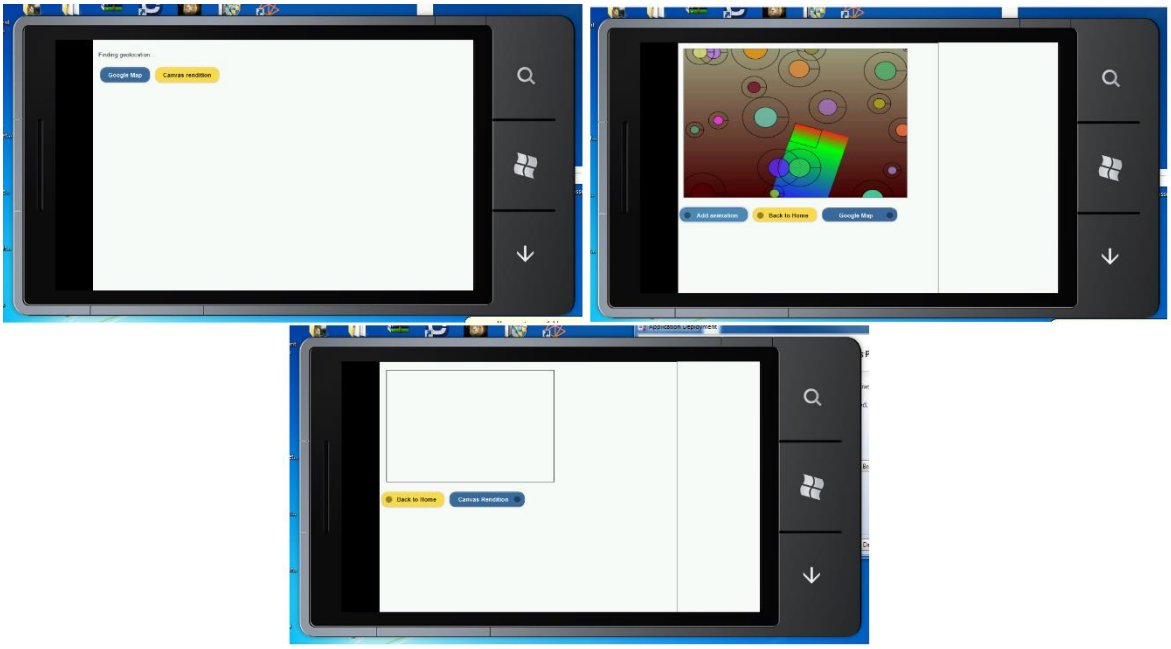

FIGURE 13. Screenshot of an application on Windows Phone 7 emulator

11.4 Findings

The testing of the application verified the HTML5's cross platform nature and PhoneGap's ability to build hybrid application using its online compiler. The testing indicated not only that the HTML5 features needs to be compatible with OS but also that the device's hardware should be capable enough. Since

PhoneGap works in the mechanism of triggering the browser's "Web View" within the application, for the same type of application, the native application always uses less hardware resources. Therefore, even though HTML5 has an ability to work cross platform, it is recommended to use the native approach to build an application in cases where targeted devices are less hardware (CPU/GPU) capable.

Minor tweaks in resolution settings are always necessary while building a hybrid application with web technologies as not all devices have the same resolution screen. A unique resolution needs to be declared for each target device. Even though the native IDE and SDK are not required for the PhoneGap built hybrid application, the emulator set up in such environment is required for testing. Online emulators such as "Ripple Extension" for Google Chrome browser are available but are not always working, being a relatively new technology.

Overall, the HTML5 technology is the best to build an application when an extensive use of hardware resources are not needed. The developer's ID and API keys are always needed to publish the application in app-store for re-distribution.

FURTHER DEVELOPMENT

This thesis was dedicated to the research to find a new mechanism to develop cross platform mobile applications. The application was built with the pure HTML, JavaScript and CSS to test the findings and mechanism. The third party animation frameworks for canvas rendition were not used to note pure performance issues. Based on the findings of this thesis, game or animation applications using different frameworks can be made. Applications like navigation and geo-tagging software, bundled with maps for multimedia can also be developed based on the findings of this research.

CONCLUSION

This bachelor's thesis was very challenging due to a limited availability of the resources and changing drafts of HTML5, for HTML5 still being at a drafting stage. An immense research was carried out during the process and the findings are very satisfying with interesting facts.

The research on HTML5 geo-location showed that it is already in a full matured state. The Possibility to use Geo-location services such as navigation, speed and map services with HTML5 is in full effect. API was able to access geo-location co-ordinates of the device located outside of "Web View" abstraction layer. It was also able to auto generate the confirmation for use of geo-location services from the user, without having to use additional native permission codes in the application.

