

Communicating Beacons using the Hybrid Mobile App (IoT)

BITS ZG628T: Dissertation

by

PAVITHRA H M

2014HT13053

Dissertation work carried out at

Tyfone CDI, Bangalore



**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN)**

November 2016

Communicating Beacons using the Hybrid Mobile App (IoT)

BITS ZG628T: Dissertation

by

PAVITHRA H M

2014HT13053

**Dissertation work carried out at
Tyfone CDI, Bangalore**

Submitted in partial fulfillment of M.Tech. Software Systems degree programme

Under the Supervision of
Manjunath Prasad, Technical Architect,
Tyfone CDI, Bangalore



**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN)**

November, 2016

CERTIFICATE

This is to certify that the Dissertation entitled Communicating Beacons using the Hybrid Mobile App (IoT) and submitted by Pavithra H M having ID-No. 2014HT13053 for the partial fulfillment of the requirements of M.Tech. Software Systems degree of BITS, embodies the bonafide work done by her under my supervision.



Signature of the Supervisor

Place : Bangalore

Date : 14/10/2016

S. R. Manjunath Prasad, Software Architect

Name, Designation & Organization & Location

Birla Institute of Technology & Science, Pilani
Work-Integrated Learning Programmes Division

First Semester 2016-2017

BITS ZG628T: Dissertation

ABSTRACT

BITS ID No.	:	2014HT13053
NAME OF THE STUDENT	:	PAVITHRA H M
EMAIL ADDRESS	:	pavithra.hm21@gmail.com
STUDENT'S EMPLOYING ORGANIZATION & LOCATION	:	Tyfone CDI, Bangalore
SUPERVISOR'S NAME	:	Manjunath Prasad
SUPERVISOR'S EMPLOYING ORGANIZATION & LOCATION	:	Tyfone CDI, Bangalore
SUPERVISOR'S EMAIL ADDRESS	:	prasad@tyfone.com
DISSERTATION TITLE	:	Communicating Beacons using the Hybrid Mobile App (IoT)

ABSTRACT: This dissertation explores the maintenance and development of a mobile application that works on multiple mobile operating systems such as Android and iOS with the same source code. The mobile application was implemented with HTML5 programming and wrapped with PhoneGap.

Hybrid mobile applications are like any other applications you'll find on your phone. Like the websites on the internet, hybrid mobile applications are built with a combination of web technologies like HTML, CSS3, and JavaScript. The vital difference is that hybrid mobile applications are hosted inside a native application that utilizes a mobile platform's WebView(Device default webkit). This enables them to access device capabilities such that we can talk to device hardware. Application assets like HTML, CSS3, JavaScript are packaged through the tool made available by Apache Cordova to target platform SDKs. A beacon is a wireless device that continuously transmits a signal informing the recipient about its position. The "Retailers App" is a Hybrid mobile application which can be built for multiple platforms (Android & iOS) with the same source code. The app will talk to the beacons located in the stores and provides its accurate position. Based on the location, the app will fetch the appropriate details/offers from the server. The challenge is to use the hybrid mobile application to talk to the beacon using native plugins. The native plugins which act as an interfaces between the web technology and the native OS.

The outcome of the dissertation validates that HTML5 programming as a considerable approach to develop cross-platform mobile application and communicating with beacons.

Broad Academic Area of Work: Wireless and Mobile Computing.

Key words: Android, AngularJS, Beacons, Cordova and Ionic Framework.



Signature of the Student

Name: PANITHRA HM

Date: 19/09/2016

Place: Bangalore.



Signature of the Supervisor

Name: S.K. Manjunath Prasad

Date: 19/09/2016

Place: Bangalore

ACKNOWLEDGEMENTS

I have taken efforts in this dissertation. But, it would not have been possible without the kind support and help of many individuals and my organization. I would like to extend my sincere thanks to all of them.

I am highly indebted to Manjunath Prasad for his guidance and constant supervision as well as for providing necessary information regarding the thesis project and also for his support in completing the project.

I would like to convey my gratitude towards my parents & member of Tyfone for their kind co-operation and encouragement which help me in completion of this project.

I would like to convey my very profound gratitude and thanks to industry persons for giving me such attention and time. My thanks and appreciations also go to my supervisor in developing the thesis project and people who have willingly helped me out with their abilities.

TABLE OF CONTENT

CHAPTER 1	1
1 INTRODUCTION	1
1.1 Project Goal	2
1.2 Project Scope	2
CHAPTER 2	3
2 BACKGROUND ON MOBILE APPLICATIONS AND BEACONS	3
2.1 Rising Demand for Mobile Applications And IoT	3
2.2 Strategies for Developing a Mobile Application	5
CHAPTER 3	7
3 REQUIREMENT ANALYSIS	7
CHAPTER 4	9
4 PROJECT MODEL AND DESIGN	9
4.1 Project Model	9
4.2 Design Features	10
4.2.1 High Level Design	10
CHAPTER 5	12
5 IMPLEMENTATION	12
5.1 Implemented Languages	12
5.2 Implemented Libraries and Frameworks	13
5.3 Implementation of Screen Views	18
5.4 Storage Implementation	19
5.5 Fetch and View Offers	20
5.6 Change Profile and Invite Friends	23
CHAPTER 6	26
6. TESTING AND RESULT	26
SUMMARY	27
CONCLUSION	28
BIBLIOGRAPHY / REFERENCES	29

APPENDICES

Appendix 1. HTML code for form without Ionic Framework	30
Appendix 2. HTML code for form with Ionic Framework	30
Appendix 3. Data storage	31
Appendix 4. Navigating between views	32
Appendix 5. Offer object with the properties	32
Appendix 6. HTML code for angular validations	33

LIST OF FIGURES

Figure 1 Use case diagram that captures the requirements of the Retailers App	8
Figure 2 Iterative model diagram	10
Figure 3 High Level Design Retailers App	11
Figure 4. Mobile operating systems supported on PhoneGap framework	15
Figure 5. PhoneGap Build compiles and deploys mobile application	17
Figure 6. Responsiveness	19
Figure 7. Fetching offers with the Google Places API	20
Figure 8. Fetching offers with the Proximity Beacon API	21
Figure 9. Offers View	22
Figure 10. Detailed Offer View	23
Figure 11(a). Change Profile Settings	24
Figure 11(b). Change Profile Settings with validations	24
Figure 12. Invite Friends View	25

LIST OF TABLES

Table 1. Worldwide Market Share by Operating System	4
Table 2. Native code language fragmentation	6
Table 3. Supported features on PhoneGap	16
Table 4. Testing on different devices and platforms	27

LIST OF ABBREVIATIONS (OR) SYMBOLS

API	API stands for Application Programming Interface.
IoT	Internet of Things.
CSS	CSS stands for Cascading Styling Sheets.
HTML	HTML, which stands for Hypertext Markup Language, is the predominant markup language for web pages.
IDE	Integrated Development Environment.
JSON	JavaScript Object notation.
OS	Operating System.
SDK	Software Development Kit.
UI	User Interface.
app	Application.
iOS	Apple's Mobile Operating System
GPS	Global Positioning System

CHAPTER 1

1 INTRODUCTION

The mobile applications development took a vital turn when Apple and Google launched the Apple app store and Android Google Play Store in 2008. Both app stores were set up to allow the developers to maintain and publish applications for consumers of appropriate mobile operating system platform. Companies, such as Nokia, RIM, Microsoft and others, followed the same trend to catch up the competition for mobile market share.

