

# **“Cross-Platform Mobile Application Development”**

Alternative approach to native development

**Muhammad Tahir Akbar**

Student ID: **21196470**

Module Leader: Dr Wei Jie

Supervisor: **Dr Samia**

**Dissertation**

**MSc Network and Mobile Computing**



**The University of West London**



## Table of Contents

1	Introduction.....	7
1.1	Motivation.....	7
1.2	Aims and objectives .....	8
1.3	Methodology.....	9
a.	Literature Review .....	10
b.	Experimental Study .....	10
2	Mobile Designing Principles .....	11
2.1	Mobile Mind Set.....	11
2.2	Mobile Contexts .....	11
2.3	Global Guidelines .....	12
2.4	Navigation Models .....	12
2.5	User input.....	13
a.	Input Type is Not Always “text” .....	13
2.6	Gestures .....	14
a.	Invisible: .....	14
b.	Two hands: .....	14
2.7	Orientation .....	14
2.8	Communications .....	16
2.9	Summary .....	16
3	Related Cross Platform Technologies.....	17
a.	Rhodes.....	17
b.	DragonRad.....	18
c.	MOSync .....	19
4	Phone Gap Development and Design.....	22
	Fig 4.1.....	22
4.1	Phone Gap Development Component .....	23
a.	HTML5 .....	23
b.	Cascading Style Sheets (CSS) .....	25
c.	Mobile JQuery.....	25

d.	Device APIs .....	25
4.2	PhoneGap Strength .....	26
4.3	PhoneGap Limitation.....	26
4.4	Phone Gap Security .....	26
a.	Insecure Cryptographic Storage .....	27
b.	Application Security .....	27
c.	Injection Security.....	27
d.	Reverse Engineering.....	27
5	PhoneGap Implementation (Application) .....	29
5.1	Functional Requirements .....	29
a.	Actors/Users.....	29
b.	Use Case Diagram.....	30
5.2	Non-Functional Requirements .....	32
	<b>Requirement</b> .....	32
	<b>Description</b> .....	32
	<b>Priority</b> .....	32
	Portability .....	32
	Response Time.....	32
	High.....	32
	Reliability .....	32
	Medium.....	32
	Availability .....	32
	Usability .....	32
	High.....	32
	Security .....	32
	High.....	32
5.3	Development Environment.....	32
5.4	Application Code .....	<b>Error! Bookmark not defined.</b>
5.5	Testing .....	33
a.	White Box Testing .....	33
b.	Black Box testing .....	33
c.	Basic Path Testing.....	33

d.	Data Flow Testing .....	33
e.	Loop Testing .....	33
5.6	Levels of Testing .....	33
5.7	Test Case .....	33
5.8	Results .....	33
6	Evaluation of Application Design.....	34
7	Discussion and findings .....	35
8	Conclusion and future work .....	40
9	References & Bibliography .....	41
10	Online References.....	45

## **Abstract**

# 1 Introduction

Mobiles are now integrated in everyday life, and the users expect versatile application and ease of access in less time. It has been a great challenge in the industry to deliver such high standard applications and meeting every single customer needs. There are many mobile operating systems working hard to deliver their best for their customer in terms of versatility, user friendly and ease of access.

With the fast technology innovations in hardware as well as software, together with World Wide Web and internet broadband internet, mobile computing become ubiquitous. Variety of mobile device getting popular for multipurpose such checking heart-beat, sending email, game, and learning apps. Now mobile become smartphones and use more Internet. As just mobile penetration increase up to 47% that is 12 time more since 2007. This all user of Internet is to download the application and use is for different purposes. Mobile application developer will "appify" pretty much every cooperation you can consider in your physical and computerized world. The expansion of mobile application to each perspective individual and business will be one of the signs of the new decade with colossal open doors for vertically every business divisions.

With the increased development of technology, there is major development in mobile platforms as well. Vendors have their own software architecture. Therefore, application development with single platform at any particular time is getting more difficult. To overcome the difficulties of developing mobile application on different platform with easy, concept for cross platform comes. This research will provide the study about cross platform. This chapter will explain about motivation, research question, methodology and structure of the dissertation.

## 1.1 Motivation

The production of smart phones are increasing gradually, every manufacturer has its unique hardware and software. All mobile providers has their development tools to build applications. These application are based on several libraries that support the interface for their application.

For the development of application user need to follow many task which involve, making mobile application such as requirement gathering, data flow, user interaction design, backend coding, testing, bug fixing and deploying in real environment. In this research, we will explore the design issue such as utilization of screen space, interaction mechanisms and data grouping. As mobile devices has smaller view as compared to desktop, so these device has limited space for focus. Mobiles has also compact view for displaying information. It need to be prioritise the task which user has to concentrate. There are number of platforms (Android, IOS, windows phone) that has unique design principles which may not be support each other. So, it is quite difficult to find someone who has knowledge and can work all these platform.

There are plenty of tools that allow developer to develop application for many platforms (with or without doing minor changes in the code). Such tools are known as cross platform tools. Mostly tools are based on different design principles and only support few platform and their basic feature such as camera, contacts etc. Phone Gap is also a cross platform mobile application development tools that use to develop application for more than seven platforms. Phone Gap is hybrid technology and have different features. PhoneGap is based on HTML, CSS and Javascript that is most commonly used to develop web interface. PhoneGap allow developers to use basic design principles for user interface. Can these design principles helpful for developing mobile application in the phone gap environment? The problem discussed earlier has encouraged this study to explore the design principles of native applications in phone gap.

## 1.2 Aims and objectives

Today, Application development getting popularity in all business. As there are several vendor that provide platforms for these application to execute. The design for mobile application become a big challenge for developer/designer to support multi-platform such as Android, iPhone, and Windows etc. There are several solution that provide the cross-platform functionality but how these tools are helpful in designing an application?

According to Christopher Reynolds (2014), that phone gap is second best cross platform mobile application development that is based on open source Apache Cordova project that is free to use. It is further studied that Phone Gap developer can write application in HTML, CSS and JavaScript, deploy on multiple platform with out losing the native application functionality.

The main objective of dissertation is to explore the designing principles of user interface that were develop for mobile application to support native tools. Cross platform tools are getting popular in the market, an investigation need to be taken on Phone Gap that may support such principle. To achieve the objective and conduct dissertation, following question need to be answered.

**Does the cross platform tools such as Phone Gap support the design principles that were develop for native applications and what are the perspective benefits of using Phone Gap?**

To analyses the above statement, research will also have following supporting arguments

- Which are the basic principle that develop to support for native mobile applications such as android, IOS and Windows Phone?
- Discussion on Phone Gap as cross platform mobile application development tools and its design feature.



- At what extend phone gap can help to achieve user interface as in native application?
- Developing mobile application in phone gap to analyse the principles

### 1.3 Methodology

The research will be based on mobile design principles that will also helpful in cross platform tools. Furthermore, a cross platform tool such as Phone Gap will be explored on these principle. The discussion will review background for these principles, will these rules be useable in the Phone Gap environment. This will further leads to implementation of user interface in the Phone Gap.

This research is based on two different aspects i-e analysis of principles and implementation of design. In order to complete dissertation, research will be based on Design Science research methodology (DS). Design Science methodology is widely use in the mobile computing research due to its cycle shown in fig 1.0. According to Vijay & Bill (2013) "Knowledge is generated and accumulated through action. Doing something and judging the results is the general model . . . the process is shown as a cycle in which knowledge is used to create works, and works are evaluated to build knowledge;".

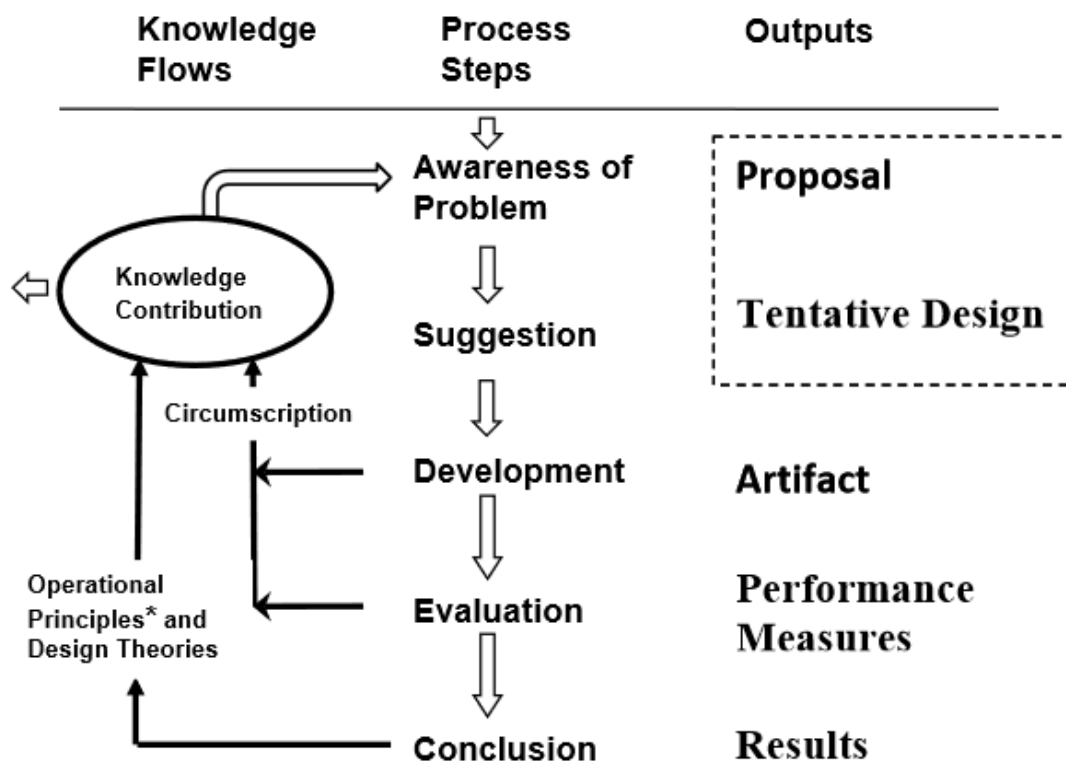


Fig 1-1 Design Science Research Process Model (Vijay & Bill 2013)

In the above diagram, primary five steps of design science methodology are displayed. The first two steps (awareness of problem and suggestion) of methodology will help us in the literature review and the last three steps (development, evaluation and conclusion) will cover

the development part of mobile application in Phonegap. We will divide our dissertation into two parts.

**a. Literature Review**

The process of gathering knowledge about design principle of different platforms will help to understand the problem area and also help to achieve the first phase of research. Literature review will include searching, reading and evaluating various section of journals, articles, book and online resources such as IEEE Xplore, ResearchGate, ACM Library and google scholar. While searching on these resources, keywords plays important role. It keep emphases to choose correct keywords for relevant data. Following keywords is selected for searching article on online resources.

“Mobile application development”, “Mobile User Interface”, “Designing Principle for Mobile Applications”, “Mobile application platform”, “Hybrid mobile application”, “background application development”, “Cross platform mobile app tools”, “Phone Gap” and “Phone Gap Design Guidlines”.

**b. Experimental Study**

Experimental study is development phase in which research will be based on practical knowledge. An application will be develop by using Phone Gap as cross platform tool. Firstly, a sample application for mobile devices will be design and discussed. The application will be design such a way that design will easy to use and user can navigate through the application without any assistance or difficulty. Furthermore, application will be design on Phone Gap using HTML CSS. After deploying the application it need to be evaluated on different factor of mobile design principles which will discussed in literature review.

The above mention methodology will help to complete the dissertation and also it indicate that Phone Gap will supports design principle that were available for the other mobile platforms.

## 2 Mobile Designing Principles

Although computing devices are placed together for different purposes but desktop and mobile are actually different such as big screen vs small screen, plugged inn vs battery power, reliable connectivity vs intermittent, high bandwidth vs low and so on. By above comparison, one might think about smart devices are not as powerful version as real but it would be a mistake.

In fact, smart phone are more powerful as compare to desktop in many behaviours. These devices are more personal, always connected with us, mostly power on and directly address by user. The most important, they have different feature like powerful sensors that has capabilities to detect the location, orientation, environmental conditions, movement and acceleration etc.

