Armoneam Tesfaye

Using PhoneGap to Create a Wish List Application

Helsinki Metropolia University of Applied Sciences
Bachelor of Engineering
Information Technology
Thesis
21 April 2017
The aim of this thesis project was to explore different mobile application development approaches and give more emphasis on a hybrid mobile application development process. The hybrid application development approach was compared to the native application development and web application development approaches and the advantages and disadvantages of each approach were explored.

The development framework that was used was Apache PhoneGap, which is an open-source mobile development framework. The user interface was built using HTML, CSS and JavaScript programming languages.

The outcome of this project was a hybrid application which is capable of storing user data on the local storage of a device. The application uses features such as camera and social share and it can be installed on multiple platforms.

From this project it can be concluded that hybrid application development has remarkably changed the field of mobile application development. The benefits and drawbacks of a cross-platform application over their rivals were discussed.

The application that was developed for this project can be further improved by the use of a remote server to host user data and implementation of unique user profiles for users, so that it will be possible to access stored information from another device.

| Keywords                      | hybrid application, Android, iOS, Apache Cordova, PhoneGap, mobile platforms |
7.1 Testing Using Emulator
7.2 Testing Using a Real Phone
  7.2.1 iOS Device
  7.2.2 Windows Mobile Device
  7.2.3 Android Device

8 Result and Discussion
  8.1 Positive Outcomes
  8.2 Challenges
  8.3 Future Improvements

9 Conclusion

References

Appendices
Appendix 1. Code Listing from config.xml
Appendix 2. Code Listing from index.html
Appendix 3. Code Listing from camera plugin camera.js
1 Introduction

The time is for mobile devices. Mobile devices are the new normal in our life affecting every aspect of it. There are different mobile devices ranging from mobile phones and tablets to wearable devices. The popularity of these devices is growing, and the number of users around the world has increased dramatically in the last decade. All these mobile devices have something in common: they all have applications running on them so that they will be easy and efficient to use.

Among mobile devices the most popular one is a smart mobile phone. Smart mobile phones can be used to make calls, send text, take pictures, shop and virtually most of the things that are possible to do on the internet. Even though all smart phones have a similar purpose, they have different operating systems. The different operating systems have different ways of working and are built using different programming languages making the applications installed on them to have different source codes.

The fact that different operating systems demand different source code has created a challenge for application developers to develop applications separately for each operating system. This process demands a longer time for application development, additional costs for companies and an upkeeping challenge for developers. Many companies want their applications to get to as many users as possible, which means a need for their applications to be available on as many platforms as possible. This causes additional costs for the development and the maintenance of the applications.

Thus, the challenge has created the solution of developing applications which can run on different platforms. These applications are called hybrid applications. Even though hybrid applications have the advantage of running on many platforms, they also have their drawbacks compared to native applications. In this thesis, I have explored different platforms, different application types and finally the development process of a hybrid application.
2 Mobile Phone Applications Development

Mobile applications development is the process of developing applications software which is specifically designed for mobile operating systems and intended for handheld devices such as mobile phones or tablet computers [1]. Mobile users use mobile applications to access many different services online for their day-to-day activities. Mobile applications are preferred to be used instead of access through mobile browsers, because of their ease of use and more user-friendly user interfaces.

The number of mobile phone applications has risen meteorically in the last few years thanks to their popularity and the rise of the number of mobile phone users. The growth of the number of mobile users has been significant while the number of desktop users has been in decline. Mobile technology trends report from KPCB (Kleiner Perkins Caufield & Byers) that the time users spent on mobile phones is outstripping the desktop time by a 9% margin. That is the time spent on mobiles is 51% compared to the 41% of the time that is spent on desktop computers. [2]

![Time Spent per Adult User per Day with Digital Media, USA, 2008 – 2015YTD](image)

Figure 1. Time spent on devices compared. Reprinted from Meeker (2016) [2]

As Figure 1 shows the amount of time spent on mobile devices by mobile users has grown significantly in the USA through the past few years compared to the time spent on using desktops or laptops.
3 Mobile Phone Platforms

A mobile phone platform also known as a mobile phone operating system is a collection of data and software instructions on mobile phone hardware. It manages the hardware and enhances the ability of mobile applications software on the hardware. It controls and manages the different functions of the device such as multimedia, internet connectivity, Bluetooth connectivity and several phone accessory connectivity.

Table 1. Different mobile phone platforms. Reprinted from IBM (2012) [f13,6]

<table>
<thead>
<tr>
<th>Language</th>
<th>Apple iOS</th>
<th>Android</th>
<th>Blackberry OS</th>
<th>Windows Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td>Objective-C, C, C++</td>
<td>Java (some C, C++)</td>
<td>Java</td>
<td>C#, VB.NET and more</td>
</tr>
<tr>
<td>Tools</td>
<td>Xcode</td>
<td>Android SDK</td>
<td>BB Java Eclipse Plug-in</td>
<td>Visual Studio, Windows Phone development tools</td>
</tr>
<tr>
<td>Packaging format</td>
<td>.app</td>
<td>.apk</td>
<td>.cod</td>
<td>.xap</td>
</tr>
<tr>
<td>App stores</td>
<td>Apple App Store</td>
<td>Google Play</td>
<td>Blackberry App World</td>
<td>Windows Phone Marketplace</td>
</tr>
</tbody>
</table>

Table 1 shows the different mobile platforms and the corresponding programming languages associated with them. It shows also tools for development as well as the App stores where the applications are available.

