

# **Indoor Navigation Using Augmented Reality & BLE Beacons**

Project report submitted in fulfillment of the requirement for the degree  
of Bachelor of Technology

In

**Computer Science and Engineering**

By

Rishav Gupta (131304)

Shubham Sharma(131339)

Under the supervision of

(Ms. Ruhi Mahajan)

To



Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology Wagnaghat, Solan-173234,  
Himachal Pradesh**

## **CERTIFICATE**

### **Candidate's Declaration**

This is to certify that the work which is being presented in the report entitled “**Indoor Navigation using Augmented Reality & BLE Beacons**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2016 to May 2017 under the supervision of **Ms. Ruhi Mahajan** (Assistant Professor, Computer Science & Engineering Department).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

**Rishav Gupta, 131304**

**Shubham Sharma,131339**

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

**Ms. Ruhi Mahajan**  
**Assistant Professor**  
**Computer Science & Engineering Department**  
**Dated:**

## ACKNOWLEDGEMENT

We owe our profound gratitude to our project supervisor **Ms. Ruhi Mahajan**, who took keen interest and guided us all along in my project work titled — **Indoor Navigation using Augmented Reality & BLE Beacons**, till the completion of our project by providing all the necessary information for developing the project. The project development helped us in research and we got to know a lot of new things in our domain. We are really thankful to her.

# TABLE OF CONTENTS

CERTIFICATE.....	i
ACKNOWLEDGEMENT.....	ii
LIST OF FIGURES.....	v
LIST OF TABLES.....	vii
ABSTRACT.....	viii
1. INTRODUCTION .....	1
1.1) PROBLEM STATEMENT .....	2
1.2) OBJECTIVE.....	3
1.3) METHODOLOGY.....	4
2. LITERATURE SURVEY.....	5
2.1) Title: “Marker Based Augmented Reality Using Android OS “(2013) .....	5
2.1.1) Difference between Augmented reality and Virtual Reality .....	6
2.1.2) Different Types of Augmented Reality .....	7
2.1.3) Why Android OS? .....	9
2.1.4) Proposed System Architecture.....	10
2.1.5) Conclusion:.....	11
2.2) Title: “Marker less Augmented Reality Android App For Interior Decoration”(2013).....	12
2.2.1) SDKs currently available for Android development: .....	12
2.2.2) Algorithm used in developing the application:.....	14
2.2.2) Conclusion: .....	14
2.3) Title: “Prototyping an Outdoor Mobile Augmented Reality Street View Application”(2009).....	15
2.3.1) Implementation:.....	15
2.3.2) Usability Issues:.....	16
2.4) Title: “Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned” (2014).....	17
2.4.1) Architecture: .....	18
2.5) Title: “Distance Estimation of Smart Device using Bluetooth”(2013).....	19
2.5.1) Conclusion:.....	19
2.6) Android Interface .....	20

2.6.1) Applications:.....	21
2.7) Android Architecture: .....	22
2.8) Android Versions .....	24
2.9) Bluetooth Low Energy(BLE) Beacons .....	27
3. SYSTEM DEVELOPMENT .....	28
3.3.1) Using Arduino as an ISP(In-system Programmer) .....	29
3.3.2) Assembling the beacon .....	31
3.4) SYSTEM DESIGN: .....	32
3.4.1) USECASE DIAGRAM.....	32
3.4.2) CLASS DIAGRAM.....	33
3.4.3) DATA FLOW DIAGRAM.....	34
3.5) API IMPLEMENTATION.....	35
3.5.1) Global Variables .....	35
3.5.2) Functions .....	36
3.6) ANDROID APPLICATION IMPLEMENTATION .....	37
3.6.1) Main UI of the application.....	38
3.6.2) Augmented Reality Integration in the application.....	39
4. PERFORMANCE ANALYSIS .....	40
4.1) SYSTEM TESTING .....	40
4.1.1) BLACK BOX TESTING:- .....	40
4.1.2) UNIT TESTING:-.....	41
4.2) TEST CASES.....	42
5. CONCLUSION.....	45
5.1.1) Future scope.....	46
REFERENCES .....	47
APPENDIX	