We have already discuss do many difference between smart phone and desktop computing. It will not be surprising that the design for mobile devices is very difficult task as compare to desktop computers. In the following section we will discuss several principles that involves while design for mobile devices.

### 2.1 Mobile Mind Set

Before start mobile application, it's important to understand that the key difference between the mobile and desktop devices. Designer need to be imperatively get into the mobile mind set and think about application.

- Don't put all the feature into the application. Reduce application feature ruthlessly and take only key feature. To may feature will not work for mobile applications. **Be focus** on the objective of application.
- You will find so many application that may relate to your idea, make sure you have highlighted the feature which unique your application. If there will be no special feature then may be no one pick it.
- Usually developer include only those feature which they like or feel that user will like it. Be sure the developer put them self in the shoe of end user.

### 2.2 Mobile Contexts

Smart phone is not designed to be specific purpose. People use this device for several purpose, a person in a busy airport holding his luggage in one hand and smart phone in another hand. It is certainly one situation, but not the only use, in this section we need to focus on three major context

- Busy
- Bored
- Lost

## 2.3 Global Guidelines

While thinking about the design. There are some global principles that need to be consider. Mobile device are of different sizes therefore the application develop on these devices my call for different purpose, designs and techniques. These techniques are based on

### ➤ Responsive

According to Margaret (2012), Responsive design is a method that provide flexible layouts, flexible images and cascading style sheet media queries. The aim of these designs to build pages that detect the user screen and orientation and adjust the layout of the design.

As mobile devices are different screen sizes, responsive design technology make design easier. While designing, developers can use X and Y coordinates to display the content on exact location and use of percentage parameter for image width and height instead of fixed. Furthermore Media queries allow cascading style sheet (CSS) to be different on depending on screen resolution.

### ➤ Thumbs

Most of the smartphone are integrated with touchscreen. To use the interface, mostly people ask to use finger but in reality its use of thumbs. When design for mobile device, make the design clear and appropriate size. So user don't need zoom functionality to read it.

## 2.4 Navigation Models

There are approaches of navigation that are used in mobile application. As Annarita (2014), describes about main three approaches i-e top navigation (do-nothing approach), select menu, toggle approach and bottom approach.

### ➤ Top Navigation (Do-Nothing approach)

Most regularly utilized answers for navigation is to keep it at the top. Due to its simplicity of usage, it's the one included by the greater part of sites. This method does not have too many tricks to play with java script and CSS, however this solution is not suitable for mobile application

### ➤ Select Menu Approach

'Select menu' is common and simple approach. This menu can be accessible in every mobile, easy to integrate, and familiar to everyone. The menu free plenty of space and make interaction to the header. It also clearly label about the navigation or menu.

### ➤ Toggle Approach

Toggle approach is one of the best approach for navigation menu. In this approach, whole menu link to a button in the header, when you touch that button the menu slides open and comes onto the screen. It's scalable, user-friendly and good-looking approach. It is also not difficult to implement and allow to stick with your whole menu and user can navigate easily.

## 2.5 User input

As Anthony (2011) explains in an article that typing is simple for most clients when they do it on a PC. Yet, when they do it on a cell phone, it's difficult for them due to that they can't see every one of the keys on a console. Most of mobile OS have composed separate keyboard for particular structure field inputs to make clients' lives easy. These advanced keyboard are there to utilize, however most designers neglect to incorporate the right info sorts in their code to exploit them. This prompts slower finishing rates when clients fill out field on their cell phone

### a. *Input Type is Not Always "text"*

Most of the developer do common mistake and put "text" type in all fields of the forms. The input information of every field is not the same as alpha-numeric characters, some required numeric characters, and some are just alphabets. When the input type set to text, it force user to make use of alphabet key to enter numeric data. By doing this developer make typing hard for users.



Fig 2-1 Phone Input Keyboard

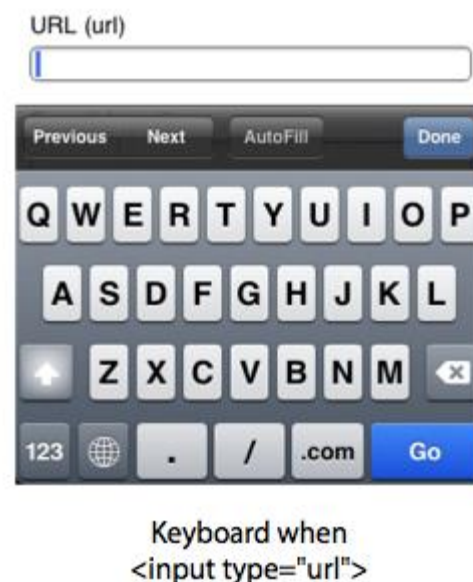


Fig 2-2 URL Input Keyboard

There are different types of keypad shown in the figure below.



Keyboard when `<input type="text">`

Keyboard when `<input type="email">`

Fig 2-3 Text and Email Input Keyboard

## 2.6 Gestures

Most powerful feature of today's smart phone is touch screen which support gestures base user input. There are several thing that need to be consider

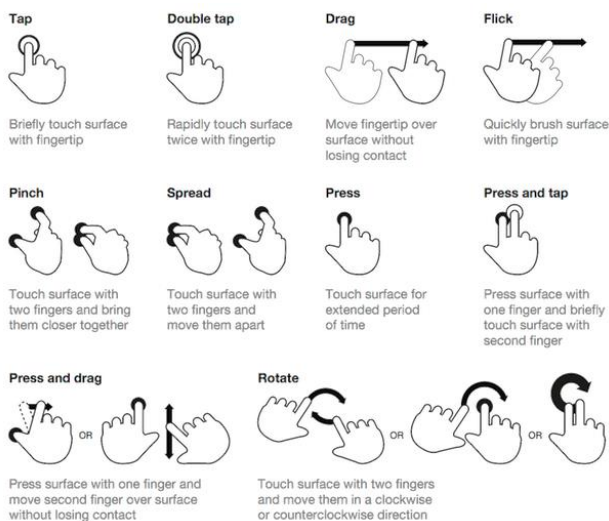


Fig 2-4 Gestures & Animations (source: uxmag.com)

## 2.7 Orientation

The accelerometer set in our mobile gadget is used to adjust view depending on the alignment of the device, for instance, exchange of portrait to landscape modes. This option give better opportunity to create best user experiences because additional layout with simple turn of gadget, and without user input. However, designing for such device bring challengers and careful thinking. This transition must be as transparent as possible. Avi Itzkovitch (2012) define 4 major categories of the orientation layouts.



### ➤ Fluid

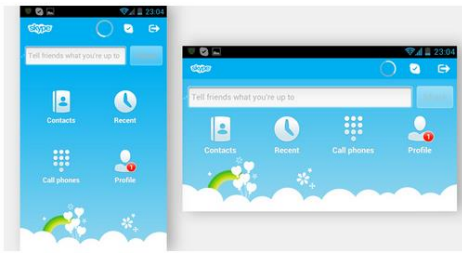


Fig 2-5 Fluid Orientation (Source: smashingmagazine.com)

In fluid layout, the view simply adjust to new orientation size, as shown below figure.

### ➤ Extended

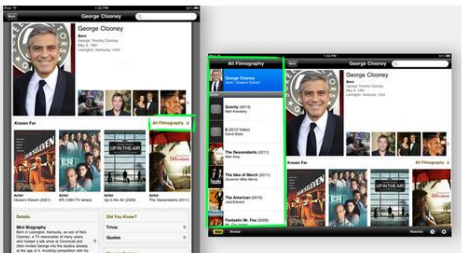


Fig 2-6 Extended Orientation (Source: smashingmagazine.com)

The view adjust the new template by adding or removing some section of the layout widgets according to screen dimensions. Following example, IMDb add new side widget in the view to adjust the landscape orientation.

### ➤ Complementary

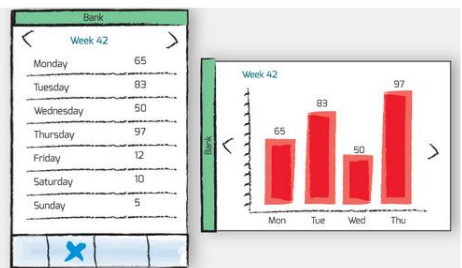


Fig 2-7 Complementary Orientation (Source: smashingmagazine.com)

Interface display on left define different view of a supporting screen with the information relevant to the supplementary screen. For example, most of the financial mobile application shows data in the portrait mode but when we change it to landscape mode it gives user a graphical view of the data displayed in portrait mode.

### ➤ Continuous

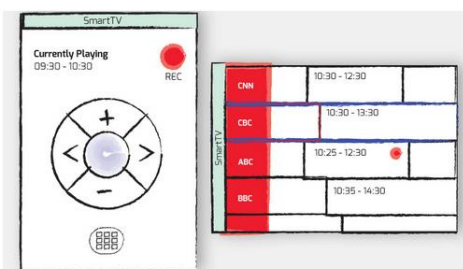


Fig 2-8 Continuous Orientation (Source: smashingmagazine.com)

By simply rotating the device user can access secondary interface, this is called continuous design. Biggest example for continuous design is YouTube. Also, using smartphone as remote control of smart tv is another good example. Main idea of continuous design is when you are browsing in portrait which provide you less information such as just heading for program on YouTube but when you browse same application on landscape mode, it will provide you brief information regarding program with its title and timings as well.

## 2.8 Communications

While developing mobile application, communication is necessary between the user and the application. Some key communication method are discussed below

### ➤ Give feedback:

Feedback is a common communication method between the user and the application. Feedback should be provided instantly to the user, if failing to do that user may think the application is crashed or it is stuck. Feedback could be in any form such as thumb vibration in android, change the colour of highlighted area. Furthermore in case the action of user take long, show some sort of progression like spinner or progress bar.

### ➤ Modal alerts:

Model alerts may annoy the user because it interrupts the user flow. Only user these popup's when the user did serious mistake (use friendly language and assure user).

### ➤ Confirmations:

"Confirmation dialog" box appears to get confirmation from user such as "Are you sure you want to delete this draft?". Based on the response of the user, action will be taken place therefore confirmation are less disturbing compare to the alerts. Also, user need to make sure while making selection for default button to avoid any accidental damages.

## 2.9 Summary

Design is an important part of application development. Several principle for mobile application design was discussed in this section. Those design principles help to generate interactive applications. We will use some of those principle such as Navigation, user input, gestures, orientation and communication while generating the mobile application. Navigation principle will help to put menu in the header section. "User input" principle guide to categorise the information that will be entered by users. We will use "gesture" principle to give use flexibility to use the maps. To interact with the user and get view of them regarding the application, communication principle will be useful.



### 3 Related Cross Platform Technologies

#### a. Rhodes

Rhodes is formally a product of Motorola Solutions Inc developed under Institute of Massachusetts (MIT). It is developed to support all major device components such as Apple, Android, Blackberry, Windows Mobile and Symbian. It relays on high-level productivity and portable in mobile programming. It is programmed in Ruby mobile platform, a popular open source.

Rhodes files can be converted to native applications, which can be executed, to real time or a part of virtual devices, which is simulated to run a java byte code applications can be executed on Blackberry or Ruby 1.9 byte code for other compatible OS.

