Bachelor Thesis

Spring 2013

School of Health and Society
Department Design and Computer Science

HTML5

A Serious Contender to Native App Development or Not?

Writer

Mark Powell
Yuesong Li

Instructor

Sandra Nilsson

Examiner

Christian Andersson

School of Health and Society

Department Design and Computer Science
Kristianstad University
SE-291 88 Kristianstad
Sweden

Author, Program and Year:
Mark Powell, Computer Software Development, 2010
Yuesong Li, Computer Software Development, 2010

Instructor:
Sandra Nilsson, Lecturer, M.Sc., HKr

Examination:

This graduation work of 15 higher education's credits is part of the requirements for a *Degree of Bachelor in Computer Science* (as specified in the English translation).

Title:

HTML5, A Serious Contender to Native App Development or Not?

Abstract:

Many desktop applications have moved away from heavy client-side programs to lighter web-based solutions served from the cloud. Mobile applications have also been traditionally client-based in the form of native applications. With the use of HTML5, however, there is a growing shift towards web apps as opposed to native. HTML5 technologies are enabling mobile apps to run in the browser with some native app functionality. HTML5 has received a great deal of attention since its release in 2009, there are numerous articles and discussions on the Internet. Some developers are very negative and some are very positive towards the idea that it is a serious contender in mobile application development. This report presents research carried out to get the views of developers as to their current use of HTML5 and how much they expect to be developing mobile apps using HTML5 in the future. Judging by the level of response gained and comments given, this is a very hot topic right now. The research shows that HTML5 is a viable alternative to native app development but it has some limitations and has some way to go before it could begin to replace native app development altogether.

Language:

English

Approved by:

Christian Andersson Date
Examiner

Table of Contents

1 Introduction

1.1 Background

In 2012, Facebook decided to move away from HTML5. On the other hand in the same year the Financial Times (FT) moved to HTML5 from previously developing very successfully natively with iOS. The FT used HTML5 to create a solution that was cross-platform compatible. The change in strategy by these Internet giants poses many questions, of which first and foremost, “is HTML5 ready” and “is it capable enough”. As intriguing as these questions are, it was believed that app development businesses, schools and students as well the community would benefit from the research to finding these answers.

1.2 Aim and Purpose

It was the intention of this research to look at mobile development past and present and evaluate what significance HTML5 and cross-mobile publishing will have on the development of mobile apps in the future. The aim was to form an informed and knowledgeable opinion regarding the future of mobile development by researching the benefits and pitfalls of the two, and to test the hypothesis. It is hypothesized that comparing HTML5 and native development of mobile applications will show that HTML5 is already a viable alternative to native for some types of mobile applications.

1.3 Method and Resources

A questionnaire was created and implemented as a web form to gather the views of developers. An application written in native Android code and then the same application using HTML5 were built.

1.4 Presentation of the Company

The name of the company that the Android application was created for is called AppInMed and is based in Lund. AppInMed is focused on creating applications for smart phones and tablets within the area of medicine and health. They help publishers, businesses, hospitals, organizations and associations to make their communications more accessible and mobile. The company consists of both clinically experienced and research active doctors, as well as young engineers and programmers.

1.5 This Document

This thesis report serves a couple of functions. First, it introduces both HTML5 and native approaches of mobile application development. Secondly, by comparing the advantages and disadvantages and by examining the current situation of both approaches, a conclusion, mixing positive with negative points of view regarding HTML5, was drawn. Thus, it is meant to serve as a guideline for future use of HTML5 when developing mobile applications.

1.6 Acknowledgements

Both authors: This thesis would not have been possible without the help, support and patience of our instructor, Sandra Nilsson, not to mention her understanding and advice which has been invaluable on both an academic and a personal level, for which we are extremely grateful.

We are also very grateful for the advices and support of our examiner Christian Andersson.

We would like to thank all the persons that took part in the research and those that were kind enough to take time to share their valuable comments that made this research so much more meaningful.

Li: Besides the kind people above, I would also like to thank those who have been generous and kind enough to cook me food. It is not trivial to do the trivial. I am very grateful.

And my very dear friend Mark Powell, who has been nothing less than a fighter and a brilliant person to work with, I thank you too.

Powell: Due to some very difficult family health circumstances, it would never have been possible to write this thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

Above all, I would like to thank my wife Viveca for being the fighter that she is and helping as best she could and giving her personal support and great patience at all times. My own family and that of my wife who have given both my wife and I their unequivocal support throughout, as always, for which my mere expression of thanks does not suffice. My wife's parents and my good friend Per, who helped with the children and made sure that the children missed out as little as possible. Also a big thank you to all our other friends and family that have been there for us.

Last, but by no means least, I would like to thank Yuesong Li for partnering with me on this research paper and for always being a great friend.

2 Analysis

2.1 Technical Background

This chapter provides background information of HTML5, the programming languages that make it up and how they are used to create mobile applications that can run on multiple platforms. It also explains what Hybrid application development is, together with an explanation of the software applications and process used to create hybrid applications. It also looks at native application development and the different platforms and operating systems that are available.

2.1.1 HTML

HTML stands for Hypertext Markup Language and is the language of the World Wide Web (1). Hypertext refers to the ability to link to other pages and resources and by marking up the content with tags, content can be displayed aesthetically within the web browser. Originally, HTML pages were only accessed using computers with large monitors but now web pages are accessed using many different kinds of devices such as smartphones, tablet computers and even screen readers for people with disabilities. The evolution of HTML has been slow and many years passed between each version of HTML being released, shown in Figure 1. (2)

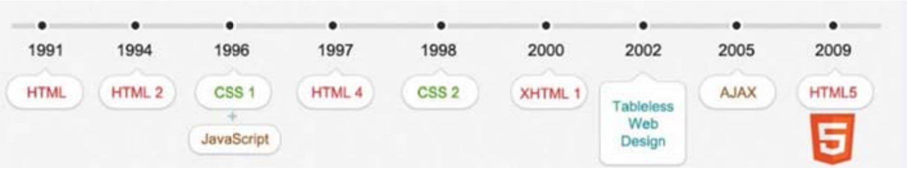


Figure 1. Versions of HTML

2.1.2 HTML5

The latest version is HTML5 and it builds on previous versions of HTML with some improvements. It is more than just HTML markup; it consists of a group of related technologies of which HTML is just one part. The other parts are JavaScript and CSS (Cascading Style Sheets). Other technologies that are included are a number of DOM (Document Object Model) interfaces that can typically be accessed within JavaScript such as geo-location, camera and video support, etc. Older versions of HTML relied heavily on third-party plugins such as Flash, Silverlight and Java to provide much of the multimedia integration (3). HTML5 adds many new features and limits the need for these processor-intensive plugins for much of the common functionality needed. It is now possible to embed audio and video, charts and animations, high quality drawings and other rich content and all this is now built straight into the browser (4). The addition of these new features not only makes it easier and faster to develop highly interactive applications, it also means that applications are less processor hungry and therefore consume less battery power which is very beneficial and desirable on devices such as smartphones (5).

Another new feature of HTML5 that didn't exist in previous versions of HTML is the introduction of semantic tags. Many developers were already using the existing <div> tags and calling them header and footer e.g. <div id="header">, so the World Wide Consortium (W3C) decided to add semantic tags to be used specifically for these elements. This makes a lot of sense because the page is now divided up into logical parts that help save developers time when trying to understand other developers' code. It also solves the problem with being able to see where a section ends. If the div tag is used, a comment would often need to be written to make it obvious which closing tag belongs to the header div. If this comment is omitted and the code is lengthy, it could be difficult to identify the correct closing tag quickly. By using the semantic tags, it can be opened and closed simply by using <header></header> (6). Also due to its logical markup it is much easier to be understood by crawlers such as Google bot and also screen readers (7). By making them more understandable to the Google bot, web pages could have advantages, for example giving more applicable search results to the user. It makes the web more accessible to users with disabilities enabling screen readers and other assistive technology to use the semantic tags to describe content and break the page up into important components. As regards mobile applications, the semantic tags may possibly be utilized to create programs that are more assistive such as recognizing voice commands for navigation elements. Even though the adoption of HTML5 has been slow for desktop browsers, smartphone and tablet support has been much stronger and nearly every smartphone and tablet device being sold already supports HTML5. (8) Figure 2 below shows the results compiled from automatically submitted tests. (9)

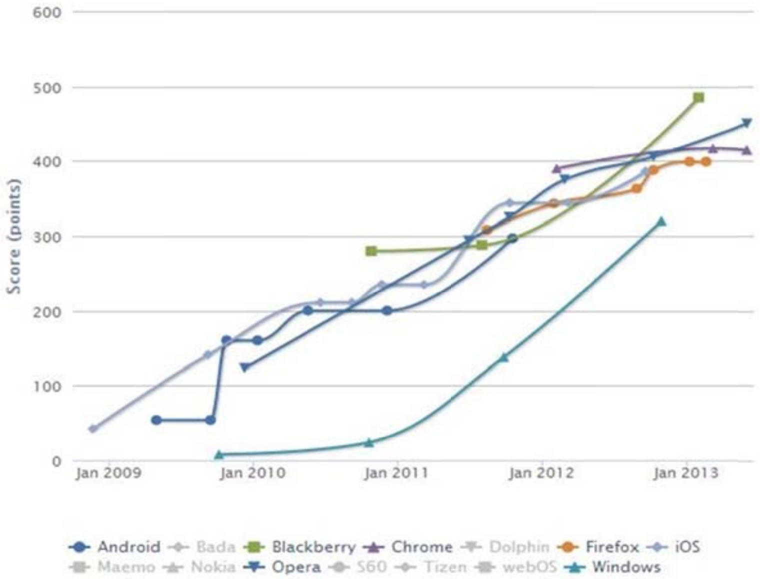


Figure 2. HTML5 scores on mobile platforms over the years

2.1.3 HTML5 - Open Web Platform

The World Wide Web (W3) has always been a publishing platform and with the introduction of HTML5 the W3 is becoming more and more important. W3 is even now used by technology that is being integrated into vehicles (10). All this enables an interface to the web almost everywhere and opens the door for even more consumer applications to be created.