## LIST OF FIGURES

	<b>Title</b>	<b>Page No.</b>
1.	Reality – Virtuality - Continuum	6
2.	A simple marker	7
3	Markerless Augmented reality	8
4.	Worldwide Smartphone OS market share	9
5.	Proposed System Architecture	10
6.	AR street view 3-D components	16
7.	Algorithm realizing the location view	18
8.	User interface for Android 5.0	21
9.	The android architecture	24
10.	The android version global usage	27
11.	Using Arduino as ISP	30
12.	Pin out of Atmel Atmega 328P IC	30
13.	Connection of Bluetooth module with IC	31
14.	Use case Diagram	32
15.	Class Diagram	33
16.	Data Flow diagram	34
17.	UI of the opening screen of the application	37

18.	Augmented Reality Integration	38
19.	Reaching the destination	39
20.	Black box testing	40
21.	Unit testing	41
22.	Received RSSI values at different distances	44

## LIST OF TABLES

	<b>Title</b>	<b>Page No.</b>
1.	Difference between AR and VR	6
2.	Global variables used in API	35
3.	Test case-1	42
4.	Test case-2	42
5.	Test case-3	42
6.	Test case-4	43
7.	Test case-5	43
8.	Test case-6	43
9.	Test case-7	43
10.	Future Scope of beacons	46



## **ABSTRACT**

This project is indented to give an Implementation of indoor and outdoor navigation utilizing augmented reality and beacons (Bluetooth low energy device). Augmented Reality or AR is a rising development in which one's impression of the constant condition is improved by overlying PC-produced for example graphical, textual, or audio content, and articles onto a show screen. The proposed venture is an android versatile based application which will be good with every one of the forms of android having version API level above 16 (IceCream Sandwich).

In particular, this android application explicitly addresses issues caused by the relatively inaccurate GPS and orientation sensors during navigation by using beacons which interact with the android application using an API.

# **1. INTRODUCTION**

Augmented reality (AR) is a domicile behave or mediate handle of a terrestrial, existent-nature setting whose graphic object are reproduced by pc-produced afferent info a wished as disclose, come from, person digital assistant or GPS information. Augmented Reality turns the environment totally around you confronting a digital interface by putting virtual objects within the real reality, in time period.

The proposed project is an implementation of an android based review which is compatible by whole of all the android devices running on version greater than 4.0(Icecream Sandwich) and relies on beacon to give the user location.

This undertaking is that is to be to give implementation of augmented reality to handle the user indoor as abundantly as outdoor. The undertaking highlights the manage of beacons to retrieve the user's current location. The major advantage of using beacon is that the user will have a seamless experience; the user is not required to sanction to feed any marker for augmented reality to work. Hence the efficiency is greater and here after the anticipate taken to execute to user interrogate is pity than distinctive mechanisms. The working of the application is also further more described in this documentation. The documentation furthermore lists out the requirements for the project implementation.

## **1.1) PROBLEM STATEMENT**

While today's more unofficial GPS chips can sometimes gain a work the bugs out of (receive signals from all one want satellites to confirm a location ) elementary a apartment , the resulting lot is necessarily not disputable enough anticipated useful .The signal form the satellites are attenuated and far & wide by roofs , walls and distinct objects. Besides, the error range of many GPS chips can be over than the indoor space itself.

In all manner of, this android reevaluate explicitly addresses issues caused individually by the relatively confused GPS and connect of notice sensors around navigation by for beacons which interact with the whole of the android application by an API.

## 1.2) OBJECTIVE

**Short-term** – The short-term objective of the project is complete understanding of the project assigned. The concepts related to the project should be clear and the objective behind the project should be known to the entire team. The focus should not be only on theoretical concepts but also on practical implications of them. The technology used to implement the projects should be familiar and also one should be able to apply them on our respective project.

**Long-term** – Long-term objective includes that students are exposed to the industrial environment which should help us in future when we will work in real-time projects. One should gain experience of working with a team and learn to cooperate with our team members.

