Master's thesis

Two years

International Master's Degree in Computer Engineering

Cross-Platform Solution for Mobile Application Development

MA level, 30 credits

Aditya Polisetti
Abstract

With the possibility to increase the smart phone market, the mobile application development is experiencing a rapid improvement in terms of both revenue and innovation. Considering the demand in the application development market, there is an urge to develop applications faster and smoother with the growing number of different platforms and technologies. Many companies and brands are making their marketing strategies more efficient by using mobile applications.

The constant growth in mobile devices and operating systems raise several problems and challenges in mobile application development. With this growth, a developer finds it difficult to build applications keeping the native technologies in mind with respect to the device platform. Application development for a mobile device mainly deals with native development platforms such as C, Objective-C, Java, C#, J2ME, C++ etc., and is incompatible with the cross-platform support. HTML5 is a solution that incorporates cross-platform capability in all these devices.

The purpose of this thesis is to research the different approaches of mobile application development and to find a favourable environment for a cross mobile platform development. The smart phone application development can built in three environments and the applications can differ such as native applications, cross-platform applications and hybrid applications.

The research work includes the study of several cross-platform frameworks, responsive web designing and location-based web services. The objective of this thesis is to develop a full-fledged application of classified services which also includes location-based services keeping in mind the new web technologies such as HTML5, CSS3, jQuery etc., and to use the frameworks by investigating and choosing different tools and APIs that incorporate the development of cross-platform applications. The work also includes the development of the look and feel prototype models for the classified ad mobile applications in terms of native application and hybrid application respectively for Android mobile phones. The developed cross-platform application is then compared with the two look and feel prototypes and the results are presented accordingly. The results can give us a better idea of the choice of environment for
application development based on the user requirements such as ease of development and cost constraints.

**Keywords**: PhoneGap, Titanium Appcelerator, jQuery Mobile, Sencha Touch, cross-platform development, HTML5, CSS3, W3C, Aurigma Up, JavaScript, responsive design, mobile web, hybrid app design, mobile web applications.
Acknowledgements

I would like to thank my beloved parents for their love, support and motivation. I would like to thank my supervisor, Prof. Tingting Zhang for approving and giving me the opportunity to carry out this thesis work at Mobidoo Technologies Ltd. I would like to thank Peter Lindberg, Project Manager at Mobidoo Technologies Ltd. for giving me this opportunity to carry out my thesis and providing me valuable input and necessary resources to complete my thesis successfully. I would like to thank Niclas Andreasson, CTO at Mobidoo Technologies Ltd., for supervising and guiding me through the entire work, which helped me to complete this thesis without delay or fail.
# Table of Contents

Abstract ........................................................................................................................................... ii

Acknowledgements ......................................................................................................................... iv

Terminology ....................................................................................................................................... vii

1 Introduction .................................................................................................................................. 1
   1.1 Background and problem motivation .................................................................................. 4
   1.2 Overall aim ........................................................................................................................... 4
   1.3 Scope ...................................................................................................................................... 5
   1.4 Concrete and verifiable goals ............................................................................................... 6
   1.5 Outline .................................................................................................................................... 6

2 Theory ......................................................................................................................................... 8
   2.1 The history of the smart phone market development ......................................................... 8
       2.1.1 Android OS .................................................................................................................... 10
       2.1.2 iOS .................................................................................................................................. 11
       2.1.3 Windows Phone ............................................................................................................. 12
   2.2 Mobile application development environment ................................................................... 12
   2.3 Native Application Development ....................................................................................... 16
   2.4 Cross-Platform application development .......................................................................... 22
   2.5 Hybrid mobile application development ............................................................................ 38
   2.6 Location-Based Services ..................................................................................................... 43

3 Methodology ................................................................................................................................. 48
   3.1 Considerations for the proposed solution .......................................................................... 48
   3.2 Modules in the application ................................................................................................... 49
   3.3 Approach ............................................................................................................................... 49
   3.4 Choosing the native development platform ......................................................................... 50
   3.5 Choosing the framework for hybrid app development ....................................................... 50
   3.6 Choosing the environment that satisfies the requirements ............................................... 51
   3.7 Choosing the user interaction framework for HTML5 to develop a cross-platform application .......................................................... 52
   3.8 Choosing a framework for location-based services ........................................................... 52
   3.9 Choosing a JavaScript framework ......................................................................................... 53
   3.10 Selecting a responsive design framework ......................................................................... 54
   3.11 Prototyping methods ........................................................................................................... 54

4 Implementation ............................................................................................................................. 56
# Cross-Platform Solution for Mobile Application Development

Aditya Polisetti

## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Identification of requirements</td>
<td>56</td>
</tr>
<tr>
<td>4.2 Tasks to be accomplished</td>
<td>60</td>
</tr>
<tr>
<td>4.3 Proposed system functionality</td>
<td>61</td>
</tr>
<tr>
<td>4.4 Proposed system architecture</td>
<td>62</td>
</tr>
<tr>
<td>4.5 Application Design</td>
<td>63</td>
</tr>
<tr>
<td>4.6 Implementation</td>
<td>71</td>
</tr>
<tr>
<td>5 Results</td>
<td>89</td>
</tr>
<tr>
<td>5.1 Module 1: Research results</td>
<td>89</td>
</tr>
<tr>
<td>5.2 Module 2: Development results</td>
<td>95</td>
</tr>
<tr>
<td>5.3 User Testing</td>
<td>113</td>
</tr>
<tr>
<td>5.4 Functional Testing – test cases</td>
<td>116</td>
</tr>
<tr>
<td>5.5 Summary and analysis of the results</td>
<td>120</td>
</tr>
<tr>
<td>5.6 Future work</td>
<td>122</td>
</tr>
<tr>
<td>6 Conclusions</td>
<td>124</td>
</tr>
<tr>
<td>6.1 Discussion</td>
<td>125</td>
</tr>
<tr>
<td>6.2 Ethical Considerations</td>
<td>126</td>
</tr>
<tr>
<td>References</td>
<td>128</td>
</tr>
</tbody>
</table>
## Terminology

### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADT</td>
<td>Android Development Tools</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ASF</td>
<td>Advanced Systems Format</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>LBS</td>
<td>Location-Based Services</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>WHATWG</td>
<td>Web Hypertext Application Technology Working Group</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
</tbody>
</table>
1 Introduction

With the rapid growth in smartphone mobile industry, the mobile application development market has become a key to raise the standards. According to the latest forecasts from Canalys [1], the mobile device market is about to reach 2.6 billion units by 2016 which means there are more smart phone devices than personal computers in the world. During 2013, the smart phones will outnumber mobile phones [1]. Several smart phone operating systems have recently been evolving for example Firefox OS, Ubuntu, Sailfish etc., in addition to the current popular mobile platforms Android, iOS and Windows. These will draw attention in the coming years and the development of applications for all platforms will be even tougher to manage and deploy. The potential of the mobile market is increasing year by year. With this improvement, the development of mobile applications is generating work opportunities and interest among the developers.

Platform fragmentation [2] will be one of the main problems in developing smart applications for mobile devices. For all the different mobile platforms described above, there will a least be support for devices older hardware and the updated software cannot always support with that of older devices. Also, developing mobile applications for several platforms demands the knowledge in their respective programming languages and frameworks that they support, which would be time consuming and costly. Some of the companies are compromising and forcing the developers to stick with any platform for application development, excluding opportunities to market at this juncture, where currently, the smartphone sales are phenomenal [1]. Hence, there is a need to think seriously towards the cross mobile development approach. This means, there is a need for the development of a framework that helps users to use these frameworks in any of the mobile devices regardless of platform and version.
1 Introduction

Cross-Platform Solution for mobile application development
Aditya Polisetti

2014-11-10

Figure 1.1: Canalys Estimates and forecast- CAGR [1]

Figure 1.2: Canalys 2012 shipment estimates [1]
Cross-Platform Solution for mobile application development

Aditya Polisetti

1 Introduction

2014-11-10

Figure 1.3: Canalys 2016 shipment estimates [1]

Figure 1.4: Gartner Mobile Operating System market share [3]
1.1 **Background and problem motivation**

Mobidoo Technologies Ltd. is the company that authorized this thesis. The company needs to develop a light-weight solution for a multi-platform mobile application for classified services which should support location-based services. The application needs to be light-weight and maintained. Hence, a proposal has been made to develop a prototype of the classified application considering the three development environments discussed above.

The initial purpose of this thesis is to carryout research and find a feasible solution for developing an application that meets the requirements which should be chosen among the three development environments. The application has to be user-friendly in terms of functionality and graphic user interface interactions. It should be able to use web services for location tracking. It must maintain sessions and online data for the registered users and should provide smooth interface. The prototypes build for the three environments were taken into consideration to choose one among them which meets the requirements. The finalized developed product is to be tested and deployed as a public end-user version. Also, this thesis work provides a study of several known and unknown frameworks as well as APIs.

1.2 **Overall aim**

The main objective of this thesis is to show the ability of HTML5 as a framework to develop a cross-mobile application and to provide Mobidoo Technologies with arguments as to why they should embrace HTML5 as their mobile application development platform. This study proposes the development of a mobile classified services application which covers selecting environment, frameworks, testing and deployment of a web application development. In short, the purpose of this thesis is:

- To demonstrate the purpose and possibilities of migrating the native application development to cross-platform mobile application development.
- To develop a fully functional mobile web application for Mobidoo Technologies.
Other objectives include:

- To study the support of HTML5 in several mobile browsers and devices.
- Listing the differences between the three development environments.
- Search for a suitable HTML5 framework which provides a cost effective light-weight graphical user interface.
- Research among several JavaScript frameworks which helps build the web-based or cross-platform mobile application.
- Study several frameworks for location-based services and choose which is suitable to use in the application.
- Research in hybrid application development and the frameworks which supports hybrid mobile development.
- Compare the three environments based on factors such as application support, application performance, cross-platform support, development process, difficulties, availability and development cost.
- Develop the chosen application based on the criteria of the proposed solution and the prototypes necessary.

1.3 Scope
The thesis work covers most of the topics in web programming which provides a fully functional cross-mobile web application. This thesis also covers the prototypes of native Android application and hybrid Android application. The work does not cover the native application development for mobile platforms other than Android. This work is divided into two parts, a research part and an application development part. The research part comprises of studying, choosing and using the frameworks that are used for development of the proposed solution, whereas the practical work includes developing the application and prototype. The final deliverables consists of the live functional cross-platform application and the prototypes of both Android and hybrid mobile application.

- Environment 1: Native application
- Environment 2: Web application
- Environment 3: Extended native/Hybrid application

1.4 Concrete and verifiable goals

The challenges and goals of this thesis work are listed below

- What problems arise in cross-mobile application development?
- What are the main benefits of using web technologies as a standard for cross-platform application development?
- Can the usage of HTML5 and related web technologies minimize the implementation cost and provide effective user interaction regardless of mobile platforms?
- Which environment is best suited for the development of the proposed solution?
- What are the alternatives in cross-mobile application development?

1.5 Outline

This thesis work is organized as follows:

Chapter 1 gives a general overview of the topic that includes an introduction to the topic, the problem statement, research goals and the scope of the thesis.

Chapter 2 provides the theoretical part of the thesis which includes a basic introduction to topics that are necessary to mention. It also provides several frameworks that are studied and the development of the application.

Chapter 3 provides the methodology which includes the environment, modules for developing the application etc.

Chapter 4 provides the design and implementation of the system which includes the wire-frames and mock-ups of the system.

Chapter 5 presents the results of the proposed solutions and the research questions.
Chapter 6 provides a conclusion for the proposed solution and discusses possible areas for future development.
2 Theory

This chapter discusses the theoretical background information and the work related to this thesis paper. As the paper concentrates mainly on the mobile application development, this chapter gives a brief introduction to web terminologies. This section also discusses the frameworks and development tools related to the thesis. Section 2.1 discusses the history of smart phone market and its development. Section 2.2 introduces the design environments of the mobile app development. Sections 2.3, 2.4 and 2.5 discuss the respective environments in terms of architecture, frameworks, examples etc. Section 2.6 discusses the location-based services and its frameworks.

2.1 The History of the smart phone market development

With the increasing mobile usage, most of the business companies are using the strategy of marketing their products by effectively deploying its applications into mobile market stores. The international telecommunication union [4] estimates that there were six billion mobile cellular subscriptions which is equivalent to 87% of the world’s population.

![Figure 2.1: ITU estimates - Internet users and mobile subscriptions [4]](image_url)
According to Gartner’s prediction [5] about the future mobile market, Android operating system devices are the most used mobile devices in the world with 49% of usage followed by Windows mobile with 19% and Apple iOS with 17% of usage.

Figure 2.2: Gartner’s Prediction on OS usage – 2015 mobile market [5]

Based on Figure 2.2, it is clear that the mobile application market boom is mainly revolving around the three popular mobile platforms, Android OS, iOS and Windows Mobile OS. The applications for Android devices are available in Google Play, and for iOS the applications are available in App Store. For Windows it is Windows Market Place.
2.1.1 Android OS

Android is an open source mobile operating system founded in 2003 by Andy Rubin [6] and purchased by Google in 2005 [7]. It was made open source under the Apache License [8]. It is currently the biggest in the smartphone operating system market. Figures for the future are still high enough for Android to rule the mobile operating system environment. Since the year of its open source license, several new versions have been successfully released. Several mobile manufacturing devices are using this operating system to keep the advantage of open source, which has been a revolution in the smartphone mobile industry in terms of usage, cost and availability.