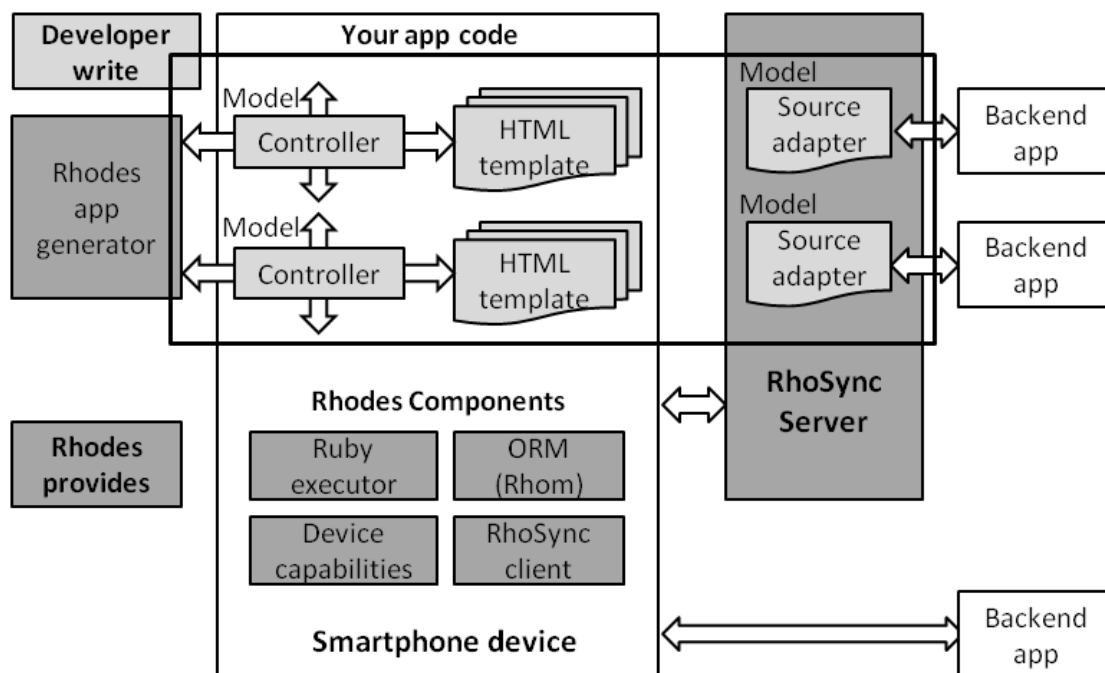


Fig 3-1 Rhodes Arch

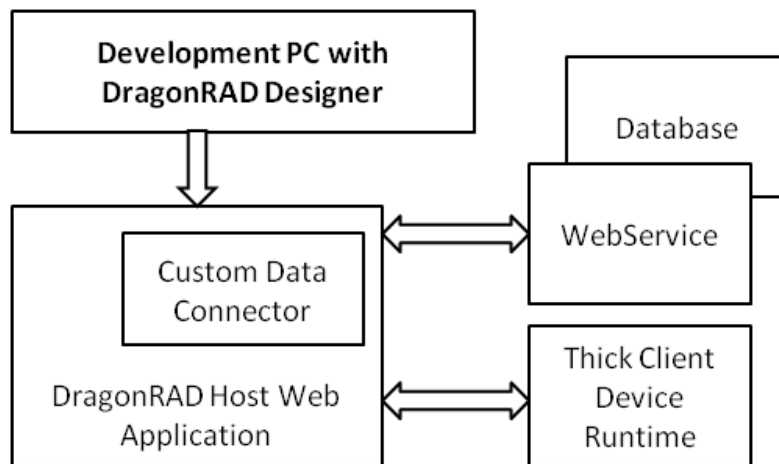
Languages included to view the elements are HTML, CSS and JavaScript and ruby for the backend process using SQL as a database structure. MVC patterns also included writing applications for the purpose of view element. It is designed and could be maintained the sites required for complexity at low level.

**b. DragonRad**

This is a native application to run and interrupt code by the designer. It is an emulator, which runs to perform and debug within the application. It has got many features for icon change, change of application name.

It has a feature of connecting to the defined host directly when connected to the DragonRad host. It has got a unique feature of compiling the programs instantly online. The features include;

- Getting information of the data query in the device.
- Execution of retrieved data query from the device.
- Acting as a medium for sending the data on request
- Acting to update the device and the database
- Comparing the data packets and comparing them in the device.



*Fig 3-2b. DragonRad Arch*

The above figure illustrates the development PC with DragonRad designer communicating with the custom data connector in the DragonRad Host Web Application.

This system is connected with the web service (i.e. SOAP / RESTFUL) with a client device runtime. The database server acts to store the incoming data and to retrieve it.

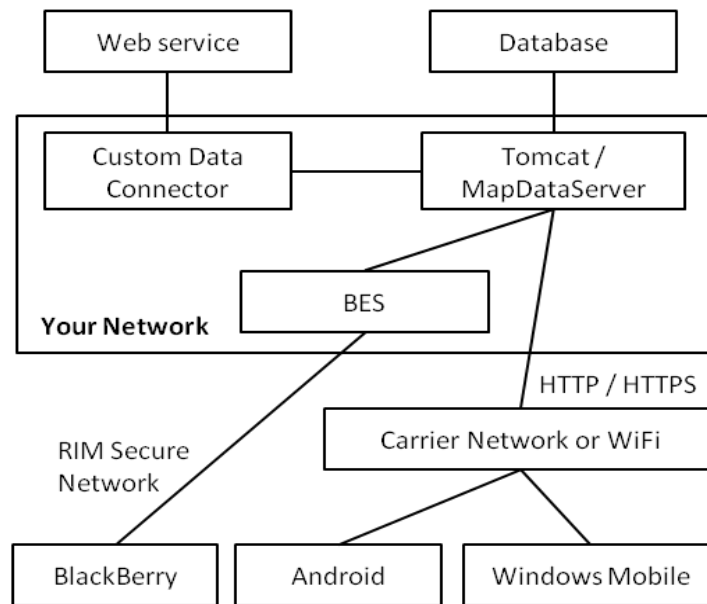


Fig 3-3 DragonRad application connection

The above figure illustrates the DragonRad application connected with different components. The data connector is featured as;

- Designed to transfer and update and synchronize with the database in the device.
- Receiving the queries as a data packet from the device.
- Executing or processing the retrieved data packets.
- Getting information from the device
- Sending back all the retrieved data packets back to the device.
- Re-query the data in the database and sending back to the device.

### c. **MOSync**

MoSync is a mobile platform emulator targeted for mobile market as an open source solution. It is determined as a full stack STK component designed to deliver an ultra pack mobile application for any complex code and application.

It is coupled with libraries, database, compilers and runtime components together using standard object oriented programming language such as C/C++.

It was also well structured with the popular web components like HTML5, CSS and Javascript for multitasking and cross platform.

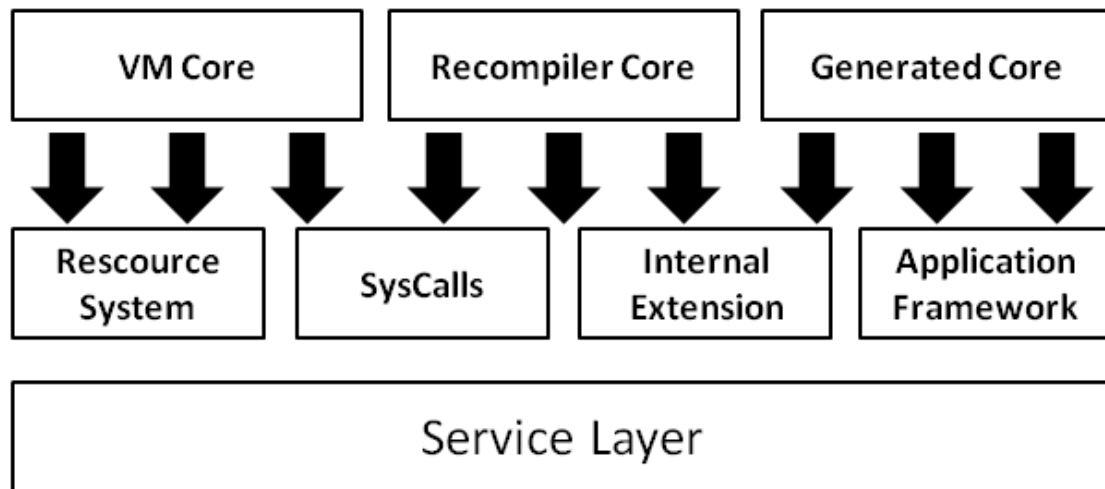


Fig 3-4c. MOSync

#### Basic Feature of Cross-Platform Applications:

The below tabular describes the Operating system and mobile device support for the application platform.

The comparison table illustrates the methods handled and language support to develop the applications and native API's, IDE and Plug-in accessibility for cross-platform support on major mobile platforms.

Tool Name	Mobile OS Support	OS Support
Rhodes	Android, BlackBerry, iOS, Symbian, Windows Mobile, Windows Phone	Linux, Mac, Windows
PhoneGap	Android, BlackBerry, iOS, Symbian, WebOS, Windows Phone	Linux, Mac, Windows
DragonRad	Android, BlackBerry, iOS, Windows Mobile	Linux, Mac, Windows
MoSync	Android, iOS, BlackBerry, JavaME, Symbian, Windows Mobile	Linux, Mac, Windows,

The below table describes the development features and their accesabilty to native API's for the applications.

<b>Name</b>	<b>Language</b>	<b>Accessibility to native API's</b>	<b>IDE</b>	<b>Plug-in Extendibility</b>
RhoMoblie	HTML, HTML5, CSS, JavaScript,	JavaScript	RhoStudio RhoHub, *	Yes
PhoneGap	HTML, HTML5 CSS, CSS3 JavaScript	JavaScript	IDE native of the mobile OS (e.g. Eclipse, Xcode)	Yes
DragonRad	D&D	na	DragonRad Designer	No
MoSync	HTML, HTML5, CSS, JavaScript C, C++	JavaScript, C, C++	Based on Eclipse	Yes

## 4 Phone Gap Development and Design

Phone Gap is a mobile cross platform framework founded by Nitobi and owned by Adobe Systems INC. The core structure of the framework is formed of HTML5, CSS3 and Javascript and the APIs are developed in the respective platforms.

The native applications are developed in the structural method to support most of the devices .

The environment is basically developed to support IOS and Android version in their version 3.0 and commonly made to support all the basic environmental platforms such as Blackberry, Symbian an WebOS.

The below architectural figure describes the model of Phone Gap technical environment which allows the native users to build their applications with the helop of their library functions and IDEs and basic plugins to supprtt the applications build with the common user interface languages.

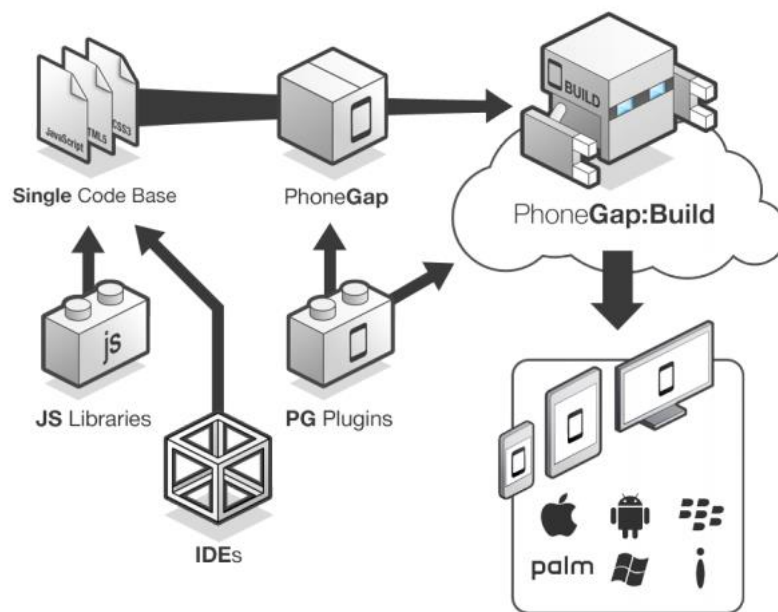


Fig 4-1 PhoneGap Structure

**Fig 4.1**

- Phone Gap is build to upload in its cloud based service, where one can upload the web assets to support the major devices.
- The services may include either Github or SVN services to upload or modify the web data within the service.
- It doesn't require any IDE or SDK to install and develop the application and it is fully formed to support IOS without OSX based devices.

- At first native shell is launched and uses the browser and then launches the web application.