However, an application published in Apple app store which can be only native to the Apple's iOS. Which means it is implemented, compiled and built in Objective C/Swift language and developed in Apple's SDK and IDE. To publish the same app product on Android would require a different language, SDK and IDE. The challenge task for developers is that the most of the mobile operating system requires a different language, SDK and IDE to develop native applications to a specific platform. Therefore, publishing the same application product for the consumers on multiple platforms can be a very challenging job to the developers because it can lead to variances in user level experience. The concept of cross platform application is writing an app with a language that is supported on a cross-platform SDK. The good news is that technologies to allow develop once app and publish with same user experience across multiple mobile operating system platforms are prominent. A Cross-platform SDK does not necessarily support all available mobile operating systems but can at least support the major ones such as Android and iOS. Another limitations is that a functionality that is accessed through a pre defined native interface may not be supported on a cross-platform SDK but still we can achieve by writing the appropriate cordova plugins(native interfaces). Hence, when developing a cross-platform application, it is very important to check if the constraints should not affect a user level experience across the targeted mobile operating system platforms. The PhoneGap/Cordova provides an implementation with that we can avoid any limitations occurred.

The “Internet of Things” (IoT) is a rapidly vast increasing topic of conversation in internet technology, which leads to new business income/ opportunities. A beacon is a wireless device that continuously transmits a signal informing the recipient about its position (location). Compared to devices based on GPS, a beacon provides better accuracy in terms of location. Range of a beacon can go up to 70m. A beacon can constantly sends out radio signals in form of alphanumeric characters to nearby

smartphones and tablets. The “Retailers App” using the kontakt beacons api.

1.1 Project Goal

The goal of this thesis is to develop a “Retailers App” that provides the same user level experience on majority of the targeted mobile operating system platforms and communicating with beacons using the same source code and by creating additional IoT interfaces. The application should run on multiple mobile operating system platforms without making any changes to the source code. It also hopes to produce an application with a simple user interface in order to make it easy for the user.

1.2 Project Scope

The dissertation covers the development and implementation of the application on PhoneGap/Cordova development framework. At the end, the working features of the app are:

- Four screens - Home, Details, Profile and Invite
- Ability to view the offers/details on demand.
- Ability to invite others of using an app.
- Ability to change the profile details (Name, Email and Phone Number)

CHAPTER 2

2 BACKGROUND ON MOBILE APPLICATIONS AND BEACONS

2.1 Rising Demand for Mobile Applications And IoT

Mobile applications usage is becoming an very increasingly popular activity of daily life around the world. Similarly, users of smart phones have also rapidly increased over the last decade. The proliferation of smart phones targeted towards consumers has led to a vast expansion of mobile apps development. This new era in mobile applications development creates a new business/opportunities for developers to earn money in two ways. One way that generates business for developers is to publish apps to consumers/users through app market stores. The developers earn their income by selling the apps or offer them for free and generate income/value from ads. The other is that it generates business for developers is developing applications to offer services to their clients/organizations. For example, a bank may have a mobile banking application through which their clients can perform the bank transactions.

There has been observed a massive increase of iOS and android mobile operating system platforms which has been a major factor in the rise of mobile applications development. Android and iOS account for 94.6% of the smart phones worldwide shipment in 2015 Q2. While Blackberry, Symbian, Windows Phone and other platforms accounted for the rest 5.4% of smart phones shipped worldwide. The worldwide smartphone market grew more than 13.0% year by year in 2015 Q2, with 341.5 million shipments, according to data from the International Data Corporation (IDC press release 2015 Q2).

Android acquires the largest market share with an 82.8% share in 2015 Q2. Samsung reacquired its global mobile leadership with a renewed focus on lower-cost smartphones. Apple enjoyed success thanks to consumers supporting for the larger screened iOS devices. The popularity of the iPhone's with the bigger screens such as iPhone 6 & 6 Plus continued in many key markets including China, where the overall smartphone market saw a revival in growth by 6.7%.

Table 1 Worldwide Market Share by Operating System

Period	Android	iOS	Windows Phone	Blackberry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

The facts stated above indicate that the demand for application development for these various mobile operating system platforms is on the increase. Prior to the introduction of iPhones by Apple, the majority of the mobile applications developed were sold with mobile devices as predefined packages of mobile devices before selling them to consumers.

Internet of Things

The Internet of Things (or the IoT for short) is a term for a various terms of technologies that describe a number of possible different ways to connect/communicate the physical things(smart devices) to the Internet. You've heard about smart cars, homes and even cities that can identify their users, predict their needs and address to them ahead of particular time. As a result, any item can become smart device and context-aware, will provide highly personalized experiences and gathers rich data about what's happening in the real world. That's what Internet of Things is all. It is not a particular technology; rather, it's a set of multiple technologies that can change the way that we're interacting/communicating with the world. And proximity API is one of them.

A beacon start repeatedly transmits a single signal that other Bluetooth devices can see. A beacon broadcasts a radio signal that's made up of a combination of alpha numeric characters transmitted on a regular interval. A Bluetooth-equipped device like a smartphone can identify a beacon which is monitoring and once it is in range, much like users using a GPS to know where they are.

2.2 Strategies for Developing a Mobile Application

Many people define mobile application as any application that runs on a mobile device which is usually a mobile web application or a native application. This thesis defines mobile application as native application. A native application is one for which need to create an app and sign that is available for purchase in mobile stores (e.g., iTunes, Android market).

A mobile application can be developed using one of the three main coding strategies. They are

- Native code
- HTML5 code
- Hybrid code

Native Mobile Application

A native code mobile application uses the native code of the targeted mobile device operating system platform for development. Table 2 shows a list of the native code languages for each mobile operating system. Applications developed with native codes have the best look and performance. They also have the fastest graphics which is very important when playing graphic intensive games. Developing an application with native code provides a developer with access to all the built-in features required to provide a user experience that most people expect on that device

Native code-based mobile applications require an IDE environment which provides the tools for debugging, managing the project etc. A comprehensive knowledge of the native code language is very important when developing an application. Hence, it is not uncommon for a developer to be limited to just one mobile device platform.

Table 2 Native code language fragmentation

Mobile OS	Native Code Language
Apple iOS	Objective C and Swift
Google Android	Java
RIM Blackberry	Java (J2ME flavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows Phone	.NET, C#
HP Palm webOS	HTML/CSS/JS

HTML5 Mobile Application

An HTML5 code mobile application is an application developed by using HTML5 code combined with JavaScript and CSS. The greatest advantage with an application purely based on HTML5 is that it can easily be adapted to various mobile operating system platforms. Most developers are already familiar with using HTML5, JavaScript and CSS, thus , it is relatively easy to use it to develop a mobile application. The drawback of developing an application with pure HTML5 coding is that it cannot access all the built in features that are available on a device. HTML5-coded mobile apps can be developed on any of the cross-platform software development platform such as PhoneGap. The thesis project adapts this strategy to develop a mobile application in order to produce an application that can run on multiple mobile device platforms.

Hybrid Mobile Application

A hybrid code mobile application is an application that is developed by combining native code and HTML5 code. The purpose of integrating native codes with HTML codes is to leverage on the best of both strategy of app development. Integrating native codes can be used to address the limitations that are faced with HTML5 limitations. However, a developer would still require native code language skills to successfully implement an hybrid code approach. Just like HTML5 code mobile apps, hybrid apps can be developed on any of the cross-platform SDK , such as PhoneGap.

CHAPTER 3

3 REQUIREMENT ANALYSIS

The requirement analysis encompasses the tasks involved in determining user expectations for a new or modified product or software application (Rouse 2007). The tasks involve getting to know the functional requirements which means what the system is required to do. Thus, a good understanding of the application requirements is needed in order to determine the specific features that should be implemented. These features must be quantifiable, testable, and relevant to the identified requirements and defined to a level of detail sufficient for designing the application.