Figure 2.3: Android platform version history [11]
2.1.2 iOS

This mobile operating system was developed by Apple Inc. for its mobile (iPhone) and music device (iPod Touch) in 2007 [11]. Since then, it has extended its support to other Apple devices. It is based on the desktop operating system Mac OS X. The name of this operating system has changed from iPhone OS to iOS, as the OS is intended to run on several Apple devices. Apple’s iOS is not an open source platform. It is not licensed for non-Apple hardware installation. So, the operating system can only be installed on Apple hardware only. iOS allows the devices to install supported applications through App Store only. These smart mobile devices are of touch based. The version history of iOS is as follows:

- iPhone OS 1.x
- iPhone OS 2.x
- iPhone OS 3.x
- iOS 4.x
- iOS 5.x
- iOS 6.x

<table>
<thead>
<tr>
<th>iOS Version</th>
<th>Xcode Version</th>
<th>Date</th>
<th>Devices initially installed with the iOS version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td>2007-06-29</td>
<td>First iPhone released</td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td>2008-07-11</td>
<td>iPhone 3G</td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td>2009-06-17</td>
<td>iPhone 3GS</td>
</tr>
<tr>
<td>3.2</td>
<td>3.2.1</td>
<td>2010-03-02</td>
<td>First iPad released</td>
</tr>
<tr>
<td>4.0</td>
<td>3.2.3</td>
<td>2010-06-21</td>
<td>iPhone 4</td>
</tr>
<tr>
<td>4.1</td>
<td>3.2.4</td>
<td>2010-08-17</td>
<td>iPod Touch 4th generation, Apple TV 2nd generation</td>
</tr>
<tr>
<td>4.2</td>
<td>3.2.5</td>
<td>2010-11-15</td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td>3.2.6</td>
<td>2011-03-11</td>
<td>iPad 2</td>
</tr>
<tr>
<td>5.0</td>
<td>4.3</td>
<td>2011-10-12</td>
<td>iPhone 4S</td>
</tr>
<tr>
<td>5.1</td>
<td>4.4</td>
<td>2012-03-16</td>
<td>iPad 3rd generation, Apple TV 3rd generation</td>
</tr>
<tr>
<td>6.0</td>
<td>4.5</td>
<td>2012-09-19</td>
<td>iPhone 5, iPad 4th generation, iPad Mini</td>
</tr>
</tbody>
</table>

Figure 2.4: iOS version history [12]
2.1.3 Windows phone
Microsoft presented its new mobile version of Windows in 2010 [13]. This new series has been introduced prior to the older versions of the mobile platform, Windows Mobile. It is now emerging as one of the most popular mobile platform devices. The design of this platform mainly emphasizes design and user experience and its light-weight integrated applications [13]. There is a steady increase in the sales and popularity of the mobile devices using Windows as platform [5]. Up until now, six versions have been released. The version history is listed below.

- Windows Phone 7
- Windows Phone 7.5 Mango [14]
- Windows Phone 7.5 Tango [15]
- Windows Phone 8 Appolo [16]

2.2 Mobile application development environment
According to the Gartner research company [17], a 62% increase is expected in the global revenue to $25 billion dollars from mobile application downloads. As of now, both Google Play from Android and App Store from iOS provides more than 700,000 applications to download. This is a big market, which is why a choice in terms of mobile environment to develop has to be made, as the cost for the development differs from platform to platform and also depends on the type of the application.

2.2.1 Categories of mobile application development
A decision needs to be made by a firm to develop a mobile application with a support for features, platform and performance at a cost that is affordable. The mobile development environment can be divided into three categories. Several factors will depend on the selected category of development. The factors include cost, speed, performance, ease of use, ease of development etc. The division of the development environment is listed below.

- Native mobile application development
- Cross-platform/web mobile application development
- Hybrid mobile application development
These three environments are discussed individually later in this section.

![Figure 2.5: Categories of mobile application development](image)

### 2.2.2 Benefits of mobile application development

A user can access and share any kind of information anytime from any location which makes the user more social within the network. With the current animated and graphical features, the applications ensure user experience and quality. The business of a company will increase by staying in touch with the employees and customers building healthy relationships and improving communication with clients. It is even possible to advertise the business with the help of other applications.

### 2.2.3 Criteria for choosing mobile platforms

This section examines the criteria for selecting mobile application development environment. Here is a list discussed precisely for a mobile app development.

**Purpose of the application**

This criterion decides the purpose and usage of the application. Creating an application without a clear idea is not a good approach to proceed further. There are several categories of mobile applications such as games, news, education, business, entertainment, finance, food, drink, lifestyle, health, fitness, music, medical, technology, productivity, social networking, marketing, travel, utilities and weather.
**Budget**

Developing an ideal application in different platforms will be expensive. A concern could be if a small budget is assigned to a project after deliberations an environment can be opted that suits best. For example, if we are going to build an application that will not be using system hardware or sensor data, a web application will be the best option and it will be more economical.

**End user**

There is a need to plan the application flow by considering how a user thinks of it. An application needs to be discovered by the user in order to use it. Here, we need to consider the marketing strategies of the product/application and also needs to decide the exact category of the application. For instance, if we plan to design a video sharing application, this belongs to the categories of education or entertainment or social networking. These are useful for filtering an application search by the user. Also we must consider the navigation from the application to the web browser and vice versa. If a website has its own application, it may insist the user to download that app and use it in the device.

**User experience**

A positive first impression is essential for a successful mobile application. User interaction with the application will be the key factor for deciding the development environment. Considering the goals of a system, we must decide the steps of choosing environment without ignoring the user experience. Beautiful eye-caching user interface design, features, presentation and content of the application will withstand the success. In addition to this, user reviews, user rating, UI elements on the screen, multi-language features etc., will boost the application for a better user experience.
2.2.4 Factors for choosing a development environment

The factors considered for the development environment for mobile devices are listed below:

**Usability**

We need to consider the fact that a mobile user will be interrupted all the time. So, we make sure that our application should run in the background or to resume after interruption such as phone calls or messages. Also, the design context should be considered based on the category of the application. For instance, chatting applications do not need to be present at all times. The notifications should be sleek and should not dominate other applications.

**Ease of development**

We need to decide the development span with the budget it has assigned. The development team, subdivision of the team etc., considering all possible ways of development, plays a major role in deploying an application without fail.

**Development time**

The development time will depend on several factors such as the complexity of the system, proficiency of the developers and of the team assigned. If there is a need to develop an application for more than one platform for a native application, obviously the development time will increase. Also, the application that is developed needs an approval time to be a part of their respective application stores.

**Maintenance**

Updating and maintaining a mobile app will be a tough task for a native non-web application as it also includes users who download and update the features. The developer is needed to submit the updated version for submission approval. For a web application, this task is easy as one can update the features etc., from their own server.
Performance

In addition to all other factors, performance is considered a very important factor. The developer needs to ensure that the app loads very quickly without any lag in terms of navigation. We need to keep in mind that mobile users have little patience.

2.3 Native application development

A native application is written in a particular language for developers and is developed to run specifically on their respective operating systems. If we want to have a native application in several platforms, we need to program the same application so that the same functionality can be replicated for another platform. These applications are downloaded and installed on the platform and are platform dependent. For instance, a mobile application that is developed using iOS platform is available to install on the devices that run iOS only. It is native, as the code deploys only for the platform where it is intended to be used.

The basic features provided by native software development kit (SDK) are as follows:

- Development platform virtual machines optimized for the devices
- Mobile emulators/simulators
- Integrated development environment with respect to SDK
- Database storage and access
- File structure
- Hardware access of the device such as compass, GPS, accelerometer, camera, Bluetooth, Wi-Fi etc.
- Debugging tools
- Multimedia support (audio/video/image)
- Access to phone features such as address book, contact list, task manager, notifications etc.

The top three mobile platforms mentioned from Gartner’s prediction [5] are discussed in this section.
2.3.1 Android operating system

The Android operating system is built for touch-interface mobile systems and is built on the Linux kernel [8]. It is open source under the Apache license [9]. So it is inexpensive for a developer to develop applications based on Android OS. This makes the Android operating system more popular than the rest. Being an open source platform, is well supported in terms of development. The developers have very good documentation available in order to build applications [10].

Architecture

The architecture of Android is shown in Figure 2.6. It is based on the Linux kernel which has several layers in its structure, each referring to several tasks such as framework, libraries, etc. The middleware libraries are written in C language and these are device specific [19]. These libraries provide the essential functionalities of the platform. Dalvic virtual machine helps to run the applications and is optimized to several runtime environments [19]. The dalvic virtual machine (DVM) is used to convert the class files into “.dex”. These are compiled Android application code files. The application framework [19] deals with the applications by managing content, activity, life cycles, resources, location and much more. The application layer comprises of default installed applications such as browser, contacts, message system etc.

Integrated development environment

The Android applications are coded by most of the Java standard edition APIs replacing by a few graphic development libraries. The applications run on DVM instead of JVM. The Android development toolkit can be configured in all the three popular desktop platforms, Windows, Linux and Mac. The integrated development environment (IDE) tool used for development purposes is Eclipse. The developed product can be debugged using the emulator as a virtual device or can be tested directly in the device itself.

For a developer to submit the developed applications, s/he must be a registered Android developer by paying a small fee. The developer can use the Android SDK for development and can submit the completed application to the Android market.
Cross-Platform Solution for mobile application development
Aditya Polisetti

Figure 2.6: Android architecture [19]

Figure 2.7: Life cycle of the Android application development
2.3.2 iOS

Apple iOS is built primarily for mobile touch-interface systems and it is loosely based on the successful desktop platform, Mac OS X. User interaction plays a key role and is an attractive quotient for iOS which is loosely based upon the human computer interaction concept using multiple touch gestures [20]. The interface games and applications for iOS are the main attraction for the user. This is one of the main reasons iOS became the most popular mobile platform in terms of application development.

Architecture

The architecture of iOS is represented in a set of layers. It is similar to that of the architecture of Mac OS X. The applications communicate with the device hardware with the help of intermediary sets of interfaces. The architecture of iOS has four layers which are presented in Figure 2.8.

The core OS layer communicates with the system kernel. It deals with IPC, threading, memory management, security, hardware optimization [20] etc.

![Figure 2.8: iOS architecture [21]](image-url)
The core service layer manages the database SQLite support, XML support and manages its frameworks for services such as accounts, address book, CF network, core data, foundation etc. The media layer manages the support for all the media elements such as graphic, image, audio and video. It also provides support to 2D and 3D graphics with its graphic engine. The cocoa touch layer includes some high-level features and key frameworks which are used to build iOS applications [21]. The support of the high-level features include document support, multitasking, storyboards, printing, push notifications, gesture recognizers, peer to peer services etc.

**Integrated development environment**

The iOS application development has its software development tool kit (SDK) and it uses the cocoa programming environment. Native applications are developed with the help of iOS frameworks and the language, objective-C [21]. The applications are available through iTunes and the user must have an account to install any application on the supported device. The objective-C programming language is used for the development of all applications.

Figure 2.9: iOS application development life cycle
2.3.3 Windows Phone operating system

Windows phone operating system is the latest mobile platform from Microsoft. The main attraction of Windows Phone is its approach to user interaction. The application development can be carried out using Silverlight and the tools are available for free. This platform has been increasing its extent and according to Gartner’s prediction, it will be the third most popular mobile platform world-wide [5].

Architecture

The architecture is intended to provide user experience with rich media support. The kernel layer usually deals with security methods, networking and storage management. The next level includes the management of applications, updates, license, graphical output, session management, notifications etc. The application level, presents the application to enhance user experience with several frameworks such as Silverlight, XNA and web technologies. Figure 2.10 shows the architecture of the Microsoft Windows Phone platform.

![Figure 2.10: Windows Phone architecture](image-url)
Integrated development environment

The developer tools for the Windows Phone application development are available for free and provided by Microsoft [22]. The platform uses the dot net environment and C# for developing apps. The set of tools provided are listed here.

- Visual studio express as the integrated development environment
- Express Blend for the user interface design
- Silverlight
- XNA Game Studio
- Virtual Emulator

Figure 2.11: Windows Phone application development life cycle

2.4 Cross-platform application development

The biggest challenge for mobile application developers nowadays is to design and develop mobile applications across different platforms. This development is termed as cross-platform/web app development. To ensure this multi-platform development strategy, developers are keen to focus on the technologies which enable applications for multi-platform support. Frameworks that enable this feature are becoming crucial for the developers as their design and development are less time-consuming eventually leading to low development costs.
2.4.1 Development frameworks

With the help of some developed frameworks, it is possible to develop these cross-platform applications. The frameworks are unique in their way of functioning and behaving. There are several key factors for choosing a development framework. With their diverse capabilities and functionalities, this area will be challenging for developers; to explore the possible ways to develop a multi-platform application using these tools and frameworks. There is a need for testing these frameworks on all the platforms as well, which we expect to run without bugs. An application which works perfectly on an Android device may not work exactly the same way on an iOS device. As the frameworks need to run mostly on touchscreen devices, certain characteristics are needed for these frameworks.