The research of HTML5 canvas showed that not all features are available to use in mobile devices yet. Major functionalities like Full Screen mode and WebGL for 3D animation are not supported. A 2D animation is fully supported with minor exceptions in few browsers. The application development showed that even though HTML5 is lighter and much faster than Adobe Flash, due to the PhoneGap mechanism, it requires more CPU and memory to execute the same animation than in the browser or native application itself.

HTML5 technology with PhoneGap is not a silver bullet in itself. Though it has a wide range of support to be cross platform technology for mobile application development and it indicates a promising future, it is not matured yet. Native mobile browsers need to advance more with updates to support a wider range of HTML5 elements. The first final draft of HTML5 is being expected by 2014, but some of the major features are still missing. Tweaks in codes are always needed according to targeted variety of devices. Overall, HTML5 is a preferable solution for a cross-platform mobile development, until the required feature for application is fully supported and the use of hardware resources are not excessively required.

This thesis helped me in gaining knowledge of different programming languages, mobile operating systems, mobile browsers, business modules of mobile industry, a PhoneGap architecture for hybrid application, mobile web technologies such as jQuery mobile, HTML5's canvas and geo-location elements. I am looking forward to implement the knowledge gained during this research in my professional career and further studies.

LIST OF REFERENCES

@Fyrd 2012a. Can I use. Date of retrieval 20.4.2013
http://caniuse.com/#agents=desktop&cats=HTML5

@Fyrd 2012b. Can I use. Date of retrieval 20.4.2013
http://caniuse.com/#cats=SVG

@Fyrd 2012c. Can I use. Date of retrieval 11.11.2012
http://caniuse.com/#cats=CSS,HTML5,JS_API,PNG,DOM,CSS3,CSS2,Canvas,[object Window],Summary

@Fyrd 2012d. Can I use. Date of retrieval 11.11.2012
http://caniuse.com/#feat=geolocation

@Fyrd 2012e. Can I use. Date of retrieval 24.4.2012
http://caniuse.com/#feat=geolocation

Aaron Brandy 2012. Facebook developers. Date of retrieval 18.5.2013
https://developers.facebook.com/blog/post/2012/05/09/introducing-the-app-center/

Addison Wesley Longman 1998. A History of HTML. Date of retrieval 15.11.2012
http://www.w3.org/People/Raggett/book4/ch02.html

Adobe Systems Inc. 2012a. PhoneGap. Date of retrieval 10.11.2012
http://phonegap.com/about

Adobe Systems Inc. 2013b. PhoneGap Build. Date is retrieval 18.5.2013
http://build.phonegap.com/images/marketing/build-diagram.png

Apple Inc. 2010. Safari HTML5 Canvas Guide. Date of retrieval 2.7.2012
http://developer.apple.com/library/safari/#documentation/AudioVideo/Conceptual/HTML-canvas-guide/AddingText/AddingText.html

Apple Insider 2012, Date of retrieval 25.10.2012
http://appleinsider.com/articles/12/09/17/apple_on_pace_to_sell_1_billion_ios_devices_by_2015

CBS Interactive 2012. HTM5-enabled phones to hit 1 billion in sales in 2013. Date of retrieval 10.10.2012
http://news.cnet.com/8301-1023_3-57339156-93/html5-enabled-phones-to-hit-1-billion-in-sales-in-2013/

Ethertrank 2011. Canvas tutorial. Date of retrieval 12.2.2012
https://developer.mozilla.org/en/docs/Canvas_tutorial

Forbes.com LLC 2013. Roundup of SmartPhone and Tablet Forecasts and Market Estimates. Date of Retrieval 20.4.2013
http://www.forbes.com/sites/louiscolumbus/2013/01/17/2013-roundup-of-mobility-forecasts-and-market-estimates/

Google Developer I/O 2011a, Date of retrieval 20.12.2011
http://youtu.be/4f2Zky_YyyQ

Google Developer, 2012b. HTML5 versus Android: Apps or Web for Mobile Development. Date of retrieval 25.5.2012
https://developers.google.com/maps/documentation/javascript/maptypes?hl=fr

Google Developer, 2012c. Geolocation. Date of retrieval 26.5.2012
https://developers.google.com/maps/articles/geolocation

Html5marketplace 2010-2011. HTML5 Canvas. Date of retrieval 2.7.2012
http://www.html5marketplace.com/html5/canvas.shtml

HTML5 Rocks 2012, Date of retrieval 10.11.2012
http://www.html5rocks.com/en/features/storage

IBM 2011, PhoneGap Day – IBM, PhoneGap and the Enterprise. Date of retrieval 20.5.2012
http://www.slideshare.net/drbac/phonegap-day-ibm-phonegap-and-the-enterprise