Existing applications related to the dissertation app were analyzed to identify the functional requirements. In addition, user feedbacks on the applications that were analyzed were also reviewed. This helps to develop the features of the thesis project application in line with what users expect

Use Cases

A use case captures the functionality that the application provides in fulfilling one or more of the users requirements. The use case diagram in Figure 1 is an instrumental model in the beginning of the project because it describes what actions the app users can perform in an overview. However, it is important to point out that it does not provide details that are sufficient in the actual designing and exact understanding of how the requirements will be met. More details and changes will be needed later during the design and implementation stage.

Retailers App mobile application requirements

The use case diagram in Figure 1 can be detailed into understandable terms as given below:

- The user shall be able to view the list of offers.
- The user shall be able to view the details of the offers.
- The user shall be able to invite friends or family members to use the application.
- The user shall be able to modify the profile settings.

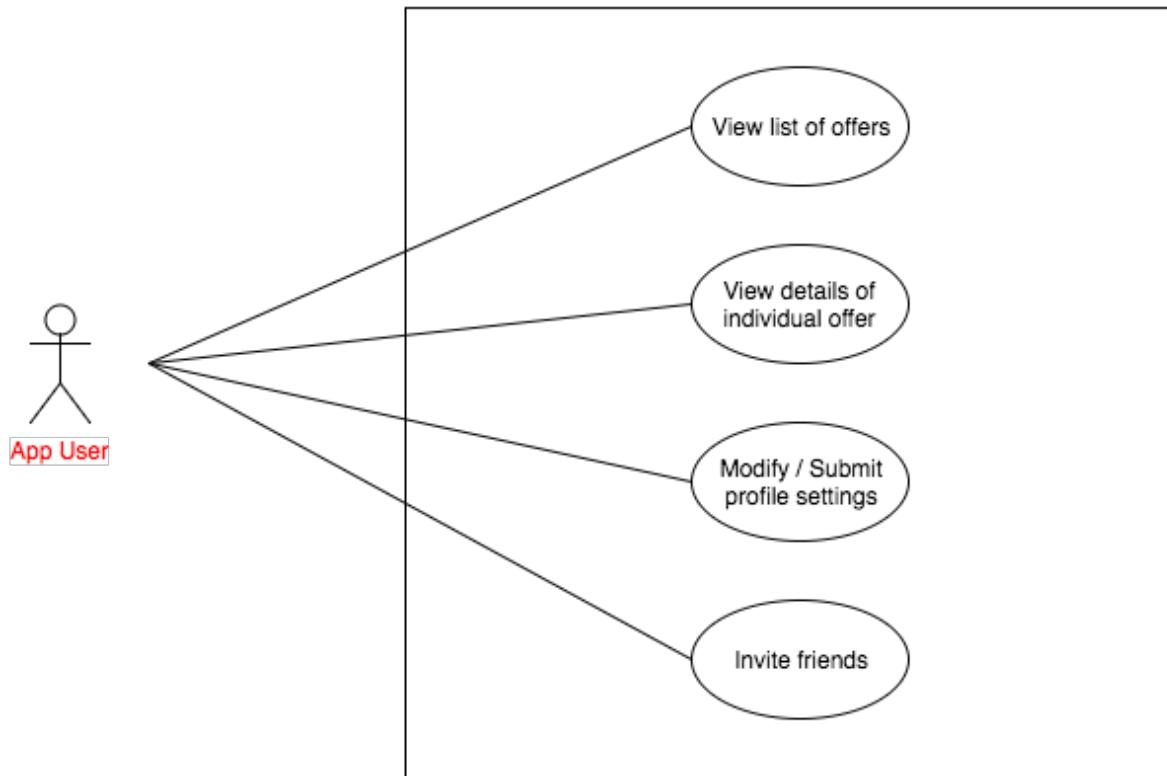


Figure 1 Use case diagram that captures the requirements of the Retailers App

User struggles to find the details / offers which are applicable in stores. Until user visit the store, unable to find the applicable offers.

User can view the details such as it can be dynamic content which can get from server or it can be an external webpage where user can find the needed info. we collecting the user contact details for to notify the details / hot offers based on user's favorite search. User can submit either phone or email, it's either one be an optional to submit the contact info.

Why invite, invite friends is an option providing to the user, to share the application details to the friends, who can be install the application and can experience the same.

CHAPTER 4

4 PROJECT MODEL AND DESIGN

4.1 Project Model

The approach to the design of the project is iterative development. As seen in Figure 4, there is a cycle of requirements, design, implement, test and evaluate. These five components make up the model that is adapted in developing the thesis project. The potential success of the mobile apps lies in the functionalities and good user experience. It is also important to distinguish the app in certain aspects from other apps available. Hence, the cycle of the five model components iterates till there is a reasonably successful outcome of an application that fulfills the goal of the thesis project. The discussions on design and implementation will focus on the final version of the thesis project. A brief overview of the five model components of the thesis project is give below:

Requirements: Details of this component have been discussed in the literature review chapter. Iterating through this stage produces further requirements till the final specification for the app is developed.

Design: This involves a series of storyboards, sketching and mockups that were used in detailing the screen view and user interactions of the app. Several iterative changes were made before the actual Implementation: This helped to save time, however, the process is revisited several times after a need for changing some design features is discovered during evaluation. This chapter is focused on the design stage

Implementation: This involves the activities that are undertaken to execute the requirements and design.

Test: Testing is performed to verify the requirements and design specification is met.

Evaluate: The product of the implementation is examined to determine if some additional features should be added or removed.

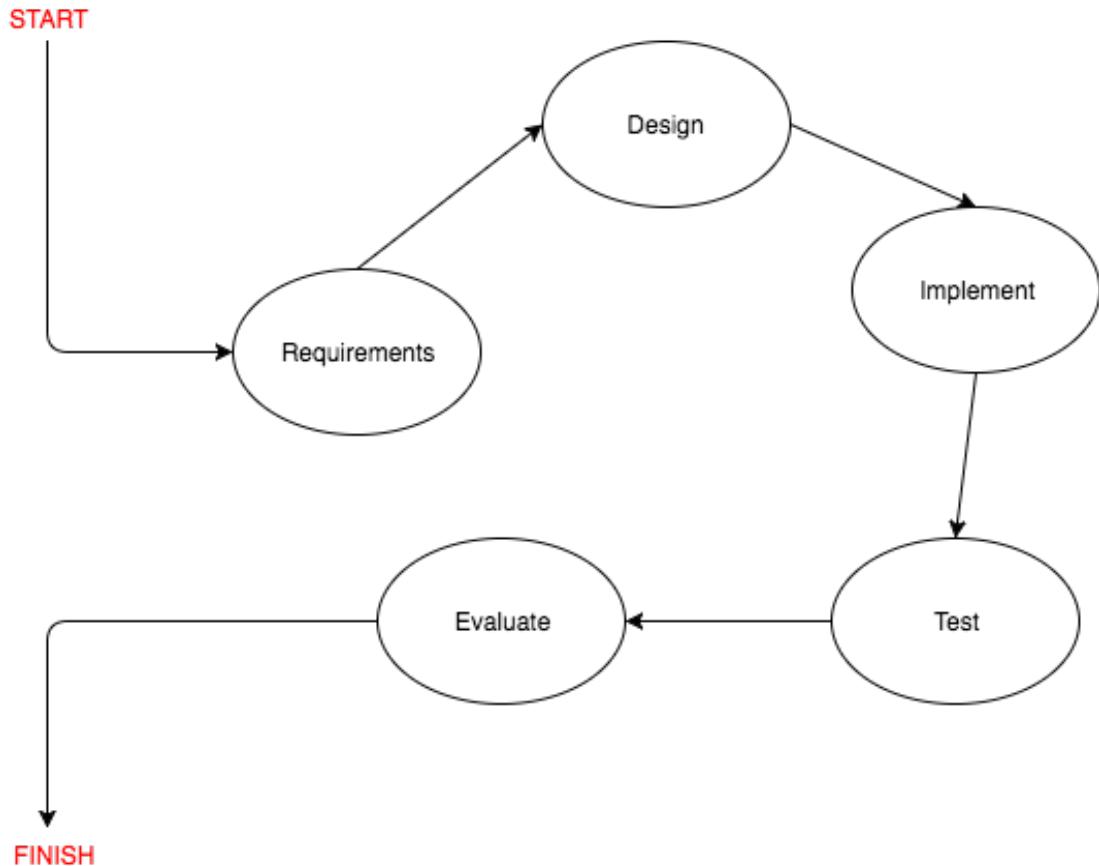


Figure 2 Iterative model diagram

4.2 Design Features

4.2.1 High Level Design

The high level design of Retailers App covers the system architecture and database design. It describes the relation between various modules, functions and interfaces of the system data flow, flow charts and data structures are covered.