- They must support multi-platform devices for a wide audience.
- They must use HTML5 and CSS3 capabilities as all smart mobile devices have web browsers and many devices support these web standards.
- They must have touch screen optimization support by providing basic user interface elements and by handling events.
- They should be light-weight to support lower bandwidth connections.
2.4.2 HTML5

HTML5 is a web standard hypertext markup language which helps provide several frameworks for designing and developing websites. It is the fifth version of the W3C standard of hypertext markup language [23]. HTML5 is evolved from the Web Hypertext Application Technology Working Group, simply called WHATWG [24] which is a community focused on developing HTML and related web technologies. Prior to the previous version, this version had a vast number of updates and features for better web browsing capabilities. This version has introduced many features and mainly focused on multimedia capabilities. These features are focused on eliminating the use of Flash content in web browsers. As many mobile devices do not support Flash within their system, this could be the best choice in order to extend its support towards the trending mobile devices. A code snippet of the basic HTML5 page is shown in code 2.1.

```html
<!doctype HTML>

<HTML lang="en">

<head>

<meta charset="utf-8">

<meta name="description" content="Sample HTML code snippet">

<meta name="author" content="nynepride">

<title>Code Snippet of HTML5</title>

<link rel="stylesheet" href="style.CSS">

<script src="jquery.js"></script>

</head>

<body>

<div id="wrapper">

<header>

</header>

</body>

</HTML>
```
Code 2.1: Code snippet of the basic HTML5 page

```html
Header section

<header>
</header>

<nav>
Page Navigation
</nav>

<section id="content">
<article>
Any text formatted Article
</article>
</section>

<aside>
Sidebar box
</aside>

<footer>
Footer content
</footer>

</div>   </body>
</html>

Code 2.1: Code snippet of the basic HTML5 page
HTML5's Browser Support

With the introduction of HTML5, several new elements and layout markups have been introduced; all are not equally supported by browsers. Some browsers may support some features, while other may support all of them. It all depends on the browser and version. As the current trend favours HTML5 evolution, almost all updated versions of popular browsers support the newly added features of HTML5 with CSS3, a new version of cascade style sheets for better user experience. As mentioned earlier, the new HTML5 tags and elements are supported in all the updated versions of modern browsers in both desktop and mobile devices. There are several online tools for testing the HTML5 elements for a particular browser. The below charts can indicate which functionality is not supported in a specific browser. HTML5test [25] is one such tools that tests and provides results for HTML5 browser support. The given charts provided by the HTML5test online tool [25], gives us an insight into the score of HTML5 in several browsers and platforms.
Cross-Platform Solution for mobile application development
Aditya Polisetti

2 Theory
2014-11-10

Figure 2.13: HTML5- Mobile browser support score [25]

Figure 2.14: HTML5- Tablet browser support score [25]
Some useful HTML5 components

Several new tags related to media, graphics and animation are introduced in HTML5 which are the main reasons for its success. These components are capable of developing very efficient web applications both in performance and functionality. Some of the important components are described in this section.

Geolocation

This technology makes it possible for a web page to know the user’s current position based on the latitudes and longitudes. This API collects information consisting of latitude, longitude and also altitude, but in rare cases, to determine the current position. The API comprises of three functions which are described below [26].

getCurrentPosition() – the current position of the user is retrieved. For a mobile browser device GPS is used for this and for a desktop device, the Internet service provider’s base station location will be taken into account.

watchPosition() – the position of the user will be tracked and the movements are recorded just like a GPS device. This function is of no use for a desktop device.

clearWatch() – removes or clears all the recorded tracks from watchPosition() function

The code is embedded within the JavaScript to act accordingly. The code snippet for geolocation position is given below in code 2.2.

```html
<script>
if(!navigator.geolocation){
  alert("Browser does not support");
  document.getElementById("testchk").style.display="block";
}
</script>
```
```javascript
else{
    navigator.geolocation.getCurrentPosition(success, error);
}

function success(position){
    var lat = position.coords.latitude;
    var lng = position.coords.longitude;
    //alert(lat);
    document.createForm.lattxt.value = lat;
    document.createForm.lontxt.value = lng;
    var ua1 = navigator.userAgent;
    if (ua1.indexOf("iPhone") > -1 || ua1.indexOf("iPad") > -1) {
        document.createForm.phoneimg.value = "yes";
    } else {
        document.createForm.phoneimg.value = "no";
    }
}

function error(error){
    switch(error.code)
    {
    case error.TIMEOUT:
        alert ('Timeout');
        break;
    case error.POSITION_UNAVAILABLE:
        alert ('Position unavailable');
    ```
break;

case error.PERMISSION_DENIED:
  alert ('Permission denied');
  break;

case error.UNKNOWN_ERROR:
  alert ('Unknown error');
  break;
}
}
</script>

Code 2.2: Code snippet of the basic HTML5 geolocation position on a page

Multimedia

Multimedia is necessary for current web applications as most of the multimedia components such as images, audio, video etc., are embedded within a web page to meet modern requirements. HTML5 has defined APIs for all embedded content which minimizes the use of third party plugins or content in the browser, for example Flash. Also, HTML5 enables user to draw elements in 2D, simply called canvas. It is included in the 2D drawing API. There are many functions for drawing a canvas object. Filling colours, borders, defining shape, thickness etc., are some functions. This API is very useful for developing games and infographic content [27].

Some code snippets for these specifications are shown below in codes 2.3 and 2.4

<video controls poster="sample.png" width="500">
  <source src="sample.mp4" type="video/mp4">
</video>
<audio id="samples" src="sample.mp3" controls>
</audio>

**Code 2.3:** Code snippet for HTML5 audio and video JavaScript

```javascript
<script type="text/JavaScript">
window.onload = draw;

function figure(){
    var canvas = document.getElementById('myCanvas');
    if (canvas.getContext){
        var context = canvas.getContext('2d');
    }
    else{
        // Put code to execute if browser doesn’t have canvas support here
    }
}

</script>
```

**Code 2.4:** Code snippet for HTML5 Canvas - Drawing object

*Drag and drop*

HTML5 has a feature enabling dragging and dropping elements from position to position. This API is also called DnD support (drag and drop support). This DnD support provides a versatile user interface that simplifies the code.
Web storage

HTML5 comes with two mechanisms in terms of data storage. Session storage and local storage that will enable the storage of data like cookies on the client side. In session storage, the data is stored until that particular page in the browser is closed. Local storage is similar to cookies; it will be stored in a temporary location within the system and can be removed when the cache is cleared.

2.4.3 CSS3

The introduction of cascading style sheets version 3 (CSS3) is a revolution in design world which has so many eye-catching features that ensure user experience, presentation and styling. With the introduction of new components such as media queries, animations, transformations, selectors, web fonts etc., the web page looks promising and attractive. These components are described in detail here.

CSS media queries

CSS3 media queries are responsible for mobile optimization of websites. This component helps the style sheet of a web page to change its size and content depending upon the screen size of the browser. In short, with the help of these CSS3 media queries, a user can design a web page that suits well for any type of device fitting to the viewport. The specification [28] consists of several properties that include width, height, orientation, aspect-ratio, resolution, colour etc. A code snippet is given below that represents the CSS media queries code.

```css
@media screen and (min-width: 600px) and (max-width: 900px) {
      .class {background: #333; }
}

@media screen and (max-device-width: 480px) {
      .class { background: #000; }
}
```
Code 2.5: Code snippet for CSS3 media queries - responsive design

**Transforms**

CSS3 has the ability to animate an element. With the help of this feature, an element can be transformed into 2D or 3D which can rotate, transform or scale. With the help of transforms in CSS3 and canvas in HTML5 we can build simple games within the browser.

**Web fonts**

Introducing embedding fonts within the website has been a best practice to minimize the content of images for typography. With this feature, the user can load fonts dynamically that are attractive and stylish. The use of the `@font-face` feature enables access to declare, load and manipulates the font within a CSS file. A sample code snippet is given in the below code 2.6.

```css
@font-face {
    font-family: cecilia;
    src: url('fonts/cecilia.ttf');
}
```

Code 2.6: Code snippet for CSS3 web fonts

### 2.4.4 JavaScript

JavaScript is a very popular light-weight, object based and platform-independent web scripting language. It can be implemented both in client-side and server-side environments. JavaScript is mainly used in web browsers, web applications, PDF documents and widgets. It was originally developed in Netscape. Later its support has been extended to each and every currently known browser. It is a very successful development in terms of web application. With the help of its light-weight APIs and frameworks. The web technology has seen a new level of success. With the collaboration of Ajax technology and CSS, this interpreted language creates an easy and promising environment in web programming and development. The sample code snippet of JavaScript is given in code 2.7.
<HTML><head><script>
document.getElementById("p2").style.color="blue";
</script></head>
<body>
<p id="p2">Hello World!</p>
</body>
</HTML>

Code 2.7: Code snippet for sample JavaScript

JavaScript syntax is quite familiar with programming languages such as C and PHP. There are many programming forums and much literature available online for a developer which helps to test, deploy and debug applications. With a basic JavaScript, we can do many things. Some of them are listed below

- Increase speed from client-side web
- Easily implement many functions such as date, time, location etc.
- Reduce the load on the server
- Perform all kind of mathematical calculations
- Perform form validation from client-side
- Design effective web pages collaborating with CSS and HTML
- Manage cookies
- Check the web browser name and version
- Develop simple games.

jQuery

jQuery is the most popular library of JavaScript, which is used mainly for client-side web page style scripting by adding animation effects without burdening the system. This free open-source library is licensed under MIT [30]. There are many frameworks and plugins available for JavaScript to create a web page to provide better user experience which draws the user’s attention to visit a web page regardless of content. A well-structured documentation is available for developers from their official website, which helps developing applications and plugins using the language.
Responsive design

With the overwhelming usage of the web and the internet in all available devices ranging from desktop to mobile devices, designing and rendering of web pages has been highlighted and draws the attention of developers. Before introducing the new trends HTML5, CSS3 and frameworks of JavaScript, the developers used to design different versions of a single website for different devices. One version for desktop browsers and another for mobile browsers. With the introduction of several JavaScript frameworks and CSS3 media queries, this work has become simple for designers and developers they can design and develop once and the page will adjust to the size and type of the device.

A website with responsive design runs based on the browser, device and platform. With the help of media queries, a designer can use the resolution and screen-size of targeted devices and code accordingly. The year 2013 has been termed a responsive year with the advancements technologies and its reception in market. “Now more than ever, we’re designing work meant to be viewed along a gradient of different experiences. Responsive web design offers us a way forward, finally allowing us to ‘design for the ebb and flow of things’“ says Ethan Marcotte, the author of Responsive Web Design [32].

Figure 2.15: Responsive design in three scenarios - desktop, tablet and mobile [32]
The use of the same code and content for different-size layouts is the main advantage of using responsive frameworks.

**Frameworks**

There are several frameworks available to embrace responsive web design. These frameworks use JavaScript and CSS functions to build predefined methods and libraries that can easily be integrated into web pages to perform accordingly. Some of the open source and free responsive design frameworks that have been chosen and discussed in this section are listed below.

- Twitter Bootstrap
- Responsive Grid System
- Skeleton Grid System
- HTML5 Boilerplate

**Twitter Bootstrap:**

Twitter Bootstrap is one of the famous responsive design frameworks which provides a number of design and develop components like widgets, images, CSS and JavaScript files. It is free, open source and licensed under Apache License [33]. This development project can be downloaded for free from GitHub repositories [33]. The grid system of Bootstrap has 12 columns and supports responsive media queries. There are rich JavaScript plugins available in the package used for several tasks such as dropdown, navigation menu, accordion, tooltip, image slide, popups, collapse, button animation, tables, transitions etc. The responsive grid system and the file structure of Twitter Bootstrap are represented in figures 2.16 and 2.17 respectively. The advantages of using Twitter Bootstrap are listed below.

- Free, open source
- Good literature and development support available online
- Well-structured files defined
- Responsive and rich in UI components
- Ease of use
- Light-weight and user-friendly
Skeleton grid system:

Skeleton grid system is another popular responsive framework which uses a small collection of CSS files and media queries to develop a responsive grid system [35]. The grid of this system is divided into 12 layouts. The media queries segment in the CSS files are used effectively for all sizes of devices including tablets and mobiles of both orientations. This code can be downloaded via GitHub and is open source under the MIT license [36]. The framework is only 25kb in size and does not contain any JavaScript files. No java plugins are available in this system. The code is easy to understand and flexible.
Responsive grid system

Responsive grid system [37] is a set of CSS files used for developing a responsive web system. It is light-weight, only 1KB size. It is open source and free to use. This system has three grid systems. The user can choose any of the available 12 column, 16 column and 24 column grid systems. It does not consist of any UI elements, images or JavaScript files and plugins.

HTML5 boiler template

This framework is another popular responsive based grid system which includes all types of files like JavaScript, CSS, icons, plugins and HTML [38]. This framework is widely used by popular websites like Google, NASA, Microsoft, Nike etc. Well documented code, online tutorials and videos are available for this framework which makes it easy for the developer to follow the code and build applications. It includes the most popular rendering libraries for JavaScript and CSS such as modernizer.js and normalize.CSS respectively. This framework is open source and free to use. This framework has a smaller number of JavaScript plugins and yet it is very rich in graphical UI elements.

2.5 Hybrid mobile application development

A hybrid application has the positive features of both native and web applications for a better application development. Usually a web application has restrictions in terms of using device hardware and sensor devices. It has no place in application stores which means it cannot be downloaded on the device. By using these web technologies with some supported frameworks and APIs, a hybrid application is developed which is able to run inside the device, just like executing a native application. It has the advantage of using web-view control in order to run the hybrid applications. This web-view control is used to deploy the web technologies such as HTML, CSS and JavaScript files within the device by rendering the native browser engine without directly using the web browser.
Without the users’ knowledge, the hybrid apps can interpret these web technologies within the web-view and run normally as a native application. With the help of some frameworks that are used to develop hybrid applications, a developer can connect the code of JavaScript to that of native development components in order to access the device’s native capabilities. With the hybrid applications we can achieve offline capabilities too. In addition to multi-platform and native API access.

2.5.1 Frameworks

As hybrid apps are designed to combine the advantages and features of native and web app development, some intermediary frameworks are needed in order to wrap the native code with HTML, CSS and JavaScript. With the help of the wrappers of these frameworks, the application can have access to use native APIs and hardware which is normally limited to web applications. There are several frameworks available in the market for hybrid mobile application development. Some of the frameworks are considered and discussed in this section. The frameworks that are discussed are free with good documentation and have good learning curve.

**PhoneGap**

PhoneGap [39] is one of the most popular mobile development frameworks. It is an open source framework which was contributed by Apache Software Foundation (ASF) under the Apache License, version 2.0 [39]. This framework is popular among developers and many applications are built using this framework successfully. It was originally developed by Nitobi and acquired later by Adobe in 2011 [40].

PhoneGap applications are built using web technologies such as HTML5, CSS3 and JavaScript which are linked to the native API specification with the help of PhoneGap API support. The code is highly compatible, easy to use and learn, follows W3C standards and it is highly flexible as a cross-platform development API.
Cross-Platform Solution for mobile application development

Aditya Polisetti

2014-11-10

PhoneGap is the most popular framework that supports a number of platforms. Accesses to hardware features in its applications do not merely look like native applications in terms of look and feel as the framework lacks themes or UI elements of a native application. But it still useful for a developer who would like to build hybrid applications with all the supported features in limited time and budget.