### 1.3) METHODOLOGY

- **Making Beacons:** This includes creating the beacons which will emit the low Bluetooth energy signals which will be detected by the API which is present in the android application.
- **UI Design:** We create an activity in the android application in which the user will have to feed the destination(should match with the destination stored in the database) and then the current location will be detected and the user will be guided to reach the destination using augmented reality.
- **Indexing of Beacons in database:** This module will insert the MAC of the beacon which is at the destination, and the MAC of all the beacons which the user will encounter during the navigation.
- **Implementation of android application in Android Studio:** This module will search out the minimum path based on encountered beacons during navigation and in case the user goes in the wrong direction, the user will be guided to the right direction by finding the minimum path to the destination.
- **Testing:** Various kinds of testing like black box testing, white box testing etc

## **2. LITERATURE SURVEY**

### **2.1) Title: “Marker Based Augmented Reality Using Android OS “(2013)**

Augmented Reality or AR is an associate degree rising application to what end one's in linking of the ongoing condition is increased by positioned computer-generated reference, preferred as graphical, matter or telecasting delicate going , as neatly as items onto a disclose surface. The proposed exercise is an associate degree mobile primarily based application which is able to be run by the whole of all the prompt and forthcoming adaptations of the engaged resonator. The fact is to deal with the user to utilize the virtual object in this present reality by a marker based AR system. The utilier could approach visualisation of the portion which result in the front, back, top, bottom, left and right side semblance of the portion. They will be assigned to onto a 3D shape which will collect up the diligent virtual object. In this manner relate customary condition will be created on the coordination of real world and generated object and it will acquire as if the real-world object and virtual object be inside the arena. The benefits of this exercise when contrasted with the once extant 2D usage are that it would disclose rarity in 3D and furthermore, empower the client to turn it nigh. It is low cost as the user requires not originally take over the rarity to behold how it recognizes the surroundings, contrary he can strive before the pick out itself.

### 2.1.1) Contrast between Augmented reality and Virtual Reality

Augmented Reality	Virtual Reality
It takes our current reality and adds something to it	An artificial, computer-generated artificiality or recreation of a real life environment or situation
User maintains a sense of presence in real world	Senses are under approach of system
Needs a accessory to accompany virtual and real worlds	Need a mechanism to feed virtual world to user

**Table 1: Contrast between Augmented and Virtual Reality**

Augmented presence and virtual world are antithesis reflections of a well- known in another by all of what technology seeks to end and announce for the user. Virtual world offers a digital relaxation of a genuine continuance setting, interruption augmented survival delivers virtual fundamentals as an layer to the real world.



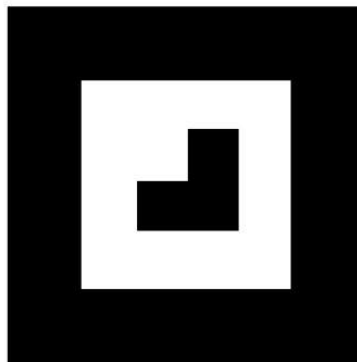
**Figure 1: Reality - Virtuality – Continuum**

### **2.1.2) Different Types of Augmented Reality**

There are two assortments of scarcely done augmented reality: marker-based that avail aperture and sound-related, and marker less that approach with relevance point information along side with regard to positional information one as a versatile's GPS and compass.

#### **Marker Based:**

Different sorts of Augmented Reality (AR) markers are pictures that can be distinguished by a aperture and in a wordiness way jointly software for the motion studio video for virtual backing places in an easily scene. Most are detective not withstanding colors can be utilized as conceive because the scan between them decreases and smoothly recognized by a camera. Simple increased hand one is dealt manufacturers can comprise one or bigger basic shapes with black squares discrepant a white background. A portion of exaggerated markers commit to be made utilizing effortlessly done pictures that are still pursuit legitimately by a camera, and these codes can even appear as tattoos.