Figure 3 clearly describes the Retailers App communication with the beacons and the server.

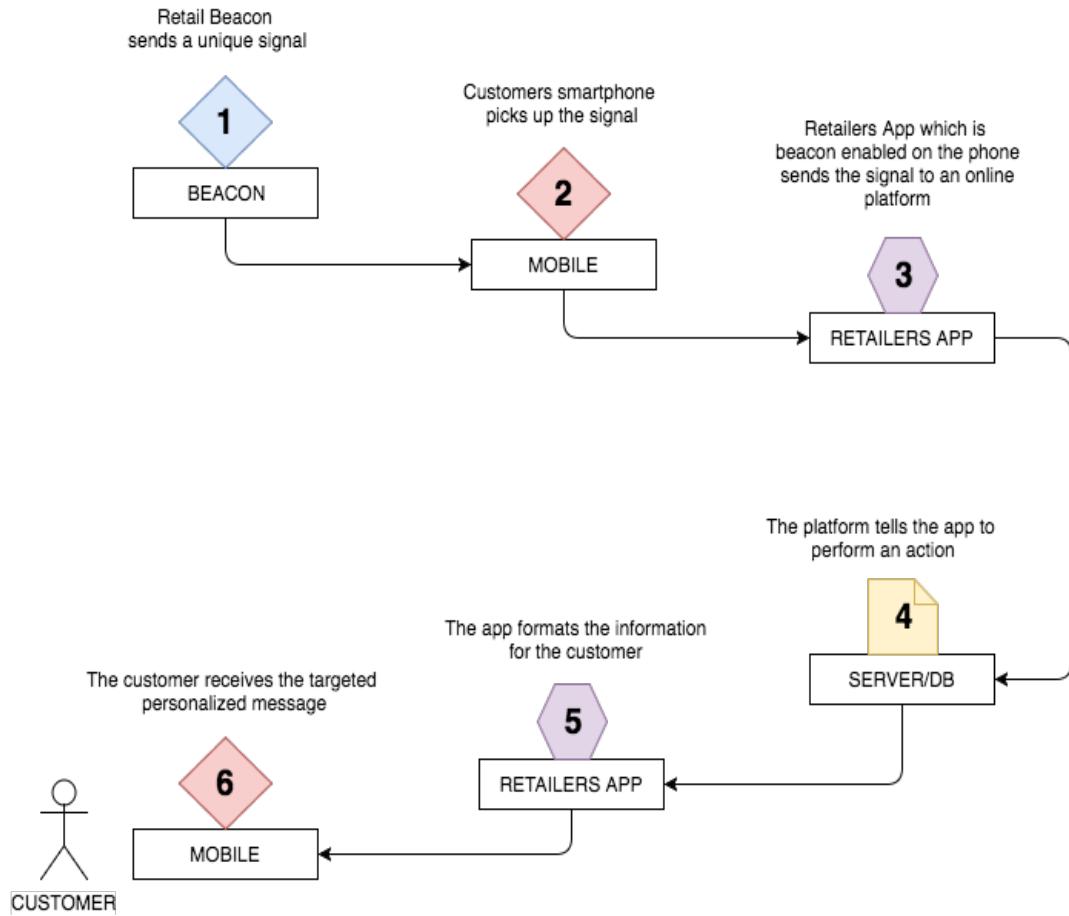


Figure 3 High Level Design Retailers App.

The following description explains the high level design and the functional of Retailers App. A beacon which is place at the store front. The beacon will holds the info relates to location and it keeps monitoring.

When user enters to the store, the app detects the beacon and will start communicating. The beacon share the info relates to location and Unique static ID. When app receives this signal, with the shared info it hit the server to fetch the details need to shown to the user.

CHAPTER 5

5 IMPLEMENTATION

5.1 Implemented Languages

The key principle behind the development of the cross-platform app is using the HTML markup language to structure content, the CSS style sheet language to style the content and the JavaScript scripting language to determine the behavior and logic of the app.

Android and iOS native plugins which talk to the beacons and transmit the data to the front end.

An overview of these technologies in relation to the thesis project is given below:

HTML5

HTML is the publishing language of the World Wide Web. The latest version of HTML is HTML5. It is currently supported by virtually mobile browsers in the market today. As HTML5 is a cross-platform, this makes it a good choice to build a cross-platform application. HTML5 has new added tags and attributes that did not exist in previous versions of HTML. HTML5 also supports some APIs that work with it but are not included in its specification. An example of those APIs is the local storage API which stores data locally on the client device. A number of these features were implemented in the thesis project. However, only the most relevant features that were used in the thesis project are discussed in this thesis.

CSS3

CSS is responsible for styling how the structure of HTML contents should look. HTML elements can be styled individually but this approach is very limited in design options and cumbersome when changes are to be made. Subsequent changes to the HTML style will need to be made on individual HTML elements. CSS offers an efficient way to style individual HTML elements by creating a style in one sheet that is applied to the individual elements of all HTML pages associated with the sheet. The project makes use of CSS3 which is the latest version of CSS. A summary of the advantages

why CSS is used in the styling is given below:

- Styles are easier to read and maintain.
- CSS offers a central point of changes to the styling of multiple pages by editing a single file.
- CSS offers better control to the styling of page elements.
- The new modules introduced in CSS3 work very well with HTML5 to bring out the best page look.

JavaScript

JavaScript is chosen as the scripting language that will add interactivity or behavior to the app for the following reasons:

Extensive support: JavaScript is the primary script language of HTML5. It is also the most common client side scripting language that is supported in available browsers.

Simplicity: JavaScript does not require a special development environment for it to be executed. It is executed in web browsers which are present in virtually all devices. This makes it easy to write and implement.

Extensive Libraries: JavaScript has so many libraries that make it easier to develop and extend its capabilities. There are lots of pre-written libraries that provide simple solutions to many complex or common tasks written in JavaScript code. Libraries such as Jquery greatly reduce the inconsistencies in JavaScript across browsers. Most of the web technologies have built JavaScript libraries that are integrated into developed applications to achieve a higher level task. An example is YUI JavaScript Library used to build sleek and interactive user interface for web apps.

5.2 Implemented Libraries and Frameworks

A library is a group of data and programming code that is used to develop software programs and applications while a framework is a platform for developing software applications. This chapter section discusses the libraries and frameworks that were implemented to achieve the objective of the thesis project.

Ionic Framework

Ionic is open source framework used for developing mobile applications. It provides tools and services for building mobile User Interface with native look and feel. Ionic framework needs native wrapper to be able to run on mobile devices.

Since Ionic is built on top of AngularJS and Apache Cordova. Need to have basic knowledge about the mentioned technologies. You also need to be familiar with the standard web technologies such as HTML, CSS3 and JavaScript, so that u can completely understand all the information provided.