Appcelerator Titanium

Appcelerator Titanium [42] is another cross-mobile framework for applications. It was developed by Appcelerator Inc., and is free for both personal and commercial use under the license to Apache Public v2. This framework includes SDK and titanium studio which has an extension with Eclipse IDE. The SDK is interpreted with web languages like PHP and is used to write HTML, CSS and JavaScript files. Currently the application supports only iOS and Android platforms which is a disadvantage. The support for Blackberry OS is still developing and is in the beta phase. It supports the web standards and has very good user interface elements that will provide native look and feel. It has good integrated support of JavaScript plugins and the advantages of JavaScript frameworks can be used with care during the development.
The applications developed using this framework is not confined to mobile or tablet devices. It will also support a desktop version with web languages like PHP, Ruby and Python which can run on desktop platforms including Mac OS, Linux and Windows. The application performance and speed of this framework is high as the code is compiled into its native language.

**jQuery Mobile**

jQuery Mobile [43] is a touch based JavaScript mobile framework with a prime focus on the touch gesture and native look of the application. This is an extension of the popular jQuery framework optimized for mobile devices. This open source framework is distributed freely under the MIT license. The main feature of this platform is its compatibility with all the touch-enabled mobile platforms. There is a limited access to native features with this framework. It is optimized to use with other frameworks such as PhoneGap for better results. This framework is lightweight and optimized for better performance in mobile devices. This framework also includes theme builder in which themes can be built according to the platforms to enhance the look and feel of the application.

```html
<div data-role="page">
    <div data-role="header">
        <h1>My Title</h1>
    </div>
    <section data-role="content">
        <p>Hello world</p>
    </section>
</div>
```

*Code 2.8: Code snippet of jQuery Mobile with data-role*
This framework not only includes the transition and interactive effects of the UI but also consists of the libraries that include both JavaScript and CSS files. It is easy to understand and develop using this framework. Well-structured documentation is available online along with good community support.

Figure 2.19: jQuery Mobile basic format
**Sencha Touch**

Sencha Touch is another hybrid framework which relies mostly on the look and feel of the application. This framework also uses web technologies such as HTML5, JavaScript and CSS. It has two licenses, commercial software license (free) and open source license. The open source license is under the GNU GPL license v3[44]. This framework does not run in browsers such as Firefox or Internet Explorer as it is built to run on WebKit. As Sencha Touch focuses more on the look and feel of the application, it does not support most of the native access APIs. Creating applications by combining the likes of Sencha Touch and PhoneGap give better results. Sencha Touch provides a touch bundle service that helps to build mobile applications using the built in applications. This help in designing HTML5 prototypes with powerful code editors. The eclipse plugin makes it possible to run it in IDE and is used as a powerful debugging tool for building big applications [44]. It supports iOS, Android, Windows Phone and Blackberry OS.

### 2.6 Location based services

Locating a user’s position geographically and delivering services accordingly is called location-based services. Location-based service in mobile devices has become so popular nowadays with the increase in the support of technologies in mobile like the internet, GPS and GIS. According to Gartner, applications using location-based services was number two [45] on a list of the top ten consumer mobile applications in 2012 with its high user value. The services provided based on location can be adopted in an application itself such as emergency, location tracking, traffic, weather, news etc. There are many types of location-based services; each service provided depends on the type of application. Some of the types are explained here.

The location search service is used to discover any place, which is nearby such as restaurants, bus stops, schools etc. Some popular applications are Foursquare, Yelp and Neet.

The navigation service is used for directions, distance, street view etc., at the location. Google Maps, Bing Maps etc., provide us with this kind of service.
Social sharing services are used for public connection and to share our current position with friends or in public at for example Foursquare, Facebook etc.

Geo-tagging services such as tagging pictures and videos are used as a context by providing the location.

2.6.1 Libraries and frameworks
With the growth of advertisement in LBS applications, several libraries and frameworks emerged to use these services by adding support with the help of defined libraries, plugins and APIs. This section describes the geographical database of the locations and also the mapping libraries that support JavaScript.

Geographic database
In order to retrieve or implement a geographic information system, a database needs to be maintained, this in order to fetch the location from the query, maintain the updates of data, retrieving information, analysing data and storing location data such as name, postal codes, coordinates etc.

Geonames database
Geonames.org is a free geographical database and is available under creative common attribute license [46]. This database can be downloaded freely and it contains over 10 million of geographical names. This organization was founded by Mark Wick in Switzerland [46]. It provides web services in several programming languages and supports JavaScript. It includes geocoding, place search, postal code search, nearby places and reverse-geocoding. Geonames uses REST web services. These are available both in XML and Json formats. The client libraries have vast support for several existing programming languages and frameworks.

JavaScript Mapping Libraries
As discussed above, several JavaScript mapping libraries exists for LBS. Some of them are discussed below.
Google Maps

Google Maps API is the most popular mapping library service which is embedded and used in many websites. The Google Maps JavaScript places library [47] makes it possible to search for geographical locations. In order to access this library, there is a need to load the bootstrap URL of the maps API such as in the code 2.9.

```html
<script type="text/JavaScript"
    src="http://maps.googleapis.com/maps/api/js?libraries=places&sensor=true_or_false"></script>

Code 2.9: Code snippet of bootstrap URL for Google Maps Places
```

The API is not totally free as it varies with the usage. The users who use the services more than intended per map view, need to pay a certain amount and there will probably be advertising displayed.

MapQuest

MapQuest [48] is an online free mapping service which provides mapping libraries that include finding local information, geocoding, directions and street level for different countries. It also provides a mobile suite, MapQuest mobile, which is a cross-platform mobile web application. This application includes several features like place search, driving directions, location search, points of interest etc. A user can also personalize the user interface in MapQuest by using the "my maps personalization" feature [48]. The data in the database has two solutions. The first is licensed data, it provides commercial data resources and user support for mapping applications. The second is open data and it provides data from the OpenStreetMap [49] data source by using unique database API sets.

```html
<script
    src="http://www.mapquestapi.com/sdk/js/v7.0.s/mqa.toolkit.js?key=YOUR_KEY">
</script>

MQA.EventUtil.observe(window, 'load', function() {
    var options={
        elt:document.getElementById('map'), zoom:10, latLng:[lat:39.743943, lng:-105.020089], mtype:'map', bestFitMargin:0, zoomOnDoubleClick: 1;
    }
```
window.map = new MQA.TileMap(options);

});

</script>

Code 2.10: Code snippet - Basic MapQuest map setup [50]

**OpenLayers**

OpenLayers is a library in JavaScript, which is used to display maps dynamically in a web page. It is open source, free under the 2-Clause BSD license [51]. It has a strong JavaScript API which helps to build vast web-based applications in terms of location and geography. It is configured easily and can easily be integrated into applications with its simple JavaScript API. It has a vast data source support from Google, Web Feature Service, Microsoft, Yahoo etc. With its huge data source, it is not recommended for a mobile application to work with OpenLayers libraries in offline mode as it needs to access the data. It has nice graphical interface with popups and position markers.

**Leaflet.js**

Leaflet [52] is another open-source JavaScript library for maps that fit the bill. The light-weight libraries, easy customization, free plugins and friendly interface of the leaflet are the reasons for its increasing popularity. It is licensed under the BSD open-source license. Despite being light-weight, it includes several features for map, visual, interaction and controls. It supports CSS3 which makes customizing its graphical interface. The visual features run smooth even in HTML5 supported mobile devices.

Embedding Leaflet in a page is done by including the JavaScript and CSS files providing the code lines within the page and adding a JavaScript function to initialize the map in the page. The basic coding for Leaflet is shown below in code 2.11.

<link rel="stylesheet" type="text/CSS" href="leaflet.CSS" />
Code 2.11: Code snippet- Embedding Leaflet in a web page and rendering map
3 Methodology

This chapter describes the methods chosen for the proposed solution by discussing the pros and cons of the frameworks that are mentioned in the above chapter.

3.1 Considerations for the proposed system

A mobile classified services application is chosen and named Mobitrade. This service allows users to publish classified ads and to view them. The application needs to be designed keeping mobile users in mind which means the application must be user-friendly and it must enhance user experience. The application may be accessed by the user using his personal computer as well. The application needs to be analyzed, developed and deployed with the required functionality, features and general experiences of the location-based services and JavaScript frameworks. The user experience features such as icons, logo, widgets, and animation effects need to be consistent and effective i.e., the user interaction should be rich and smooth. The things need to be considered to develop this application is as follows.

- Using a feasible layout
  - Using a one column layout
  - Limiting the scrolling unless it is necessary
  - Visible text and graphical buttons
  - Application flow needs to be maintained.
- Must be a light-weight application.
- The application should run on major platforms. A desktop version would be an advantage.
- Rich UI presentation and it should interact with touch gestures.
- Access location of the consumer.
- The application must be user-friendly.
3.2 Modules in the application

This section describes the functionality of the proposed application, Mobitrade classified services. At a conceptual level, it has three roles:

- **Public Role:** In this role, the users are guests who browse the application and search for the classified advertisements that are available. A user can browse an advertisement from the displayed categories or by using the search functionality. The displayed results can be browsed in brief for the item cost, location and description. If the user is interested in the offer, s/he can show it by adding an interest button. By doing this, a pop-up window will be displayed prompting the user for details and asking the user for permission to access the location.

- **Publisher:** This role is for registered members. Any user can register in the application by submitting the registration form. By doing this, s/he is able to create a classified and to publish it. S/he can pin point his or her location or the item location on the map, which in turn saves the physical location of the address from the map pointed to. S/he can view the number of users who showing interest for that particular advertisement and the list of advertisements that s/he published.

- **Administrator:** The administrator role is to control the whole system right from configuring server, database, sending notifications, proof reading the content, editing, deleting, modifying users and classified ads. The administrator will monitor the content submitted by the publisher, reviews it and validates the classified. S/he can notify the publisher any modifications of the content.

3.3 Approach

This section describes the approach and the selection of several frameworks to develop the application in the three mentioned environments. The frameworks that are introduced in chapter 2 are studied and shortlisted making sure that the considerations are satisfied for developing the application. In the process of moving forward to build an application, a number of criteria is listed below keeping the considerations in mind.
3.4 Choosing the native development platform

This section describes the selection of platform to develop a prototype of native application.

As discussed in the above chapters, according to Gartner’s prediction for the future mobile market, the Android operating system devices are the most used all over the world with 49% of usage followed by Windows Mobile with 19% of usage [5]. This is represented in Figure 2.2. This would be an advantage as it would mean reaching almost half of the mobile users all over the world, if we develop the application in the Android mobile platform. So, the prototype of the application will be developed in Android platform rather than in iOS and Windows Mobile. The application needs to meet the above mentioned considerations. The requirement to design a layout of mobile classified application includes access with the GPS system of the device in order to know the user location.

3.5 Choosing the framework for hybrid app development

From the listed cross-platform tools mentioned in the chapter 2, PhoneGap is chosen for the development of a prototype for the following reasons:

- It was at the top of the frameworks, used by 34% of the developers [54].
- It supports seven platforms for development.
- Excellent community support with very good documentation and tutorials available online.
- Easy to develop.
• Free to use.
• Supports most of the native features of mobile devices especially in the top three mobile platforms.
• We can use other popular UI framework jQuery within PhoneGap for native look and feel.

3.6 Choosing the environment that satisfies the requirements
As discussed in chapters 1 and 2, a development environment is chosen by considering the factors which are listed in section 3.1 that satisfy the requirements of the proposed solution. The best suited environment to meet the above considerations will be developing in a cross-platform environment.

Figure 3.1: Choosing a mobile application development environment
3.7 Choosing the user interaction framework for HTML5 to develop a cross-platform application

To develop an application in the cross-platform environment, to the best HTML5 framework of those available must be chosen, while considering the application for desktop browser support.

There is a need to check a feasible way to use the application in desktop browsers. To achieve this, the application must be responsive which means the application can change its scale and limit its display depending on the size of the browser. Several frameworks are discussed in chapter 2 which incorporate responsive web designing. Among them, Twitter Bootstrap is chosen for the project based on its popularity, user support, availability, light-weight feature etc., the reasons are discussed in section 3.10.

3.8 Choosing a framework for location-based services

As the application needs to receive the user location, a suitable light-weight framework for location services and MAP user interface should be considered. Among the several frameworks discussed in section 2.6, Geonames is chosen as geographical database services embedded with GoogleMaps’ responsive JavaScript plugin, because of the advantages listed below:

- It is free of charge under a creative commons attribution license.
- The services run on faster servers and guarantee fast and reliable response time.
- It covers all countries’ geographical database information; over eight million place names.
- Light-weight and flexible when using the particular geographical information we need.
- It has libraries associated with reverse geocoding.
- Uses REST web services
- Uses JSON services which is helpful to use Geonames with JavaScript, as most browsers restrict the call of XML services from server to server.
- The service client libraries gives support to PHP, making it easy to develop the web app.
3.9 Choosing a JavaScript framework

The application needs to have professional look in terms of user interface and extensibility. The system needs a light-weight JavaScript framework which should have a wide range of plugins available for specific purposes. Light-weight framework is preferable. It should be easy to use and must have support for HTML5 and CSS3 markups. Keeping all these considerations in mind, jQuery and jQuery Mobile is chosen rather than other available frameworks. The reasons for choosing the jQuery framework are listed below:

- Large development community and wide range of plugins available.
- Easy to develop and understand.
- Numerous documentations and tutorials available.
- Supports Ajax.
- It has wide support in mobile devices and with HTML5 and CSS3.
- Better User Experience with its graphical elements and animations.
- Light-weight.
- Chaining capabilities are very powerful.

The reasons for choosing jQuery Mobile are listed below:

- Light-weight and the framework is especially optimized for mobile devices.
- Incorporates already designed widget elements, graphical elements, layout and control.
- Sleek animation effects that provide better user experience.
- Supports touch gestures.
- Navigation and transition effects provides a feel of using a native application.
- Less complex in terms of code and design.
- Large development community and several plugins available.
- Custom themes can be created by using Theme Roller.
3.10 Selecting a responsive design framework

Responsive web design is touted to be the current emerging trend, being used in all sorts of web development. It continues to draw designers’ attention with the motto one design for all devices. From the many frameworks discussed above, Twitter Bootstrap is chosen considering its advantages, which are listed below:

- Popular framework for responsive design and it is developed by the founders of the famous social networking website Twitter.
- Open source and is a popular repository in GitHub.
- Cross-browser compatibility.
- Reusability of components.
- Excellent documentation and online tutorials.
- Easy to follow and develop the code.
- Contains different web user interface components such as animations, sliders etc., as needed by the application.
- Includes a lot of HTML5 features such as typography, elements, forms and graphical buttons.
- Consists of several light-weight JavaScript frameworks such as tooltip, navigation, dropdown, accordion, carousel, image slider etc.
- Testing pages are very flexible.