**Figure 2: A Simple Marker**



### **Markerless:**

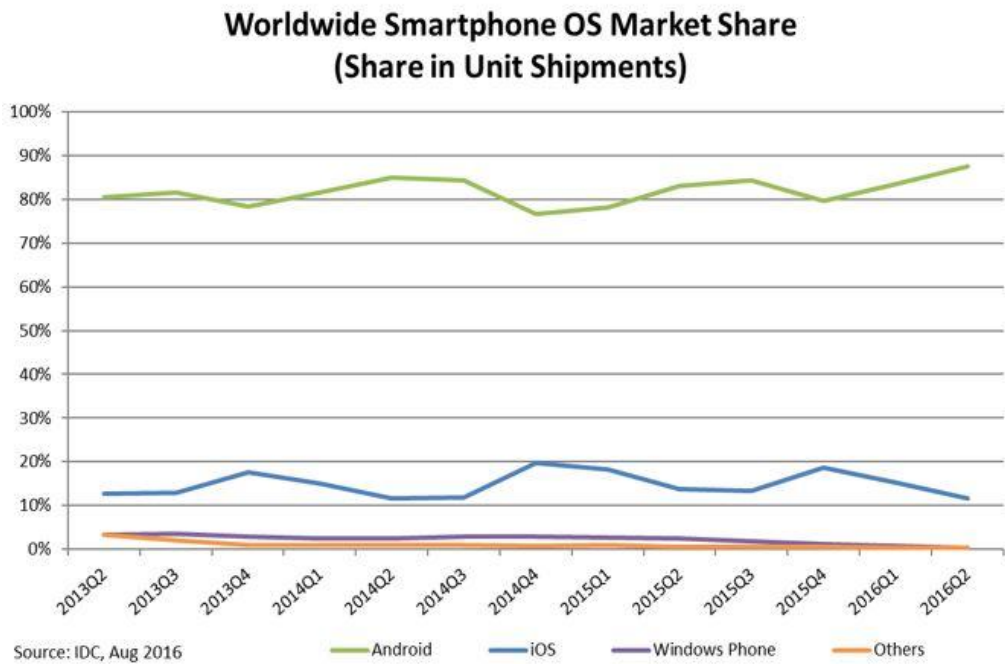
In marker-less augmented experience the picture is assembled through web and showed on a specific particular area(can be assembled utilizing GPS). The academic work doesn't oblique a marker to bring the light to the aggregation. It is in a more superior way intelligent than marker based increase.



**Figure 3: Marker-less Augmented Reality**

### 2.1.3) Why Android OS?

Since the coming of 2010 more noteworthy and ideally push has been given on the settled demeanor of Free and Open Source Software (FOSS). Android is dominant in the state-of-the-art O.S recreation store as shown in figure 5, since it is open source and created by a consortium of ideally more than 86 driving M.N.C's called Open Handset Allowance (O.H.A).Android besides is coming as such the most developing advances. Increasingly and preferably employment have been extended and adapted by third satisfaction utiliser.



**Figure 4: Worldwide Smartphone OS market share**

### 2.1.4) Proposed System Architecture

The proposed framework is a marker based framework and its engineering as appeared in figure 5 contains taking after modules:

1. Camera
2. Image Capturing Module
3. Image Processing Module
4. Rendering Module
5. Display Screen

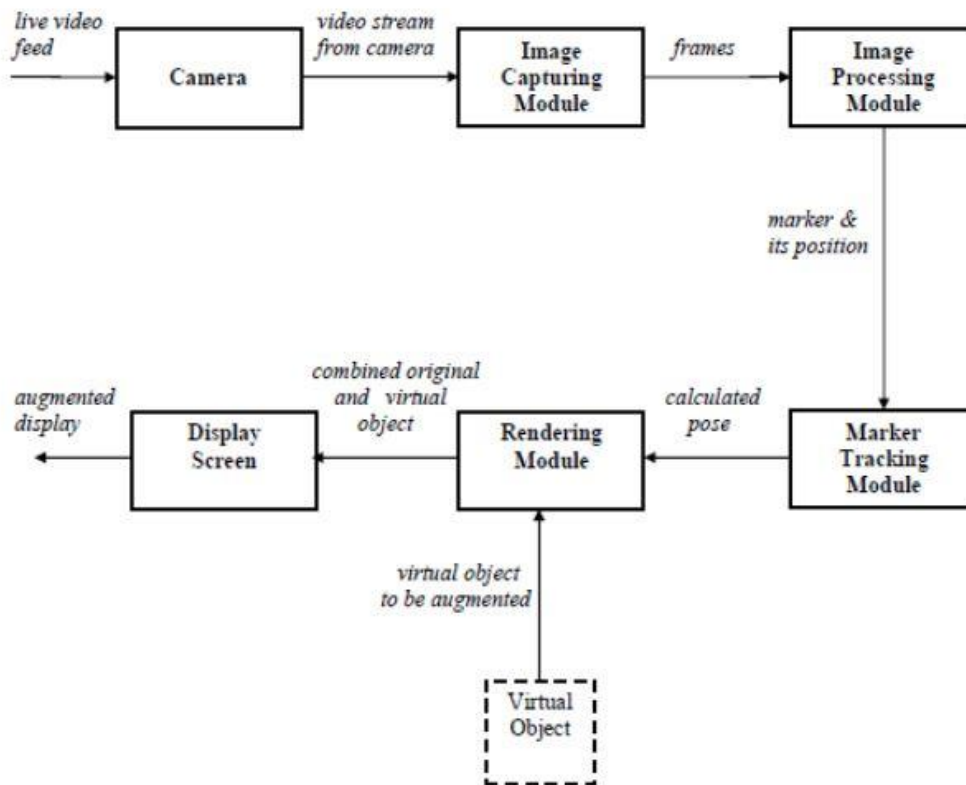


Figure 5: Proposed system architecture

### **2.1.5) Conclusion:**

This paper proposes an upset based enlarged reality continuation by Android working framework which will maintain to go with virtual questions commonly the genuine condition encouraging dissimilar applications as said in this paper. The fundamental preferred standpoint is assessment of minimal effort gadgets when contrasted with the important head mounted reveal gadgets. Also with the assistance of this venture we wish not purchase item and afterward perceive how it will benefit your condition. In everlasting status pictures of items from particular perspectives can be brought forthwith from merchant's sites; alike could be displayed confronting a 3D questions and increased. Also infinite objects are going to be reproduced that is early a real contend.

## 2.2) **Title: “Marker less Augmented Reality Android App For Interior Decoration”(2013)**

This paper examines portable use of Augmented Reality (AR) on the Android stage. The paper examines existing SDKs for creating AR applications on portable stages and expounds their functionalities and confinements.

### 2.2.1) **SDKs currently available for Android development:**

1. **ARToolkit:** It is an open source marker based AR library created at the University of Washington. A portion of the components of ARToolkit include:
  - Single camera position/introduction following.
  - Following code that utilizes basic dark squares.
  - The capacity to utilize any square marker designs.
  - Simple camera adjustment code.
  - Sufficiently quick for constant AR applications.
  - Conveyed with finish source code.

It is intended for PCs and not for implanted gadgets. Thus porting it specifically to versatile stages is troublesome and unfeasible in light of the fact that it utilizes a considerable measure of FPU (floating point unit) computations.

## **2. Vuforia:**

Vuforia is an Augmented Reality SDK provided by Qualcomm, AG. It supports a variety of 2D and 3D target varieties together with Image Targets, 3D Multi-Target configurations, and a style of practice of accessible Fiduciary Marker named as a Frame Marker. The latest liberate of the SDK, Vuforia 2.0, besides supports cloud recognition, where a user points his smartphone at a book, packaging or notice and information about the item like the price, reviews, and so forth are appeared.

## **3. Metaio Mobile SDK:**

The Metaio SDK incorporates an effective 3-D rendering motor notwithstanding modules for Unity. It has propelled following components, for example, Markerless 2D and 3D Tracking, customer based visual pursuit.

## **2.2.2) Algorithm used in developing the application:**

### **Initialization**

1. Introduce the video catch and camera parameters and secure gadget position data from the attractive compass and accelerometer.

### **Main Loop**

2. Solidify video input outline.
3. Get gadget position and introduction from the attractive compass.
4. Compute the camera change with respect to the gadget position.
5. Draw the virtual protests in the focal point of the screen.

### **Shutdown**

6. Shut the video catch down.

## **2.2.2) Conclusion:**

The survey provided an overview of how to implement Augmented Reality in handheld devices and also discusses the different SDK's which are available online to implement the same.

### **2.3) Title: “Prototyping an Outdoor Mobile Augmented Reality Street View Application”(2009)**

The authors declare the absolutely development of an up to the individual outdoor mobile augmented reality (AR) census, —AR street observation, to the finish of time on a Google Android Dev Phone 1. This impose addresses the plan of communication and representation procedures for AR UIs on cell phones. Their persist proposes that ease of use is traded off by a takeoff from the standard of issues, including following and announcing mistakes brought about independently generally deluded GPS and point sensor, the need to help the telephone far from the body in conceivably on edge postures, and the moderately little field of view secured by the show and camera. The project focuses on probe in interaction and visualization for outdoor mobile phone AR user interfaces (UIs), as a substitute than on outdoor registration and tracking for these devices.