IDC Worldwide Mobile Phone Tracker 2013. Worldwide Mobile Phone Growth Expected to Drop to 1.4% in 2012 Despite Continued Growth Of Smartphones, According to IDC. Date of Retrieval 20.4.2013
http://www.idc.com/getdoc.jsp?containerId=prUS23818212#.UPeTeSdi58F

John Bristowe 2011. Intro to HTML5 Canvas. Date of retrieval 12.2.2012
http://vimeo.com/26165167

Kravchick, O. 2011. Native vs. HTML5 Application Development. Date of retrieval 31.1.2012
http://myok12.wordpress.com/2011/06/29/native-vs-html5-application-development/

AOL Tech 2013. Even As HTML5 Gaming Skepticism Remains, Ludei Pushes An App To Seven Stores Simultaneously. Date of retrieval 18.5.2013
http://techcrunch.com/2013/02/22/ludei/

MacWorld 2007. Apple unveils iPhone. Date of Retrieval 14.11.2012
http://www.macworld.com/article/1054769/iphone.html

Marko Gargenta 2010. Android for Java developers. Date of retrieval 20.10.2012
http://www.slideshare.net/marakana99/marakana-androidjava-developers

Mihai Sucan 2010. SVG or Canvas. Date of retrieval 6.2.2012
http://dev.opera.com/articles/view/svg-or-canvas-choosing-between-the-two/

Mozilla Developer Network 2005-2012. Introduction to HTML. Date of Retrieval 10.11.2012
https://developer.mozilla.org/en-US/docs/HTML/Introduction

MSDN Microsoft 2011a. Introduction to HTML. Date of retrieval 15.5.2012
http://msdn.microsoft.com/en-us/library/ie/gg193983(v=vs.85).aspx

MSDN Microsoft 2012b, Date of retrieval 05.11.2012*
http://msdn.microsoft.com/en-us/library/ff637514(v=vs.92).aspx)

MSDN Microsoft 2012c. Introduction to the Geolocation API. Date of retrieval 20.8.2012
http://msdn.microsoft.com/en-us/library/ie/gg589513(v=vs.85).aspx

Nokia 2013. HERE Maps API. Date of retrieval 24.4.2013
http://www.developer.nokia.com/Community/Wiki/Nokia_Maps_API_-_Locate_user_with_HTML5_Geolocation_API

Open Handset Alliance 2012. Android. Date of retrieval 20.2.2012
http://www.openhandsetalliance.com/android_overview.html

Pearson Education, Que Publishing 2012. Getting Started with Windows Phone 7. Date of retrieval 18.5.2013
http://www.quepublishing.com/articles/article.aspx?p=1824248&seqNum=

PhoneGap 2012a. PhoneGap: Building & Testing Mobile Apps with Web Standards. Date of retrieval 20.4.2012
https://www.youtube.com/watch?v=u6YBDqIFlvg

PhoneGap 2013b. PG Build App. Date of retrieval 21.5.2013
https://build.phonegap.com/apps/400573/collaborators

Refsnes Data 1999-2012a. HTML5 Inline SVG. Date of retrieval 2.5.2012
http://www.w3schools.com/html/html5_svg.asp

Refsnes Data 1999-2012b. HTML5 Canvas. Date of retrieval 23.5.2012.
http://www.w3schools.com/html/html5_canvas.asp

Refsnes Data 1999-2012c. HTML5 Geolocation. Date of retrieval 23.8.2012
http://www.w3schools.com/html/html5_geolocation.asp

Research in Motion Limited 2012. BlackBerry 10 Launch Event. Date of Retrieval 14.11.2012.
http://press.rim.com/newsroom/press/2012/blackberry-10-launch-event-to-be-held-on-january-30th-2013.html

Richard Lawer 2011. Adobe reportedly will announce the end of Flash Player for mobile devices .Date of retrieval 12.2.2012
http://www.engadget.com/2011/11/09/adobe-reportedly-will-announce-the-end-of-flash-player-for-mobil/

Rob Hawkes 2011a. The future of graphics on the Web. Date of retrieval 12.2.2012
http://vimeo.com/19965866

Rob Hawkes 2011b. Gaming with HTML5 and Javascript. Date of retrieval 16.2.2012
http://www.slideshare.net/robhawkes/html5-javascript-games

Sergey Mavrody 2011. HTML5 Taxonomy and Status. Date of retrieval 20.5.2012
http://upload.wikimedia.org/wikipedia/commons/f/f7/HTML5-APIs-and-related-technologies-by-Sergey-Mavrody.png