## 4.1 Phone Gap Development Component

### a. HTML5

Html5 is a cross platform markup language that provides structure base document. It is enhanced version of old HTML standard which removed a lot of unwanted features and introduce new features. It also recognise officially for documenting standard for web platform. (Pilgrim,2010)

Mid of 2000s, World Wide Web Consortium (W3C) created some standard for XForms and XHTML, that will be used for web in future. Those standards were required very strict rules and did not support the old technology. Being disappointed with this vision that was seen as unrealistic for this present reality, a community of web sellers and other invested individuals had a contending vision without bounds of the Web: developing HTML4 to incorporate extra elements keeping up in reverse similarity. W3C individuals did not concur with this vision, and therefore, the WHATWG(Web Hypertext Application Technology Working Group) born and characterizes itself as a "free, informal, and open coordinated effort of Web program makers and invested individuals" (Pilgrim,2010)

With the support of development community of web. W3C and WHATWG come to one point and start building HTML5 as new standard however there are still some difference between W3C and WHATWG. Although there are many feature are available for developer but the language is site not completed. As a disadvantage, in any case, the usage may change between browsers, and develop need to change their implementation.(Pilgrim,2010)

In the below section, some new feature for HTML5 will be discussed

#### ➤ Semantic MarkUp



Fig 4-2 Semantic Markup

(Source w3schools.com)

HTML5 define several new elements, classes and attributes. The main aim is to provide content processors for search engines and the markup are more meaning full to developers. This semantic structure HTML give more presentation feature (Hickson, 2012).

**Embedded content:** embed, audio, img, iframe

**Flow Content:** a, article, header, div, audio, nav, form, p

**Heading content:** h1 – h5, hgroup

**Interactive content:** button, menu, a, select etc

#### ➤ Multimedia



Fig 4-3 Multimedia Formats

Before the HTML5, embedding of videos, audio was a big problem. Usually developer use 3<sup>rd</sup> party plugins, however HTML5 has built inn feature to support the multimedia contents. According to w3schools.com, multimedia has many form and format, anything you can hear or see such as: music, sound, pictures, records, videos, animations, films, and more. HTML5 provides video and audio tags. Furthermore, the major issue with HTML5 audio or video is the question of codecs between browsers. For instance, Mozilla and opera supports Theora which safari does not support it, however safary support H.264 which is supported by IE9. Both iPhone and Android likewise supported H.264 on their portable browser. W3C prescribes OggVorbis and Ogg Theora for sound and video individually.

### ➤ Drag and Drop



Fig 4-4 Google Drop Drag Functionality

Drag and Drop is a common interaction technique where elements can be moved within the user interface from one place to another. Older browsers have had proprietary solutions for this interaction pattern, but HTML5 standardizes the API. The specification defines the element attributes and DOM (Document Object Model) events for easily enabling and controlling draggable elements and drop targets.

Custom cross-browser JavaScript solutions have enabled this interaction before, but little JavaScript code is needed with the new API. The browser handles the interaction and the dynamic rendering, reducing the interface lagging and the need for extra processing.

### ➤ Local Storage

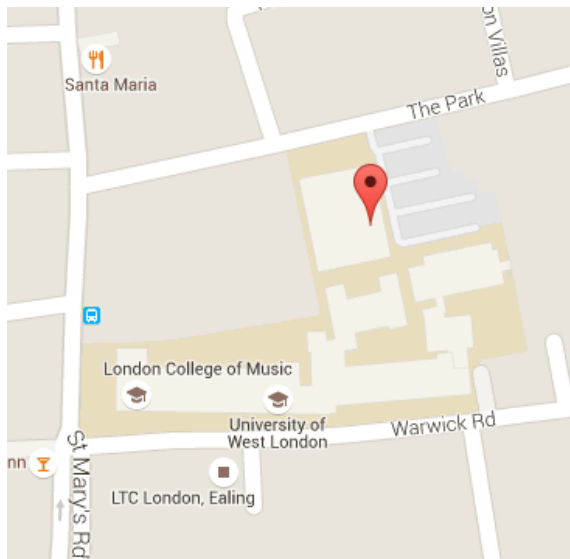
```
if(typeof(Storage) !== "undefined") {
    // Code for localStorage/sessionStorage.
} else {
    // Sorry! No Web Storage support..
}
```

Fig 4-5 Sample Code For local storage

One of the biggest limitations of HTML was the absence of any data storage capability. Client Side storage is the basic requirement for any interactive modern day application. This was the area where native applications had a huge advantage over web applications. HTML5 solved that issue by introducing many data storage capabilities. One of those is local storage which is also referred as Web Storage as per HTML5 specification. Local Storage provides a way for web pages to store data as key/value pair within the browser. This key/value pair data persist even after we close the browser.



### ➤ **Geolocation API**



HTML5 provides Geolocation API to developers which facilitates in adding location based features to their applications. Location Based Services are quite popular these days in native applications. HTML5 Geolocation API provides the geographical position of a user by finding its latitude and longitude. The HTML5 geolocation API finds the location of a user through GPS or using Network signals like tower or using other sources like IP Addresses etc.

#### **b. *Cascading Style Sheets (CSS)***

CSS Level 3 specifications introduce lots of new functionality for web application styling. Well separated layout layer keeps the document structure clean, and rich styling and effects capabilities reduce the need for scripting and provide graceful fallback functionality for older user agents. By letting the browser handle, for example, rich user interface animation effects allows developers to easily optimize the responsiveness and performance of their applications since the browser can use the most efficient techniques of the platform to handle these effects. Below I list the main components and specifications of the W3C CSS working group<sup>1</sup>.

#### **c. *Mobile JQuery***

JQuery Mobile is a touch-optimized mobile framework currently being developed by the JQuery project team. The development focuses on creating a framework compatible with a wide variety of smart phones and tablet computers, made necessary by the growing but heterogeneous tablet and smartphone market. The JQuery Mobile framework is compatible with other mobile app frameworks and platforms such as PhoneGap, Worklight and more. The advantages of JQuery Mobile: 1. It is a light-weight framework. 2. The loading speed of mobile application is fast. 3. Support more web browsers. 4. It is easier to make function extensions. The disadvantages of JQuery Mobile: 1. The performance of UI is relatively slow. 2. The implementation of gestures is difficult. 3. Some parts of the documents is spotty and wrong.

#### **d. *Device APIs***

## 4.2 PhoneGap Strength

Architecture of PhoneGap is small, simple and flexible: the PhoneGap team has intentionally implemented only the lowest common denominator of native APIs for the web browser-based app (programmers can choose or develop plugins as needed, they can also choose UI frameworks freely such as JQuery Mobile and Sencha Touch).

PhoneGap supports almost all platforms and devices. Basically any native platform that supports a web view or web runtime can be a PhoneGap platform.

## 4.3 PhoneGap Limitation

Development with PhoneGap tool give user attraction and also promises to support several platform without compatibility issue. It expects to turn into an accepted versatile application development framework by utilizing HTML5 technology. But as studied with the about abilities offered by the framework is limit to web technology. There are some limitation listed below.

CSS limitation for styling, in few cases it work such as rounded corners appear in android but same code look jagged in IOS

As for webView, it has limited memory, so code for HTML, JS and JSON need to be optimised to get better result.

While working in in multi-platforms development environment, developer should know the phone gap capabilities to support the platform. For different platform, phone gap has different behaviour, for instance – new version of IOS supports web sockets while android, it's still lacking behind.

For BlackBerry, PhoneGap supports latest version, some of the previous version are not properly working and not fully supported by phone gap.

## 4.4 Phone Gap Security

As your app runs inside WebView (aka a browser), you should not allow untrusted third-party code to execute in the page, because this can compromise the security, the JS code downloaded from internet will suddenly have access to several device features that they can make substantial damage to user's privacy and security. Do not load an iFrame in your app, or redirect to some other websites when the app is loaded, as this will likely get your app rejected in Apple app store, because Apple does not like applications opening public websites inside the app as it makes it difficult or almost impossible to enforce policies regarding content.

“PhoneGap is generally limited to the security features of the platform on which it is running” Jordanbtucker (2014). In this section, several security features will be discussed with respect to operating system and hardware level. Hardware encryption is a very important part to secure mobile application or data, now almost every mobile operating system provides some sort of encryption to communicate with the hardware level. There are many solutions available that support the storage encryption such as password and sensitive data. Encryption is enhancing day by day because there are a number of security approaches that have been compromised. Some of security concerns are discussed below.

**a. *Insecure Cryptographic Storage***

Cryptography is a term that is used to encode or decode the data, so that data will be secure. There are several methods by which a data can be encoded. Usually keys are used to encode the data, those keys known as public and private key. Public key is known to both sender and receiver whereas private key is kept safe for decryption of data.

As PhoneGap is a tool that deals with the data, so it is the responsibility that the developer must ensure about cryptographic. According to Jordanbtucker (2014), some developers use this technique but it leads to risk. Reason for this risk is that developers do a common mistake, the encrypted data is kept in clear text. Usually the developer does not care about password to be salted, just encrypt and stored. This will help the attacker to decrypt and use the passwords.

**b. *Application Security***

Since PhoneGap applications use HTML and JavaScript, many of the security concerns around PhoneGap applications are similar to those concerns on the web at large. While the fact that cross domain network requests are possible from a PhoneGap application may seem like an additional security concern, there has always been a number of different ways to do cross domain requests from web applications running in browsers that respect the same origin policy (e.g. sending data in the querystring of a GET request as the result of loading an image, script etc).

**c. *Injection Security***

We use `JSON.parse` to prevent malicious content from being executed in the context of your application using something like `JavaScript eval(...)`. There is no built in prevention mechanism against JavaScript injection.

**d. *Reverse Engineering***

Reverse engineering is a concern of many people that use Phone Gap since one can simply open an application binary and look at the JavaScript source code of the application. One could even go so far as to add malicious JavaScript code, re-package the application and re-submit it to app stores / markets in an attempt at app phishing. This practice could be

undertaken with any application whether it is written with Phone Gap or otherwise since it is a similarly simple task to decompile either Java or Objective-C.

Phone Gap can actually get around this security concern since application developers can download JavaScript in their application at runtime, run that JavaScript, and delete it when the application closes. In that way, the source code is never on the device when the device is at rest. This is a much more difficult prospect with Java or Objective-C let alone the restrictions in the App Store around dynamically running Objective-C code.

## 5 PhoneGap Implementation (Application)

There are several mobile applications in the market for user convenience to store and retrieve data, trace the geo locations, play videos, capture images and save important notes. Mostly application provides individual functionality such as you can search location but cannot take picture (Google Map application), add notes but cannot play videos. This situation encouraged to think about an application that has some basic feature such as gallery with ability to take photos, playing videos, search location and save important notes. So user can perform all in one application.

The application is designed in such a way that any user could get the normal functionality in convenient way. Application will be easy to navigate, use high quality image and user friendly environment. We have named it as “AppForAll”. In the following sections, we will illustrate, what are the key functionality of the application, how the design principle help in PhoneGap in developing the mobile application.

### 5.1 Functional Requirements

In this section, functional requirements will be defined in the use case diagrams and corresponding description will be discussed. Actors/user will discusses, how user will use this application.

#### **a. Actors/Users**

Application has different feature, it can be helpful for different users.

##### ➤ **Students**

Application is specially design for students, the feature such as notes and video recording can help student to make notes and record lectures during the class time. These files can be helpful for student while preparing for exams.