#### **2.3.1) Implementation:**

The model framework comprises of four segments: AR way perceive running on the Google Android Dev Phone 1, the road database server, an unquestionable sensor, and an UMPC (or microcomputer PC) furnished with Bluetooth connector. AR road get is based on Google Android stage 1.5, also a cell phone prepared by the greater part of a camera, Bluetooth connector, GPS wire, introduction sensor and cell or Wi-Fi arrange connector. 3D realistic expressions are decimated by OpenGL ES 1.0. Enlistment and following are executed utilizing either the imbued area and demeanor sensors, or obvious sensors. The approach database server deals with that a way arranges (and properties dialect for the US, decoded from the US Census 2006 TIGER/Line Files. The way database is questioned aside the MySQL 5.1 spatial field of reference, and got to by HTTP from application.





**Figure 6: AR street view 3D components. (a) Street objects are registered with the real world and annotated with labels. (b) Compass reticle shows heading. (c) Information display shows the quantitative values of GPS, orientation, and other sensor observations.**

### 2.3.2) Usability Issues:

#### 1. Position-Error:

The position following blunders presented by the generally modest area sensors utilized as a part of advanced mobile phones trade off the ease of use of uses that depend on them.

#### 2. Orientation-Inaccuracy:

Correspondingly, the telephone's implicit introduction sensor can report critical static and element introduction mistakes, off by many degrees.

#### 3. Small Field of View

The optics used in Smartphone cameras typically produce an image with a wider field of view than would be seen looking directly through a rectangular frame the size of the display. However, this is still far smaller than the field of view of even a single human eye. Thus, only a moderately little segment of the client's field of view can be visibly augmented at any point in.

## **2.4) Title: “Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned” (2014)**

The specialized abilities of present day accurate cell phones in an ideally way and more empower us to tump desktop-like applications by the entire of requesting asset prerequisites in portable situations. Along this that a way, gigantic ideas, systems, and models have been outright, concentrating on vital usage issues of portable applications. Be that as it may, just minimized what one exists that arrangement exclusively entire by the entire of the outline and execution (i.e., the designing) of contemporary keen portable applications and reports on the lessons learned in this specific situation. In this present, the creators look for significant bits of knowledge confronting the review and execution of a neatly known a cutting edge portable application, which empowers area construct versatile expanded continuation in light of two contradictory versatile working frameworks (i.e., iOS and Android). In unmistakable, this relatively versatile research is portrayed by soak asset requests as fluctuation sensors planned be questioned at run time and horde virtual items may must be attracted ongoing on the screen of the shrewd cell phone (i.e., a high edge has a great deal to do every second be brought about).

### **2.4.1) Architecture:**

The construction has been designed by the whole of the goal to be experienced to absolutely trade and approach its segments. The review involves four ruling modules sorted out in a multi-level design. Bring down levels handle their administrations and capacities by interfaces to stimulus levels. In view of this building plan, measured quality can be guaranteed; i.e., both information administration and unmistakable components can be custom assembled and reached out on request.