But for this ever increasing range of devices to be able to use the richness of HTML5 successfully, there needs to be standards. *"The World Wide Consortium (W3C) is an international community that develops open standards to ensure the long-term growth of the web".* (11) The core aims of the W3C have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices. (12) W3C standards define an Open Web Platform which is the umbrella term that refers to the collection of specifications which together make up the rich multimedia web that is evolving and making possible the next generation of web applications. (13) The standardization process (14) for HTML5 began in 2004 with Mozilla Foundation and Opera Software submitting a position paper at a W3C workshop. Later work based on that position paper was moved to Web Hypertext Application Technology Working Group (WHATWG) and this HTML5 specification was adopted as the starting point in 2007. In 2008 WHATWG published the First Public Working Draft. In 2011 W3C extended the charter for the HTML Working Group with clear milestones for HTML5. In 2012 a degree of separation was decided upon where WHATWG would continue to work on HTML5 as a "Living Standard" which is always being updated and Improved. W3C would focus on a single definitive standard that can be used by developers. W3C preferred a modular approach to hasten progress and hence the use of a number of separate specifications and APIs. HTML5 is therefore made up of many more technologies than just markup. It also consists of other technologies that the W3C and its partners are creating such as Scalable Vector Graphics (SVG), Web Open Font Format (WOFF), the Semantic Web stack, Extensible Markup Language (XML), and a variety of APIs. Figure 3 (15) below shows the APIs.

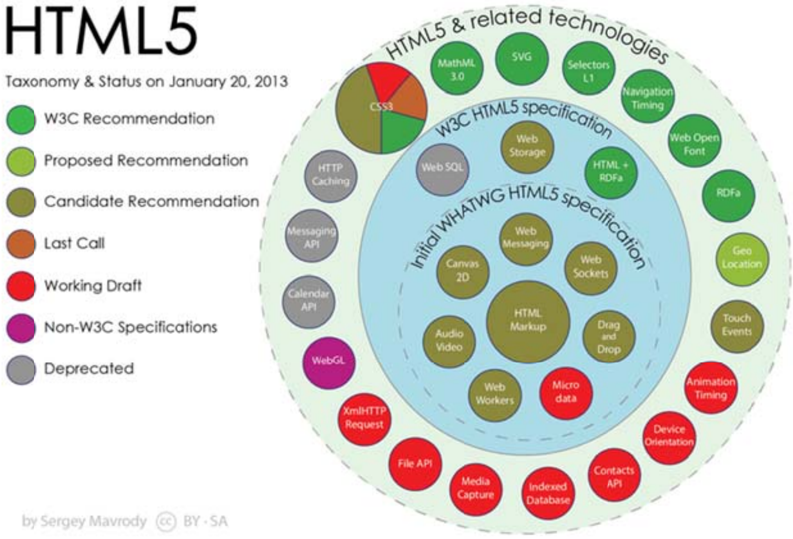


Figure 3. Currently available and under-development APIs

2.1.4 *CSS*

CSS3 is a big part of HTML5 along with JavaScript. CSS3 is the latest version of CSS and is a style sheet language used to describe the look and feel of markup documents such as HTML web pages. (16) The big advantage of using CSS is that the presentation semantics are separated from the content. This provides benefits such as multiple pages being able to share presentation formatting, which can speed up development, site-wide changes can be made from one file and these changes will affect the whole site. The HTML markup is also easier to read as it is less cluttered.

2.1.5 *JavaScript*

JavaScript is a scripting language and has been used for a long time together with web pages to add functionality and make web pages more dynamic and eventful. (17) It helps create visual effects on static pages as a result of user controlled events because it is a language that can be interpreted client-side by the browser. It enables dynamic updates to the browser with user interaction, asynchronous updating of parts of the web page and much more. JavaScript is very powerful and when combined with CSS and HTML it can help build some very good looking and highly dynamic web pages.

By combining the aforementioned technologies together it is possible to create powerful web applications that can be viewed using any kind of device that has a supporting browser installed. As shown in figure 2, support for HTML5 in mobile browsers is extremely good and all the big companies' browsers have a high level of support for HTML5 applications. Responsive web design (18) is creating web sites that respond to the device that is accessing them and render the page appropriately; using this approach, web sites that will work optimally on any device can be created. This means that apps are not limited to one single operating system but are available across all devices that have a browser.

2.1.6 Native Application Development

A native application is an application that has been developed for use on a particular platform or device. It is a locally installed piece of software that having been written in code that is specifically designed to run on that operating system means that it is therefore optimized for the running environment. The fact that the apps are written specifically for the target platform gives it the ability to access device-specific hardware and software. By having full access to the device in this manner the app can take advantage of the hardware and use it in the best manner possible and offer the user the best user experience available. A developer will need to write a separate native app for each platform they are targeting. Native apps can either be installed on the device out of the box or be downloaded and installed from an app store.

2.1.7 Hybrid Application Development

Like native apps, hybrid apps run on the device not in a browser; like web apps, hybrid apps are written using web technologies. (19)

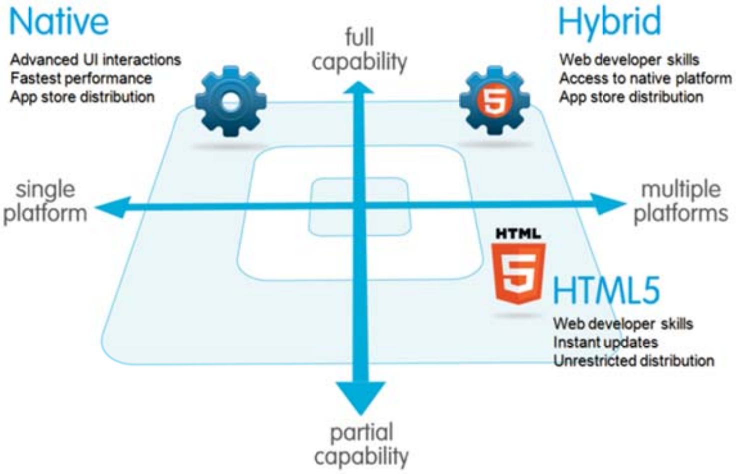


Figure 4. Native, HTML5, and Hybrid

What separates hybrid apps from native apps is the use of a native container as a wrapper. A container is a webview control (UIWebView on iOS, WebView on Android and others) that uses the native browser rendering engine (not the on-device browser itself) to present the content in a full-screen and browserless fashion. Figure 4 on the previous page illustrates this. (20)

What separates hybrid apps from web apps is the implementation of an abstraction layer, which exposes the device functionalities such as GPS, accelerometer, camera, etc., to the hybrid app as JavaScript APIs. That makes it possible for hybrid apps to interact with the device's native functionalities in the way web apps cannot or could not, due to security barriers between the browser and the device APIs. However, this gap in between is being closed up.

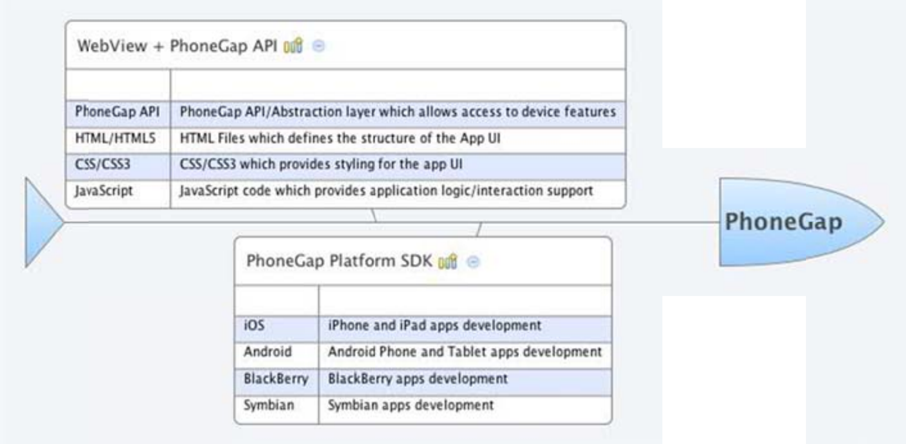
2.2 Comparison

2.2.1 Feature Richness

Native apps excel in both the in-app experience and the way they interact with the devices' system such as widgets and sending notifications. Inside the application, all the on-device accessories, such as the GPS, accelerometer, camera, various sensors, etc, are open to utilization; the hard keys' events (the "back" and the "context" button on a typical Android phone or the "home" key on an iPhone) are handled via respective native APIs; gestures, such as swiping, multi-touch, multi-finger gesture, pinch-zooming, are supported by the native SDK (21).

HTML5: Web standards aren't all there and ready yet, but they are evolving rapidly: W3C has a Device API Working Group (22) that is working on client-side APIs that enable the development of Web applications and Web widgets that interact with devices' services such as Calendar, Contacts, Camera, etc.

Whilst waiting for the standards to mature, an alternative way of hooking up into the device system and utilizing system accessories is to wrap up the web app in a shell native app. This is the essential approach of the PhoneGap framework (23), as shown in Figure 5. (24)



WebView + PhoneGap API

PhoneGap API	PhoneGap API/Abstraction layer which allows access to device features
HTML/HTML5	HTML Files which defines the structure of the App UI
CSS/CSS3	CSS/CSS3 which provides styling for the app UI
JavaScript	JavaScript code which provides application logic/interaction support

PhoneGap Platform SDK

iOS	iPhone and iPad apps development
Android	Android Phone and Tablet apps development
BlackBerry	BlackBerry apps development
Symbian	Symbian apps development

Figure 5. How PhoneGap works

However, this approach does add a level of complexity to the building process of the software.

2.2.2 *Performance*

Native apps have no web runtime barrier; their performances are not limited within that of the browsers. Also native apps can take advantage of GPU acceleration and multi-threading (25).

As regards to web development, the web runtime has been accelerating in the recent years: JavaScript Engine V8 has been a major development that ships with Google's browser Chrome (26); Web Worker APIs are making multi-threading possible; Trends indicate that desktop advances are on their way to the mobile. Besides, the majority of apps aren't bleeding-edge 3D-games but information-based apps (27).