3.11 Prototyping methods

The system needs to develop the prototype for the development environments. In this case, as we have chosen to develop the application using web technologies that support cross-platform functionality. The prototypes are designed for the native and hybrid applications. According to Stephanie Houde and Charles Hill, there are four dimensions of developing a prototype the role prototype, the implementation prototype, the look and feel prototype and the integration prototype [53].
The role prototype estimates, determines and implements the functionality and behaviour of the system.

The implementation prototype checks for the design flaws within the methods and implementation cost. It implements the methods and makes sure that the functionality works.

The look and feel prototype includes mainly the user interface interaction. For instance, for a website, it focuses mainly on the static web pages, their design, colours, images etc., and checks the support of the frameworks. The approach considered for prototyping the two environments is the third one, the look and feel prototype.

The integration prototype integrates the above three prototypes to build a complete system. The third environment, cross-platform development application, uses this approach.
4 Implementation

The design process enables mobile developers to proceed with developing either from any of the three development environment. This section is divided into subsections. First, we need to identify the requirements for the proposed system design followed by the task division of the system. The proposed system architecture is explained in the graphical layout of a flowchart in the below sections. The application subsection discusses the scope, structure and the functionality of the proposed application. The following subsections discuss the approach of each of the environments. As the thesis work focuses mainly on one chosen environment. The cross-platform, the look and feel prototypes of the remaining two systems include the basic structure, mockups and consideration of frameworks. The cross-platform development environment is focused and implemented in detail describing all steps of the app development. Finally, testing considerations are discussed with respect to the developed application and the results are shown in Chapter 5.

4.1 Identification of requirements

The requirement identification is the crucial part in developing this application. A table with requirements and a description is presented in Table 4.1. This will help to design, develop and test the product. In this case, we have already chosen the development environment, which is the cross-platform based mobile application.
### Cross-Platform Solution for mobile application development

**4 Implementation**

Aditya Polisetti

---

<table>
<thead>
<tr>
<th>S No.</th>
<th>Requirement Identification</th>
<th>Description of the requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Environment</td>
<td>The development environment is finalized for the application (native, web, hybrid) for mobile classified ads.</td>
</tr>
<tr>
<td>2</td>
<td>Prototype</td>
<td>The prototype scopes are defined for the remaining two development environment modules.</td>
</tr>
<tr>
<td>3</td>
<td>Server programming</td>
<td>A server language is chosen for the development. This includes the connectivity language PHP and a database, MySQL.</td>
</tr>
<tr>
<td>4</td>
<td>Frameworks</td>
<td>Frameworks have to be selected with respect to the proposed solution considerations and the developed environment.</td>
</tr>
<tr>
<td>5</td>
<td>Web markup</td>
<td>The application is identified to develop using web standards such as HTML5, CSS3, JavaScript, AJAX and JSON.</td>
</tr>
<tr>
<td>6</td>
<td>Graphic design</td>
<td>Icons, buttons and logos are designed according to the requirements.</td>
</tr>
<tr>
<td>7</td>
<td>CRUD</td>
<td>The application designs the basic four functions create, read, update and delete. According to the proposed design functionalities and follows the logic of the proposed system.</td>
</tr>
<tr>
<td>---</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>8</td>
<td>Location</td>
<td>The location access has to be checked with respect to HTML5 technology within a browser and is accessed after user’s permission and stored in the database for future usage.</td>
</tr>
<tr>
<td>9</td>
<td>Alerts</td>
<td>Alert messages such as forgot password and newsletter are to be included, as well as alerts for any problems occurring during the application usage.</td>
</tr>
<tr>
<td>10</td>
<td>Touch navigation</td>
<td>The application needs to be touch friendly and should fit in both orientations.</td>
</tr>
<tr>
<td>11</td>
<td>Online</td>
<td>The server needs to be online as the application does not work without the internet.</td>
</tr>
<tr>
<td>12</td>
<td>Media</td>
<td>The data needs to be uploaded from the file system to the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td></td>
<td>server. Also, the location of the user needs to be accessed and stored.</td>
</tr>
<tr>
<td></td>
<td>Search</td>
<td>Search functionality needs to be flexible and accurate.</td>
</tr>
<tr>
<td></td>
<td>Links</td>
<td>The links should not open in a new window. Make sure the flow of the application needs to be followed by eliminating the unnecessary popups.</td>
</tr>
<tr>
<td></td>
<td>Design</td>
<td>The design of the application should be pleasing and must be user friendly.</td>
</tr>
<tr>
<td></td>
<td>Animation/Styles</td>
<td>Use JavaScript and CSS UI frameworks whenever needed to enhance the application look and feel.</td>
</tr>
<tr>
<td></td>
<td>Map APIs</td>
<td>Choose the better API which meets the requirements of the proposed solution.</td>
</tr>
<tr>
<td></td>
<td>User testing</td>
<td>The system needs to be tested for the appropriate results to reach better user experience.</td>
</tr>
</tbody>
</table>

Table 4.1: Identification Requirements
4.2 Tasks to be accomplished

Each task can be a requirement or a group of requirements. The tasks of the application are mentioned in Table 4.2 for the cross-platform development application.

<table>
<thead>
<tr>
<th>S no.</th>
<th>Tasks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application landing page</td>
<td>Needs to maintain minimal design and ensure user interaction.</td>
</tr>
<tr>
<td>2</td>
<td>Database</td>
<td>Setup database tables and check server configurations and connections against the interface.</td>
</tr>
<tr>
<td>2</td>
<td>Search</td>
<td>Search for classified ads from database.</td>
</tr>
<tr>
<td>3</td>
<td>Location access</td>
<td>Access the user location with the help of location-based tracking mechanisms and frameworks.</td>
</tr>
<tr>
<td>4</td>
<td>Login</td>
<td>User authentication in the system.</td>
</tr>
<tr>
<td>5</td>
<td>Newsletters</td>
<td>Email weekly updates to the members.</td>
</tr>
<tr>
<td>6</td>
<td>Cookies</td>
<td>Cookies management from the client side.</td>
</tr>
</tbody>
</table>

Table 4.2: Tasks to be accomplished
4.3 Proposed system functionality

The goal of the system is to build a user-friendly classified application. This application covers the functionality of user authentication, search for an item, registration, accessing geolocation, pagination, cookies management, picking the address of a location using map interfaces, databases and libraries, responsive user interface with respect to the browser type/version and the basic CRUD operations. As the application needs communication between client and server, it uses PHP language as a server programming interface and MySQL as the database. As mentioned in Chapter 3, the proposed system has three modules, the end-user module, member module and admin module.

4.3.1 End-user module

This module comprises of a list of items that are to be selected by the user, which are then divided into categories. The search functionality is deployed in each page s/he visits and the user can use this feature to filter the items. The items are listed in categories and subcategories. Each subcategory is filtered by selecting the correct category of items. The displayed advertisement has two views, one to display the result of the search list and another to display the complete details. The result list needs to display the items entry date, image title, price and a description in a well-structured format. The pagination is mentioned after displaying a certain number of items, for example 12 items. The display of the advertisement page gives us detailed information of the classified which includes title, date, price, description, and address. The address will be in written format and on a map. If the user would like to show interest in that particular advertisement, s/he can push a button, “add interest”. After pushing the button, a pop-up is displayed asking the user to enter the details such as name, number, location and email id. This information is stored in a database table.
4.3.2 Member Module

The member module is for those who wish to publish classified ads in the application. For that the user needs to register and authenticate. After authentication, the member is redirected to the member’s home page, where s/he can publish advertisements and view the list of items s/he published and the status of the classified is shown in three sub types. If the status is ‘accepted’, then the classified is ready to display in the list of items and in the search list, if the status is ‘modify’, the classified needs to be modified according to the instructions given by the administrator. If the status is ‘delete’, then the item will be removed from the database. The publisher can add a classified by filling out the form which includes title, description, image, category, subcategory, price, currency and item location. A publisher can upload a maximum of three images from the storage. There are issues while accessing the data such as images of some devices using a web browser. Those are discussed in section 4.6.

4.3.3 Administrator module

The admin module manages the CRUD mechanism within the functionality which can create, view, update and delete the data as well as modify the database details. The admin can send weekly notifications to the members regarding status, user interests etc. The admin is responsible for the data within the advertisement and can send comments to a particular publisher in order to modify certain fields in the particular advertisement posted earlier.

4.4 Proposed system architecture

The architecture of the proposed system is divided into three parts. The first part is the designing of user interface and mobile user interaction. This includes the user perspective view within the mobile devices. The main assets in this part are HTML5 pages, logos, graphical buttons, CSS files, JavaScript files and images. The second part consists of the application specific logic. This includes the connectivity of native API with JavaScript files and the business logic. All dynamic files will be included in this part. This part also includes the application logic, framework files, system configuration and functionality. The third part includes the web server, the database handling, requesting web services, sending and receiving requests, creating tables, updating and modifying data.
4.5 **Application design**
In this subsection, the design process of the application, wireframes, flowchart and the implementation is discussed.

4.5.1 **Design process of the application**
The design process of the application is represented in Figure 4.2. It starts with research and literature study followed by the design of the system, implementation of prototype and finally the end-user application.
4.5.2 Wireframes

After planning the application design, some mock-ups are created in order to create the basic model of the application. The mock-ups are planned for both mobile view and desktop view of the application. This will follow the functionality of the proposed system. The mock-up model images are shown below for different categories and pages.
Design - mobile view

Figure 4.3 shows the model of mobile view of the home page.

![Homepage model design](image)

Figure 4.3: Homepage model design

Figure 4.4 shows the model for the classified view of the application.

![Classified view model design](image)

Figure 4.4: Classified view model design

Figure 4.5 gives the mock-up model for the classified details. It shows the complete details of a classified which includes price, description, show interest button and images followed by the map page which shows the address of the item.

Figure 4.5: Classified details model design
Figure 4.5: Classified details and model design

Figure 4.6 gives the mock-up model for the add interest page where the user adds interest after viewing the classified. The page is followed by the confirmation pop-up window. Both are shown here.

Figure 4.6: Add interest model design
Design- desktop view

The mock-ups of the application which can be viewed in the desktop browser are designed to present the basic idea of the application model in desktop mode.

Figure 4.7 gives the home page model of the desktop application.

Figure 4.7: Home page desktop model design

Figure 4.8 gives the publisher login page model of the desktop application. It contains all the necessary data and links are displayed without any confusion.
Figure 4.8: Publisher login desktop model design

Figure 4.9 gives the search page model of the desktop application. It contains the search results and panels text as well as links.

Figure 4.9: Classified details and model desktop design

Figure 4.10 gives the advertisement information page model of the desktop application.
Figure 4.10: Classified view model desktop design

Figure 4.11 gives the user interest model of the desktop application, which is self-explanatory.

Figure 4.11: Add interest model desktop design
4.5.3 Flowchart

The flow chart of the application model is shown in the figure 4.12.

![Flowchart](image)

**Figure 4.12: Flow chart of the system**
4.6 Implementation
Both the prototypes and the web application are developed under the Windows platform. The implementation of the system in different scenarios is discussed in this section. It includes the native, hybrid and web application development respectively.

4.6.1 Native application development
The look and feel prototype model of the native application has been designed. The prototype is developed under the Android development environment using development tools such as Eclipse IDE, emulator and all the essentials of Android SDK. The documentation from Android developer tools [10] is followed to set up the ADT bundle for Windows.
The development environment of the system is shown below. It includes IDE, system tools, graphic tools etc.

Operating System:
Windows 7- 32 or 64 bit

Android Development and IDE tools:
- Eclipse 3.6.2 (Helios) or greater [10]
- Java Development kit 6 (JDK 6)
- Eclipse Java Development Tool plugin
- Android plugin tools (ADT)

Screenshots
Screenshots of the system is shown in the following figures.
Figure 4.13: Home screen

Figure 4.14: Search items
Figure 4.15: Item details

Figure 4.16: Publisher login screen
4.6.2 Hybrid application

For developing the look and feel prototype of the hybrid application, the frameworks like jQuery mobile and PhoneGap is used to fulfil the requirements of a hybrid mobile application. The usage of the demos and docs from the official websites for both jQuery [43] and PhoneGap [55] helped to develop the prototype by using required JavaScript features and PhoneGap libraries for the application. The development environment of the system is shown below:

Operating system:

- Windows 7- 32 or 64 bit

Android development tools:

- Eclipse 3.6.2 (Helios) or greater [10]
- Java Development kit 6 (JDK 6)
- Eclipse Java Development Tool plugin
• Android plugin tools (ADT)

PhoneGap configuration tools:

• Latest PhoneGap API copy of Apache Cordova.
• JQuery Mobile API

Web server package:

• Xammp
  o Apache HTTP Server
  o MYSQL database
  o PHP

Integrated development environment (IDE) tools:

• Notepad++
• Aptana Studio 3
• Adobe Dreamweaver

Screenshots
The look and feel prototype of the hybrid application model is shown in the following screenshots.
Figure 4.18: Home screen

![Home screen](image)

Figure 4.19: Login screen

![Login screen](image)

Figure 4.20: Register screen

![Register screen](image)
4.6.3 Cross-platform application

The proposed system under cross-platform development has to implement the functionality which is discussed in section 4.3 (proposed system functionality). The system uses geonames web service for getting the address of a particular item or the address of the member. With the help of geolocation functionality in HTML5, the system gets the coordinates (latitude and longitude) which in turn, with the help of reverse geocoding concept, is managed to get the location. With the help of the database from geonames API, this is achieved.
The development environment of the system is shown below. It includes IDE tools, graphic tools etc.