3.1 Android OS

The Android operating system was invented and developed by Android Inc. and was later acquired by Google Inc. in 2005. Google in turn came together with other companies to form the Open Handset Alliance (OHA), which is now responsible for the ongoing development and growing of the Android operating system. [3.]

The Android OS is based on a Linux kernel and its code is written in Java for user interface, C programming for core programmes and C++ programming language. The software code is under the open-source software license. According to Android Inc. Android OS powers hundreds of millions of handheld devices in more than 190 countries. It is the largest installed operating system and the fastest growing platform [4].
3.2 iOS

Apple Incorporation’s iOS is a mobile operating system which is designed and developed by Apple Inc. for its own hardware mobile devices. It is currently deployed on devices such as iPhones, iPads and iPods. It is the second most popular platform next to the Android platform.

According to Apple Inc. 27th WWDC in 2016 there are more than two million applications in the iOS App store and those apps has been downloaded more than one hundred thirty billion times since the app store was launched more than ten years ago in 2007. The iOS has up to seventeen million registered developers for its platform making it the second popular platform for developers. The software is a proprietary software which is written in C, C++, Objective-C and Swift. It has gone through different versions and the latest one is the iOS 10 [5].

3.3 Windows Mobile OS

The Windows mobile OS is a platform which is developed by Microsoft based on the Windows CE kernel, and is designed and implemented in a way that the mobile device will operate and feel like a Windows desktop operating system [6,229-231].

The first Windows mobile operating system started in the year 2000 for the device called pocket PC. It was named “Windows mobile” for the first time in 2003. Windows Phone, the successor of the Windows mobile, which was designed to target primarily the consumer market unlike its predecessor which was targeting the business market.

Windows Phone OS was superseded by Windows 10 Mobile in 2015. The Windows 10 mobile tends to give more attention to the unification of the Microsoft’s PC and mobile user experiences and platforms. The software is written in C and C++ and its applications run on the mobile .NET version.
3.4 Blackberry OS

The Blackberry operating system for mobile phones is developed by Blackberry Limited or also known as Research in Motion Limited (RIM) for its mobile hardware devices. The operating system is written in C, C++ and Qt and its kernel is based on QNX Neutrino RTOS kernel.

The software is proprietary software for Blackberry smartphones. The latest version is called the Blackberry 10 and was released in 2016 [6,253-256].

3.5 Firefox OS

The Firefox mobile operating system is an open-source licenced operating system which was developed by Mozilla. It is based on the Linux kernel and the rendering power of the Firefox web browser. It is known for using open standards and highly utilises the standards of JavaScript and HTML5. The software is written in HTML5, CSS, JavaScript and C++.

The Firefox mobile was first officially released in the year 2013 but in December 2015 Mozilla announced that it will stop the development of smart phones [7]. Moreover, in September 2016 Mozilla announced the end of the mobile OS software development [8].

3.6 Sailfish OS

Sailfish OS is a Linux-based operating system developed by Jolla, the Sailfish and Mer-project communities and various open community members. The user interface was developed by using QML (Qt Modelling Language). [9]

Sailfish operating system has also the ability to run Android Applications making it more flexible and easy to use. Since the Sailfish’s libraries are based on the Android libraries, making it almost identical to run Android applications on Sailfish compared to running Android native applications on Android operating system that they were designed for [9].
3.7 Ubuntu touch OS

The Ubuntu touch mobile operating system is the mobile version of the Ubuntu operating system. It was designed to target smart phones which are capable or have the feature of a touch screen. It uses the Qt5-based touch user interface [10].

The Ubuntu touch is developed by Canonical Ltd. and uses Linux kernel making it easy to be implemented on many Android devices. Being open-source software and high flexibility in the programming language makes the Ubuntu touch an attractive choice to many developers and users [10].

The above mentioned platforms are not the only ones which have been developed through the time of mobile OS platform developments. It is good to note that how they are different and vary in many aspects, which makes it hard to choose one platform to develop an application. It will be time and money consuming to develop an application which can run on all platforms. In the future, I believe that the technology will come together to allow an application to run on all platforms without the need to worry about platform-specific issues.

4 Mobile Applications

Mobile Applications also known as Mobile Apps are application software programmes which are either pre-installed or later installed programmes which run on mobile phones so that users will be able to use their handheld devices to the fullest potential or to different purposes. Most mobile apps have a desktop counterpart. Even though most mobile Apps have the same purpose as their desktop version, it does not mean that they have the same way of working. In mobile phone devices the screen area is much smaller than the traditional desktop screen, making it more difficult for users to precisely touch the target that they want if the links or the buttons are very small and very close to each other. [11,41-44.] The simplicity that comes with the limitation of the screen makes mobile Apps easier and simpler to use, making them more user-friendly in this respect.
Figure 2 shows the pre-installed native Applications for an Android and an iPhone mobile phone platforms.

4.1 Web Applications

Mobile web applications are applications software which are developed using web technologies. Even though they are used on mobile devices they are not hosted or installed on the mobile device rather they are hosted on a remote server and served to the user using standard protocols such as HTTP [12]. Mobile Apps typically run on web browsers and might differ how they look from one browser to the other because they are platform dependent. Those platforms support different versions of the HTML5. [11,41-44.] The most important feature that these web applications give to the user is that they tend to give the feeling and look of being the same with native apps, making web applications easier to use and more user friendly than their desk top counter parts.
As can be seen from Figure 3 A, the web app version of the online retailer company eBay is much simpler compared to the desk top version of the website. But on the negative side, the web app version might be difficult to navigate since all the navigation links are not upfront like the desktop version in Figure 3 B. Since the code that is used to write web apps adapts a standardized programming language, it makes it easier for fast development, easy maintenance and overall low cost for developers and owners [13].