There is however an array of areas which Web development is currently weak at. Scrolling, along with similar gesture-based user actions, for example, is one of them. There are many attempts to address this weakness such as iScroll, TouchScroll, GloveBox, Sencha, jQuery Mobile.

2.2.3 *Developer Experience*

Native apps use robust programming languages (Java, Objective-C, etc.) that are designed for complex application development and have proven their competence and usability. In addition to the programming languages, the native APIs were designed ground-up specifically for the platform at-hand (28), which naturally is the reason why they would fit their respective platform the best.

The downside is that there are various OS and different platforms of SDKs, sometimes even different versions of the same platform can have relatively significant differences.

In the world of Web development, JavaScript has become a serious contender nowadays; most of the computational logics that native code can perform, such as data processing, transition, animation, etc., can be performed by JavaScript, with the help of various libraries and CSS. Besides, the biggest advantage of JavaScript is that it can run on any given platform (given that it ships with a browser, which is normally the case). Of course, due to the incomplete development of Web standards and because each browser has a slightly different way of interpreting JavaScript code (or parts of it, different on different browsers), you'll need to tweak a thing or two in order to get the experience right for each platform, but you'd have to do that in native code too, and there are different versions and different devices that you have to deal with in the native world. (28)

2.2.4 Monetization

Native apps can make their profit via various ways offered by the marketplace business model, including one-time payment to unlock an app for its entirety, in-app payment and subscription mechanisms, in-app ads and sponsorship. (29)

However web apps do have their own ways without the marketplace. Web has always been a place where profit can be made; otherwise it wouldn't have lasted this long and still be prosperous. Approaches such as online ads, affiliating links, sponsorships and cross-promotion, etc., are a small portion of the whole business world of Web. Besides, one can always choose to make a native wrapper app to make use of the marketplace, with the help from technologies like the PhoneGap framework to address the problem of cross-platform publishing of these different native apps (30).

2.3 Conflicting Views

2.3.1 The Hype

Many mobile application developers have been listening to the hype (31) about HTML5 over the past few years and have been wondering if HTML5 really is sophisticated and capable enough to deliver mobile applications as well as native applications do.

2.3.2 Ready or Not?

In May 2012, the Financial Times ditched their award-winning iOS app (32) and embraced HTML5; in March 2013, Dark Sky announced their revolutionary global weather app (33) in HTML5, Forecast.io; However, during the same year in April, LinkedIn decided to turn to native development and abandoned their pursuit of their HTML5 web app (34); in September 2012, even Facebook admitted that HTML5 was a huge strategic mistake, quoted from Facebook's CEO Mark Zuckerberg, "*It's one of the biggest mistakes, if not the biggest strategic mistake we made. We're coming out of that*". (35)

While Mark Zuckerberg, in the long-term, does have high expectations of HTML5, he thinks *"it just wasn't there yet"*. (36) However, Sencha, a leading framework vendor for the mobile web industry, had different views on this. They think that the problem was not with HTML5 but rather how Facebook used it. (37) The problem, as they suspected, was that Facebook took a traditional web development approach to building an app, *"and often don't use the right tools and architectures for application development"*. So they went ahead and did their own version of Facebook called Fastbook. The purpose of building Fastbook, according to Sencha, was to prove that HTML5 was ready.

As much as Sencha, along with the several big players like Google, Mozilla, Adobe, etc., (38) support HTML5, even Sencha admits that right now HTML5 isn't ready for just everything just yet; when one of the developers was asked to comment on the weaknesses HTML5 has at the moment, he said, *"… the easiest one that I think of, today, is games"*. (39)

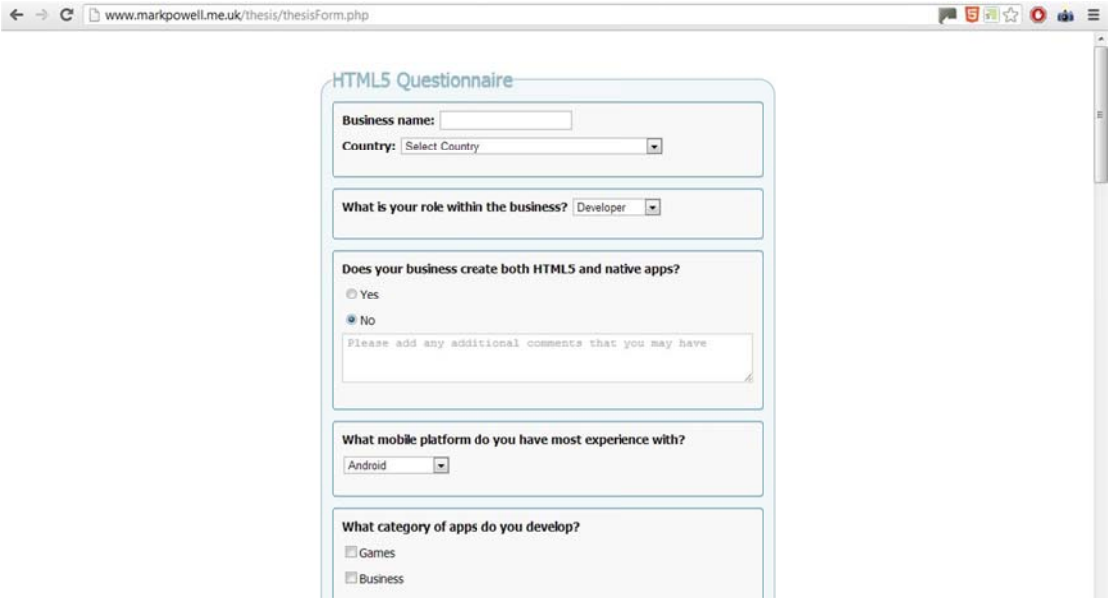
3 Realization

3.1 A Twofold Approach

The method that was chosen to use to conduct the research was twofold. This was derived from the consideration that both collecting professionals' opinions and conducting practical experiments were desired.

3.2 Questionnaire

It was decided to contact developers and app development businesses to get their views as to how much they use HTML5 presently and how they expect to be developing using it in 5 years' time. Thus, a questionnaire was collated and a web form containing the questions was created. Figure 6 below shows part of the online questionnaire.



www.markpowell.me.uk/thesis/thesisForm.php

HTML5 Questionnaire

Business name:

Country: Select Country

What is your role within the business? Developer

Does your business create both HTML5 and native apps?

○ Yes

● No

Please add any additional comments that you may have

What mobile platform do you have most experience with?

Android

What category of apps do you develop?

□ Games

□ Business

Figure 6. HTML5 Questionnaire (see Appendix for the complete form)

Past experiences of filling out questionnaires ourselves have made it clear that it is highly necessary to keep the number of questions to a minimum to get people to participate. It was therefore decided, as an absolute maximum that the questionnaire would contain no more than 15 questions. This has posed a limitation on the amount and the range of information that could be collected.

It was also decided to put more general than specific questions in the questionnaire, as it was not certain if developers could be reached directly, because it was often a contact form or an info@website.com address that was public and not individual developers' e mail addresses.

It was also very obvious that the questionnaire had to be built as professionally looking and intuitive for people to read and fill in as possible. The aim was to take the surveyed as little time as possible hence encouraging maximum responses and increasing the validity of the survey.

In order to keep the process as simple as possible, an e mail was sent to each company containing an explanation of what the research subject is and why it was being researched together with a link to an online form where they could fill in the questionnaire. When the link was clicked on they were presented with the form which was incorporated of mostly questions with dropdown answers, check boxes or sliders for percentages. Open text boxes were included where comments could be added.

The web form took the answers to the questions given by the participants and sent them to our email addresses using PHP.

Our empirical results are not only based on the observations while building the applications using both HTML5 and Android but also by collecting the data from the results of the questionnaire. The results of the questionnaire were parsed using a Java program that was written specifically for this purpose. The parsed data were then used to chart the results graphically using pie charts and the like.

3.2.1 Information Gathered

Business name: *business name*
Country: *country name*
Role within the business: *Developer/Management/Other*
Do they already develop HTML5 apps? *Yes/No*
Do they already develop HTML5 apps? *Comments*
What platform they have most experience with: *Android/* iPhone */Windows Mobile/BlackBerry/HTML5*
Categories of app they currently develop:
Games/Business/Communication/Education/Entertainment/Finance/Health & Fitness/Medical/Music/News/Other
Percentage of apps currently developed with HTML5: *percentage%*
Advantages of developing apps with HTML5: *Faster development/Works across multiple platforms/Easier to update the app no need to update on device/Save 30% on app store costs/Better security/Cheaper to develop/Not applicable (No experience with HTML5)*
Limitations of developing apps with HTML5: *less rich user experience/Less features (can't access as many devices as native apps can)/HTML5 not updated as quickly as native OSs/Less security, code is open for all to see/Can't be used*

offline as easily as native apps/No app store for HTML5 apps/Not applicable (No experience with HTML5)
Does HTML5 development of mobile apps pose particular problems as opposed to native app development? *Yes/No*
Problems comments: *comments*
Will HTML5 overtake native in following categories?
---News and blogging apps: *Yes/No*
---Map Navigation apps: *Yes/No*
---Games: *Yes/No*
---Photo apps: *Yes/No*
Expected % of apps developed with HTML5 in 5 years: *percentage%*
Would you like a summary of results emailed to you: *Yes/No*
E mail: *example@example.com*
Additional Comments: *comments*

The full version of the comments can be found in Appendix B.

3.3 Same Application, Different Technologies

After considering the problem, many questions were soon raised. As the other part of the twofold approach, a test involving practically building an app was carried out.

The application was an audience response system that is used in conferences to create interactivity between the presenter and his/her audience. It was first built using Java for Android and the Eclipse IDE.

Then the same app was built but this time using HTML5 and its related technologies.

During the process, thorough notes were made, noting the difficulty of tasks and time taken. The results were then used to help prove or disprove the hypothesis stated.