Speed: The size of the library is relatively small which makes it work fast.

Easy to learn: There are a lot of materials and documentations on Ionic. The learning curve is really short because it is easy to understand and most developers are familiar with it, thereby making it easy to get assistance.

Lots of Plug-ins: There so many Ionic plug-ins available for free that make it easier to reuse nice features and effects already created. There is no need to re-invent the wheel. Some features and effects in this thesis project are courtesy of plug-ins.

Compatible with CSS3: It works pretty well with CSS3 styling sheet that is used in the helps to produce good user interface and design. The standard HTML markups with the basic style. The HTML source code can be found in Appendix 1. Ionic provides the predefined markups, its in a different style. The HTML source code can be found in Appendix 2.

PhoneGap

PhoneGap, also referred to as Cordova, is a mobile development framework that enables a mobile app to be developed with HTML, CSS, and JavaScript. The sets of device API in PhoneGap allow a mobile developer to access native device features such as the camera or storage using JavaScript. JavaScript APIs are built on web standards that are consistent and widely supported across various device platforms (Apache Cordova 2016). The Retailer App developed in the thesis project is implemented using JavaScript APIs within the PhoneGap framework. As a result, the app can run across the mobile operating systems that are supported on PhoneGap as seen in Figure 5.1. However, the native

features that can be implemented on a mobile device using PhoneGap are limited to the features that are supported by PhoneGaps' API. Table 5.2 shows the API features that are currently supported by PhoneGap. It can be seen in Table 5.2 that Notification (Alert), Connection and Device APIs are colored with a red. The red line marks them as the core APIs that are implemented in the thesis project. They have to be widely supported in order to ensure a successful implementation of the app across multiple device platforms.

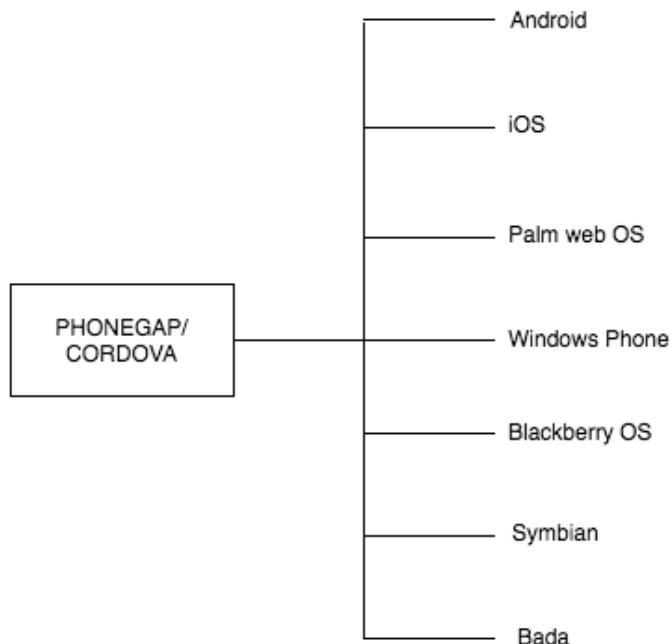


Figure 4. Mobile operating systems supported on PhoneGap framework.

Table 3. Supported features on PhoneGap.

	amazon-fireos	android	blackberry10	ios	Windows Phone
Accelerometer	Y	Y	Y	Y	Y
Camera	Y	Y	Y	Y	Y
Capture	Y	Y	Y	Y	Y
Compass	Y	Y	Y	Y	Y
Connection	Y	Y	Y	Y	Y
Contacts	Y	Y	Y	Y	Y
Device	Y	Y	Y	Y	Y
Events	Y	Y	Y	Y	Y
File	Y	Y	Y	Y	Y
Globalization	Y	Y	N	Y	Y
In App Browser	Y	Y	Y	Y	Y
Media	Y	Y	Y	Y	Y
Notification	Y	Y	Y	Y	Y
Splash screen	Y	Y	Y	Y	Y
Storage	Y	Y	Y	Y	Y

Y - Yes, N - No

There are other open source mobile development framework such as Rhodes, Sencha Touch, Appcelerator Titanium and MoSync that are available for free. However, the thesis project is implemented in PhoneGap because of its high level of popularity and support among the developers community.

Implementing within the PhoneGap framework also allows the source codes to be easily compiled and deployed. The source code of the app is simply uploaded in a single zip file into a cloud service called PhoneGap Build. PhoneGap Build then generates a native app for each platform that it supports. Prior to September 2012, a PhoneGap app was compiled on each native SDK platform of the targeted device. For example, an app that is

targeted to run on iOS, Android and Windows phone would require XCode, Eclipse and Visual Studio respectively. Furthermore, developing an iOS app requires an Apple computer and a Windows phone a Windows computer. Therefore, using PhoneGap Build reduced the resources and complexities that would have been required in implementing the thesis project. An overview of deploying an app with PhoneGap Build is seen in Figure 5.

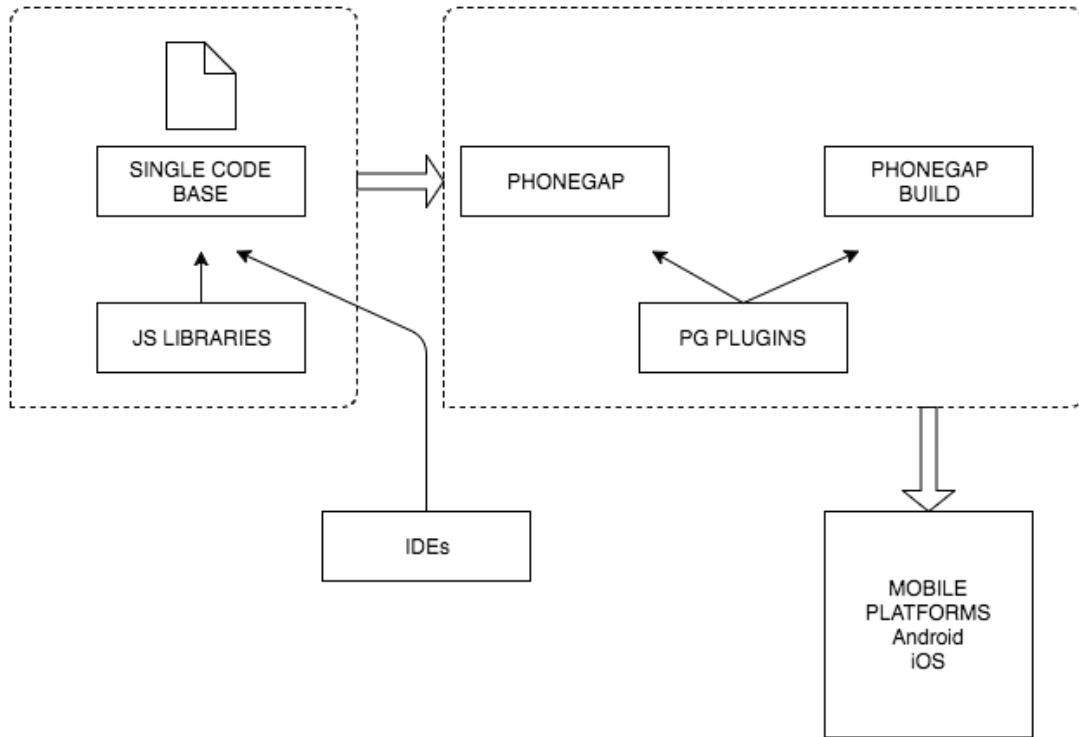


Figure 5. PhoneGap Build compiles and deploys mobile application

Kontakt Beacon

The Kontakt.io Proximity REST API allows you to programmatically access all your devices, venues, actions etc in the Kontakt.io Cloud.