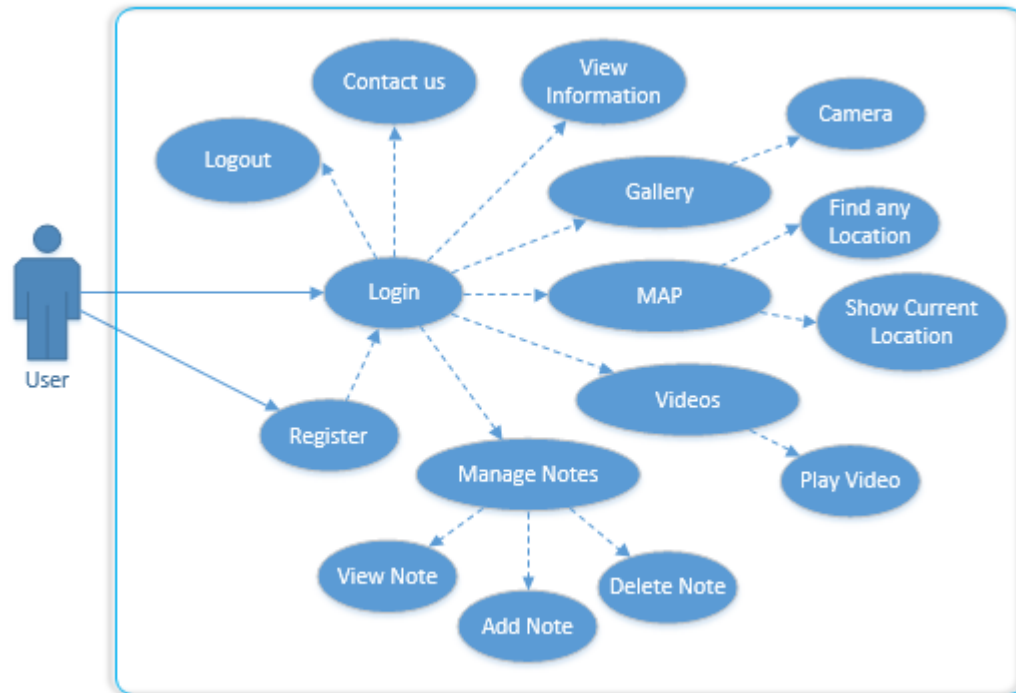
##### ➤ **Tourist**

There are two major functionality which can be helpful for tourist such as map and camera. When tourist visit new places, it is difficult for them to go around without using map. So they can use map feature to find their current location and also get path from their current location to the destination. Also they can find out popular places to visit around in particular city. Furthermore they can use camera to take photos and also make note where they have visited and what is special thing for it.

##### ➤ **General Users**

Application has varieties of feature, it can be adjusted for anyone such as offices workers, tourist, visitors and students. This application has both features like, you can save your data and retrieves data.

**b. Use Case Diagram**



*Flg 5-1 Use Case Diagram*

In the above Use Case diagram, the general functionality for user is shown. Following section will briefly explain the features of application that will be implemented. Furthermore chapter 6 (Evaluation of Application Design) will have more detail “how these feature are being implemented with the design principles”. Some feature are listed below.

➤ **Registration Section**

Before using the application, all user need to register with the application. After registration, user can use their login details to access all the feature of the application. This login details can be used for gathering data and analyse for different purpose such as, how often the system is used by the user. Registration data must be store in an encrypted form, without encryption the data will be at high risk.

➤ **Login**

After loading the application, login section will be displayed. Infect, successfully completion of the registration section leads to login section. User can fill their credential to access the application features. Without login, user can not have access to the any functionality of the application. These function will also be valid for the web version.

➤ **Dashboard / Sidebar**

According to Alexander (2015), To achieve all objectives and display important information, dashboard play vital role. In this application successfully authenticate user will be redirect to a page (Dashboard) where they can access all the features of application. When the user is on dashboard, he can navigate to any screen of the application and also access the sidebar where different option are available as listed below.

**My Profile** – Show the information about the user such as user image, username, email, user personal information etc.

**About Us** – Give detail idea about the feature and functionality of an application.

**Contact Us** – User can contact to administrator in case of any issue or suggestion.

**Logout** – After use of application, user can logout from the application and close application.

#### ➤ **My Gallery**

Gallery shows collection of memories taken by user through camera. User can take pictures and also use front camera to take selfie using application without navigating to the in-built camera application.

#### ➤ **Map**

Navigation is one of the key feature of application. By using this feature, user can find their location and surrounding areas on the map and location by just opening the "map section" of application. There is another feature of called street view, user can use this feature to view the physical existence (pictorial view) of the street and also can move through the streets. Furthermore user can also find any address on the map and plan their visit.

#### ➤ **My Videos**

Video Section is used to list all the latest videos which are separated by the category. User can enjoy videos in their spare time such as at home, travelling in a bus etc. These videos is live streamed form YouTube. User can save it against its profile.

#### ➤ **My Notes**

Application has notes section, to list user notes that help to remember thing for user. Furthermore user can add, modify and delete noted form the list.

### ➤ Email

The feature allows user to send email to anyone from their email address without login to their personal emails. User can put his email address, subject and message body and press the "send" button to send it. But before using this function user need to verify the email address through a verification process.

## 5.2 Non-Functional Requirements

Non-functional requirements of the system are shown below.

*Table 1 Non-functional Requirements*

Requirement	Description	Priority
Portability	Application can be install on any mobile platform such as IOS, Android, Window phone etc.	High
Response Time	Response time of one request from users should be 1. For Login: less than 3 seconds. 2. For register: less than 5 seconds. 3. For gallery: less than 20 seconds. 4. For Map: less than 25 seconds. 5. For Videos: less than 5 seconds. 6. For Notes Lists: less than 3 seconds.	High
Reliability	Application downtime depend on the internet connection. Most of the communication of data is relay on the internet. On average downtime must not be more than 60 minutes in a year.	Medium
Availability	The availability of the application function is almost upto to 95%.	Medium
Usability	It is possible to reach the feature of application in 3 touches from the main screen. In case of delay more than 5 second, a progress bar will be displayed	High
Security	Due to data handling in the application, it is necessary to encrypt the data, so it is protected.	High

## 5.3 Development Environment

IDE	Eclipse
Mobile SDK & Tool	Android upto 4.2V IOS XCODE 4.6.2 PHONEGAP
Web Server	Apache CXF .7.4
Database	Mysql 5.3.1
JDK	JDK 1.9
Web service container	Tomcat 7.0.39
Operating System	Mobile application: Windows 7 Professional



## 5.4 Testing

There are several methods used to test a prototype and the basic and effective ways of testing a system are described below;

### **a. *White Box Testing***

White box testing is the detailed investigation of internal logic and structure of code. White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately

### **b. *Black Box testing***

The technique of testing without having any knowledge of the interior workings of the application is black box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

### **c. *Basic Path Testing***

Basis path testing or structured testing is a method for designing test cases intended to examine each mathematically possible path of execution at least once. By creating a set of basis paths, creating and executing tests for these paths, 100% statement and 100% branch coverage can be guaranteed.

Basis path testing is a white box testing technique that is used to test the code based on the control flow. The method uses a control flowchart and a control flow graph to convert the code into a model and then derive independent test paths from it.

### **d. *Data Flow Testing***

Data flow testing is a family of test strategies based on selecting paths through the program's control flow in order to explore sequences of events related to the status of variables or data objects. Dataflow testing focuses on points at which variables receive values and the points at which these values are used.

### **e. *Loop Testing***

Loop testing is a white box testing technique that exercises program loops. Several kinds of loop testing include simple, nested, concatenated and unstructured loops.

## 5.5 Levels of Testing

## 5.6 Test Case

## 5.7 Results

## **6 Evaluation of Application Design**

## 7 Discussion and findings

The implementation of application is complex enough to get potential benefit for design principles and also understanding of hybrid technology of cross-platform tools. Below screenshot displayed application difference of interface on iPhone and Android. All these devices has different screen size (aspect ratios) and fonts.



We have separated our findings into three main categories such as user interface, development and platform specific findings.

### **a. *Finding for User Interface***

By developing this application we have notice that user interface generated with common code is almost same on popular mobile platforms because in this application we have used Mobile jQuery framework which provides different UI components and also supported by different mobile platforms. During the test on mobile, it is notice that some fonts changed on different platform because of defaults fonts. In addition to fonts, different size of mobile screens impact the user interface. In this project we have used CSS3 and its media query to manage different orientation of the screens, so different screens will adjust and have similar display. However some template related issue cannot be resolve but we can solve it through platform specific style.

There are quit few issue with UI that exaggerated for few platform, for example animation for page transition flicker on android and not visible on Windows Phone 8. We have also facing difficulty in windows 8 while scrolling down to the end of the page, it is not showing clear content. Majority of the issue has been fixed by the latest version of Mobile jQuery and PhoneGap and it also assume that these UI issue will be fixed in future by these vendors as well.

There is another issue that relate to the keyboard. Keyboard are simulated with the individual platforms and has compatibility issue. Each keyboard has its own size on the screen and also underlying interaction with the application. So it's necessary to test application thoroughly on all the platform with the input fields.

The major issue with the cross platform mobile application is about touch input. Occasionally the touch input of smart devices are a bit slow or missed the event either it pressing of button or swapping around. The touch interruption is rare but it will not give the user a better experience, so need to investigate the issue. Probably this slow response of touch event is due to HTML5, CSS3, JavaScript or DOM tree. May be it need to be optimise.

Usually, support for mobile platform are different with respect to the platform, when dealing with cross platform mobile application, its necessary to verify the touch event for all the platform because in some cases the Mobile jQuery does not support swipe event for Window Phones. It also suggested to test the multi-navigation in mobile platform to confirm.

**b. Findings for Platform Specific**

In addition to user interface issues which were caused by the differences in browser rendering engines, there were a couple of platform specific issues we could not solve or circumvent by modifying the application.

By request from a Cloud Software Program partner, we briefly experimented with the possibility of porting the application to use another backend. During our trials with the second backend which used HTTPS we came across a problem with SSL certificates. The development installation of the backend used a static IP address without a domain name, which meant that browsers could not ensure the authenticity of the certificate. On desktop browsers we could add an exception, and on the Android PhoneGap version we observed no issues. However, we could not get iPhone to create an exception for the server. This meant that the iOS application could not be run against that backend. While the problem is eliminated when the certificate is tied to a domain name, it could be a problem during development as in our case. Certificate handling was not tested using Windows Phone 8.

Another issue we could not solve from within the application was with browser cookies on Windows Phone 8. Our application uses a session cookie received from the server at login to identify the user during subsequent operations. PhoneGap obtains the cookie settings from the browser, but these settings vary between platforms. On Windows Phone 8, we had to change the system wide cookie settings manually on the browser of the device in order to get the application to store the session cookie. This, of course, is not acceptable for a consumer application. The need for cookies could be averted by implementing an authentication token scheme on the client and the server but that would require extra work.

Overall, however, we found cloud-based resource access straightforward and uniform across all platforms.

Native plugins are also a source of platform specific differences. It should be noted that even the plugins that ship with PhoneGap are not supported on all platforms, so the need for native support should be considered early on in a cross-platform development project. We

implemented a native application settings screen on each platform and passed the settings to the HTML application via the plugin interface. Activation of the settings screen was also done via the interface. We found the plugin interface to work quite well. The native side of the plugins can be debugged on platform specific development environments like any native code.

**c. *Finding for Development Method and Tool***

JavaScript is an interpreted language, meaning that without a compiler, the role of the editor in finding programming errors is emphasized.

While all native development environments (Xcode, Eclipse, Visual Studio) support the development of HTML5 applications, none of them in our opinion match the best of dedicated HTML5 editors. Also, the use of a common editor for the HTML5 application by all developers in a project is justifiable in order to establish, e.g., common practices and file templates. While significant parts of an application can be implemented against a desktop web browser, deploying the application on a device, however, requires the native development environment. This causes extra steps and switching between applications in the development process.

We found automated unit testing useful in detecting problems in program logic earlier. Running unit tests with a framework such as Jasmine is quick and isolates program logic issues well. We ran a limited set of unit tests on a desktop browser and because of the ease of running the test suite, unit testing was useful in detecting programming errors quickly. Unit testing frameworks typically provide means for writing stubs, spies and mocks that enable the separation of, e.g., network code from the UI. This helps in isolating program logic issues and programming errors, but in our experience, automated unit testing frameworks are of limited use in exposing issues related to the target platform.

We also found the SW project structure to have significance in cross-platform development. Since in our case the common application code project was included as a subproject in each of the native projects, we occasionally ended up with subproject version conflicts. In the Git version control system the only links between the main repository and the submodules are submodule IDs which are saved in the main repository, and in some situations changes in the IDs are not automatically reflected into the submodules. As a result, we ended up cloning the common module as a separate project into the appropriate directory in each native project, and excluding the directory from version control in the native projects. Automatic refreshing of the subproject during native project refresh was thus lost, but in our case extra manual work caused by that was negligible since the native projects were changed much less frequently than the common project. Native project updates were mostly PhoneGap version updates. In our experience, however, they need to be done with care as PhoneGap version updates usually have to be synchronized between all native projects and the common project. Occasionally, a new PhoneGap version forced us to recreate the native

projects from scratch. The documentation of the new release was also outdated at times, which caused some extra work to solve out the native project upgrade process.