3.3.1 Program Logic

In a nutshell, the client app would talk to a server and get the information about sessions first; after the user chose which session to attend, the app would then contact the server and get the data of questions associated with that session, and display it on the screen; lastly, the app would send users' choices back to server. The results would then be displayed after analysis by the presenter during the conference, or saved to a database for later reference and usages.

3.4 Similar Studies

There have been several recent studies in the same area as this report. Two big studies were conducted by Kendo UI. (40) On two occasions in 2012 and 2013 Kendo UI surveyed over 4000 then over 5000 respectively as regards their choices for cross-platform development. In general they found that the majority of developers didn't feel that HTML5 had been overhyped and that 70% of developers would adopt HTML5 to build apps to reach all platforms simultaneously, as shown in Figure 7 (41) below.

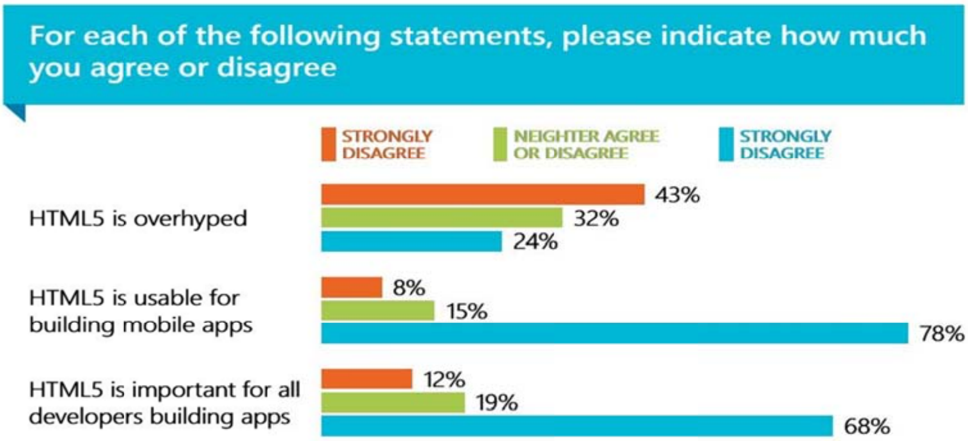


Figure 7. Results from Kendo UI

3.5 Method Discussions

3.5.1 *About the Application*

The application was a simple application that mainly performed certain client-server communications, which didn't require much of the device's functionalities and wasn't computationally intensive.

3.5.2 *About the Figures*

Before showing the results, it is necessary to lay down some fundamental figures for you as readers to grasp an idea of the scale and magnitude of the research.

A total number of 302 requests were sent, non-evenly distributed among 21 countries (most by email, some by filling contact forms on companies' websites). A total number of 82 replies were received, non-evenly distributed among 16 countries. The distribution is shown in Figure 8 below.

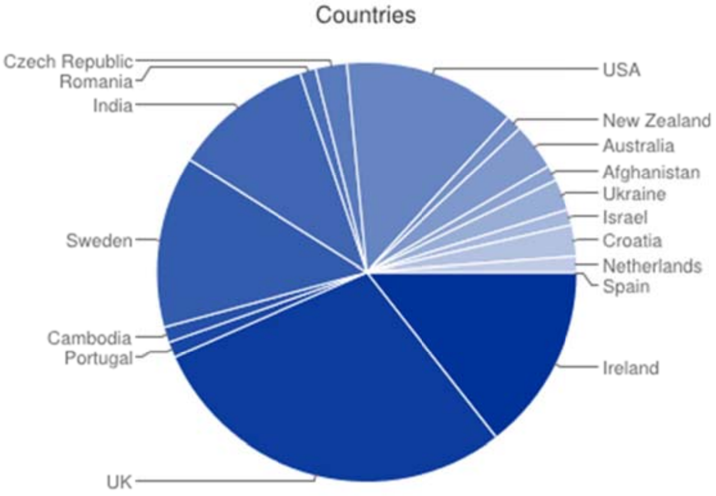


Figure 8. Geographical distribution by countries of sent request

Out of these 82 replies, 53 of them were very interested in this research and wanted the results sent to them on completion.

Among the replies, 54% were of manager-sort of positions, 42% developers, and 4% of other kinds of positions, as shown in Figure 9 below.

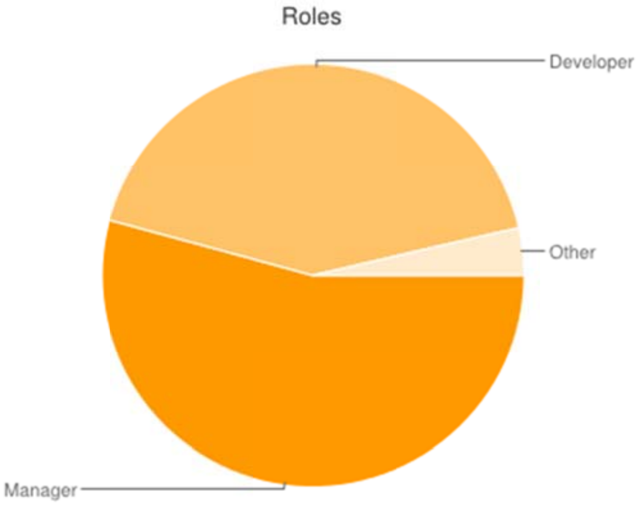


Figure 9. Composition of those surveyed

4 Results

4.1 Results from the Questionnaire

After a sufficient amount of data was collected, an analysis was conducted based on this research's purposes. Here are the results drawn from the analysis.

1. As shown below in Figure 10, the companies' main targeted platforms are iOS, 80% and Android, 18%, with 2% dedicated to HTML5 development

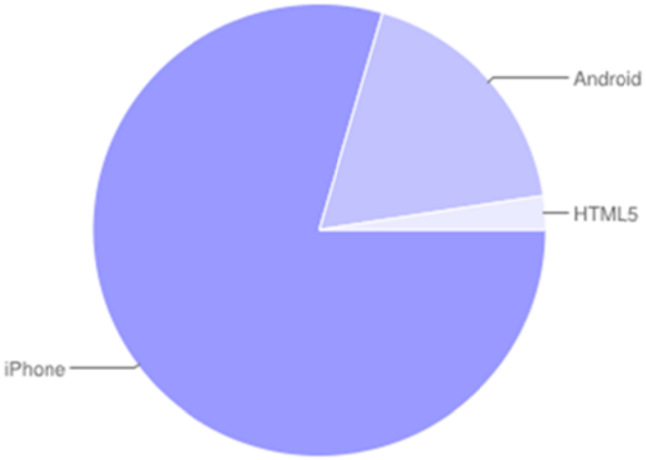


Figure 10. Companies' main platforms

2. 70% of the companies have built HTML5 apps, as shown in Figure 11

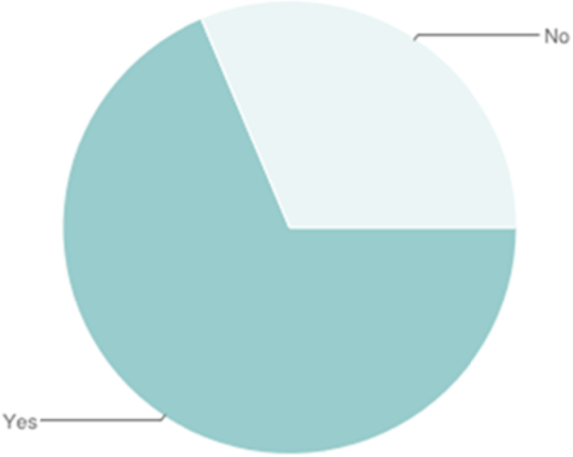


Figure 11. Companies that currently have HTML5 apps

3. The percentage of their HTML5 apps out of total is 14%, as shown in Figure 12 below

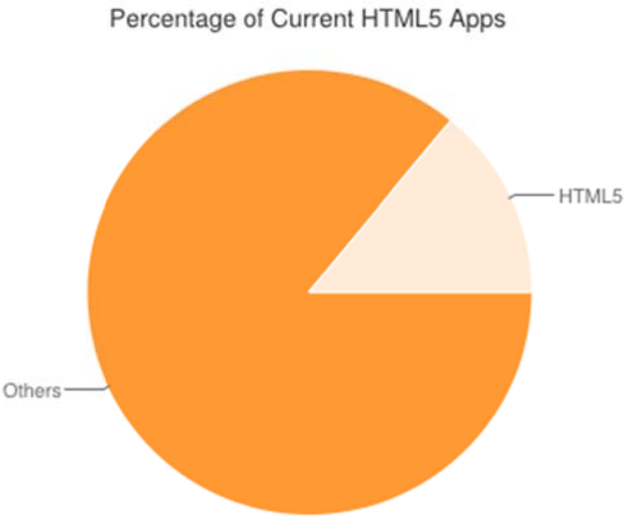


Figure 12. Percentage of current HTML5 out of total

4. 67% think HTML5 will take over the market in News category of apps, 14% in Map, 14% in Photo and 5% in Game, as shown in Figure 13 below

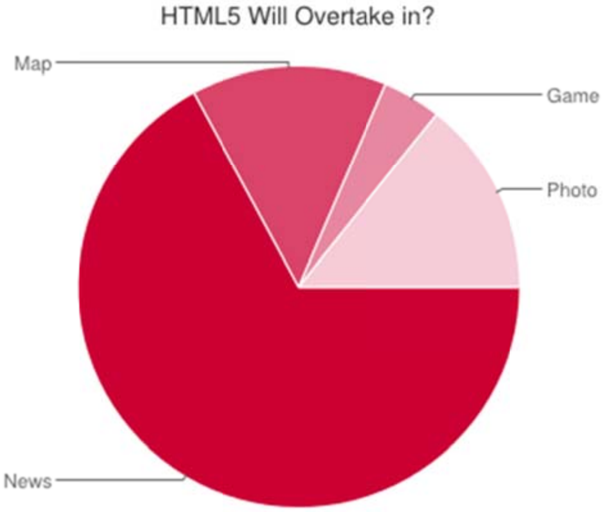


Figure 13. HTML5 overtakes in?