4.2 Native Applications

Native applications are applications that are specifically programmed to run on a specific mobile device platform. They have binary executable files that are downloaded to the device to be stored locally [13,2]. Native applications are usually downloaded from an online application store such as Apple’s App store, Android’s Marketplace or by other means such as directly from online, for example from the application developer’s site. Typically, native applications have the advantage of access of the APIs (Application Programing Interface) that are available from the operating system [13,3].
Since native application platforms differ from one another there is a negative side to developing a native application. The code which is written to develop a native application for one platform cannot be used to develop the same application for another platform, thus making the developing and the maintenance of a native application an expensive and lengthy process.

4.3 Hybrid Applications

Hybrid Applications are applications that are built with the blend of web technologies such as JavaScript, CSS, and HTML5. Their main difference from web applications which use the same technology is that they are nested inside the native application container that will allow them to access the whole set features that the device offers to native applications [13,6].

![Figure 4. Application configuration. Reprinted from IBM (2012) [13,7]](image)

Figure 4 shows different configuration styles of applications that will help them utilize the device features. Application developers have the choice to either develop their own channel between the browser and the device’s APIs or use some ready-made solutions that are on the market. For example, PhoneGap is an open source JavaScript library interface that can be used in the development of hybrid applications so that they can access native device features that are not available for simple web applications. For this thesis project, I have used the PhoneGap interface to develop my own hybrid application.
4.4 Comparison Between the Application Types

In a comparison between these three approaches, one can see that every approach has advantages and disadvantages or limitations. Choosing the right option might depend on the project type, financial limitations, time limitations and many other technical requirements. This section will try to show the advantages and disadvantages of each approach.

Table 2. Comparison of application types. Reprinted from IBM (2012) [13,8]

<table>
<thead>
<tr>
<th>Feature</th>
<th>Native app</th>
<th>Hybrid app</th>
<th>Web app</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development language</td>
<td>Native only</td>
<td>Native and web or web only</td>
<td>Web only</td>
</tr>
<tr>
<td>Code portability and optimization</td>
<td>None</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Access device-specific features</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Leverage existing knowledge</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Advanced graphics</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Upgrade flexibility</td>
<td>Low (Always by way of app stores)</td>
<td>Medium (Usually by way of app stores)</td>
<td>High</td>
</tr>
<tr>
<td>Installation experience</td>
<td>High (From app store)</td>
<td>High (From app store)</td>
<td>Medium (By way of mobile browser)</td>
</tr>
</tbody>
</table>

In the evaluation of Table 2, native application development is preferred when there is already the knowledge of the native application development language, the target of only single operating system, need of particular native functionality and requirement of high user interface like games [13,8]. Web application development on the other hand is preferred when requirements such as development of a pilot app is needed, visibility in search engines is important and when the need to avoid third party application stores and instead directly distribute the applications for the users is needed but at the same time lacks the use of native features that the device offers for native applications. [14.] A hybrid application development approach combines the best of the two approaches by combining the ability to use the devices native features while at the same time using technologies which are used to develop web applications. Hybrid applications have a
very bright future since the use of HTML5 is rapidly growing and the capability is also developing making it very attractive and simpler for developers. The capability of developing an application which can run on multiple platforms is very appealing for anyone who is constrained by financial limitations.

5 Technology Implemented

For this thesis project, I have developed a simple hybrid application which can run on multiple platforms such as iOS, Android and windows. It has been tested for the time being on these three platforms and works properly. In this section I will discuss the technologies I have used to develop my hybrid application.

5.1 HTML5

According to W3C (World Wide Web Consortium) the world wide web’s mark-up language has always been HTML [15]. HTML supports DOM APIs and many tools that support it. Because Html was developed from the beginning by many people through a long period of time, it has many features that have arisen from those different sources.

HTML5 is the latest advancement in the development of the HTML [16]. The idea of HTML5 denotes two various concepts which describes it very well. The first one is that it has new elements, attributes and behaviours making it the latest version of HTML. The second is that it has and supports larger sets of technologies, which makes it able to allow more various and powerful websites and web applications. [16.] HTML5 has added more mark-up tags than its predecessor HTML4 such as <header> and <footer> tags to separate the top and bottom of content blocks, <article> tag to identify a specific content, <nav> tag to identify navigational blocks, <section> tag to represent a specific section, <video> and <audio> tags represent the addition of video and audio content, <canvas> tag to support the graphical drawing from other scripts and <embed> tag to insert external applications and inputs into the webpage.

Because of HTML5s’ improvements from the previous HTML4, its efficiency and improved performances makes HTML5 the future of web browsing. Some of HTML5 benefits include pages will load faster, less bandwidth will be used and energy consumption will be more effective letting devices have longer battery time.
5.2 CSS3

CSS3 (Cascading Style Sheet) is the latest version the cascading style sheets formatting language. It is the latest improvement on the previous version (CSS2.1) [17]. It is used to describe the appearance and the formatting of a document such as HTML5.

There are many improvements which are incorporated in the development of CSS3. These include

- Colour
- Selectors Level 3
- Namespaces
- Media Queries

CSS can also be used on documents such as XML (Extensible Mark-up Language), XUL (XML User Interface Language) and SVG (Scalable Vector Graphics) documents making it a versatile and very useful technology.