To reduce the need for handling native projects, Adobe offers the cloud-based PhoneGap Build service which builds native applications from the HTML5, JavaScript and CSS code. There are, however, restrictions to custom plugins in PhoneGap Build.

The most significant shortcoming we experienced during development was the limited debugging ability of PhoneGap applications. The reason is that the embedded native browser PhoneGap uses is not accessible to a debugger on every platform, and thus problems that arise only on a specific platform may be very difficult to debug. At the time of writing, only BlackBerry and iOS browsers offer remote debugging that can be extended to PhoneGap applications. The Chrome browser on Android offers remote debugging but not via PhoneGap. Windows Phone 8 lacks remote debugging capability for both of the scenarios. At the time of writing, the best solution for remote debugging of hybrid web applications is Apple's development tools for iOS. Xcode in combination with Safari on Mac offers all required debugging capabilities including DOM tree manipulation, breakpoints and variable inspector.

For most of the time we used a desktop browser for debugging, occasionally augmented by the PhoneGap Emulator on Google Chrome. The emulator was useful in verifying the UI with different screen sizes and resolutions, and getting a hang of using the native interfaces exposed by PhoneGap, although the emulator mostly uses mock data for them. A good rule of thumb for hybrid web application development is to use desktop browsers so that Chrome is used as a preliminary test for Android, Safari for iOS and IE for Windows Phone. Some browsers also have built-in tools for simulating different mobile device screen sizes.

Another useful PhoneGap debugging tool we used is *weinre* that is available either as a local installation or online via [debug.phonegap.com](http://debug.phonegap.com). While *weinre* does not offer breakpoints, it does allow the inspection, highlighting and modification of DOM elements and JavaScript variables via a console.

PhoneGap can also relay the JavaScript `console.log()` output to the development environment console window. We found debug prints to console a viable debugging method, although understandably limited.

#### **d. Summary**

HTML5-based cross-platform applications rely heavily on the web browser on each platform, and differences in how the browsers implement HTML5 features were the underlying cause for most of our findings. In particular, we found occasional platform specific issues with page element layout and certain jQuery Mobile page animations, and touch event support. Most issues were solved by platform specific code and style definitions, but the intermittent problems with touch input responsiveness on all platforms were not.

Issues were also encountered in the way the browsers interact with their surroundings, namely in the visual cue the browsers give on trying to scroll past page boundaries, virtual keyboard behaviour, SSL certificate handling, cookie handling, and PhoneGap plugin support. While some of the issues were remedied via native project settings, solutions were not found during this study for the SSL certificate and cookie problems.

From a developer viewpoint, we found a dedicated HTML editor more useful than native IDEs which are typically not optimized for editing HTML5. Support for debugging on the device is only possible on iOS and Blackberry at the moment, which was found to be the biggest drawback of the approach. When device debugging is not required, desktop browsers provide good debugging options – although their use is not as seamless as debuggers on native IDEs

## **8 Conclusion and future work**



## 9 References & Bibliography

1. International Telecommunication Union, "ICT Fact and figure -- The World in 2015", ITU, 2015 [Online] Available: <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf> Accessed On 01 July 2015
2. Andersson, T., & Johansson, E. (2014). A closer look and comparison of cross-platform development environment for smartphones. *Student thesis*, 1-8.
3. Ciman, M., Gaggi, O., & Gonzo, N. (2014). Cross-Platform Mobile Development: A Study on Apps with Animation. 1-3.
4. Cowart, J. (2013, Nov 12). *Pros and Cons of the Top 5 Cross-Platform Tools*. Retrieved from De Developer Economics: <http://www.developereconomics.com/pros-cons-top-5-cross-platform-tools/>
5. Heitkötter, H., Majchrzak, T. A., & Kuchen, H. (2013). Cross-Platform Model-Driven Development of Mobile Application With MD2. *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 526-533.
6. Simone, S. D. (2014, May 28). *C++ Cross-Platform iOS and Android Development: The Dropbox Lesson*. Retrieved from InfoQ: <http://www.infoq.com/news/2014/05/dropbox-cpp-crossplatform-mobile>
7. Sommer, A., & Krusche, S. (2013). Evaluation of cross-platform frameworks of mobile application. 363-376.
8. Xanthopoulos, S., & Xinogalos, S. (2013). A comparative analysis of cross-platform development Approaches of mobile application. *Proceedings of the 6th Balkan Conference in Informatics*, 213-220.
9. Skyscanner. (February 2014). *Skyscanner Blog*. Available: <http://www.skyscanner.net/blogs/developing-mobile-cross-platform-library-part-1-exploring>. Last accessed 18th June 2015.
10. Anthony I. Wasserman (M/S 23-14) *Software Engineering Issues for Mobile Application Development*, Moffett Field, CA 94035 USA: Carnegie Mellon Silicon Valley.
11. Snecha (2002) Create native-looking HTML5 apps using JavaScript, : [www.snecha.com](http://www.snecha.com).
12. Yonathan Aklilu Redda (2002) *Cross platform Mobile Applications Development*, June 2012: Norwegian University of Science and Technology.
13. Lionbridge (2002) *Mobile Web Apps vs. Mobile Native Apps: How to Make the Right Choice*, [www.lionbridge.com](http://www.lionbridge.com) edn., June 2002: Lionbridge.

14. Suyesh Amatya, Dr. Arianit Kurti (2013-10-29 ) *Cross-Platform Mobile Development: An Alternative to Native Mobile Development* , Department of Computer Science edn., Sweden: Linnaeus University.
15. Nabil Litayem, Bhawna Dhupia, Sadia Rubab (2015) *Review of Cross-Platforms for Mobile Learning Application Development*, Computer Science and Information edn., Kingdom of Saudi Arabia: University of Carthage, TUNISIA.
16. Intel IT (July 2013) *Implementing a Cross-Platform Enterprise Mobile Application Framework*, Mobile Application Development edn., United Kingdom: Intel Corporation.
17. Christopher Reynolds (October 16th, 2014 ) *Ten of the Best Cross-Platform Mobile Development Tools for Enterprises*, <http://appindex.com/blog/ten-best-cross-platform-development-mobile-enterprises/> edn., : Appindex.
18. Chelsea Baldwin (2001) *A guide to the 10 best cross platform mobile development tools*, <http://thinkapps.com/blog/development/develop-for-ios-v-android-cross-platform-tools/> edn., Think Apps: Build Dog.
19. Blue Tube (May 16, 2014) *Cross Platform Mobile Development Tools Comparison*, <http://bluetubeinc.com/blog/2014/5/cross-platform-mobile-development-tools-comparison> edn., Appcelerator Industry Trends Mobile and Tablet Xamarin Technical
20. Gustavo Hartmann, Geoff Stead, Asi DeGani ( March 2011 ) *Cross-platform mobile development*, 1 st edition edn., United Kingdom: Tribal Mobile Development.
21. John M. Wargo (2014) *PhoneGap Essentials Building Cross-Platform Mobile Apps*, Crawfordsville, Indiana: Addison Wesley.
22. Rich Morrow (10/30/2014 ) *Sector Roadmap: cross-platform mobile development* , Oracle edn., United States: OM Research.
23. Otso Kassinen, Erkki Harjula, Timo Koskela, Mika Ylianttila ( July 2010 ) *Guidelines for the Implementation of Cross-platform Mobile Middleware*, Volume 4 edn., Finland : University of Oulu .
24. Christian G. Acord, Corey C. Murphy (March 2012) *CROSS-PLATFORM MOBILE APPLICATION DEVELOPMENT: A PATTERN-BASED APPROACH* , MONTEREY, CALIFORNIA : NAVAL POSTGRADUATE SCHOOL .
25. Mark Power (May 2012) *Delivering Web to Mobile*, TechWatch Report final version 1.0 edn., United kingdom: JISC Observatory.

26. Niyigena Jean Pierre, Fan Xiumei, Gakwaya Daniel, and Gombanero Jean Claude (May 2015) *Cross-Platform Mobile Geolocation Applications Based on PhoneGap*, Vol. 3 edn., : <http://www.lnse.org/>.
27. Opera Mobile 10 beta for Opera Mini 5 for J2ME and Blackberry: (n.d.) *Opera cross-platform UI framework*, Norway: Opera Software.
28. Andreas Sommer, Stephan Krusche (n.d.) *Evaluation of cross-platform frameworks for mobile applications*, Version 1.0 edn., denmark: <http://subs.emis.de/>.
29. Paulo R. M. de Andrade, Adriano B. Albuquerque (February 2015) *CROSS PLATFORM APP A COMPARATIVE STUDY*, Vol 7, No 1 edn., Fortaleza - CE, Brazil: College Estacio FIC of Ceara Fortaleza - CE, Brazil.
30. Simon Roses Femerling (March 2014) *The Inner Workings of Mobile Cross Platform Technologies*, Vulnex edn., Singapore: BLACK HAT ASIA .
31. Research 2 Guidance (October 2013) *Cross Platform Tool Benchmarking 2013*, <http://c-sharx.net/>: Telerik.
32. Avinash Shrivastava, Anandkumar Pardeshi (September 2013) To Study and Design a Cross-Platform Mobile Application for Student Information System using PhoneGap Framework, Volume 3, Issue 9 edn., Mumbai, India: IJeteae.
33. Matteo Ciman, Ombretta Gaggi, Nicola Gonzo (n.d.) *Cross-Platform Mobile Development: A Study on Apps with Animations*, Volume : 1 edn., Padua, Italy: University of Padua.
34. (December 11-12) *Sample Schedule Cross Platform*, Arlington, VA: modevcon.com
35. John Lyle, Salvatore Monteleone, Shamal Faily, Davide Patti and Fabio Ricciato (n.d.) *Cross-platform access control for mobile web applications*, Volume : 1 edn., Italy, Europe: Department of Electrical, Electronic and Computer Engineering (DIEEI), University of Catania.
36. Joshua Barnett (n.d.) *Building a Cross-Platform Mobile Game with HTML5*, Anglia: University of East Anglia Faculty of Science School of Computing Sciences.
37. Nielson (MARCH 2014) *AN ERA OF GROWTH THE CROSS-PLATFORM REPORT*, THE CROSS-PLATFORM SERIES | Q4 2013 edn., DOUNIA TURRILL SVP INSIGHTS: Penngood.
38. William Van Woensel, Newres Al Haider, Ahmad Ahmad, and Syed SR Abidi (n.d.) *A Cross-Platform Benchmark Framework for Mobile Semantic Web Reasoning Engines*, NICHE Research Group, Faculty of Computer Science edn., Canada: Dalhousie University, Halifax, Canada.