**Operating system:**
- Windows 7- 32 or 64 bit

**Web server package:**
- Xampp
  - Apache HTTP Server
  - MYSQL database
  - PHP

**Integrated development environment (IDE) tools:**
- Notepad++
- Aptana Studio 3
- Chrome web browser developer tools
- Adobe Photoshop CS3

**Dealing with multi-platform development**
While trying to develop a mobile application with cross-platform capability, one should identify the differences and problems of cross-platform development to find a solution. The areas where the difference persists are as follows:

- Mobile screen and resolutions
- Content fit
- Browser compatibility
- Performance variance

**Mobile screen and resolutions**
The main difference in several devices will be the screen dimensions, size and resolution. As we know, developing mobile applications we are confined to a limited screen space, we must ensure that the application will fit for all conditions and all screen sizes. For iPhones, though they use iOS as their operating system, the screen size/resolution varies from model to model. For instance, iPhone 3GS has a 320 X 480 screen size, iPhone 4 has a 640 X 960 screen size and iPhone 5 has a 640 X 1136 screen size. It is necessary to design the user interface considering many mobile screens and resolutions.

![Different screens and resolutions of different devices.](image1)

**Figure 4.23:** Different screens and resolutions of different devices.

![Screen size and pixels per inch of different devices.](image2)

**Figure 4.24:** Screen size and pixels per inch of different devices [56].
**Content fit**

As the application is intended to be develop for mobile users, fitting text, images, and widgets to the screen is a must keeping the user experience in mind. We need to consider two things to solve this issue.