Figure 5. CSS3 implementation as a separate file.

The main purpose of a CSS document, as shown in Figure 5, is to separate the document content from the way it is going to be presented, making it possible to have con-
tent page which is simple to read and maintain. Another purpose is to increase recyclable codes that can be shared, for example one CSS file can be used by more than one HTML pages. This will help in reducing the complexity of the document and the repetition that might have happened on multiple documents. The CSS specification has a priority scheme to regulate style rules when there are more than one rules which might conflict or stipulate to a single element. [18,7-9] Other factors which are affected by using CSS file include loading time, lower bandwidth and avoidance of bloated HTML codes.

CSS3 can also render multiple background images, box shadows, text shadows, hoover state animations and round corners. To sum up, by using CSS3, we can make documents load faster, use fewer resources, make documents easier to maintain and make them easy to change or modify for the future. Alongside HTML and JavaScript, CSS3 is the foundation of modern day mobile applications, web applications, dynamic and responsive websites that are developed [18,7-9].

5.3 Apache Cordova / PhoneGap

Apache Cordova is an open-source mobile application development framework [19]. It allows its users to use web standard technologies such as HTML5, CSS3 and JavaScript for the development of applications. Applications are executed in wrappers which are designed to target specific platforms and use standard APIs to access device’s features and capabilities such as camera, vibration, GPS, sensors etc. Apache Cordova is recommended for developers who want to make an application which is capable to run on multiple platforms, or developers who have the ability and knowledge only to develop web applications but want to develop an application which would be available in native application stores, and developers who want to develop applications which are able to mix features of web applications and native applications. [19.]
There are several sections of Cordova Application. The above figure shows a view of the Cordova Application Architecture. The application launches in a WebView within a native application wrapper [19]. In this container there is an important file (config.xml), which is responsible for affecting how the application is going to work.

Among important parts of the Cordova system are the plugins [20]. They are the bridge between the Cordova application and the native components to bind together and access to the standard device APIs. The Apache Cordova project team by itself develops and maintains a set of plugins (Core plugins) which are used to give access of device capabilities such as camera, sensors and contacts, to the hybrid application. Even though core plugins are developed by the Apache team, there are many third-party plugins that can be used on applications to access device capabilities.

A basic hybrid application consists of configuration files, Icons for the application and content of the application which is built using web technologies. The configuration file
consists of all the necessary instructions to compile the application. All the contents of configuration file are wrapped in `<widget>` tag. Various options such as full screen, background colour, orientation is declared using `<preferences>` tag. Any application can be designed based on the need on internet access. There are offline applications, online applications and applications which combine both. Offline applications are downloaded entirely on the device and do not need internet access to operate. These applications store offline content on the device so that there is no need of an internet connection to access the necessary content.

![Folder structures of offline and online application.](image)

It can be seen that in Figure 7 the data folder is not necessary in Online Apps because all the files and contents needed are stored in remote servers but not on the device. For this thesis project, I have chosen to develop an offline application which will use local storage to store user data.
5.4 JavaScript Language and Libraries

JavaScript Language is a high-level programing language which is used to enhance web pages [21,3-4]. The three languages of HTML, CSS and JavaScript create the bases for modern-day web page development. JavaScript is becoming so popular that all modern web browsers support it. Because of this popularity, there is no need to use any plugins when implementing JavaScript on websites. Because JavaScript is already supported by most popular operating systems.

As Figure 8 shows JavaScript works hand in hand with style sheets and marked up content to create the best possible user experience and dynamic web content. Even though JavaScript has many uses and advantages over other programming languages, it has its limitations. As a programmer, it is good to know the advantages and the limitations. JavaScript is better to use when, to get webpages to respond directly to the user, providing small database content is a user-friendly way, controlling different APIs in HTML documents, processing user data before sending to server, dynamic changing of content and style according to user preference and requesting files from server.
vaScript limitations include initiating an application on the user’s computer, writing or reading files on the user’s computer, writing on files which are located on the server, sending secure emails from users to administrators. [21,12-13.] Even if there are some similarities between Java and JavaScript, they are both different programming languages.

5.4.1 JQuery Library

JQuery is a JavaScript library which is small, fast and multi feature which makes HTML handling interesting and easy. It has high adaptability and extensibility [22]. Using JQuery enables easy handling of animations, event handlings and Ajax, JSON parsing and multi browser support.

JQuery library is a single file which is written in JavaScript programming language. There are two ways of adding JQuery library to a project. It is possible to include it either by linking the webpage to a local copy of the JQuery file or it is possible to use a remote copy of the file which can be located on a server anywhere in the world and can be accessed by using https protocol.

```html
<script src="jquery.js"></script>
Listing 1. Local JQuery file
```

```html
<script src="https://code.jquery.com/
    jquery-3.1.1.min.js"></script>
Listing 2. Remote JQuery file
```

While Listing 1 shows the code that is used to state a local JQuery file to be included in a web page, Listing 2 shows the code used to include a remotely located JQuery file. JQuery’s Architecture permits the development of plugin codes by developers. Currently there are many plugins created for jQuery library [22]. These plugins are developed for many different purposes and cover different functionalities. They can be easily found on the jQuery project website.
The jQuery Mobile is the touch-optimised version of jQuery library which is currently being developed by the jQuery project team. It is developed based on the core jQuery making it easy to learn for someone who has prior knowledge of jQuery.