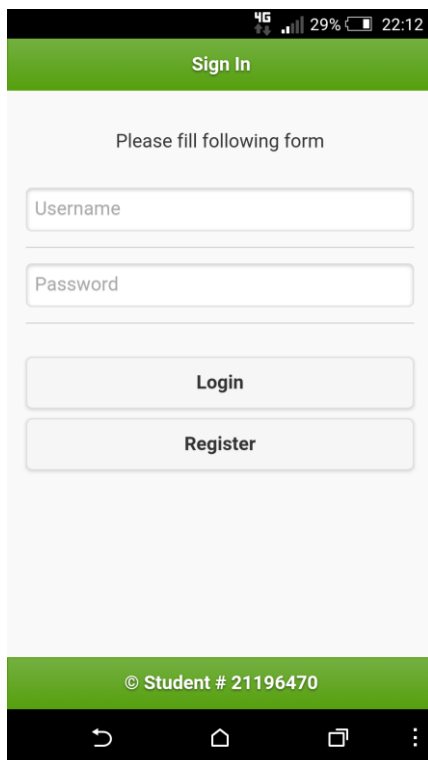
39. Ambrose A. Azeta, Ph.D. Abimbola Esuruoso, B.Sc.and A.E. Azeta, HND (n.d.) *An Automated Cross-Platform Mobile Bible Application.* , Department of Computer and Information Sciences, : Covenant University, Ogun State, Nigeria.
40. www.diamondkinetics.com (n.d.) *Cross-Platform Mobile Software Engineer Pittsburgh, PA*, Version : 1.0 edn., Pittsburgh: Diamond Kinetics.
41. José Calla Guzmán,Victor Cernea (2012) *Cross-platform development of mobile applications*, Evaluation of the PhoneGap framework edn., CHALMERS UNIVERSITY OF TECHNOLOGY, Gothenburg: Chalmers.
42. Dr. Olaf Grebner (n.d.) *Develop Mobile Cross-Platform Apps For Business (with Ionic, Angular, JS)*, : mobilion.eu.
43. Jesse Liberty (2012) *Learning to Master Cross-Platform Mobile Development with Xamarin*, Scotts Valley, California: Falafel Software.
44. Mathieu Bruning (November 29, 2011) *Native Cross-platform Mobile Application Development Using Voind*, Netherlands: Faculty EEMCS, Delft University of Technology Delft, the Netherlands.
45. Pilgrim, M. *HTML5: Up And Running.* O'Reilly Media / Google Press, 2010
46. Hickson, HTML5. W3C Editor's Draft, W3C, Jan 2012. Available at: <http://dev.w3.org/html5/spec/Overview.html>. Accessed 15-Aug-2015

## 10 Online References

1. <http://www.skyscanner.net/blogs/developing-mobile-cross-platform-library-part-1-exploring>
2. <http://www.cmu.edu/silicon-valley/wmse/wasserman-foser2010.pdf>
3. <https://www.sencha.com/products/touch/#overview>
4. <http://www.diva-portal.org/smash/get/diva2:626531/fulltext01.pdf>
5. [http://www.lionbridge.com/files/2012/11/Lionbridge-WP\\_MobileApps2.pdf](http://www.lionbridge.com/files/2012/11/Lionbridge-WP_MobileApps2.pdf)
6. <http://www.diva-portal.org/smash/get/diva2:664680/fulltext01.pdf>
7. [http://thesai.org/Downloads/Volume6No1/Paper\\_5-Review\\_of\\_Cross-Platforms\\_for\\_Mobile\\_Learning\\_Application\\_Development.pdf](http://thesai.org/Downloads/Volume6No1/Paper_5-Review_of_Cross-Platforms_for_Mobile_Learning_Application_Development.pdf)
8. <http://www.intel.co.uk/content/dam/www/public/us/en/documents/best-practices/implementing-a-cross-platform-enterprise-mobile-application-framework-paper.pdf>
9. <http://appindex.com/blog/ten-best-cross-platform-development-mobile-enterprises/>
10. <http://www.diva-portal.org/smash/get/diva2:691708/FULLTEXT01>
11. <http://www.diva-portal.org/smash/get/diva2:664680/fulltext01.pdf>
12. <http://appindex.com/blog/ten-best-cross-platform-development-mobile-enterprises/>
13. <http://thinkapps.com/blog/development/develop-for-ios-v-android-cross-platform-tools/>
14. <http://bluetubeinc.com/blog/2014/5/cross-platform-mobile-development-tools-comparison>
15. <http://www.oracle.com/us/corporate/analystreports/gigaom-sector-roadmap-2355064.pdf>
16. <http://www.dtic.mil/dtic/tr/fulltext/u2/a560748.pdf>
17. <http://observatory.jisc.ac.uk/docs/delivering-web-to-mobile.pdf>
18. <http://www.lnse.org/vol3/170-P029.pdf>

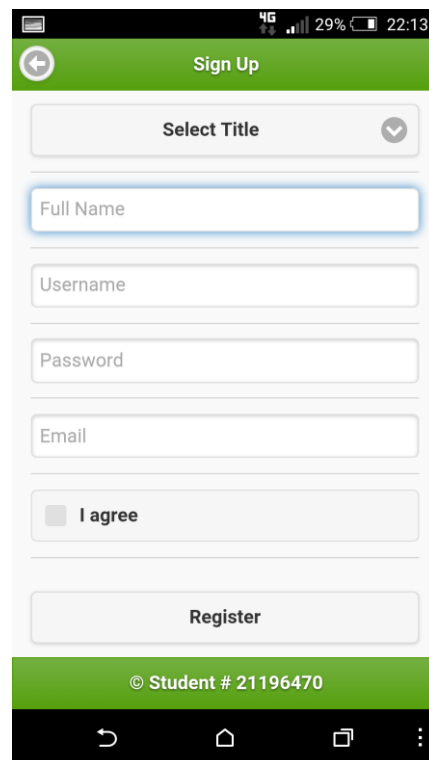
19. [http://media.opera.com/media/b2b/Opera\\_Mobile\\_Product\\_Sheet.pdf](http://media.opera.com/media/b2b/Opera_Mobile_Product_Sheet.pdf)
20. <http://arxiv.org/pdf/1503.03511.pdf>
21. <https://www.blackhat.com/docs/asia-14/materials/RosesFemerling/Asia-14-RosesFemerling-The-Inner-Workings-Of-Mobile-Cross-Platform-Technologies.pdf>
22. <http://c-sharx.net/wordpress/wp-content/uploads/2013/12/Cross-Platform-Tool-Benchmarking-2013.pdf>
23. [http://www.ijetae.com/files/Volume3Issue9/IJETAE\\_0913\\_61.pdf](http://www.ijetae.com/files/Volume3Issue9/IJETAE_0913_61.pdf)
24. <http://www.math.unipd.it/~gaggi/doc/SAC14.pdf>
25. <http://mdc.gomodev.com/files/2014/11/Sample-Schedule-Cross-Platform.pdf>
26. <https://www.cs.ox.ac.uk/files/4723/policy2012.pdf>
27. [http://www.caffeinatednightmare.com/blog/wp-content/uploads/2014/11/joshuabarnett\\_2014.pdf](http://www.caffeinatednightmare.com/blog/wp-content/uploads/2014/11/joshuabarnett_2014.pdf)
28. <http://penngood.com/wp-content/uploads/2014/03/nielsen-cross-platform-report-march-2014.pdf>
29. [http://www.akamaiuniversity.us/PJST15\\_1\\_141.pdf](http://www.akamaiuniversity.us/PJST15_1_141.pdf)
30. <http://diamondkinetics.com/wp-content/uploads/2013/08/Diamond-Kinetics-Mobile-Developer.pdf>
31. <http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/managing-multiple-devices.pdf>
32. <http://publications.lib.chalmers.se/records/fulltext/167502.pdf>

## Appendix (Application Screens)



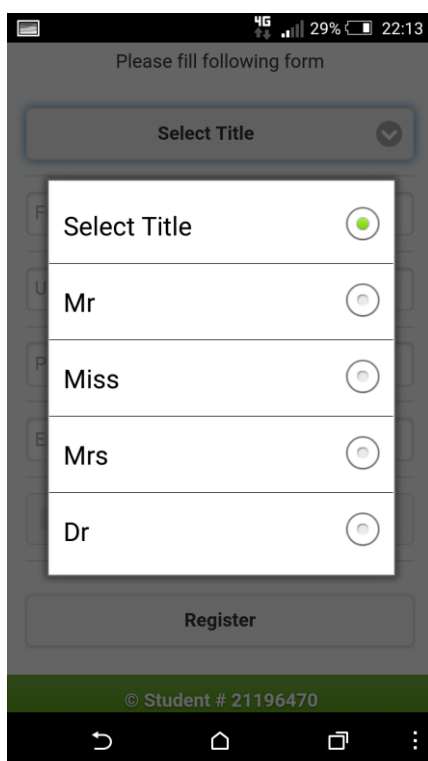
The Sign In screen features a green header with the text "Sign In". Below the header, it says "Please fill following form". There are two input fields: "Username" and "Password". Below these fields are two buttons: "Login" and "Register". At the bottom, there is a green footer with the text "© Student # 21196470".

Login



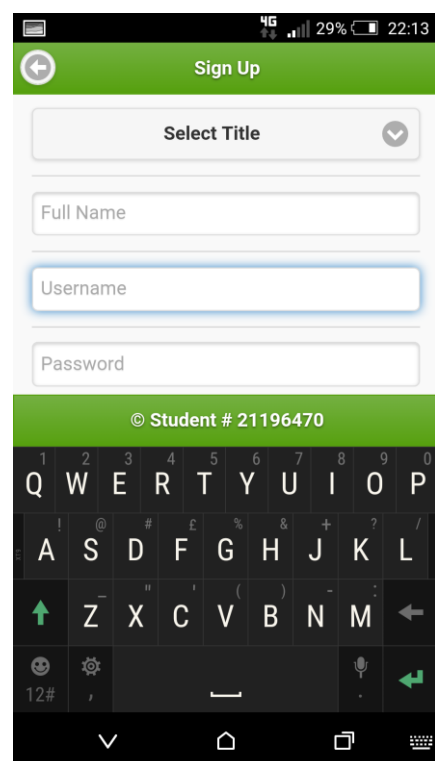
The Sign Up screen features a green header with the text "Sign Up" and a back arrow. Below the header, there is a "Select Title" dropdown menu. There are four input fields: "Full Name", "Username", "Password", and "Email". Below these fields is a checkbox labeled "I agree". At the bottom is a "Register" button. At the very bottom, there is a green footer with the text "© Student # 21196470".

Rgister



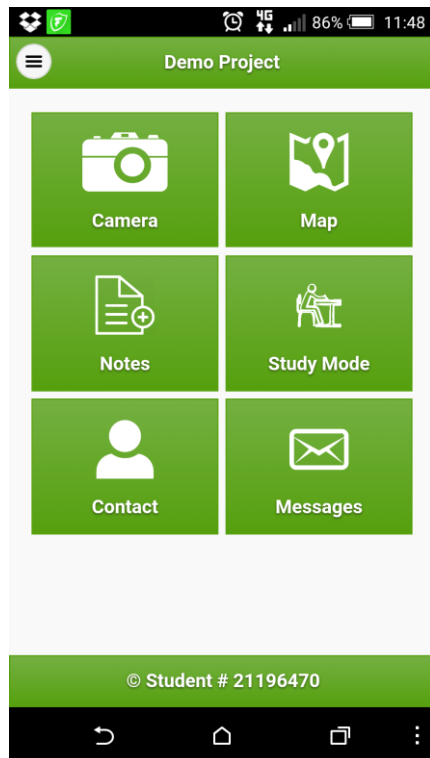
This screen shows the Sign Up form with the "Select Title" dropdown menu open. The menu lists five options: "Select Title" (with a green circle), "Mr", "Miss", "Mrs", and "Dr", each with a radio button. The "Register" button is visible below the menu. At the bottom, there is a green footer with the text "© Student # 21196470".

Register with select menu

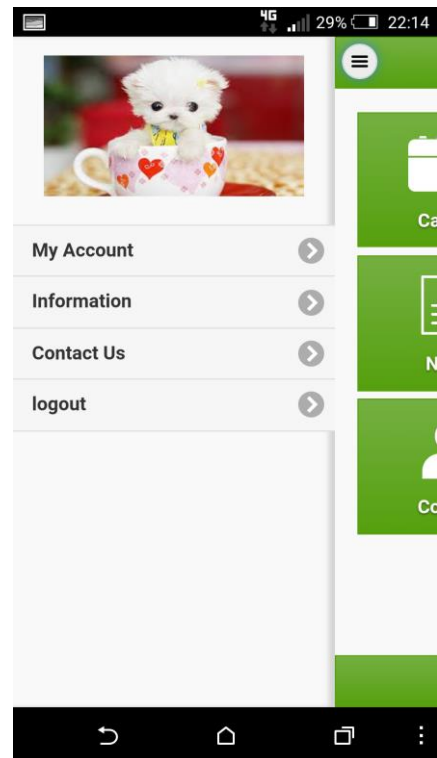


This screen shows the Sign Up form with the "Username" field selected, and a virtual keyboard is displayed at the bottom. The keyboard includes letters, numbers, and symbols. At the bottom, there is a green footer with the text "© Student # 21196470".