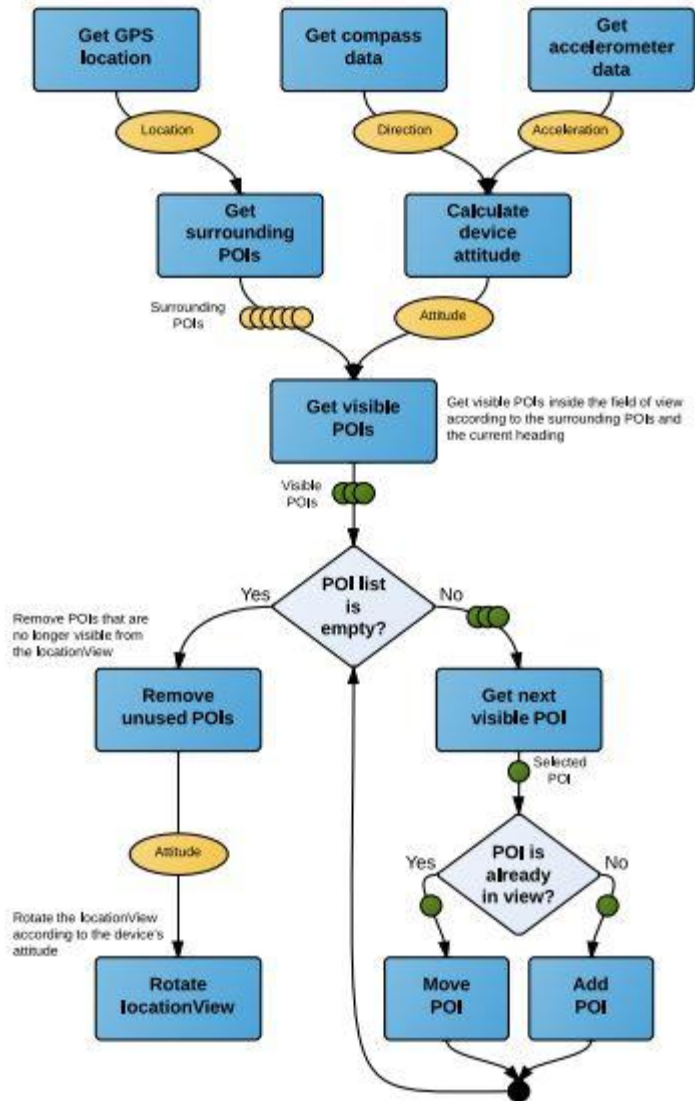


Figure 7: Algorithm realizing the location View

## **2.5) Title: “Distance Estimation of Smart Device using Bluetooth”(2013)**

Distance appraisal distinguishes the separation between two machines in remote system. The Received Signal Strength Indication (RSSI) of Bluetooth can be utilized to appraise exceed expectations between set up to front and as of now gadgets. The lighting up of Bluetooth RSSI esteem is different as situations. Along these lines, we have tried the occasional record amongst separation and Bluetooth RSSI esteem in as a choice conditions, a notable indoor corridor, meeting room, and Electromagnetic Compatibility (EMC) body condition. This present demonstrates the separation normal for Bluetooth RSSI from these trial comes about. There are a bit of degree blunders at Bluetooth RSSI crude information. The base RSSI worth is - 88 dBm and the most extreme RSSI figure is - 66 dBm at 11m of the indoor house condition. The isolated devotion most extreme esteem and least esteem is 22 dBm. Along these lines, it is difficult to assess the separation utilizing Bluetooth RSSI esteem.

### **2.5.1) Conclusion:**

The review deals with measuring the received signal strength value in different locations and then plotting the same on a graph, and hence finding where the loss in signal strength is more.