1. Advantages
   Out of 83 replies,
   a. 34% think HTML5 would enable faster development process
   b. 63% think HTML5 has the ability to "write once, run on all the platforms"

   c. 30% think that developing apps in HTML5 will be cheaper than in native environment
   d. 43% think that using HTML5 will make the updating and maintenance job of application easier
   e. 12% think that developing web apps in HTML5 will save them 30% on the app market cost (Apple’s App Store, Google Play, etc.)

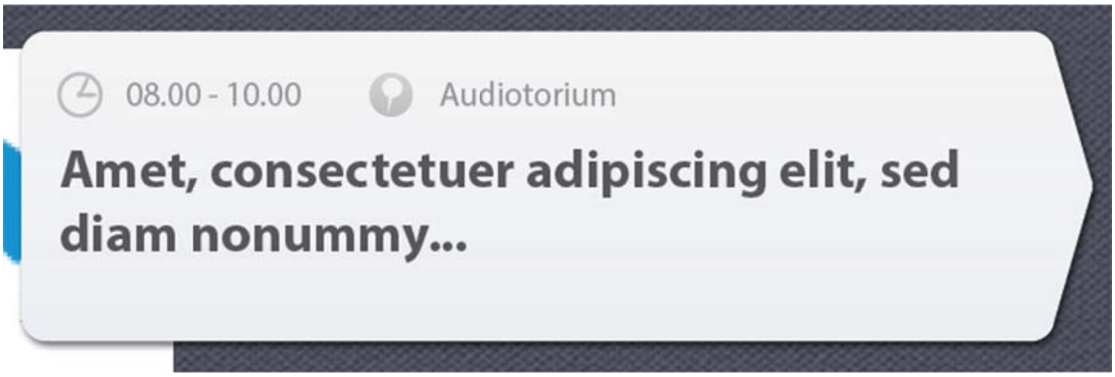
2. Limitations
   Out of 83 replies,
   a. 82% think HTML5 cannot provide as much rich user experience as native can
   b. 82% think HTML5 is not able to provide as many features as native is, regarding device functionalities.
   c. 36% think that HTML5 cannot keep up its pace with the native OSs, judging from the frequency of updates each have gotten in the past
   d. 41% think HTML5 is not as safe, due to the fact that all the markup and most of the code are open to everyone
   e. 63% think it’s not as easy to use offline and system cache with HTML5 as with native
   f. 43% think that it’s harder to make profit with HTML5 apps, due to the lack of a mature marketplace

4.2 Results of Implementation

4.2.1 Constructing the Interface

Design and implementing the interface for the Android app took 2 weeks. Because of the nature of the application, all the components and their layout had to be done programmatically and not in the layout xml file.

As shown in Figure 14 below, this is a typical layout that represents one session in a conference. It most often consists of a time, a place and a short description.



Amet, consectetuer adipiscing elit, sed diam nonummy...

Figure 14. A typical session UI

The major difficulty while doing this was that all the layouts had to be written from scratch because there was no equivalent in the system UI list. While this might not take long at all for an expert, it took a long time for a novice.

The same work, however, in HTML5, was much simpler and straightforward. It was an expectation that the interface work in HTML5 would be relatively easy and it was proven so. With the help of the JQuery Mobile framework, the similar look-and-feel UI was built in less time.

4.2.2 Implementing the Logic

Since the logic part wasn't complicated at all, little problem was encountered regarding this in both versions of the program. The same Java Servlets were used as the back-end components of both applications.

4.2.3 Performance

It was found out during the testing on Android that constructing large amount of user interfaces in the code was not an optimal choice. The tests conducted on Samsung Galaxy Note II showed that, on average, without the application having previously run before, it took 7 seconds for the system to render all the layouts. The same process in HTML5 took around 1 second counting the latency in the network.

4.2.4 User Experience

Due to the absurdly long waiting time interface-building phase in the Android version, the test results on user experience were not as optimistic as expected. However after one time of usage, all the transitions between activities were rather smooth and the long-waiting time was solved by the system itself with the cache being used.

On the other hand, the HTML5 version lived up to expectations. The transition effects and page rendering was very smooth (due to the utilization of JQuery Mobile), and the web page was responsive regardless what devices it was accessed from. It auto adjusted itself to fit the screen of Note II, Galaxy S2 and iPhone 5.

5 Conclusion

The results of the research conducted proved that our hypothesis was valid and that HTML5 is already a viable alternative to native for some types of mobile applications. This was further supported by our own personal experiences when comparing development of an application with both HTML5 and Android, during which we found HTML5 to be very capable of meeting our goals. These results together with the research of others such as Kendo UI showed that the hype about HTML5 was justified to some extent.

The fact that Facebook and LinkedIn moved away from HTML5 showed that HTML5 maybe wasn't ready for all types of applications and developers but the use of HTML5 by Sencha to create Fastbook showed that with the right expertise, HTML5 was already extremely capable.

Our sample size of developers that responded to the questionnaire was relatively small in comparison to other studies and it was limited in the number of questions we felt we could ask. We did however feel that there was a large interest in our research with many responders wanting to be contacted with a summary of the results and many took the time to write some very informative comments. The comments highlighted that HTML5 isn't quite ready for most developers with user interface and speed problems cited often. It seemed however that many developers believed that HTML5 will dominate app development in the future, at least for less demanding applications while games and more demanding apps will continue to be developed natively.

It should be pointed out that had we chosen to create a graphically intensive game instead of a simple application then we wouldn't have been able to prove that our hypothesis was valid. We still would have come to the same conclusion that HTML5 is already very capable but isn't the best choice for all types of applications.

The future of the W3 is an exciting subject to write about and be involved in researching. This report was born out of our interest in web technology and our desire to know more and get a better understanding of which direction developers and business might take as regards HTML5. Having had the chance to dig around and really immerse oneself in web technology literature and to get our hands dirty testing out these technologies has only added to the excitement. If we were to bet on the outcome we would bet that HTML5 will overtake native for mobile development within 10 years mainly because of the benefits of cross-mobile deployment but also due to the amount of web knowledge that already exists.

However, it is important for current web developers to take a different approach to mobile than that of web, in order to be successful with mobile development using HTML5. Facebook learned this lesson, coming from a company of web developers they had to abandon their HTML5 application and had to start employing native developers. However it was possible to create the app in HTML5 when programming from a mobile approach like that of Sencha with Fastbook.

6 Future Work

6.1 Restrictions

One obvious restriction was time. Time played a very important role throughout the work regarding planning, implementation and the approach and depth of the research.

Another restriction was the type of application we could choose to implement. Because we were not experts in Android and knew almost nothing about HTML5 before starting, we had to be very carefully when choosing the type of application, so that we could both finish it in time and fulfill our research. In our case, these two factors complemented each other, which made them equally important.

6.2 Deeper Research and Face To Face Interviews

We would have liked to implement an application that had more complexity regarding the program logic and system device utilization. In that case we could have compared more aspects of the two technologies.

We would have also liked to actually meet face-to-face and interview local app companies, because that way we could be sure that we were talking to the right people and we could ask more specific questions and collect more accurate data.

Unfortunately we were unable to do these due to the time constraint.

6.3 Work Suggested for Other Projects

For those considering working in the same field or continuing the work, the suggestion is researching the practicality and benefits of using hybrid mobile application development frameworks such as PhoneGap and Appcelerator Titanium. These use a mixture of both HTML5 and native programming and have been described as the best of both worlds. With the explosion in the sale of mobile devices, businesses can’t afford to ignore the possible money savings that creating apps that run on multiple platforms provides. By building apps using hybrid frameworks the HTML5 is wrapped in a thin native container that provides access to native platform features. It may be interesting to research whether it is safer and more efficient to use hybrid development until HTML5 is finished or maybe until HTML6 is released.

7 References

1. HTML, Wikipedia. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Html.
2. **C.R. Venkatesh**. (2012). HTML: Past, Present and Future. Retrieved from http://www.exeideas.com/2012/07/HTML5-past-present-and-future-via-infograph.html.
3. HTML5, Wikipedia. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/HTML5.
4. **Crawford, James.** What is HTML 5 and why should I care? *Jupix.* [Online] March 6, 2013. http://www.jupix.com/what-is-html-5-and-why-should-i-care.
5. **Ubl, Malte.** HTML5 paragraph after smartphones. *HTML5 Rocks.* [Online] Februray 14, 2011. http://www.html5rocks.com/en/tutorials/speed/html5/.
6. **Nair, K.G. Sreeju.** Semantic Tags in HTML 5. [Online] February 24, 2013. http://weblogs.asp.net/sreejukg/archive/2013/02/24/semantic-tags-in-html-5.aspx.
7. **Pilgrim, Mark.** WHAT DOES IT ALL MEAN? [Online] 2011. http://diveintohtml5.info/semantics.html.
8. **HTML5test.** HOW WELL DOES YOUR BROWSER SUPPORT HTML5? *HTML5test.com.* [Online] October 2012. http://html5test.com/results/mobile.html.
9. **HTML5test.com**. HTML5test.com score over the years. Retrieved from http://html5test.com/results/mobile.html.
10. **Osborne, Charlie.** Vehicles to have Internet access as standard? *Smartplanet.* [Online] January 5, 2012. http://www.smartplanet.com/blog/smart-takes/vehicles-to-have-internet-access-as-standard/21133.
11. **Ian, Jacobs.** W3C mission statement. *W3C.* [Online] 1999. http://www.w3.org/Consortium.
12. **Berjon, Robin.** HTML5 — Smile, it's a Snapshot! *W3C.* [Online] December 17, 2012. http://www.w3.org/QA/2012/12/html5_smile_its_a_snapshot.html.
13. Open Web Platform, Wikipedia. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Open_Web_Platform.
14. HTML5 Standardization process, Wikipedia. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/HTML5#Standardization_process.
15. **Mavrody, Sergey**. (2012). HTML5 APIs and Related Technologies: taxonomy and specification status. Retrieved from http://en.wikipedia.org/wiki/File:HTML5-APIs-and-related-technologies-by-Sergey-Mavrody.png.
16. CSS, Wikipedia. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Css.
17. JavaScript, Wikipedia. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Javascript.
18. Responsive Web Design, Wikipedia. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Responsive_web_design.