Steve Jobs 2010. Thoughts on Flash. Date of retrieval 12.2.2012
http://www.apple.com/hotnews/thoughts-on-flash/

The jQuery Foundation. 2012. jQuery Mobile. Date of retrieval 18.5.2013

http://www.jquerymobile.com

Third Door Media 2012. Apple Earnings. Date of retrieval 25.10.2012
http://marketingland.com/apple-earnings-36b-revenue-44m-iOS-devices-sold-25003

Tutorials point 2012a. HTML5 Canvas – Composition. Date of retrieval 4.7.2012
http://www.tutorialspoint.com/html5/canvas_composition.htm

Tutorials Point 2012b. HTML5 – Geolocation. Date of retrieval 28.8.2012
http://www.tutorialspoint.com/html5/html5_geolocation.htm

TNW 2012. Angry Birds Space and Cut the Rope have both landed in the Windows Phone Store. Date of retrieval 05.11.2012
http://thenextweb.com/microsoft/2012/11/07/angry-birds-and-cut-the-rope-have-both-landed-in-the-windows-phone-store/

Vox Media 2012. Google I/O highlights: Nexus 7 tablet, Android 4.1 Jelly Bean, and more. Date of retrieval 11.11.2012
http://www.theverge.com/2012/6/27/3121011/google-i-o-highlights-news-2012-nexus-7-nexus-q

WhatWG 2012a. HTML Living Standard. Date of retrieval 12.2.2012
http://www.whatwg.org/specs/web-apps/current-work/#introduction

WhatWG 2012b. HTML Living Standard. Date of retrieval 2.7.2012
http://www.whatwg.org/specs/web-apps/current-work/multipage/

Wikipedia 2012a. HTML5. Date of retrieval 15.11.2012
http://en.wikipedia.org/wiki/Html5

Wikipedia 2012b. HTML5 in mobile devices. Date of Retrieval 10.11.2012
http://en.wikipedia.org/wiki/HTML5_in_mobile_devices

Wikipedia 2012c. iOS. Date of Retrieval 5.11.2012
http://en.wikipedia.org/wiki/IOS#cite_note-WWDC_2012_Keynote-3

Wikipedia 2012d. Android (operating system). Date of retrieval 20.10.2012
http://en.wikipedia.org/wiki/Android_(operating_system)

Wikipedia 2012e. Windows Phone. Date of retrieval 11.11.2012
http://en.wikipedia.org/wiki/Windows_Phone

Wikipedia 2012f. BlackBerry. Date of retrieval 05.11.2012
http://en.wikipedia.org/wiki/BlackBerry#cite_note-2

Wikipedia 2012g. Scalable Vector Graphics. Date of retrieval 2.5.2012
http://en.wikipedia.org/wiki/Scalable_Vector_Graphics

Wikipedia 2012h. W3C Geolocation API. Date of retrieval 24.5.2012.
http://en.wikipedia.org/wiki/W3C_Geolocation_API

Wikipedia 2012i. jQuery mobile. Date of retrieval 18.5.2013
http://en.wikipedia.org/wiki/JQuery_Mobile

W3C 2010a. SCALABLE VECTOR GRAPHICS (SVG). Date of Retrieval 2.5.2012
http://www.w3.org/Graphics/SVG/

W3C 2011b. Scalable Vector Graphics (SVG). Date of Retrieval 2.5.2012
http://www.w3.org/TR/SVG/

W3C 2010c. Scalable Vector Graphics (SVG) Tiny. Date of Retrieval 3.5.2012
http://www.w3.org/TR/SVGTiny12/

W3C 2012d. HTML Canvas 2D context. Date of retrieval 2.7.2012
http://www.w3.org/html/wg/drafts/2dcontext/html5_canvas/

W3C 2012e. Geolocation API Specification. Date of retrieval 24.5.2012
http://dev.w3.org/geo/api/spec-source.html

LIST OF FIGURES

Number	Figure Description
1	Screenshot of a mobile operating systems' market share and future prediction.
2	Screenshot of a taxonomy and status of HTML5 elements.
3	The best use case scenario for SVG and Canvas.
4	Browser support for video formats.
5	Screenshot of a support table for HTML5 Geo-location API on mobile OS.
6	Screenshot of a support for HTML5 Geo-location API across mobile operating systems.
7	Screenshot of a PhoneGap Architecture.
8	Screenshot of a PhoneGap Build mechanism.
9	User Interface of the application utilizing canvas and geo-location.
10	Screenshot of a downloadable PhoneGap build application with QR code.
11	Snapshot of an application running on Samsung gt-i9000.
12	Screenshot of an application asking for permission before installation on mobile device Samsung gt-I9000.
13	Screenshot of an application on Windows Phone 7 emulator.