5.5 Ripple Emulator

A mobile application development process has a completely different path compared to the desktop version. While desktop applications are developed on a device which they are going to be installed, the mobile versions are developed on desk top machines and tested on mobile devices later. This process makes it challenging to test mobile applications side by side on the process of developing them. Because of these challenges, emulators play an important role in the development of mobile application.

Ripple Emulator is one of the most popular emulators which supports different platforms such as Android, iOS and Blackberry. It gives developers the ability to automatically test applications without the need to use a mobile device and restart the emulator, since it supports live automated refresh capability. The Ripple emulator operates as a Google chrome browser extension and therefore there is the need to install a Google chrome web browser to use the Ripple Emulator. [23,45-48.]

![Ripple Emulator extension in a chrome web store.](image)

As Figure 9 shows Ripple Emulator is available in chrome web store’s extensions section and can be easily added to the browser as an extension. After it is installed, it will be available by a button which will launch it. In Figure 9, it can be seen how the emulator will be available to users. There will be a button in the right corner of the web browser after installation.
Figure 10. Chrome web browser with installed Ripple Emulator

Figure 10 shows how it is possible to launch the Ripple Emulator. The platform gives different options to choose from, so that it is possible to get the option of available mobile platform frameworks. Figure 11 shows the current available mobile platform frameworks supported by the Ripple Emulator.
The reason for me to choose the Ripple Emulator was the fact that it supported Apache Cordova / PhoneGap mobile framework as shown in Figure 11. Ripple has many features such as setting geolocation, shaking the device, device orientation and different screen-size mobile devices. The downside of using emulators is that there is no guarantee that a tested application on the emulator will perform in the same way as it did in the emulator when it was installed on the real device.
6 Implementation

In this chapter I will discuss the application that was developed for this thesis project. The name of the application is Bucket List. Its main purpose is to provide a platform for users, where they would be able to save their wishes and keep track of them if they have been achieved or not. In the event of a wish being accomplished, the application will give the chance to take a memento picture and the ability to share that achievement with the rest of the world through either social media, email and text messages.

The first step in this application development was the installation of the PhoneGap (Apache Cordova) framework. Before the installation of PhoneGap, there are two basic requirements that need to be in the development workstation. Node.js and git are prerequisites for the PhoneGap application to work properly. Node.js is a JavaScript runtime environment which is used to build JavaScript code. Git on the other hand is used by a command line interface to download resources.

Figure 12. PhoneGap initial page
Figure 12 shows the start page of the PhoneGap application which can be used to develop cross-platform applications. After installation, the project can be created by using the ‘Create a new PhoneGap project’ button. It will automatically suggest a list of simple templates or a blank project for the use of the new project. The next step will be providing the correct pathway for the project and the name of the project.

Figure 13 shows a sample project created on PhoneGap. It shows the name of the application and the path of the project where the files are located. It also shows that the server on the application is running and the server address which can be used to connect with a device to test the project and shows the PhoneGap application for iOS which can connect with the desktop version enabling live testing of the applications developed without the need to install the application on the device.
6.1 Description

As the name of the application describes, the application is used for the purpose of saving users’ bucket list of wishes in order for it to be easily saved, edited and easily accessed for referencing. The application has features such as the ability to take a memento picture when a certain wish is accomplished and save it in the pictures gallery or in the camera roll. In addition to this it gives the opportunity to directly share the achievement to social media or directly through email or a text message.

Figure 14. Overview of Bucket List cross platform application

Figure 14 shows the overview of the application developed. The figure which is on the left shows the vertical view of the application while the other one shows the landscape portrayal of the Application. As it can be seen the application is simple and user-friendly. The addition of new wishes is easily achieved by the use of the add a wish button. The user interface is designed by using the HTML code and rendering it using a CSS file.
Listing 3: Code snippet from HTML file

Listing 3 shows part of the code used to implement part of the front page of the application. It shows how the form was implemented, which will help in the retrieving of data which is stored in the local storage of the device.

6.2 Main Components

The Bucket List application has four main components that are displayed to the user so that it will be used for the intended purpose. They are simple and easy to use for any user whether they have been using it for a while or using it for the very first time. In the following part I will discuss these components in detail.

6.2.1 Add a Wish

The Add a Wish feature is used to add new wishes to the bucket list. The added list is stored in the localStorage facility of the device. It is key-value pair database stored locally. In this storage we can store any string and that data will be available on that de-
vice later on. The downside of this method is that the data stored on this device will not be available from any other device. For this application, I have used a single key in the local storage and the value will be a stringified version of the wish list items that are stored.

```javascript
wish.add = function(event) {
  // Read the task from the input
  var wish=$('#input').val();
  if (wish) {
    // Add the task to array and refresh list
    wish.list[wish.list.length] = wish;
    wish.refresh_list();
    // Clear the input
    $('#input').val('');
  }
  event.preventDefault();
};

// Store back the list
localStorage.todo_list = JSON.stringify(todo.list || []);

// Load the list by parsing the JSON from localStorage
todo.list = JSON.parse(localStorage.todo_list);
$('#add').bind('vclick', todo.add);
$('#task_list').on('vclick', 'li a', function() {
  todo.selected = $(this).data('task');
});
```

Listing 4 Code snippet from wish add function

Listing 4 code snippet shows the way how a new wish was added and the method used to store the added wish. JSON.parse was used to convert the JSON string back to JavaScript data and return that to the front page of the HTML.
6.2.2 Wish List Update

After exploring the Add a Wish feature comes the Wish List Update Page. Its use is simple and easy for the user. It is accessed by clicking on the arrow sign which is found on every wish list component. It gives the choice of updating the wish list by marking the selected wish as either Achieved or Not Achieved. It also gives the option of removing the wish completely from the wish list.