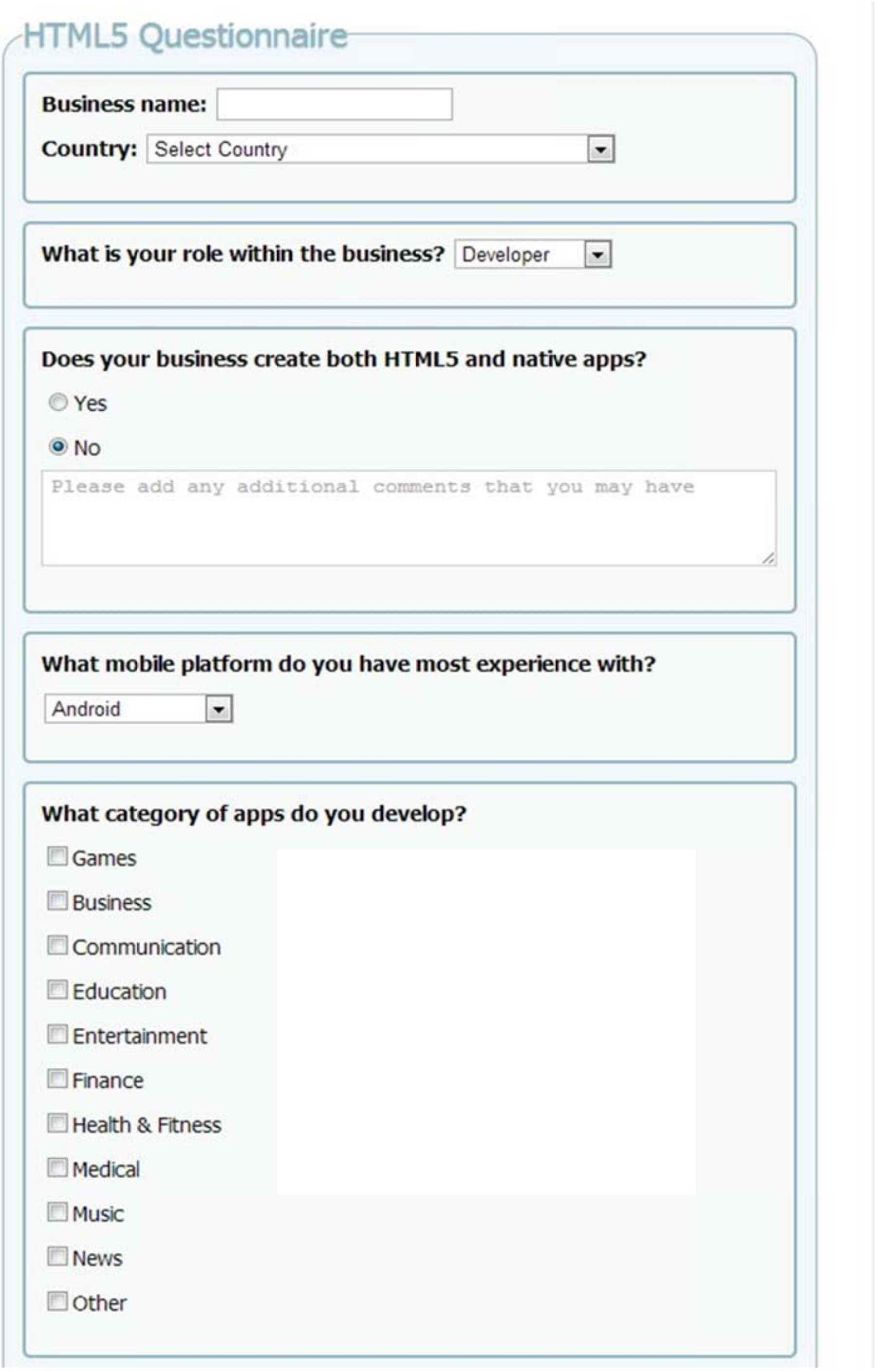
19. **Korf, Mario and Oksman, Eugene.** Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options. *Developer Force.* [Online] http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options.
20. **Developer Force**. Native, HTML5 or Hybrid? Retrieved from http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options.
21. **Mahemoff, Michael.** HTML5 vs Native: The Mobile App Debate. *HTML5 Rocks.* [Online] June 3, 2011. http://www.html5rocks.com/en/mobile/nativedebate/.
22. **Hirsch, Frederick.** Device APIs Working Group. *W3C.* [Online] August 2011. http://www.w3.org/2009/dap/.
23. **Chakraborty, Arnab.** PhoneGap - How Does It Work. *Arnab Chakraborty's blog.* [Online] December 5, 2011. http://arnab.ch/blog/2011/12/phonegap-how-does-it-work/.
24. **Chakraborty, Arnab**. (2011). PhoneGap Architecture. Retrieved from http://arnab.ch/blog/2011/12/phonegap-how-does-it-work/.
25. *Mobile Development: Campfire Discussion, Mertechdata.*
26. **Craig.** Why the V8 Javascript Engine is so good. *Developer Knowhow.* [Online] November 30, 2012. http://www.developerknowhow.com/why-the-v8-javascript-engine-is-so-good/.
27. **Gordon, Mary Ellen.** The iOS and Android Two-Horse Race: A Deeper Look into Market Share. *Flurry Blog.* [Online] Janurary 6, 2013. http://blog.flurry.com/?Tag=App%20Usage.
28. **Charland, Andre and Leroux, Brian.** Mobile application development: web vs. native. Communications of the ACM. *Communications of the ACM.* May 2011, pp. 49-53.
29. *HTML5–bridging the mobile platform gap: mobile technologies in scholarly communication.* **Padley, R.** 2011, The Journal for the Serials Community, pp. 32-39.
30. **Natu, Nakul Vishwas.** *JavaScript Game Engine for Mobile using HTML5.* 2012.
31. **Ronwinski, Dan.** HTML5: Don't Believe the Hype Cycle. *ReadWrite.* [Online] August 21, 2012. http://readwrite.com/2012/08/21/html5-ready-for-prime-time-dont-believe-the-hype-cycle.
32. FT wins Apple Design Award for its iPad app. *Financial Times.* [Online] June 9, 2010. http://www.ft.com/intl/cms/s/2/76a037fc-73c4-11df-bc73-00144feabdc0.html#axzz2VSWILSc9.
33. **Grossman, Adam and Turner, Jack.** Announcing Forecast.io. *Kickstart.* [Online] May 26, 2013. http://www.kickstarter.com/projects/jackadam/dark-sky-hyperlocal-weather-prediction-and-visuali/posts/438101.

34. **O'Dell, Jolie.** Why LinkedIn dumped HTML5 & went native for its mobile apps. *Venture Beat.* [Online] April 17, 2013. http://venturebeat.com/2013/04/17/linkedin-mobile-web-breakup/.
35. **OLANOFF, DREW.** *TechCrunch.* [Online] September 11, 2012. http://techcrunch.com/2012/09/11/mark-zuckerberg-our-biggest-mistake-with-mobile-was-betting-too-much-on-html5/.
36. **Langel, Tobie.** What Mark Zuckerberg's really said about HTML5 at Disrupt yesterday. *W3C.* [Online] September 12, 2012. http://lists.w3.org/Archives/Public/public-coremob/2012Sep/0015.html.
37. **Nguyen, Jacky and Avins, Jamie.** The Making of Fastbook: An HTML5 Love Story. *Sencha.* [Online] Decemeber 17, 2012. http://www.sencha.com/blog/the-making-of-fastbook-an-html5-love-story.
38. **O'Reilly, Tim.** Google Bets Big on HTML 5: News from Google I/O. *O'Reilly radar.* [Online] May 27, 2009. http://radar.oreilly.com/2009/05/google-bets-big-on-html-5.html.
39. **Nguyen, Jacky and Avins, Jamie.** *Sencha demonstrates HTML 5 can be faster than native iOS or Android apps (Facebook example).* [interv.] Robert Scoble. December 17, 2012.
40. Global Developer Survey 2013. *Kendo UI.* [Online] http://www.kendoui.com/surveys/global-developer-survey-2013.aspx.
41. **Kendo UI**. (2013). Hype you can believe in. In *Global Developer Survey Kendo UI 2013*, p. 8.
42. **Mertechdata.** *Mobile Development: Campfire Discussion.*
43. HTML. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Html.
44. HTML5. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/HTML5.
45. What is HTML 5 and why should I care? *Jupix.* [Online] March 6, 2013. http://www.jupix.com/what-is-html-5-and-why-should-i-care.
46. HTML5 paragraph after smartphones. *HTML5 Rocks.* [Online] Februray 4, 2011. http://www.html5rocks.com/en/tutorials/speed/html5/.
47. **Pilgrim, Mark.** WHAT DOES IT ALL MEAN? [Online] http://diveintohtml5.info/semantics.html.
48. Mobile browser support. *HTML5test.com.* [Online] http://html5test.com/results/mobile.html.
49. [Online] www.w3.org.
50. Open Web Platform. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Open_Web_Platform.
51. HTML5 Standardization process. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/HTML5#Standardization_process.
52. CSS. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Css.
53. JavaScript. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Javascript.
54. Responsive Web Design. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Responsive_web_design.

55. "Android", Github. [Online]
https://github.com/search?q=android&ref=cmdform.
56. "HTML, JavaScript", Github. [Online]
https://github.com/search?q=html5&type=Repositories&ref=searchresults.

Appendix A

Questionnaire



HTML5 Questionnaire

Business name:

Country: Select Country

What is your role within the business? Developer

Does your business create both HTML5 and native apps?

○ Yes

● No

Please add any additional comments that you may have

What mobile platform do you have most experience with?

Android

What category of apps do you develop?