It provides the Android and iOS SDK. The new simplified Ranging & Monitoring API based on fluent interfaces. The SDK provides developers with components to help build applications using [kontakt.io](#) proximity Beacons. It covers the following areas of functionality:

- Devices Ranging & Monitoring
- Device connection
- Device actions
- Communicating with Kontakt.io Rest API

5.3 Implementation of Screen Views

Mobile devices come in varying screen sizes. It is essential that the screen view of the app adapts to the screen size of the user's mobile device. Thus, the app can have same look irrespective of the screen size. To address this problem, the statement `<meta name='viewport' content='width=device-width, initial scale=1.0'>` is included within the head tag of the index.html file.

The design chapter detailed the components of the offers list screen view and the details screen view of the app. The reader should notice we refer to them as screen views and not pages. Screen view is the appropriate word that conveys the implementation of what the user refers to as a page in the context of the mobile app. In the context of the thesis project, it implies that only some parts of the index.html page are presented on the screen for the user to see. All the parts of the two screen views are defined within div tags in the index.html. A JavaScript function is defined to display the parts that make up the active screen view and hide the rest parts of the index.html.

Hence, we are able to reuse an HTML code by referring to the id of its tag and thereby reduce the amount of HTML code required to run the app.

The app is implemented such that the screen adjusts automatically depending on the size and resolution of the device's screen. The app supports both the Portrait and Landscape mode.

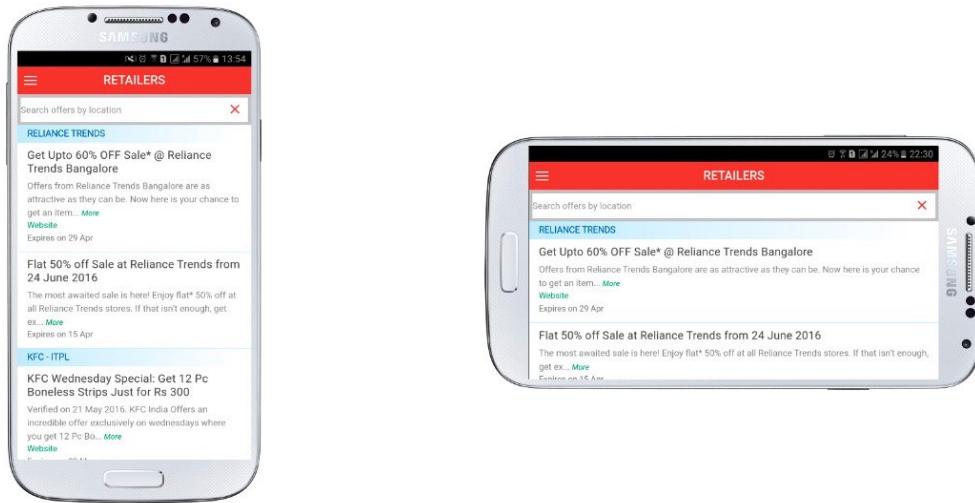


Figure 6. Responsiveness

5.4 Storage Implementation

The data of the application is stored locally on the mobile device. The PhoneGap Shared Preferences facilitates the storing of data locally on a device that supports HTML5 web browser. The storage API is one of the new APIs that is supported by HTML5 but not included in its specification. Prior to the introduction of local storage API, cookies were the only means to store data locally on a device. However, cookies were allowed to store data of maximum 4KB. Unlike cookies, the local storage API can store persistent data to a limit of 5MB or 10MB on a device. Hence, the author leveraged the Storage API in the implementation of data storage.

The offers data has been uploaded as hardcoded JSON to the www.json-generator.com. An each offer data can be found in Appendix 3. The offers data is stored as array of objects.

PhoneGap documentation specifies that the Shared Preferences API supports the local storage on the following platforms.

- Android
- iOS

This would have enabled a storage implementation to manage the profile settings (Name, Email and Mobile Number) in the app. It is important to note that the Local Storage object can only store data in form of a string. Since the mobile app data that is stored involves storing an item's data as an object. Hence the JSON data interchange format is used to convert the item's object data into a string before it is stored. Then the stored string is parsed from a string to an object when retrieving the item's object data. The source code for storing and retrieving data can be found in Appendix 4.

5.5 Fetch and View Offers

In line with the app design, a user first views the list of offers. The user can fetch the list of offers in two ways. One with the Google place search and the other with the proximity beacon.

The user can search the offers with the Google Places API. When user chooses the location, the coordinates will be sent to the server. The server will filter the offers based on the location and provides the filtered offers to the user.. This is the manual process. The UI can be found in Figure 7.

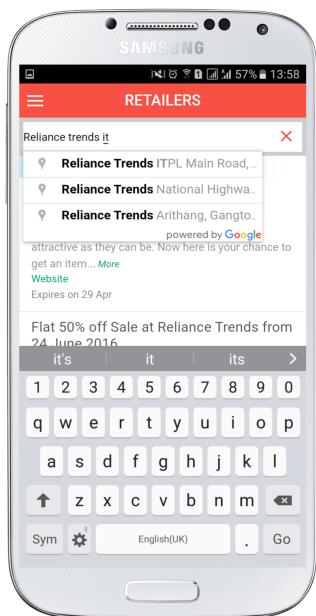


Figure 7. Fetching offers with the Google Places API.

In the automated process, as the user enters a specific location or in this case a retail store, the beacon placed at the entrance of the store will identify the device send the signal and location coordinates to the device. To work with the beacon, the app requests the user to turn on Bluetooth. A popup message asks the user to give permission to the app to turn on Bluetooth. Figure 8 represents the UI for the confirmation popup. When the user turns on Bluetooth, the app starts to scan for all the BLE devices and connects to only the BLE device (Beacon) with the namespace - RETAIL xxxxxxx. After connecting to the beacon, the app will communicate with the server to fetch the offers based on the provided location of the beacon.

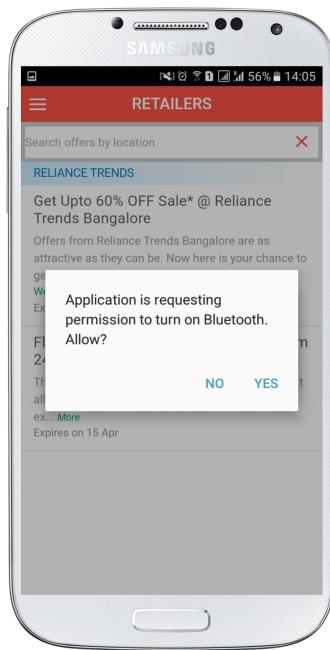


Figure 8. Fetching offers with the Proximity Beacon API.

Each offer tile displays the Title, Description, Website Link and Offer Expiry Date. The link is used to show the complete details of the offers. Its an optional link. Figure 9 will show the offers list. The expired offer will be removed at the back end with the daemon process.

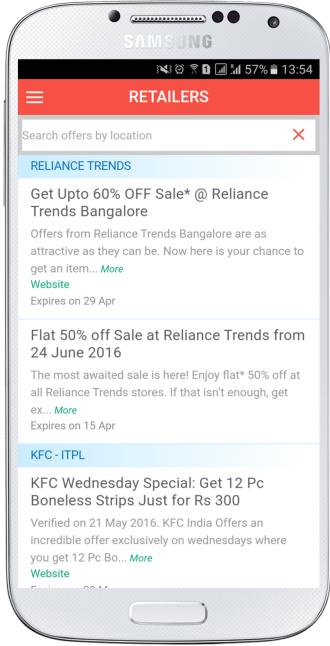


Figure 9. Offers View.