When the user updates a wish to be achieved, then the application will give a congratulatory message and an option to take a memento picture to remember the achievement or to share it to others. After this the wish will be removed from the wish list and the updated list will load on the front page of the application. In the event the user clicks on the Not Achieved button, the application will give an encouragement message and keep the wish in the wish list.

```
<div id="confirm" data-url="confirm" data-role="page">
  <div data-role="header">
    <h1>WISH LIST UPDATE</h1>
  </div>
  <div data-role="content">
    Mark this Wish as<br>
    <a class="remove_task" href="#done" data-role="button" data-icon="check" data-theme="f">Achieved</a>
    <a href="#notdone" data-role="button" data-icon="delete" data-theme="g">Not Achieved</a>
    <a class="remove_task" href="#index" data-role="button" data-icon="alert" data-theme="f">Remove Wish</a>
  </div>
  <br>
  <a href="#index" data-role="button" data-icon="minus">Cancel</a>
</div>
```

Listing 5 Code snippet of Wish List update
Listing 5 demonstrates the code used to implement the Wish List Update page of the application. In case of an accidental click or change of mind by the user, there is a button for cancelling and it goes back to the main page without the need to update the wish list.

The screen shot picture of the wish update list page is demonstrated in Figure 15. It can be seen that in both orientations the user interface remains simple and attractive for users. A gradient colour background was used to give the darker blue colour to the top and the lighter blue colour to the bottom of the application.

6.2.3 Take a Picture

The Take a Picture feature is used to take a souvenir picture to mark the moment or to share it with others. One of the core plugins for PhoneGap framework is a camera plugin. It gives access to hybrid apps to the device’s camera features.
The plugin is added by using a command line interface directly or via repo URL. It can be added by the following command.

```
$cordova plugin add cordova-plugin-camera
```

```
$cordova plugin add https://github.com/apache/cordova-plugin-camera.git
```

Listing 6 Cordova camera plugin installation

Listing 6 shows the command line that can be used in the installation of the Cordova camera plugin. After the installation, the implementation was performed by using the codes shown in Listing 7.

```javascript
capturePhoto: function(){
    navigator.camera.getPicture(onSuccess, onFail, { quality: 50,
        destinationType: Camera.DestinationType.DATA_URL,
        saveToPhotoAlbum:true });

    function onSuccess(imageData) {
        var image = document.getElementById('image');
        image.style.display="block";
        image.src = "data:image/jpeg;base64," + imageData;
    }
    function onFail(message) {
        alert('' + message);
    }
}
```

Listing 7 Cordova camera plugin installation

As Listing 7 shows, the camera.getPicture function opens the device’s camera for it to be ready to take a picture. Listing 7 shows that I have used the Base64 encoding of the image data and the image taken to be saved in the photo album by using the saveToPhotoAlbum:true parameter.
Figure 16 shows the before a picture is taken and after use of the take a picture feature. The left one shows before the souvenir picture was taken. On the right side, it shows how the picture that was taken will be shown to the user.

Listing 8 code snippet for camera and picture display

Listing 8 shows the code implementation of call for capturePhoto function and the image display section. <img> tag is used to determine the size of the thumbnail picture on the screen.
6.2.4 Social Sharing

The Social sharing plugin for this project was used from a third-party plugin developer. The social share plugin allows the use of native social sharing window of the mobile device. The plugin was installed by using the following command line interface command.

```javascript
window.plugins.socialsharing.share(message, subject, file, url, [successCallback],
{console.log('result: ' + result)}
[errorCallback]
{alert('error: ' + result);
});
```

Listing 9 Native social share plugin [24]

As Listing 9 shows it is possible to share a different combination of data from the application. Even though there are different possibility options, the only necessary for my application was to share an image accompanied by a message. The other parameters are not necessary for this project. Figure 17 shows the application using the native share sheet.

![Figure 17 Native social share sheet](image.jpg)

As Figure 17 shows I have used the native social share sheet instead of a plugin developed by a third-party. In PhoneGap plugin archives there are many social share plugins developed by third parties.
7 Testing

Mobile application development is considered very challenging amongst many developers. Mobile environment and devices bring more challenges compared to the actual development of the application itself. [25,213.] In order to check the functionality of my application it was tested on two environments; on a real phone and in a simulator or emulator environment.

7.1 Testing Using Emulator

Testing using an emulator is easy and gives different options to test different features. For this project, as I have mentioned in earlier chapters, I have used the Ripple Emulator to test the application. Ripple Emulator supports different screen sizes and different operating system mobile devices.

As Figure 18 shows the application worked as expected on the many platforms available that it was tested on. It was possible to test it on operating systems such as iOS, Android, Windows and BlackBerry. Even though testing on an emulator is easy, it has
drawbacks for testing. For example, it was not possible to test the camera feature of the application on the emulator because it does not support that feature.

7.2 Testing Using a Real Phone

To test the application on a real device the application must be built into installation package by Adobe PhoneGap build.

As Figure 19 shows an application developed by using HTML, CSS and JS is built into an installation package which can be installed on different operating systems. After this process, the application can be installed by downloading on the test device or by a simple scan of a QR code of the app which will automatically install the application.