□ Games

□ Business

□ Communication

□ Education

□ Entertainment

□ Finance

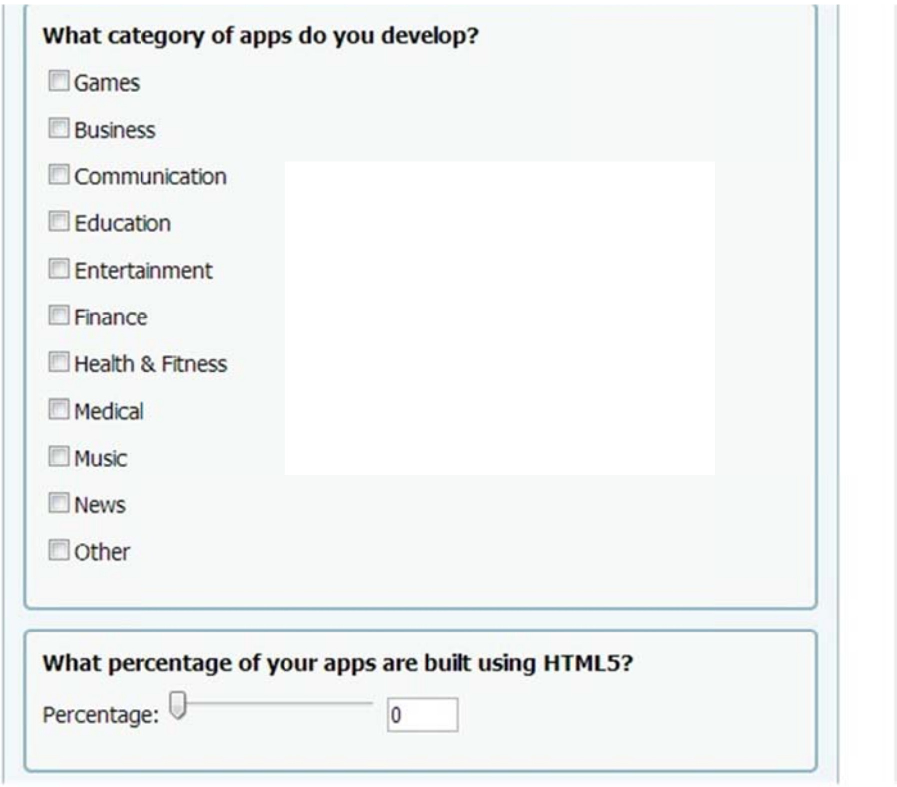
□ Health & Fitness

□ Medical

□ Music

□ News

□ Other



What category of apps do you develop?

□ Games

□ Business

□ Communication

□ Education

□ Entertainment

□ Finance

□ Health & Fitness

□ Medical

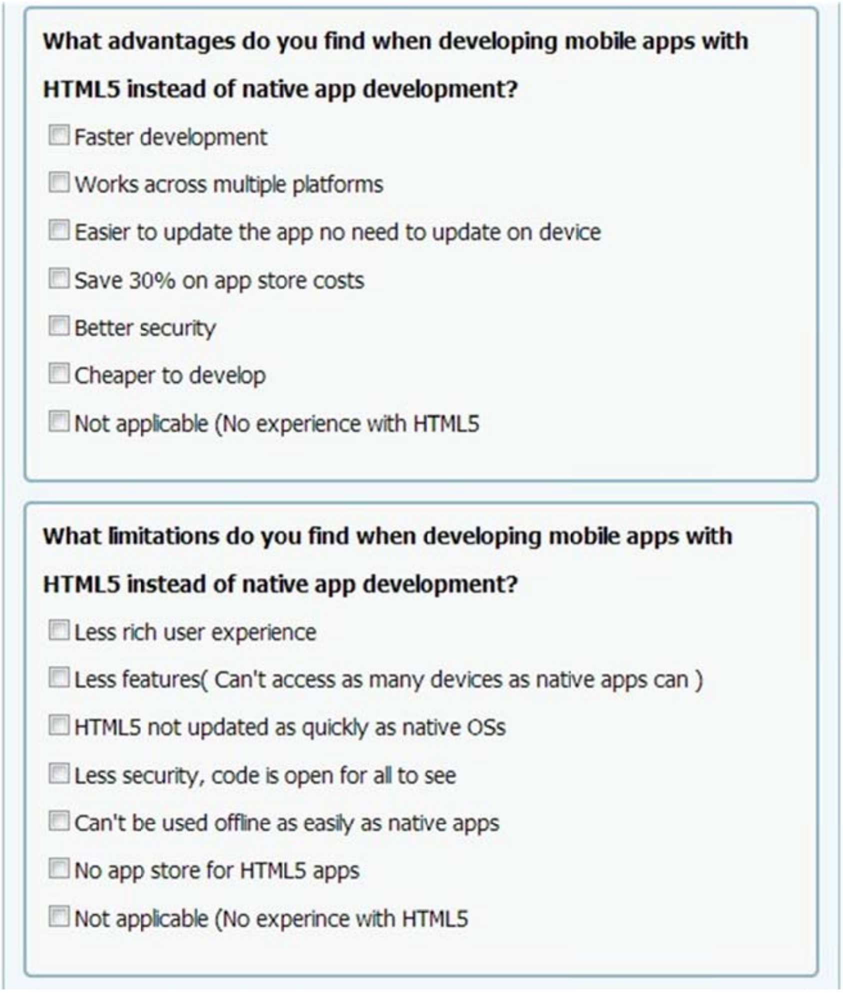
□ Music

□ News

□ Other

What percentage of your apps are built using HTML5?

Percentage: 0



What advantages do you find when developing mobile apps with HTML5 instead of native app development?

□ Faster development

□ Works across multiple platforms

□ Easier to update the app no need to update on device

□ Save 30% on app store costs

□ Better security

□ Cheaper to develop

□ Not applicable (No experience with HTML5

What limitations do you find when developing mobile apps with HTML5 instead of native app development?

□ Less rich user experience

□ Less features(Can't access as many devices as native apps can)

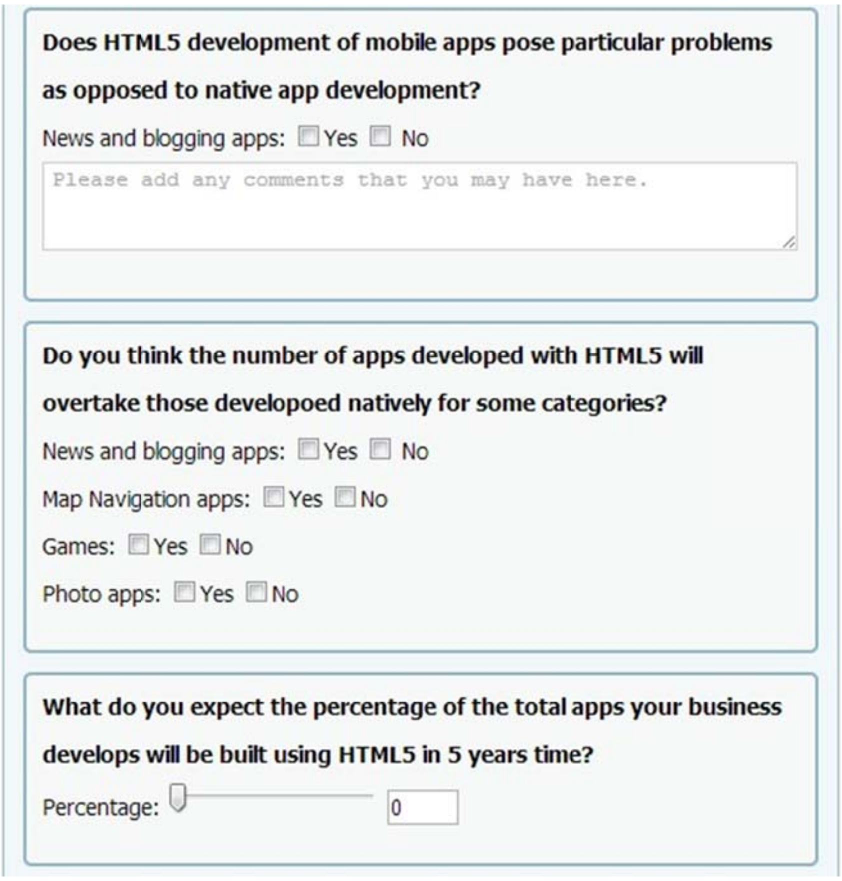
□ HTML5 not updated as quickly as native OSs

□ Less security, code is open for all to see

□ Can't be used offline as easily as native apps

□ No app store for HTML5 apps

□ Not applicable (No experince with HTML5



Does HTML5 development of mobile apps pose particular problems as opposed to native app development?

News and blogging apps: □ Yes □ No

Please add any comments that you may have here.

Do you think the number of apps developed with HTML5 will overtake those developoed natively for some categories?

News and blogging apps: □ Yes □ No

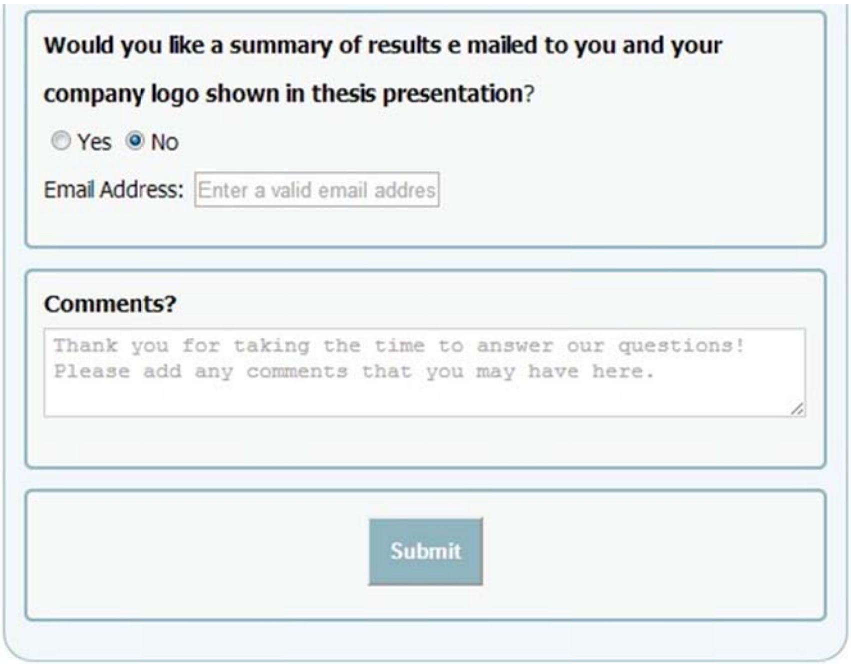
Map Navigation apps: □ Yes □ No

Games: □ Yes □ No

Photo apps: □ Yes □ No

What do you expect the percentage of the total apps your business develops will be built using HTML5 in 5 years time?