**Viewport**

In order to fix the content to fit, HTML5 has a new feature called viewport. This meta-tag is used to create a mobile friendly version of the application by confining the contents with respect to the device screen size. This meta-tag will be displayed within the header section of the markup script and is as follows:

```html
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Figure 4.25 shows the display of the content in different sizes and with different properties of viewport meta-tag in an iOS platform.
Figure 4.25: Viewport display of an iPhone with different settings.

a) Viewport with width set to 320
b) Viewport with default settings
c) Viewport with width set to 320 and scale set to 50%
d) Viewport with width set to 320 and scale set to 150%
e) Viewport with arbitrary user scale
There are several properties and values for the viewport meta-tag. Different platforms have different default viewport widths. Figure 4.26 gives us the default widths for the respective platforms.

<table>
<thead>
<tr>
<th>Phone Browser</th>
<th>Default Viewport Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opera Mobile</td>
<td>850 pixels</td>
</tr>
<tr>
<td>iPhone Safari</td>
<td>980 pixels</td>
</tr>
<tr>
<td>Android</td>
<td>800 pixels</td>
</tr>
<tr>
<td>Windows Phone 7</td>
<td>1024 pixels</td>
</tr>
</tbody>
</table>

Figure 4.26: Default viewport width of different phone browsers

Orientation
The orientation of a mobile phone changes from landscape to portrait. The system design should respond to the device with all the content within it to fit the screen. For this, responsive design is chosen, so that the application will change accordingly and is flexible in both orientations.
Browser compatibility

As the application runs through a web browser we have to make sure that the default browsers for the application supports its features. As the application needs to access user location information with the help of the HTML5 enabled feature the browsers must support this functionality. Among the top mobile platforms, the default browsers have the functionality support and using this application will not be a problem. The default browsers for Android, iOS and Windows phone are Android browser, Safari and mobile Internet Explorer respectively. The system needs to access the user data from the mobile storage, as the application asks the user to upload images. Both the Android and Windows platform supports this feature. There will be a problem in the iOS platform to enable this functionality of uploading user data from a mobile browser. For this, a third party free application is used in development known as Aurigma up [57].

Performance variance

As the application speed and performance depends upon the Internet connection and mobile hardware specifications, the performance varies from device to device and also the type of connection. Keeping this in mind, the system is developed as a light-weight application and the search functionality of the app is developed so that it should not be an overload for smaller configuration devices. The performance testing has been conducted for the same and will be discussed in Chapter 5.
4.6.4 Testing

While developing the cross-platform application for mobile devices, there is a need of testing the functionality and operations. As we know, deploying a developing Android mobile application is done in two ways. One way is to test it with the emulators that are integrated with the Android SDK and integrated development environment. The drawback of these emulators is that these cannot be tested real time, and if we want to access the sensor information, location tracking, etc. these are of little help. The second way is to deploy the application directly into the mobile device and check for testing cases. With this, it is possible to run it in the desired mobile devices where the app is intended to run. The problems that need to be addressed for testing purposes are mentioned below.

- The navigation should be visible to the user and should respond accordingly.
- Graphical elements and text alignment needs to be handled to ensure the look and feel.
- Orientation of screens needs to be taken care of for different screen sizes.
- User interface needs to be consistent in terms of widgets, buttons, keys etc. and the display needs to be clear.

It is necessary to choose appropriate browsers and test devices in order to test the key features that are necessary for the application to support for example accessing device storage files, accessing location tracking through a browser etc.

Mobile phone specific factors

We need to consider some device specific factors to test the application in order to make a good test device which the application is targeting. The factors are mentioned below.

*Touch screen:*

With the increasing popularity of touch screen devices and almost every device that associated with the top three mobile platforms have a touch interface, the application needs to be tested in a touch-enabled device.
Size and resolution:
Considering different devices with different screen sizes and resolutions will be a good choice for testing the application. This will check both the viewport of the web application as well as the representative orientation change of the device.

Performance:
The performance of the device depends on the hardware specifications such as memory, touch quality, and CPU. Devices are considered by selecting these specifications to vary.

Devices used for testing the application
Table 4.3 gives the devices that are used to test the application. As the application is cross-platform, the devices using Android, iOS and Windows phone OS are considered keeping the above factors in mind.

<table>
<thead>
<tr>
<th>Mobile</th>
<th>OS</th>
<th>CPU and Memory (RAM)</th>
<th>Display size</th>
<th>Display Resolution</th>
<th>ppi</th>
<th>Touch Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sony Ericsson</td>
<td>Android 4.0.4</td>
<td>1 GHz Scorpion, 512 MB</td>
<td>3.0 inches</td>
<td>320 x 480 pixels</td>
<td>192 ppi</td>
<td>Capacitive touchscreen, 16M colors</td>
</tr>
<tr>
<td>Xperia Active</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samsung Galaxy</td>
<td>Android 4.1.2</td>
<td>Quad-core 1.6 GHz Cortex-A9, 2GB</td>
<td>8.0 inches</td>
<td>800 x 1280 pixels</td>
<td>189 ppi</td>
<td>Capacitive touchscreen, 16M colors</td>
</tr>
<tr>
<td>Note 8.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samsung Galaxy</td>
<td>Android 4.1.2</td>
<td>1 GHz Cortex-A5, 768 MB</td>
<td>3.27 inches</td>
<td>320 x 480 pixels</td>
<td>176 ppi</td>
<td>Capacitive touchscreen, 256 colors</td>
</tr>
<tr>
<td>young</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Samsung Omnia</td>
<td>Windows 7.5 Mango</td>
<td>1.4 GHz Scorpion, 512 MB</td>
<td>3.7 inches</td>
<td>480 x 800 pixels</td>
<td>252 ppi</td>
<td>Super AMOLED capacitive touchscreen, 16M colors</td>
</tr>
<tr>
<td>Apple</td>
<td>IOS</td>
<td>600 MHz</td>
<td>3.5 inches</td>
<td>320 x 480 pixels</td>
<td>165 ppi</td>
<td>Capacitive touchscreen,</td>
</tr>
</tbody>
</table>
Cross-Platform Solution for mobile application development

Aditya Polisetti

4 Implementation

2014-11-10

<table>
<thead>
<tr>
<th>iPhone 3gs</th>
<th>6.1.3</th>
<th>Cortex-A8, 256 MB</th>
<th>640 x 960 pixels</th>
<th>330 ppi</th>
<th>Capacitive touchscreen, 16M colors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple iPhone 4</td>
<td>iOS 6.1.3</td>
<td>1 GHz Cortex-A8, 512 MB</td>
<td>640 x 960 pixels</td>
<td>330 ppi</td>
<td>Capacitive touchscreen, 16M colors</td>
</tr>
<tr>
<td>Apple iPhone 5</td>
<td>IOS 6.1.3</td>
<td>Dual-core 1.2 GHz, 1GB</td>
<td>640 x 1336 pixels</td>
<td>326 ppi</td>
<td>Capacitive touchscreen, 16M colors</td>
</tr>
</tbody>
</table>

Table 4.3: Properties of different devices used for testing system

Web application checklist

To determine the features, functionality, support and similarity a checklist is prepared. This will help to evaluate the progress whether or not the usage of web applications such as the proposed model might be a clear alternative for multi-platform deployed native applications. The checklist will make sure to find answers such as framework support, user-friendliness and performance. Table 4.4 lists these.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Checklist for web application correctness</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application landing page loads without any missing elements</td>
<td>☐</td>
</tr>
<tr>
<td>2</td>
<td>Application loads without problems and errors</td>
<td>☐</td>
</tr>
<tr>
<td>3</td>
<td>Application display fits in both orientations (viewport and landscape)</td>
<td>☐</td>
</tr>
<tr>
<td>4</td>
<td>Application text is well formatted and content fits within the screen size</td>
<td>☐</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>5</td>
<td>Application runs in offline mode</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Loads perfectly in the default mobile device browser</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Application has no broken links</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Navigation is identified easily and is displayed when needed</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Application flow is good</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Map display is fine and fits</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Number of pop-up windows have increased and are distracting the app flow</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Results displayed after using the search functionality</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Much usage of page scroll</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Buttons show an action when clicked</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Application user interface has similarities with that of native application in terms of looks</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Forms are well formatted</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Browser asking permission for location access</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Image carousel works fine in the classified details page</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Application uses camera</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Local storage (cookies) working well</td>
<td></td>
</tr>
</tbody>
</table>
### 4 Implementation

Cross-Platform Solution for mobile application development

Aditya Polisetti

4 Implementation

2014-11-10

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>User is able to pick up calls or messages in between the app use.</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Uploads images perfectly</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: Checklist for measuring the functionalities of the developed multi-platform mobile application
5 Results

This chapter will provide the results and solutions to the problems addressed in Chapter 1. This section is divided into two parts. Section 1 presents the research results which include the comparison of the frameworks, development platforms and the solutions to the objectives discussed in Chapter 1. Section 2 provides the development results of the cross mobile classified application which includes the application screenshots, UML diagrams if any, developer’s experience, performance testing, GUI testing and user testing.

5.1 Module 1: Research results

There are many reasons for having good support of HTML5 in all the recent smart mobile devices. Its mobile friendliness, light-weight effects, performance, modern UI elements in its GUI attracts the support from browser vendors of several platforms. From the study it is clear that the support of HTML5 in mobile devices is a necessary option for a successful cross-platform web application. So, to develop a mobile web application, the browser plays a prominent role as it must support HTML5. Several smart phone mobile default browsers support HTML5 and CSS3 which makes it easy to develop a cross-platform/web mobile application and this can be used as a desktop web application also. In order to have efficient user interaction for the application, the browser must have support for HTML5 and CSS3. For the application, a list of required features is listed below in Table 5.1 and the browser support is mentioned for the top four mobile platforms [58].
From the above table it is clear that there is almost no problem of using a mobile web browser for HTML5 supported cross-platform applications.

**Comparison of the three environments of mobile development**: Table 5.2 gives the comparison between the three environments. Each environment has its advantages and disadvantages. The choice of platform for development depends mainly on the requirements of the application. These are detailed in section 6 in Chapter 3.
## Cross-Platform Solution for mobile application development

Aditya Polisetti

### 5 Results

2014-11-10

<table>
<thead>
<tr>
<th>Factors</th>
<th>Native development</th>
<th>Hybrid development</th>
<th>Cross-platform development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development skills needed</td>
<td>High</td>
<td>Moderate</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Languages/tools</strong></td>
<td>Native SDK IDEs</td>
<td>HTML5</td>
<td></td>
</tr>
<tr>
<td>Java</td>
<td>JavaScript</td>
<td>CSS3</td>
<td></td>
</tr>
<tr>
<td>Objective-C</td>
<td>Frameworks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C, C++</td>
<td>JQuery Mobile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dot Net etc.</td>
<td>appcelerator etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Graphics(Animation, images, visual effects)</strong></td>
<td>APIs, Libraries</td>
<td>HTML5 Canvas, CSS3</td>
<td>HTML5 Canvas</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GIF, SVG, PNG etc.</td>
<td>CSS3 GIF, SVG, PNG etc.</td>
</tr>
<tr>
<td>Can be downloaded from App Store</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Application performance</td>
<td>Very Fast</td>
<td>Fast</td>
<td>Moderate (depends mainly on internet connection)</td>
</tr>
<tr>
<td>Look and feel of GUI components</td>
<td>Yes, native</td>
<td>Yes, native (always depends on the framework components)</td>
<td>Not native</td>
</tr>
<tr>
<td>Device access</td>
<td>Full</td>
<td>Full</td>
<td>Partial access,</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>------</td>
<td>---------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Geolocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Offline storage</td>
</tr>
<tr>
<td>Swipe touch gesture</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offline access</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
</tr>
<tr>
<td>Development time</td>
<td>Slow</td>
<td>Fast</td>
<td>Fast</td>
</tr>
<tr>
<td>Usage of app in other major platforms</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Development support</td>
<td>Yes</td>
<td>Depends on the framework chosen</td>
<td>Yes</td>
</tr>
<tr>
<td>Development cost</td>
<td>Very expensive</td>
<td>Bearable</td>
<td>Less expensive</td>
</tr>
<tr>
<td>Application Visibility</td>
<td>More, App Store distribution</td>
<td>More, App Store distribution</td>
<td>Only with mobile browser, bookmarks, icons will work.</td>
</tr>
<tr>
<td>Advanced Graphics</td>
<td>High performance</td>
<td>Moderate performance</td>
<td>Moderate performance</td>
</tr>
<tr>
<td>Maintenance and updates</td>
<td>More difficult</td>
<td>Difficult</td>
<td>Very easy to update and maintain</td>
</tr>
<tr>
<td>3D support</td>
<td>High</td>
<td>Limited</td>
<td>Limited</td>
</tr>
</tbody>
</table>

Table 5.2: Native vs hybrid vs web applications [59]

**Comparison of the JavaScript mapping libraries**

The table gives the comparison of the mapping libraries which examine the performance and the interface features that suits a mobile device the best.
Cross-Platform Solution for mobile application development

Aditya Polisetti

5 Results

2014-11-10

<table>
<thead>
<tr>
<th>Factors</th>
<th>Google Maps</th>
<th>MapQuest</th>
<th>Open layers</th>
<th>Leaflet.js</th>
</tr>
</thead>
<tbody>
<tr>
<td>Touch navigation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>View comparison</td>
<td>• Map</td>
<td>• Map</td>
<td>• Map</td>
<td>• Map</td>
</tr>
<tr>
<td></td>
<td>• Satellite</td>
<td>• Satellite</td>
<td>• Satellite</td>
<td>• Satellite</td>
</tr>
<tr>
<td>Mobile services</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mobile API pricing</td>
<td>Free for limited usage</td>
<td>Free and enterprise editions</td>
<td>Free and enterprise editions</td>
<td>Free and enterprise editions</td>
</tr>
<tr>
<td>Ease of development</td>
<td>Easy</td>
<td>Moderate</td>
<td>Difficult</td>
<td>Easy</td>
</tr>
<tr>
<td>API</td>
<td>Lightweight</td>
<td>moderate</td>
<td>Large</td>
<td>Lightweight</td>
</tr>
<tr>
<td>Template design</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>UI elements look and feel</td>
<td>Excellent</td>
<td>Moderate</td>
<td>Moderate</td>
<td>Excellent</td>
</tr>
<tr>
<td>Development community support</td>
<td>High</td>
<td>Moderate</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison of the JavaScript mapping libraries

**Comparison of the hybrid frameworks**

Table 5.4 gives the comparison of the hybrid frameworks in terms of supported features.
## Results

<table>
<thead>
<tr>
<th>Feature support vs framework</th>
<th>PhoneGap</th>
<th>Appcelerator Titanium</th>
<th>JQuery Mobile</th>
<th>Sencha Touch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform (Consider top four platforms, Android, iOS, Windows Phone and Blackberry OS)</td>
<td>Android iOS WP BB OS</td>
<td>Android iOS</td>
<td>Android iOS WP BB OS</td>
<td>Android iOS BB OS</td>
</tr>
<tr>
<td>Application type</td>
<td>Hybrid</td>
<td>Native Hybrid</td>
<td>Web app Website</td>
<td>Web app Website</td>
</tr>
<tr>
<td>- Native app</td>
<td>Hybrid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Hybrid app</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Web app</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Website</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language (Java, C, C++, C# PHP, HTML5, CSS3, JavaScript,)</td>
<td>HTML5, CSS3, JavaScript</td>
<td>PHP, HTML5, CSS3, JavaScript</td>
<td>HTML5, CSS3, JavaScript</td>
<td>HTML5, CSS3, JavaScript</td>
</tr>
<tr>
<td>Price</td>
<td>Free</td>
<td>Free</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>Documentation support</td>
<td>Rich support</td>
<td>Moderate support</td>
<td>Rich support</td>
<td>Moderate support</td>
</tr>
</tbody>
</table>
5 Results

Module 2: Development Results

Finally, the cross-platform mobile classified ad application has been developed with the implementation of all the features proposed for the application. This module gives the developed application, screenshots, performance analysis and testing.
5.2.1 The developed application

This section provides an insight of the developed application with its structure and the functions performed by the application.

Structure of the application

The application is developed under XAMPP Apache server with PHP as server connectivity technology and MYSQL as its database. The application comprises of the following folders and files which are shown in Figure 5.1. It includes folders “admin”, “CSS”, “fancypop”, “function”, “img”, “includes”, “js”, ”template” and “uploaded_file”. The admin folder has HTML, CSS, js and PHP files which deals with the admin functionality of the application. The fancypop JavaScript plugin helps for responsive a pop-up window which is used in the application. In the img folder there are images, icons, buttons etc. used for the application. The template folder consists of set of PHP files which are used repeatedly in other PHP files for the design of pages. The page structure is defined in these files. The uploaded_file folder has the images which are uploaded by the publisher in the process of creating a classified. The js file has all the JavaScript libraries and components which are used in the application. The CSS folder has ‘.css’ files which will define the style sheets for the development. This is responsible for the responsive design of the system.

![Figure 5.1: Project folders and files](image-url)
Functions performed by the application.

This section gives the system functions of the public user, member and admin. The end-user can use this application to search for a classified of choice and is able to show interest in any of the advertisement by providing details and allowing the device to access the location of the user using HTML5. The operations of the public user can be shown with the help of the following screenshot images of the application. The screenshots provided here are taken from Samsung Galaxy Note mobile device which runs on the Android platform.

Figure 5.2 gives the user accessing the category list displayed in the home page. After selecting the category the user is redirected to the subcategory of the selected list and the advertisements that are associated with the main category are displayed accordingly. It is shown here.

![Figure 5.2: Home page of the application](image-url)
After selecting the subcategory list from the dropdown menu, the list will be refined accordingly. The user can select any of the items that are displayed after navigation. The user can also search the item by using the search box which is present in almost every page of the application. Figure 5.4 provides the screenshots of searching an item, the display of the list and pagination according to it.

Figure 5.3: Selecting subcategory

Figure 5.4: Screenshots for search results
Figure 5.5 gives the advertisement description page after selecting an advertisement. It has add interest button, images of the classified which can be animated with the help of JavaScript plugins, a list of accordion text that includes the title, description and address of the item. The hyperlink map below the image divider helps to navigate smoothly towards the map location in the page. If the user shows interest in this classified, the user can click the button, add interest from the top which results in a pop-up window appearing which asks the user permission to access the location information. Then the user needs to enter the details which are stored in the database and the interest is registered.

![Figure 5.5: Screenshots of the classified display and submission of details](image)

The user can register in the database, in order to become a publisher. A publisher can create advertisements and publish it after the admin’s approval. The publisher’s access to the system will be shown in the screenshots. The registration page for the user is displayed in Figure 5.6. The validations of forms are carefully and successfully dealt with using JavaScript, PHP and Ajax technology.
After registering in the system, the page is redirected to the login page which is shown in Figure 5.7. The publisher can login into the system to perform actions like publishing an advertisement, checking the number of published classified ads, number of user interest, deleting a classified of choice, modify content when asked by the admin.
The publisher can create an advertisement by clicking the *create classified* page which redirects to the form page, as shown in Figure 5.8. After sending the form, the publisher needs to wait for admin’s approval and then the advertisement can be listed.
According to the system, a publisher can add a maximum of three images from the storage. Some browsers such as Safari will not provide access to the storage of a mobile device. Hence it will be difficult to upload any data from the device through a browser. In order to solve this, there are some free applications that provide an easy way to upload data from mobile devices to browsers. Aurigma up [57] is such an application and is free to use. So, in Apple devices, the ‘create classified’ page will be a little different from the others. This is shown in Figure 5.11. The links provide the application which is redirected directly to the appstore to download and install. If the application is already installed on the device, then the form page looks like Figure 5.10.

Figure 5.10: Screenshots of the classified creation
Until now, the results of the functionalities of public role and member role have been discussed. As per the proposed solution, there is a need for the application to have a desktop version as well. If the application needs to target PC users, then a desktop version is necessary. The screenshots in the desktop mode are shown in the figures shown below.

Figure 5.11: Screenshot prompting to install Aurigma up software for multiple picture upload from an iPhone mobile device

Figure 5.12: Screenshot of the home screen display
Figure 5.13: Screenshot of the desktop classified search page
The admin section needs to be discussed. This section controls the important functionalities of updating, deleting and adding user information. The admin section provides news digest with which automated emails are sent to the list of publishers in the database whenever the operation is performed. Screenshots are provided here.
Cross-Platform Solution for mobile application development
Aditya Polisetti

5 Results
2014-11-10

Figure 5.16: Screenshot of the admin home page

Figure 5.17: Screenshot of the admin classified list page
5.2.2 Analyzing performance