Figure 19 PhoneGap Build. Reprinted from Apache (2015) [19]

Figure 20 PhoneGap Build result
The results of the build process from the application can be seen from Figure 20. The apk file for Android and the appx file for a Windows mobile phone are ready to be installed or published in their respective application stores. When it comes to the iOS installation file it shows an error because I was not a subscribed Apple developer so I could not provide a signing certificate and keychain pair in my application.

7.2.1 iOS Device

Because I do not have an Apple developer account, it was not possible to directly install the application on an iOS device but it was possible to use a PhoneGap developer application for iOS, which will allow to run the application on an iOS device. On this test the Application performed as expected and all features were working. Most of the screen shots in the above chapters were taken from the application working on an iOS device. The device I used was iPhone 6s with iOS version 10.2.1.

![Figure 21 Bucket List Application on a iOS mobile device.](image)

Figure 21 was taken as a demonstration of Bucket List running on an iOS device. It fitted the screen properly and the fonts were rendered correctly.

7.2.2 Windows Mobile Device

With a Windows mobile phone there was the same problem as in the case of iOS and it was not possible to install the application without a developer account. Although I was not able to install it, I was able to use a PhoneGap developer application to run the app on a Windows platform. On a Windows phone the application was tested and per-
formed as it should without any major issue. The device I used was Microsoft Lumia 650 with a Windows 10 mobile operating system.

![Figure 22 Bucket List Application on a Windows mobile device.](image)

As Figure 22 shows the application was performing as expected. There was some speed issue detected but it was not that significant.

7.2.3 Android Device

Installation on an Android device was the easiest from the three devices chosen for this project. Since the operating system is open-source software it was possible to install the Bucket List Application without any problem. All it took was the QR code from PhoneGap build and then the application downloads and installs automatically. The device I used was Samsung Galaxy Ace 3 with Android version 4.2.2.

![Figure 23 Bucket List Application on an Android mobile device.](image)
From this test, as can be seen from Figure 23, it was evident that the font size was very small compared to what I had expected. Other than the font issue, the application and all its features were performing as expected.

8 Result and Discussion

In this section I will discuss the positive outcomes I gained from this project, challenges I faced and finally improvements that can be done to this application.

8.1 Positive Outcomes

The result of this project was the development of the Bucket List Application using the hybrid mobile application development method. The project was a successful one and it has helped me to know and to familiarize myself with the application development process. Using the PhoneGap development framework has helped me in making the development process easy and getting a cross-platform application without the need to write the code for each platform. I have come to understand the future is bright when it comes to HTML5’s versatility and powerful features, making it very attractive for developers around the world. The benefit of cross-platform applications is not only for developers, but also for companies who are concerned about the cost of application development for many platforms and maintaining them.

8.2 Challenges

During this project everything was not rosy and easy. I had to face a few challenges. One of the biggest challenges that I faced was familiarizing myself with PhoneGap framework. It was a new framework for me so it was challenging to study it by myself and examine how it works. Other challenges include lack of real devices to test my application on, not having a developer account for Windows mobile and iOS, dependency on an emulator to test for many devices and screen sizes, outdated information on the internet, lack of user feedback since I was not able to publish my application in app stores yet. Those were some of the challenges I faced during this project.
8.3 Future Improvements

The application that was developed is not a final product but rather a big step towards a fully functioning robust application. Some of the planned improvements were noticed during the developing process while the rest were planned after the completion of this project.

- Online storage feature since the application stores locally on the device
- User login and access of saved items from different devices
- Better user interface
- Storage of achieved wishes with accompanying pictures in the application rather than saving the picture in the photo album of the device
- Taking and saving video after achieving a wish
9 Conclusion

From the beginning the aim of this thesis project was to research different approaches of mobile device application development and make an informed choice based on merit. This project was a chance to brush up and use my HTML5, CSS3 and JavaScript knowledge and skills. Knowing a programming language is different from knowing how it can work together with another one to produce an application. Furthermore, I have found out that a hybrid mobile application is the most versatile and practical choice since it was possible to target a wide range of mobile platforms. Even though hybrid applications may not give high performance compared to native applications because of the extra layer between the code and the targeted mobile platform, they compensate it by the reduction in development time and cost.

In a rapidly growing and changing technological field it was a challenge to get up-to-date information. Some of the information that I found and thought would be helpful was outdated and old or is not supported any more. It is easy to see how fast things are changing and the use of a cross platform application is being adopted by big platform developers such as Microsoft.

To summarize my experience in this project I was able to create a working hybrid application capable of running on multiple platforms. I have tested the application on three different devices from three different platforms. In addition to that, I was able to test it on the Ripple Emulator to see how it performs on different screen sizes and platforms. The results from these different tests are very satisfactory and encouraging for future endeavours.
References