Percentage: 0



Would you like a summary of results e mailed to you and your company logo shown in thesis presentation?

○ Yes ● No

Email Address: Enter a valid email addres

Comments?

Thank you for taking the time to answer our questions! Please add any comments that you may have here.

Appendix B

Comments Gathered from Surveyed

Question: Does your business already develop HTML5 and native apps?

We only do native apps. We set up the business for that exact reason.

Mainly develop for native platforms, but have created in hybrid frameworks like PhoneGap and others.

Mostly native apps for app store success

We prefer Native development and have no intention of developing in HTML5; HTML5 gives a horrible user experience, slow and not practical.

Online web app builder called www.zapplab.net

95% of our business is in native development.

Decision depends on target audience, budget, app functionality and usability requirements.

We find it best to be flexible as every client's requirements are different

Our HTML5 apps are wrapped with PhoneGap which make access to device features and app store submission possible.

We can develop native applications, pure HTML5 web apps, and hybrid apps - web apps in native wrappers.

We've been developing native apps since 2008. More recently we've experimented with HTML5 and hybrid development. Ideally, we'd use HTML5 for everything and work from a single codebase to deploy apps across multiple platforms. Unfortunately we feel HTML5 is not mature enough to deliver a slick and robust user experience. In theory, HTML5 is great, in practice it maybe does not work so well.

I'd just like to point out that different problems may be solved by different tech. So while HTML5 may be great for a project, it may be a real bad idea for another. And then there are hybrid apps that use both native UI and HTML5...

Capable of doing HTML5, but purely native.

When we evaluate that it will benefit the client we use HTML5 or a mix of Native and HTML5.

HTML5 is promising but in the long run native apps is the way to go...

We prefer and always recommend native apps. There is a small category of apps for which a hybrid HTML5 app works well. This is typically informational or marketing style apps.

We did try HTML5 but weren't happy with the experience and quality of the App.

Our preference is always to create native apps.

Native FTW

Now we are working on native apps only. But we developed with HTML5 in past.

HTML5 does not give you (yet) the complete user experience that Native app will.

Only native Android and iOS apps. We do create HTML5 websites but not for apps.

Regular HTML5 web apps as well packaged HTML5 with PhoneGap etc. Native apps for iOS and Android.

We develop apps on iOS, Android, Windows and HTML5

Question: Does HTML5 development of mobile apps pose particular problems as opposed to native app development?

Yes, getting a handle on core device functionality is an issue. If you do, you need to write a lot of wrapper type classes to "talk" to the native OS and pass this back to the webapp. This type of development can be difficult. If you have a complex application, you are better to stay native.

Absolutely, HTML5 is great for websites not for apps

Differing browser compatibility = user experience, some device functionality is only valuable via the native OS tool kits

Again it depends on the client requirement. Often a client wants an app but doesn’t want it on the App Store, many of our apps aren’t even HTML, standard HTML and CSS2 are often good enough unless the client wants local data storage, which has its own challenges.

HTML5 performance is not good especially on Android platform. It depends on the device spec and Android version.

I have yet to see a HTML5 app of any substance that runs unmodified across multiple platforms. It is a lot harder to match the precise control of animation available in native apps. CPU consumption is a serious problem for HTML5 / JavaScript apps.

Yes it does but there are also many opportunities available using HTML5. BBC Sport and CNN are both great examples of HTML5 being used in a good way. Hybrid apps can be developed using platforms like PhoneGap. This involved developing in HTML5 and using a middleware solution like PhoneGap to create a native 'wrapper' for the app. This means the app can be downloaded via the app store and solves the burning issue of how users discover HTML5 apps. Right now the app economy is driven by native apps. In the future developers will lean towards HTML5 as the technology matures. Developing hybrid apps is a great way to bridge this gap and obtain the most positive benefits of both HTML5 and native app development processes.

The device manufacturer’s don’t take it as seriously as native, so tools aren’t as good which means we spend a lot of time on debugging issues. Performance is another issue, web browsers and web views don't offer the same performance as native.

Access to hardware

User experience

Sometimes even the browser differences between platforms (even Android vs iOS which both use WebKit) are so huge that getting the app to work in HTML5 is more expensive than a native re-implementation. Look - native is native.

Sometimes even the browser differences between platforms (even Android vs iOS which both use WebKit) are so huge that getting the app to work in HTML5 is more expensive than a native re-implementation. Look - native is native.

The biggest issue is the richness of the framework vs the native framework. It's much more limiting and it's only really appropriate for simple apps and simple games. We are asked to do much much more complex applications (image processing, data processing, hardware/device interactions...), this is just not an option with HTML5. The final issue is the overall user experience, it's not up to people's expectations yet (it's getting there though).

The Biggest problem is the User experience and access to native device features. The next issue is discovery of such apps as they are not searchable in an app store. However, we do not create pure HTML5 apps - there is a mid-way and that is called as Hybrid apps. We propose this to our clients when they do not want to spend money across all platforms.

At the end of the day, you can never go wrong with Native development.

It is way easier to develop, but from a UX perspective the html5 does not reflect differences across multiple operational systems (iOS vs Android sizes, habits..)

Mostly performance and managing client expectations for what an HTML5 app can do vs native.

Cannot access all device features.

In fact, we have limited our exposure to HTML5 based apps to news apps only, and quite happy with the outcomes.

Bad user experience, no appStore, non-personalized data - you must remember when you develop for personal phone, you want to be personal as possible..

So many different browsers to support. Behaves differently among them

It has a bad UX

Users want a stunning design, super user experience and fast performance. There is no way HTML5 apps do this better than native. Each user that has to wait a second will cost you one!.

Not as cross platform as you may think. Poor tooling support. Performance.

The user experience is not that great, on lower end handsets with less memory the experience is very sluggish. HTML 5 provides one kind of experience for all devices which is not right.

General comments

html5 sucks for apps.

To develop apps learn Objective C, Java and C#. Once you can code there is no need for HTML5. If you value your customers and love coding you will want to give your customers the best experience possible. It is not all about saving money.

I hope this helps with your research.

The current trend is native apps vs. web apps, however I believe the future of web is mobile and responsive websites. Native apps will still have their place but websites and web apps will be different strands of the same thread.

It's only a matter of time till HTML5 applications win this competition, but now HTML5 is just not ready to be the definitive technology. As for now, for an average project (in terms of scope and complexity), HTML5 doesn't support the required minimum of technology stack across the platforms. HTML5 is good for a small/simple application, but it's not enough for anything more complex. Not enough doesn't mean it's not possible, it's just pointless. For now.

We are a third party development company. We have developed hundreds of apps. Only around 20% of the apps that we build could be built with HTML5 / JavaScript. However, we build quite a few apps that embed a WebKit browser for a small aspect of the app deliverables. Fragmentation in the Android platform is a significant deterrent to adopting HTML5 for some projects. Clients want maximum audience for their apps. Too many legacy Android devices in use and clients want them addressed. WebKit in legacy Android versions is very poor. Good luck with your thesis!

This is a really interesting debate. This is the best implementation of HTML5 I've ever seen

This is a brief overview, if you want to chat in more depth feel free to give us a call, our web team would be happy to talk in more detail if it would help

In 5 years I hope to see HTML6 that will be a complete revamp of HTML. HTML itself is obsolete - if you need a JS framework to emulate what everybody needs (only very slowly) then there's a problem with the markup spec. And that's the state of things. How about native layouts? What about a newer version of JavaScript that would introduce a proper native object programming model that would make it a bit less flexible because its flexibility costs a LOT in speed. One day I hope to have enough resources to build a next-gen, distributed, web operating system. :-)

You should also differentiate between B2C apps and enterprise app. Enterprises are a better candidate for Hybrid/HTML5 apps simply because user experience does not matter much here. Also with the BYOD movements, HTML5 apps make more business sense. You should have included that in your questions.....just a thought.

To get a really rich mobile experience, you must go native.

Again we only recommend this to our clients when we deem it's the right fit and audience.

HTML5 is definitely amazing, but I don't expect it overtake native apps in next 2 years. Maybe later! Good luck!

In addition to HTML 5 there are now multi-platform templates that are available that give you the best functionality of a native experience as well deploy to multiple devices seamlessly. Our new company http://www.applacarte.com does exactly this.

Best of luck Mark and Yuesong. P.s. with PhoneGap, our advice would be that you shouldn't believe the hype. We ended up having to write our own version of PhoneGap due to how buggy it was. At the end of the project it would have been easier for us to have developed natively for both iPhone and Android. If the app is reasonably complicated and its message mission critical, we have found that PhoneGap isn't up to the task (Just our agency though).

Look forward to seeing what you find!

It is our a great pleasure to get connected with you and look forward to speak to you for mutually beneficial engagement possibilities.

Native Apps are better than HTML5

Thanks and good luck with the thesis

Note that HTML5 apps can be packaged with PhoneGap etc. It reduces some of the limitations (features, app stores etc).

HTML 5 development as per our understanding is at least a couple of years away before we see mass market takeover. HTML 5 apps are good to use for Mobile Web App to be used in the browser but not that great for Mobile Client apps which we download from the App store. There are too many platforms out there who claim to support all the devices but they too also have lot of challenges. The developer community support is that great for many, many of the platforms just focus on a particular OS like the iOS but they have issues with Android. User believe they save money by making HTML5 apps but we haven't seen any such things because the time saved in development has led to increase in time for testing. In the end you spend the same or sometime even more because in the case of Android when you have so many form factors supporting and ensuring that it works on everyone is itself very challenging.