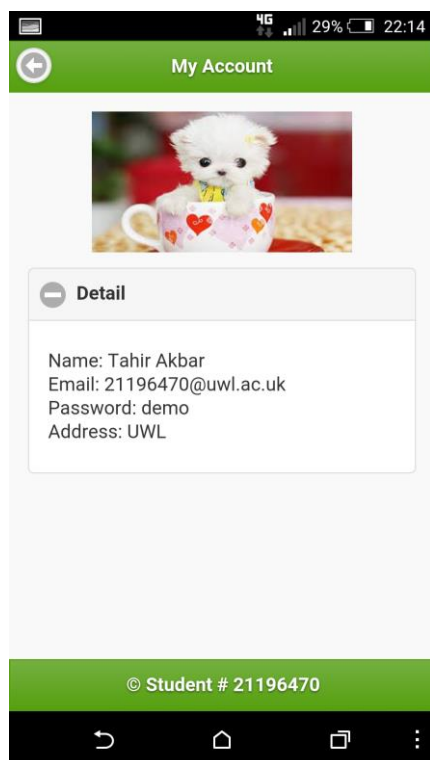
Keyboard option



Dashboard



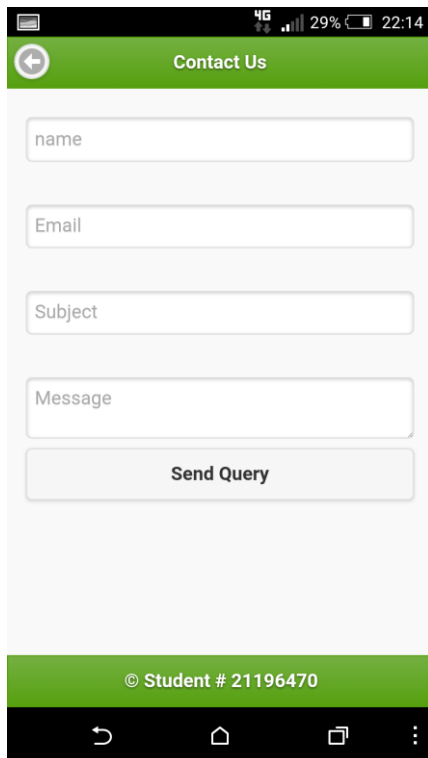
Side Menu Option



My Account Option

My Information.





Contact Us

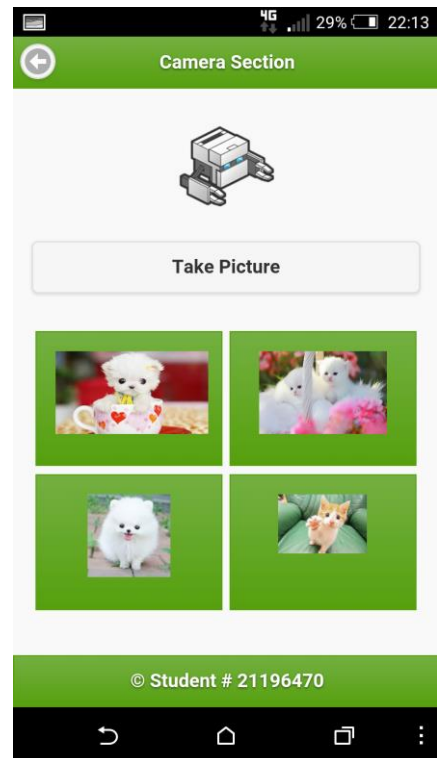
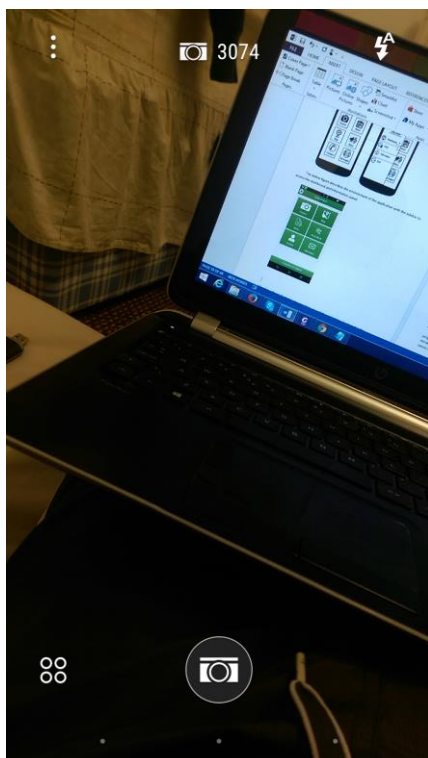
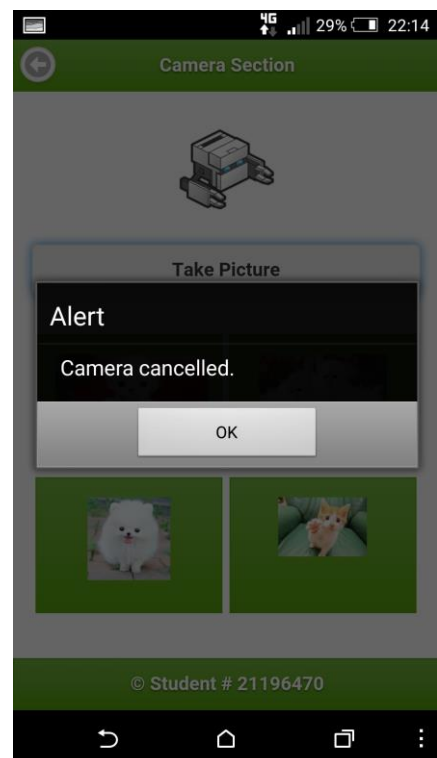


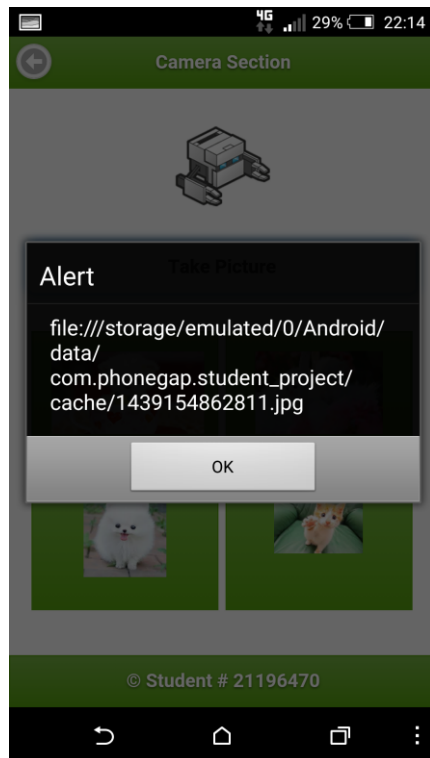
Image Gallery



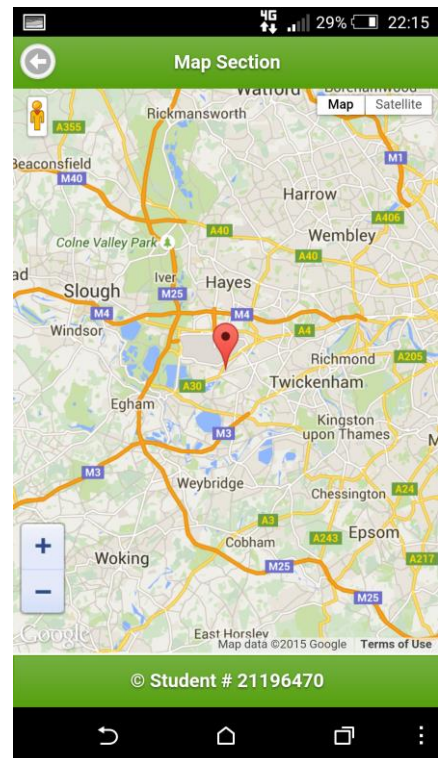
Camera Feature Image Taking



Cancel Image Taking



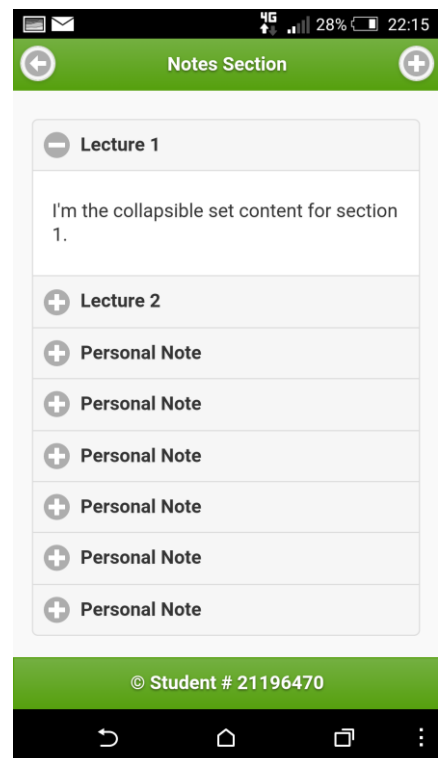
Success Image Capturing



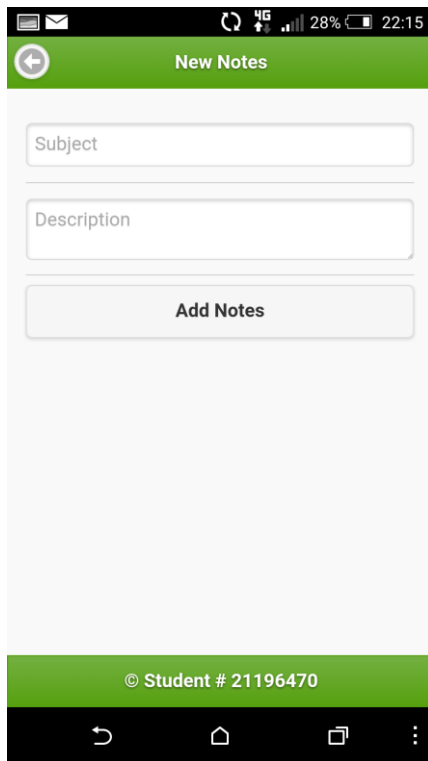
My Current location



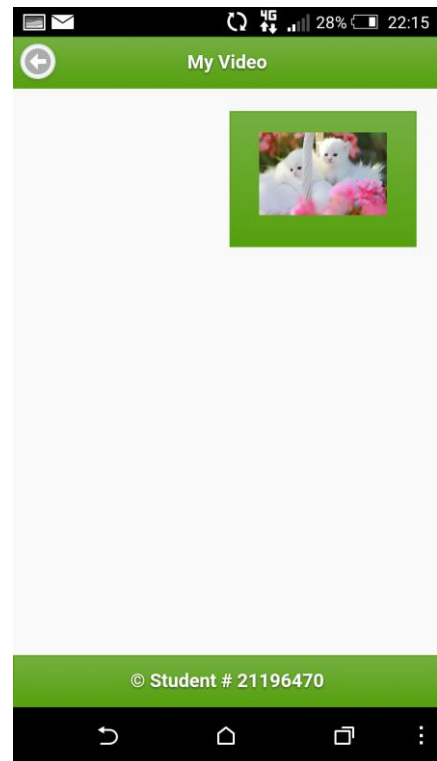
My Current Location Street View



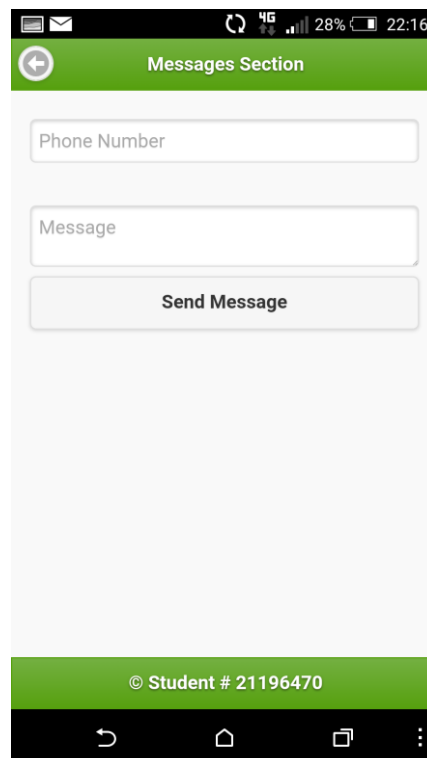
My Notes List



Add My Notes



My Video Gallery



Send A SMS section

The above figures illustrates the Android view of the application developed in the PhoneGap environment.