Detail View

When user performs an action on the offer tile in the Home screen, the app navigates to the detail view, which shows the complete details of an offer. The source code for navigation can be found in Appendix 5. This behavior was implemented in order to preserve the consistency of having unique items in the offers list. A user may wish to change the view an offer after it has been shown.

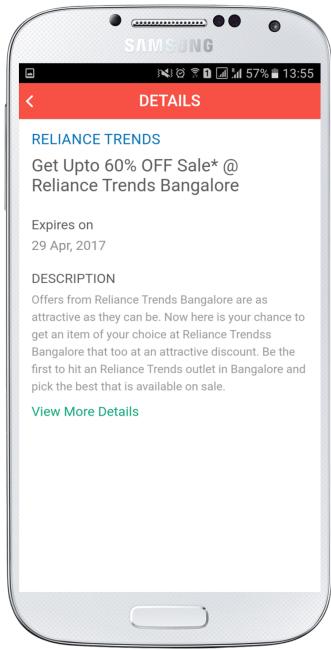


Figure 10. Detailed Offer View.

5.6 Change Profile and Invite Friends

Change Profile

A user can edit and save the details like Name, Email and Mobile Number. The App will collect the user info to send the notifications or any hot offers based on user's search behavior.. This action triggers an event that brings focus to the input name, email and mobile number. It is essential to implement a defensive design in order to prevent a user from accidentally modifying or clearing the user details. During the requirement analysis, it was noticed that many users did not like to share the user info.

To solve this potential problem, the profile change settings can be optional without submitting the user info. Still user can fetch and view the offers. The angular messages directive used for the form validations. The form validations will display as span text bottom to the appropriate input fields. The validations markup can be found in Appendix 6.

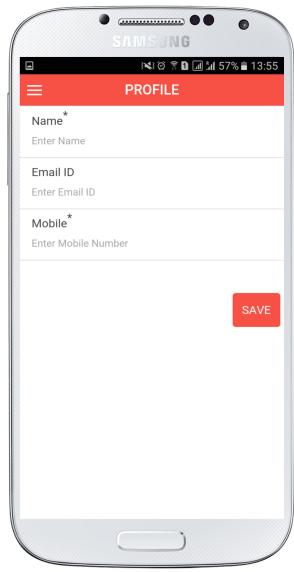


Figure 11(a). Change Profile Settings.

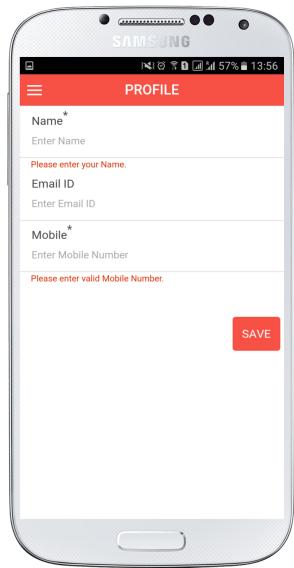


Figure 11(b). Change Profile Settings with validations.

Invite Friends

The user can invite friends to use the 'Retailers App' through an email or a mobile number. An Email has been sent to the submitted email address of the friend. In case of mobile number submission, a SMS will be sent. The Email and SMS contain the play store link and the details of the 'Retailers App'.

During the requirement analysis, it was noticed that users want to invite through Email or SMS. To solve this problem, the user can invite the friends on any one option or both. It's not mandatory that the user needs to submit both the info.

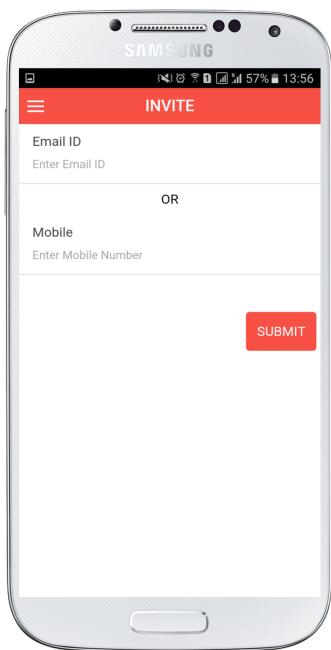


Figure 12. Invite Friends View

CHAPTER 6

6. TESTING AND RESULT

The application was tested based on the requirements and overall user interface design. The result of the test showed that the project performed well and the requirements and design specification were met. However, more improvements can be made and will be discussed in the future work section of the Conclusion chapter.

Due to the limited available resources, the testing was only carried out on Android and iOS Phone mobile operating system platforms. The application file for each platform generated by PhoneGap Build was installed on an emulator (virtual device) and a real device as specified in Table 4. Set up and installation of the application was faster on an emulator than on a real device for both Android and iOS application. Using an Android Devices enabled the test to be carried out on Android 5 to Android 6 versions. The screen shots used in this thesis writing were captured from the Android and iPhone Devices.

Emulators also have their limitations. They work very slowly and their performance may not represent the true performance of an app on a device. On the other hand, testing the app on a mobile device produces reliable results in terms of performance and usability. However, only two mobile devices were available for this thesis project which is fairly limited.

Although the app was also tested on desktop browsers, it is specifically designed to run on mobile device platforms. The nature of the app makes it suitable for just mobile devices. Testing the app on a desktop browser revealed that the alert notifications did not work. It was later discovered that the method for defining an alert function on a mobile device platform differs from that of a desktop browser. The BLE communication wont be happen with the source code. The plugins wrote as platform specific Android and iOS.

Nevertheless, few lines of code were added to the source code to apply the right method of notification alert if the platform running the app is a desktop browser.

Table 4. Testing on different devices and platforms.

DEVICE	VERSION
Samsung Galaxy On5	Android 6.0
iPhone 6	iOS 10
Kontakt Beacon	3.0

SUMMARY

Developing a cross-platform mobile application which communicates to the beacon based on HTML5, JavaScript and CSS saves time and resources. A developer can built an app in multiple platforms with the same source code. The only limitation is that the developer need to write Cordova Plugins in the native interface language which act as a bridge between the Javascript and the Native hardware. The Cordova Plugins which talks to the beacon. By using a mobile development framework such as PhoneGap, MoSync and SenchTouch, a native application that runs on various mobile device platforms was built without the use of Java, C# and other native coding languages.

The key findings in the study is

- 1.Saves Project Maintenance time.
- 2.Single source code can be used in different platforms.
- 3.Need native interface plugins to communicate to the hardware components.
- 4.UI Responsiveness
- 5.Can communicate with the physical devices through BLE (Beacon).

The outcome of the dissertation validated as the HTML5 programming is a considerable approach to develop Hybrid Mobile Application and can communicate to the physical devices (Beacon) with the same by introducing native interface layers provided that the associated constraints are acknowledged.

CONCLUSION

The thesis project achieved its goal and met the set requirements of the developed cross-platform mobile application. However, some challenges and limitations were encountered. Despite the right decision to migrate to a reliable cross-platform mobile development framework PhoneGap, there exist certain limitations. Some features of a device can only be accessed through the device's native language. Hence, this limits what feature can be implemented on a cross-platform app except a developer understands the native interface language of the targeted mobile device platform.

The requirements and design of the thesis project has distinct features in comparison with the apps that were analyzed from two app stores. This validates requirement analysis as an effective process for producing a unique or well modified software design.

The author recommends mobile application developers to assess what choice of development framework is more suitable for each cross-platform mobile application project.