Performance analysis has been carried out for the application to provide an insight into meeting the requirements and monitoring the performance of the application. The performance has been analysed by using different tools. Tests are conducted in terms of validation, page size, load time, display in several devices, response time etc. using tools like mobileOKchecker [60], mobiready [61], screenfly[62], pagespeed, gtlspeed [63] and YSlow.

Testing with mobileOK checker:

The W3C markup validator is used to check the validation of web markup documents that include HTML files, JavaScript files and CSS files. The mobile validator is based on mobile best web practices [64]. The tests include the result score, the page size and lists the detailed report which is based on W3C basic tests [64].
Cross-Platform Solution for mobile application development
Aditya Polisetti

5 Results
2014-11-10

Testing with MobiReady:

Mobiready is a popular online testing tool which is used to test the page, markup and site test. This tool determines the mobile friendliness of a web app. It will check the web app based on its screen presence. It will give us a rating on a scale of 1 to 5, regarding its overall score for a better mobile web application. It will also estimate the readiness, size and speed of the mobile application. The test results are shown in the figures given below. Figure 5.20 gives the overall rating and it is 4 for the application which means the design displays well on a mobile phone. The readiness score is 3.64 out of 5. The size of the page is 0.2k and the speed details are estimated which includes Wi-Fi, 3G and GPRS respectively as shown in the figure. The average load time from Wi-Fi, 3G and GPRS is 1.35 seconds.

Figure 5.19: Testing with mobileOK checker

Figure 5.20 (a): Testing with MobiReady
Testing with Pingdom Tools

Pingdom Tools [65] are used to test the speed of the web application by choosing servers among the three provided. Figure 5.21 gives us the performance grade, load time and the overall page size which includes the files and images of the home page. The performance grade is 79 out of 100 and is tested from the Netherlands server.

Testing with Mobitest

Mobitest [66] is an online tool which measures the performance results of a mobile web app in terms of its display, load time and page size. Also, it simulates the view of the page in the mobile we selected. It is possible to run the test up to three times simultaneously. The results tested from the tool are given below.
According to this tool, the average load time of the entire website is 1.4 seconds and the average page size is of 626.76 kb including images and files after performing tests with three times run on an iOS device.

Figure 5.22: Simulator view of the application in an iPhone 4 mobile device

<table>
<thead>
<tr>
<th>URL</th>
<th>Load Time (ms)</th>
<th>File Size (kb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>332</td>
<td>626</td>
</tr>
<tr>
<td>style.css</td>
<td>261</td>
<td></td>
</tr>
<tr>
<td>jquery-1.6.4.min.js</td>
<td>258</td>
<td></td>
</tr>
<tr>
<td>custom.js</td>
<td>265</td>
<td></td>
</tr>
<tr>
<td>main.js</td>
<td>261</td>
<td></td>
</tr>
<tr>
<td>jquery.Tweet.js</td>
<td>264</td>
<td></td>
</tr>
<tr>
<td>superfish.css</td>
<td>260</td>
<td></td>
</tr>
<tr>
<td>hoverIntent.js</td>
<td>205</td>
<td></td>
</tr>
<tr>
<td>supersub.js</td>
<td>219</td>
<td></td>
</tr>
<tr>
<td>jquery.prettyPhoto.js</td>
<td>261</td>
<td></td>
</tr>
<tr>
<td>prettify.js</td>
<td>269</td>
<td></td>
</tr>
<tr>
<td>tip-twitter.css</td>
<td>392</td>
<td></td>
</tr>
<tr>
<td>tip-yellow.example.css</td>
<td>305</td>
<td></td>
</tr>
<tr>
<td>jquery.amplify.js</td>
<td>476</td>
<td></td>
</tr>
<tr>
<td>fontslider.css</td>
<td>483</td>
<td></td>
</tr>
<tr>
<td>favs.googleapis.com.css</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>modernizr.js</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>js.../slider.min.js</td>
<td>157</td>
<td></td>
</tr>
<tr>
<td>leqlogo.png</td>
<td>194</td>
<td></td>
</tr>
<tr>
<td>reot.css</td>
<td>194</td>
<td></td>
</tr>
<tr>
<td>comments.css</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td>elements.css</td>
<td>202</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.23: Waterfall chart, load time of the files in a scale of 1.4 seconds.
5.2.3 **Browser compatibility display test**

Using two different tools, tests have been conducted to check the browser compatibility in terms of display of the page. These virtual tests are really helpful as it might be difficult and time consuming to test the application for display in each and every mobile device that it supposes to run, as the size varies for most of the devices. The mobile phone emulator [67] and screenfly [62] are such tools that are used for browser compatibility tests.

**Tests using MobilePhoneEmulator tool**

The MobilePhoneEmulator tool has its own emulator environment that includes monitor size, cell phone terminal, website to emulate sections that will direct us to perform operations of our choice. The results are shown in the following figures after the test. We can directly check the entire website within the emulator area. And hence it will be really helpful to check the entire content and area that suits best virtually in a device.

![Toolbox for emulator system](image_url)

*Figure 5.24: Toolbox for emulator system*
Figure 5.25: Emulator display for the devices iPhone 5, HTC touch and Samsung Galaxy Spica.

Tests using Screenfly

Screenfly [62] is another online emulator tool used to test mobile layout and its display in several chosen mobile devices. This tool has all types of simulator devices for desktop systems, tablets, mobile devices and televisions. A user is able to view the application by specifying the size of the choice. With all the available mobile devices, desktop and tablets, the application is displayed perfectly. The following figures give the application compatibility display of some mobile devices.

Figure 5.26: Testing web app using Screenfly
5.3 User Testing

A real context testing has been proposed in the application. User testing considered user perspective in the form of a checklist provided in Chapter 4. The tests are conducted in desktop, tablet and mobile devices. The test outcomes of the checklist are shown in a way that if a certain task fails to act as per the intention in any one of the mentioned devices, the task is resulted to fail. The browsers used for testing are all updated versions till date. The list of devices used for testing is shown in Table 5.5.

<table>
<thead>
<tr>
<th>S No</th>
<th>Testing Devices</th>
<th>Browsers used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Samsung Galaxy Note 8.0</td>
<td>Android browser, Chrome</td>
</tr>
<tr>
<td>2</td>
<td>Samsung Galaxy Young</td>
<td>Android browser, Chrome</td>
</tr>
<tr>
<td>3</td>
<td>Samsung Omnia</td>
<td>Firefox, Internet Explorer for mobile</td>
</tr>
<tr>
<td>4</td>
<td>Apple iPhone 3gs</td>
<td>Safari browser, Chrome</td>
</tr>
<tr>
<td>5</td>
<td>Apple iPhone 4</td>
<td>Safari browser, Chrome</td>
</tr>
<tr>
<td>6</td>
<td>Apple iPhone 5</td>
<td>Safari browser, Chrome</td>
</tr>
<tr>
<td>7</td>
<td>Sony Ericsson Xperia Active</td>
<td>Android browser, Chrome</td>
</tr>
<tr>
<td>8</td>
<td>iPad 2</td>
<td>Safari browser, Chrome</td>
</tr>
<tr>
<td>9</td>
<td>Personal computer</td>
<td>Firefox, Internet Explorer, Chrome</td>
</tr>
</tbody>
</table>

Table 5.5: Devices used for testing
The results of the checklist are presented in Table 5.6.

<table>
<thead>
<tr>
<th>SNO.</th>
<th>Checklist for web application correctness</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application landing page loads without any missing elements</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>Application loads without problems and errors (check internet connectivity)</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Application display fits in both orientations (viewport and landscape)</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>Application text is well formatted and content fits within the screen</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>Application runs in offline mode</td>
<td>✗</td>
</tr>
<tr>
<td>6</td>
<td>Loads perfectly in the default mobile device browser</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>Application has no broken links</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Navigation is identified easily and is displayed when needed</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>Application flow is good</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>Display of Maps is fine and fits</td>
<td>✓</td>
</tr>
<tr>
<td>11</td>
<td>Number of pop-up windows have increased and are distracting the app flow</td>
<td>✗</td>
</tr>
<tr>
<td>12</td>
<td>Results displayed after using the search functionality</td>
<td>✓</td>
</tr>
<tr>
<td>13</td>
<td>Limited page scroll</td>
<td>✗</td>
</tr>
<tr>
<td>14</td>
<td>Buttons show an action when clicked</td>
<td>✓</td>
</tr>
</tbody>
</table>
User interface has similarities with that of native application in terms of look and feel

Forms are well formatted

Browser prompts permission for location access

Image carousel works fine in the classified details page

Application uses camera of the device

Local storage(cookies) working fine

User can able to pick up calls or messages in between the app usage.

Uploading images without fail

### Table 5.6: Results of the checklist

Form validation is dealt with using JavaScript and PHP scripts. Tests have been made for form validation also. The screenshots of these are presented in the following figures.
Figure 5.27: Form validation for registration page

Figure 5.28: User login validation screenshot

5.4 Functional testing - test cases

**Test 1:** Make sure all the links are working and redirecting properly.

**Steps:**

- Open home page.
- Click on each hyperlink.
- Repeat the same for every available page.

**Desired outcome:**

The application needs to redirect the desired page to be opened without any errors.
Table 5.7: Functional test case 1

**Actual outcome:**
The application redirected the desired page to be opened without any errors.

**Test 2:** Check that all the images and content are loaded correctly without missing and the images fit perfectly.

**Steps:**
- Open home page.
- Check for missing elements such as images and also check for formatted text.
- Repeat the same for each available page.

**Desired outcome:**
The application needs to load the images and formatted text properly in each and every page.

**Actual outcome:**
The application loads its images and text is formatted properly in each and every page.

Table 5.8: Functional test case 2
**Test 3:** Check the search bar for a desired classified and pagination of it.

**Steps:**
- Open home page.
- Type anything you wish to check using the search bar in the top menu.
- Check that the results are reasonable for the search item.
- Repeat by searching with an empty text.
- Observe the number of results.
- Try to open the next page from the pagination tag below and check if it works.
- Try to click any of the resulted items and check if it works without problem.

**Desired outcome:**

The application needs to perform perfectly the above steps in terms of searching the item and retrieving the results and also the links of the searched results.

**Actual outcome:**

The application just worked fine passing all the steps mentioned above.

Table 5.9: Functional test case 3
### Test 4: Form validation and authentication test

**Steps:**

- Open login page.
- Try to login.
- Enter wrong credentials and check the validation.
- If login is successful, try to log out and again open the login page.
- Try to click the “forgot password” button.
- Now enter an incorrect email id and check for validation.
- If the above steps work fine, try to enter your registered id and check the inbox from the admin for the password.

**Desired outcome:**

The application needs to validate the form and the email alert must work without problem.

**Actual outcome:**

Everything mentioned in the steps above worked without problem.

---

**Table 5.10: Functional test case 4**
**Test 5:** Test for the publisher, try to add a new classified

**Steps:**

- Login into the system.
- Click, *add a new classified* link.
- Fill out the form.
- Check the status (pending, modify, approved).
- If the status is pending, wait for approval. If it is approved, check for the classified from the search functionality. If it is modified, check the message from admin, modify it accordingly and submit again.

**Desired outcome:**

The application needs to have mandatory fields such as images, location, name and price.

**Actual outcome:**

Everything went out smooth and perfect. There were no problems with the above steps.

| Table 5.11: Functional test case 5 |

5.5 **Summary and analysis of the results**

As per the proposed solution and research topic, using web development technologies such as HTML5, CSS3 and JavaScript is crucial for the paper. As a cross-platform mobile application, HTML5 has been chosen, followed by selecting platform. As the native applications will not have to use HTML5 for its app development, this web markup technology is used in hybrid app and web app development.
After choosing the platform for the development, the development of an application will follow considering the requirements of the proposed solution. The proposed application mainly focuses on the browser compatibility and functionality of HTML5 for mobile devices. After checking the browser compatibility with several devices on the top mobile platforms, the application to develop is considered. A choice is made for the selection of best framework among the many available for user interface and experience. Responsive design has been chosen considering its facts and advantages of a multi-platform and cross-browser use. As the proposed system needs to access location and maps, Google Maps is chosen from the selected list of APIs. Geonames is used for the location database. These all are discussed briefly in Chapter 3.

The look and feel prototype for both native application and hybrid application is designed in Android platform. The frameworks PhoneGap and jQuery mobile are chosen for designing the hybrid app prototype. For the cross-platform application a responsive framework, Twitter Bootstrap is chosen. The application is developed using the web technologies PHP, jQuery, Ajax Technology, MYSQL and CSS3 and is tested using several tools. User testing is also performed in the application which is mentioned in chapters 4 and 5.

The overall support of HTML5 features and its performance is not completely dealt with in this thesis work. The further support test of HTML5 in mobile devices will be investigated completely in future work. Though handling a responsive design is a difficult task, it is handled well with the help of frameworks that are chosen for the development. The same applies for the programming module of the classified app using PHP and MYSQL, as the server functionality code is implemented perfectly using these technologies. All the database tables are well created and the classified ad system functionality is dealt with carefully and effectively throughout the project.
After the completion of the development work, tests are made to check the overall performance, areas of improvement, user experience, and functionality. According to the conducted tests, the system works fine for a mobile web application. The resulting application meets expectations. The application areas to improve will be the optimization of images to make it light-weight. The proposed solution deals with lots of images to be uploaded from the user-side, image optimization needs to be considered for mobile devices. Several tools are used to test the performance, and the user experience using emulators of the mobile devices. Few of the mobile devices are used for user testing.

The research which was carried out to study the possibility of migrating native mobile applications to web applications and the feasibility of choosing HTML5 as a standard for mobile application development, resulted in successful solutions. An application is proposed to work in accordance with certain considerations which are mentioned briefly in the previous chapters. This resulted in examining the different types of application development environment which in turn lead to choose a platform for development. A study is carried out to select libraries for maps and HTML5 frameworks. The application has been tested in several mobile browsers to ensure compatibility and support. The developed application also passed user tests ensuring the user experience. This thesis work ensures the extension of HTML5, CSS3 and JavaScript to their next level as a mobile development framework along with its subsidiary web development technologies.

5.6 **Future work**

The developed application could be subjected to future work. As the application is developed for the Android platform only, there is a possibility of developing the native and hybrid prototypes for other platforms as well. The developed application has advanced functionality for a classified application. Security in any application is a plus and there is room for the application to provide security for attacks like sql-injection, phishing, cross-site scripting etc. The features that can be added in the future are listed here.

- Multi-language system
• Usage of PHP frameworks such as code-igniter, cake PHP etc. for better security results and plugins
• Utilizing the advantages of jQuery mobile and PhoneGap totally for a hybrid application
• Possible usage of geo tagging features for member and public user
• Adding payment system for a publisher to view the user’s interested advertisement and integrating payment gateways.
• Adding additional features for publishers to create and edit their own profiles.
6 Conclusions

With its universal acceptance and the recent successful trends in the mobile platform, the mobile application market has become an important for the mobile platforms to extend their limits. Many mobile platforms are evolving continuously. Due to this, the need for universal accessibility has been up for discussion which leads to the evolution of multi-platform mobile application development. This needs special attention due to all its advantages. The open standards that are utilized for development will be the main factor for its success.

As the overall aim of the thesis is to study, seek reasons to suggest and develop the development environment of a mobile application using web technologies, the research has been carried out for the development of cross-mobile applications. The study focused on HTML5, a standard markup language for the web. It includes the additional interaction of the technologies such as cascade style sheets, JavaScript and plugins. The study suggests that the support of these web technologies has been growing in mobile devices day by day. There will be a definite improvement in the time to come. Every mobile company and platform is considered to have the support of these technologies in their devices. As most of the people prefer surfing the web through mobile devices, a special focus and attention has been given the development and support of these web technologies in smart mobiles. This strategy aimed for increased development of cross-platform mobile applications as they have many advantages compared with native applications.

The browser compatibility and support is recognized as the major issue for HTML5 support and its related technologies. The study provides satisfactory results for the support of this technology in major mobile browsers. The major target will obviously be the default browsers on devices. This automatically means there is a good support of HTML5 and related technologies in default mobile browsers of the top platforms (Android, iOS, Windows Phone and Blackberry OS).
6.1 Discussion
The solutions provided for some of the research objectives are given here.

Q: What are the main benefits of using web technologies as a standard for cross-platform application development?

As HTML5 is emerging strongly with its support and features, there are several benefits of using web technologies as a standard. For an application to develop some of the major constraints such as low cost of development, community support, ease of development and the development time are considered. These are all achieved by developing an application in HTML5 and web technologies which makes it the most popular thing helping to migrate native apps to cross-platform apps. Almost all kinds of hybrid frameworks such as PhoneGap, Sencha Touch, jQuery mobile etc. use HTML5 extensively, making it even more popular in application development, ensuring cross-mobile compatibility.

Q: What problems arise in cross-mobile application development?

The main problems that arise during the development of a multi-platform application are:

- Size of the screen as it varies from device to device.
- Screen orientation as the application layout needs to behave according to the orientation.
- Content fit, as the text of the application should be clearly visible and should be formatted keeping several resolutions in mind.
- Response time, it should be fast. When the user tries to click any button or link the response time from the system should be there should be some significant.
- Avoiding horizontal scrolling and minimizing virtual scrolling.
- Easy to use ensuring mobile user experience.
- Performance varies from device to device.

These problems are identified and the application is dealt with accordingly.
Q: Can the usage of HTML5 and related web technologies minimize the implementation cost and provide effective user interaction regardless of mobile platforms?

Using HTML5 which is an open source for application development automatically minimized the cost of implementation. With the help of robust frameworks, we can even try to achieve the native feeling for an application regardless of mobile platform.

Q: Which environment best suited for the development of the proposed solution?

As the proposed solution needs to access location of the device, there has to be a desktop version and the application in several platform, Considering its budget constraints, web application has been chosen in addition to native and hybrid applications. Because of the constraints, hybrid mobile app also suits the proposed solution except for there being a need to have a desktop application too. Hence, cross-platform web application environment is chosen for the proposed solution.

6.2 Ethical considerations

The proposed solution prompts the users to share their location access, which puts the users’ privacy at stake by sharing their current position in the application. As the proposed solution is with cross-platform development and suggests hybrid or cross-platform development environment in the future generation mobile apps, it would be possible to decline the developers who used to code native applications. As most of the development environment deals with open source plugins and libraries, there will be a possibility to raise security issues such as phishing etc. Encryption standards need to be considered as most of the application communicates with the flow of data from a client to server and vice versa, as the system is hosted in a private server.
From this study, it is clear that there are quite a number of reasons for a developer to prefer HTML5 and cross-mobile platform as their development environment. The code-portability, low development cost, cross-platform flexibility, application reach, open source of technologies and good support community for development were among these reasons. There, however, a need for HTML5 technologies to improve on a few areas in order to compete with the native mobile application development. The improvements include the optimized devices for HTML5, debugging and error recovery support, access to the native libraries and APIs, libraries to access a few other hardware devices of the mobile. If these improvements are considered and implemented in the coming future, HTML5 and other web technologies will remain at the top of development environments for developers deriving all the positive outcomes as a benchmark tool for application development.
References


Cross-Platform Solution for mobile application development  
Aditya Polisetti  

References  
2014-11-10  


Cross-Platform Solution for mobile application development
Aditya Polisetti

References
2014-11-10


[27] W3C- HTML5 Canvas, http://www.w3.org/HTML/wg/drafts/2dcontext/HTML5_canvas/)


[37] Responsive Grid System- http://responsive.gs/

[38] HTML5 Boiler Plate http://HTML5boilerplate.com/


[41] PhoneGap build- https://build.PhoneGap.com/

Cross-Platform Solution for mobile application development
Aditya Polisetti

References
2014-11-10


[56] Adobe, different screen sizes of mobile devices http://www.adobe.com/devnet/devices/articles/design_tips_mobile_ria.HTML


[58] IDC forecast – world wide quarterly mobile tracker report, http://www.globalnerdy.com/2012/05/07/the-Windows-phone-
predictions-that-idc-gartner-and-pyramid-research-probably-hope-youve-forgotten/

[59] HTML5 compatibility testing http://mobileHTML5.org/


[64] Best web practices , http://www.w3.org/TR/mobileOK-basic10-tests/