Appendix 1: Code Listing from config.xml

```xml
<widget id="Bucket_list" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:gap="http://phonegap.com/ns/1.0">
  <name>Bucket List</name>
  <description>
    Bucket List
  </description>
  <author email="armon44@gmail.com" href="http://yourdomain.com"></author>
  <preference name="permissions" value="none" />
  <preference name="phonegap-version" value="cli-5.1.3" />
  <preference name="orientation" value="default" />
  <preference name="target-device" value="universal" />
  <preference name="fullscreen" value="true" />
  <preference name="webviewbounce" value="true" />
  <preference name="prerendered-icon" value="true" />
  <preference name="stay-in-webview" value="false" />
  <preference name="ios-statusbarstyle" value="black-opaque" />
  <preference name="detect-data-types" value="true" />
  <preference name="exit-on-suspend" value="false" />
  <preference name="show-splash-screen-spinner" value="true" />
  <preference name="auto-hide-splash-screen" value="true" />
  <preference name="disable-cursor" value="false" />
  <preference name="android-minSdkVersion" value="7" />
  <preference name="android-installation" value="auto" />
  <gap:plugin name="org.apache.cordova.camera" source="npm" />
  <gap:plugin name="org.apache.cordova.media-capture" source="npm" />
  <gap:plugin name="org.apache.cordova.console" source="npm" />
  <gap:plugin name="org.apache.cordova.contacts" source="npm" />
  <gap:plugin name="org.apache.cordova.device" source="npm" />
  <gap:plugin name="org.apache.cordova.device-momt" source="npm" />
  <gap:plugin name="org.apache.cordova.device-orientation" source="npm" />
  <gap:plugin name="org.apache.cordova.dialogs" source="npm" />
  <gap:plugin name="org.apache.cordova.file" source="npm" />
  <gap:plugin name="org.apache.cordova.file-transfer" source="npm" />
  <gap:plugin name="org.apache.cordova.geolocation" source="npm" />
  <gap:plugin name="org.apache.cordova.globalization" source="npm" />
  <gap:plugin name="org.apache.cordova.inappbrowser" source="npm" />
  <gap:plugin name="org.apache.cordova.media" source="npm" />
  <gap:plugin name="org.apache.cordova.network-information" source="npm" />
  <gap:plugin name="org.apache.cordova.splashscreen" source="npm" />
  <gap:plugin name="org.apache.cordova.vibration" source="npm" />
  <icon src="icon.png" gap:platform="android" gap:qualifier="ldpi" src="res/icon/android/icon-36-ldpi.png" />
  <icon gap:platform="android" gap:qualifier="mdpi" src="res/icon/android/icon-48-mdpi.png" />
  <icon gap:platform="android" gap:qualifier="hdpi" src="res/icon/android/icon-72-hdpi.png" />
  <icon gap:platform="android" gap:qualifier="hdpi" src="res/icon/android/icon-96-hdpi.png" />
  <icon gap:platform="android" gap:qualifier="xxhdpi" src="res/icon/android/icon-144-xxhdpi.png" />
  <icon gap:platform="android" gap:qualifier="xxxhdpi" src="res/icon/android/icon-192-xxxhdpi.png" />
</widget>
```
Appendix 2: Code Listing from index.html

```html
<!DOCTYPE html>
<head>
<meta charset="utf-8" />
<meta name="format-detection" content="telephone=no" />
<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height, target-densitydpi=device-dpi" />
<title>Bucket List</title>
<link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.4.3.min.css" />
<link rel="stylesheet" type="text/css" href="css/index.css" />
<link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.0.1.custom.css" />
<script type="text/javascript" src="js/jquery-1.11.1.min.js"></script>
<script type="text/javascript" src="js/jquery.mobile-1.4.3.min.js"></script>
</head>

<body>
<script type="text/javascript" src="cordova.js"></script>
<script type="text/javascript" src="js/todo.js"></script>
<div data-role="page">
<header data-role="header" data-position="fixed">
</header>
<main data-role="content">
<ul data-role="list-view">
</ul>
</main>
</body>
</html>
```
Appendix 3: Code Listing Camera plugin camera.js

```javascript
/* http://www.apache.org/licenses/LICENSE-2.0 */

var argscheck = require('cordova/argscheck'),
    exec = require('cordova/exec'),
    Camera = require('./Camera');

var cameraExport = {};

// Tack on the Camera Constants to the base camera plugin.
for (var key in Camera) {
  cameraExport[key] = Camera[key];
}

cameraExport.getPicture = function(successCallback, errorCallback, options) {
  argscheck.checkArgs('FO', 'Camera.getPicture', arguments);
  options = options || {};
  var getValue = argscheck.getValue;

  var quality = getValue(options.quality, 50);
  var destinationType = getValue(options.destinationType, Camera.DestinationType.FILE_URI);
  var sourceType = getValue(options.sourceType, Camera.PictureSourceType.CAMERA);
  var targetWidth = getValue(options.targetWidth, -1);
  var targetHeight = getValue(options.targetHeight, -1);
  var encodingType = getValue(options.encodingType, Camera.EncodingType.JPEG);
  var mediaType = getValue(options.mediaType, Camera.MediaType.PICTURE);
  var allowEdit = !!options.allowEdit;
  var correctOrientation = !!options.correctOrientation;
  var saveToPhotoAlbum = !!options.saveToPhotoAlbum;
  var popoverOptions = getValue(options.popoverOptions, null);
  var cameraDirection = getValue(options.cameraDirection, Camera.Direction.BACK);

  var args = [quality, destinationType, sourceType, targetWidth, targetHeight, encodingType,
              mediaType, allowEdit, correctOrientation, saveToPhotoAlbum, popoverOptions, cameraDirection];

  exec(successCallback, errorCallback, "Camera", "takePicture", args);
  // XXX: commented out
  //return new CameraPopoverHandle();
};

cameraExport.cleanup = function(successCallback, errorCallback) {
  exec(successCallback, errorCallback, "Camera", "cleanup", []);
};

module.exports = cameraExport;
```