BIBLIOGRAPHY / REFERENCES

1. BOOK

Chris Griffith. Hybrid Mobile App Development with Ionic. California: O'Reilly Media, 2015

Neil Smyth. Android Studio Development Essentials - Android 7 Edition: Learn to Develop Android 7 Apps With Android Studio 2.2. South Carolina: Independent Publishing Platform, 2016

Neil Smyth. iOS 9 App Development Essentials: Learn to Develop iOS 9 Apps Using Xcode 7 and Swift 2. South Carolina: CreateSpace Independent Publishing Platform, 2015

2. SCHOLARLY JOURNAL ARTICLES

Peter Lewis. "How Beacons Can Reshape Retail Marketing." Beacon technology. 20 (2016): 6

John M. Wargo. "Learn Apache Cordova Programming." Apache Cordova development. 3 (2015): 8

3. CONFERENCE PROCEEDINGS

Ryan Reith. Worldwide Smartphone Volumes. IDC's Worldwide Quarterly Mobile Device Trackers. 2016 Jul 28; FRAMINGHAM. FRAMINGHAM: IDC; 2016

APPENDICES

Appendix 1. HTML code for form without Ionic Framework

```
<form>
  Name:<br>
  <input type="text" name="user" placeholder="Enter Name">
  <br>
  Email ID:<br>
  <input type="email" name="email" placeholder="Enter Email ID">
  <br><br>
  Mobile:<br>
  <input type="number" name="mobile" placeholder="Enter Mobile Number">
  <br><br>
  <input type="submit" value="Submit">
</form>
```

Appendix 2. HTML code for form with Ionic Framework

```
<ion-view view-title="{{strings.profile_title}}>
  <ion-content class="margin-bottom-20">
    <form name="profileForm" novalidate="">
      <div class="list">
        <label class="item item-input item-stacked-label no-border-top">
          <span class="input-label">Name<sup>*</sup></span>
          <input type="text" placeholder="Enter Name" name="user" ng-model="vm.data.name" required>
        </label>
        <label class="item item-input item-stacked-label no-border-top">
          <span class="input-label">Email ID</span>
          <input type="email" name="email" placeholder="Enter Email ID" ng-model="vm.data.email">
        </label>
        <label class="item item-input item-stacked-label no-border-top">
          <span class="input-label">Mobile<sup>*</sup></span>
          <input type="number" name="mobile" ng-maxlength="10" ng-minlength="10" placeholder="Enter Mobile Number" ng-model="vm.data.mobile" required>
        </label>
      </div>
      <br>
      <div class="row">
        <div class="col no-padding text-right">
          <button class="app-button button" ng-click="vm.save(profileForm.$valid, data)">
            SAVE
          </button>
        </div>
      </div>
    </form>
  </ion-content>
</ion-view>
```

Appendix 3. Data storage

```
$scope.$on('$ionicView.beforeEnter', function() {
    function PSuccessHandler(value) {
        if(value) {
            vm.data['name'] = value;
        }
    }
    function ESuccessHandler(value) {
        if(value) {
            vm.data['email'] = value;
        }
    }
    function MSuccessHandler(value) {
        if(value) {
            vm.data['mobile'] = Number(value);
        }
    }
}
sharedpreferences.getString('NAME_P', PSuccessHandler, doNothing);
sharedpreferences.getString('EMAIL_P', ESuccessHandler, doNothing);
sharedpreferences.getString('MOBILE_P', MSuccessHandler, doNothing);
});

function save(valid, data) {
    if(valid) {
        if(data.name) {
            sharedpreferences.putString('NAME_P', data.name, doNothing, doNothing);
        }

        if(data.email) {
            sharedpreferences.putString('EMAIL_P', data.email, doNothing, doNothing)
        }

        if(data.mobile) {
            sharedpreferences.putString('MOBILE_P', data.mobile, doNothing, doNothing)
        }

        $rootScope.showPopup('Your details have been saved successfully.', 'OK', 'Success');
    }else {
        vm.submitted = true;
    }
}
```

Appendix 4. Navigating between views

```
function offerDetail(ev, data) {
  $state.go('app.detail', {'detailOffer': data});
}
```

Appendix 5. Offer object with the properties

```
{
  "offerCode": "FAC000009",
  "subtitle": "",
  "description": "Offers from Reliance Trends Bangalore are as attractive as they can be. Now here is your chance to get an item of your choice at Reliance Trendss Bangalore that too at an attractive discount. Be the first to hit an Reliance Trends outlet in Bangalore and pick the best that is available on sale.",
  "title": "Get Upto 60% OFF Sale* @ Reliance Trends Bangalore",
  "enddatetime": "2017-04-29T12:33:00.000Z",
  "__v": 0,
  "link": "http://www.upto75.com/Reliance_Trends/m/5846/Sale_Offer.html",
  "location": {
    "geo": {
      "type": "Point",
      "coordinates": [
        77.73645399999981,
        12.986851
      ]
    }
  },
  "startdatetime": "2016-05-21T12:33:46.177Z",
  "_id": "574055eee42227482c3737e8",
  "type": "HOT",
  "name": "RELIANCE TRENDS"
},
```

Appendix 6. HTML code for angular validations

```
<form name="profileForm" novalidate="">
  <div class="list">
    <label class="item item-input item-stacked-label no-border-top">
      <span class="input-label">Name<sup>*</sup></span>
      <input type="text" placeholder="Enter Name" name="user" ng-model="vm.data.name" required>
    </label>
    <div ng-messages="profileForm.user.$error" ng-show="vm.submitted">
      <div ng-message="required" class="error">{{strings.name_valid}}</div>
    </div>
    <label class="item item-input item-stacked-label no-border-top">
      <span class="input-label">Email ID</span>
      <input type="email" name="email" placeholder="Enter Email ID" ng-model="vm.data.email">
    </label>
    <div ng-messages="profileForm.email.$error" ng-if="vm.submitted">
      <div ng-message="email" class="error">{{strings.email_valid}}</div>
    </div>
    <label class="item item-input item-stacked-label no-border-top">
      <span class="input-label">Mobile<sup>*</sup></span>
      <input type="number" name="mobile" ng-maxlength="10" ng-maxlength="10" placeholder="Enter Mobile Number" ng-model="vm.data.mobile" required>
    </label>
    <div ng-messages="profileForm.mobile.$error" ng-show="vm.submitted">
      <div ng-message="required" class="error">{{strings.mobile_valid}}</div>
      <div ng-message="maxlength" class="error">{{strings.mobile_valid}}</div>
      <div ng-message="minlength" class="error">{{strings.mobile_valid}}</div>
      <div ng-message="number" class="error">{{strings.mobile_valid}}</div>
    </div>
  </div>
</form>
```

Checklist of items for the Final Dissertation Report

	Is the final report neatly formatted with all the elements required for a technical Report?	Yes
	Is the Cover page in proper format as given in Annexure A?	Yes
	Is the Title page (Inner cover page) in proper format?	Yes
	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	Yes Yes
	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	Yes Yes
	Is the title of your report appropriate? The title should be adequately descriptive, precise and must reflect scope of the actual work done. Uncommon abbreviations / Acronyms should not be used in the title	Yes
	Have you included the List of abbreviations / Acronyms?	Yes
	Does the Report contain a summary of the literature survey?	Yes
	Does the Table of Contents include page numbers? i. Are the Pages numbered properly? (Ch. 1 should start on Page # 1) ii. Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) iii. Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) iv. Are the Captions for the Figures and Tables proper? v. Are the Appendices numbered properly? Are their titles appropriate	Yes Yes Yes Yes Yes
	Is the conclusion of the Report based on discussion of the work?	Yes
	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	Yes Yes Yes
	Is the report format and content according to the guidelines? The report should not be a mere printout of a Power Point Presentation, or a user manual. Source code of software need not be included in the report.	Yes

Declaration by Student:

I certify that I have properly verified all the items in this checklist and ensure that the report is in proper format as specified in the course handout.



Place: Bangalore

Signature of the Student

Date: 14/10/2016

Name : Pavithra H M

ID No.: 2014HT13053