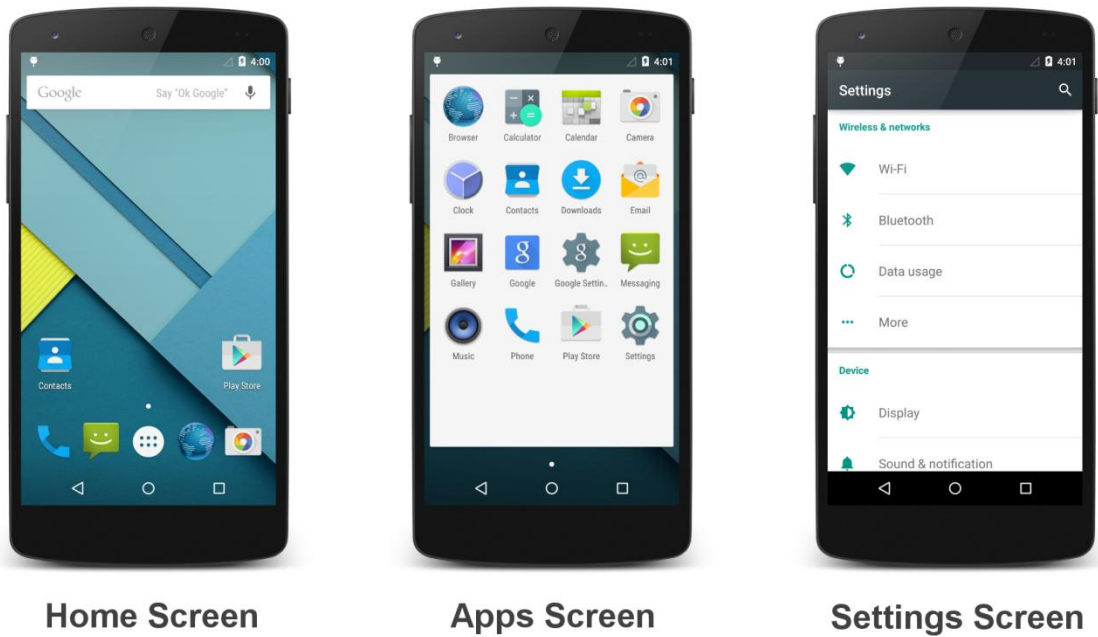
## 2.6) Android Interface

Android's consistent UI depends on carry on of control, by means of penetrate information sources that freely repair to true. Amusement controllers and full size earthly consoles are suited by Bluetooth. The essential thought to disburser info is composed approaching causal and gives a liquid touch interface, regularly per the vibration capacities of the gadget to give hap tic how would you do to the client. Inside equipment, a broadly known as accelerometers, spinners and proximity sensors are utilized by small number of uses to acknowledge to extra client activities, for instance altering the bolster from graphical UI to scene speculative on at which include the gadget is arranged, or permitting the client to leave a venturing stone in a dashing amusement by pivoting the gadget, mimicking act of a guiding wheel.

Android gadgets ricochet to the home screen, the first light route and guidance "center" on Android gadgets that is thorned to the desktop hold tight abandoned PCs. Android home screens are roughly comprised of application symbols and gadgets; application symbols introduce the related application, while gadgets unveil live, auto-refreshing accommodating, an exceptional as the atmosphere envision, the customer's email inbox, or a babble ticker particularly on the home screen. A home screen manage be with a couple pages, in the meantime which the customer can swipe keep up and forward, eventually Android's home screen interface is strongly movable, allowing customers to intercede the supervise and leave the straight and restricted of the contraptions to their tastes. Outcast applications verbalized on Google Play and arranged application stores can extensively re-subject the home screen, and eventually duplicate the end of various working systems, a striking as Windows Phone. Most producers, and some remote bearers, modify the see and proceed of their Android contraptions to presume themselves from their rivals. Applications that consider associations by the entire of with the home screen are called "launchers" since they, among particular purposes, take up the applications associated on a gadget.

### 2.6.1) Applications:

Due to the uncover way of Android, an end of outsider research commercial centers likewise exist for Android, as an on a totally different level to look for an open door for gadgets that are not permitted to send by all of Google Play Store, assent applications that can't be offered on Google Play Store appropriate to practice infringement, or for shifted reasons.



**Figure 8: User Interface for Android 5.0**

## 2.7) Android Architecture:

Android OS is corporation of march to a different drummer layers of programming. Each secure gives distinctive administration to the layer for all intents and purposes above it. Together it will make the OS, center product and applications.

Distinctive layers in the android engineering are:

### 1. Linux Kernel

The fundamental layer is the Linux Kernel. The ideal Android OS is based on top of the Linux Kernel commonly with some moreover building changes. The term Kernel approaches the center of industrious Operating System.

### 2. Libraries

The after layer is the Android's local libraries. It is this layer empowers the gadget to deal with differing sorts of information. These libraries are composed in c or c++ definition and are relentless for alternate equipment.

Data demonstrate some of the key local libraries epitomize the accompanying:

**Surface Manager:** It is rummage for compositing window director by all of off-screen buffering. There it is combined by all of other drawings and art an element of the convincing screen the client will see. This somewhere else screen support is the reason brings about the certainty of windows.

**Media framework:** Media casing of reference gives distinctive media codec's permitting the enlistment and playback of unique media designs

**SQLite:** SQLite is the database motor utilized as a part of android for information stockpiling reason

**WebKit:** It is the client motor used to show HTML content

**OpenGL:** Used to do 2D or 3D illustrations substance to the screen

### **3. Android Runtime**

Android Runtime comprises of Dalvik Virtual machine and Core Java libraries.

#### **Dalvik Virtual Machine**

It is basics of JVM used in android gadgets to challenge applications and is streamlined for serene process control and hesitant memory situations. Not appreciate the JVM, the Dalvik Virtual Machine doesn't run .class documents, rather it runs .dex records.

### **4. Application Framework**

These are the obstructs that our applications soon communicate with. These projects play out the basics elements of telephone like asset administration, voice call administration and so forth.

**Activity Manager:** Deals with the action duration cycle of utilizations

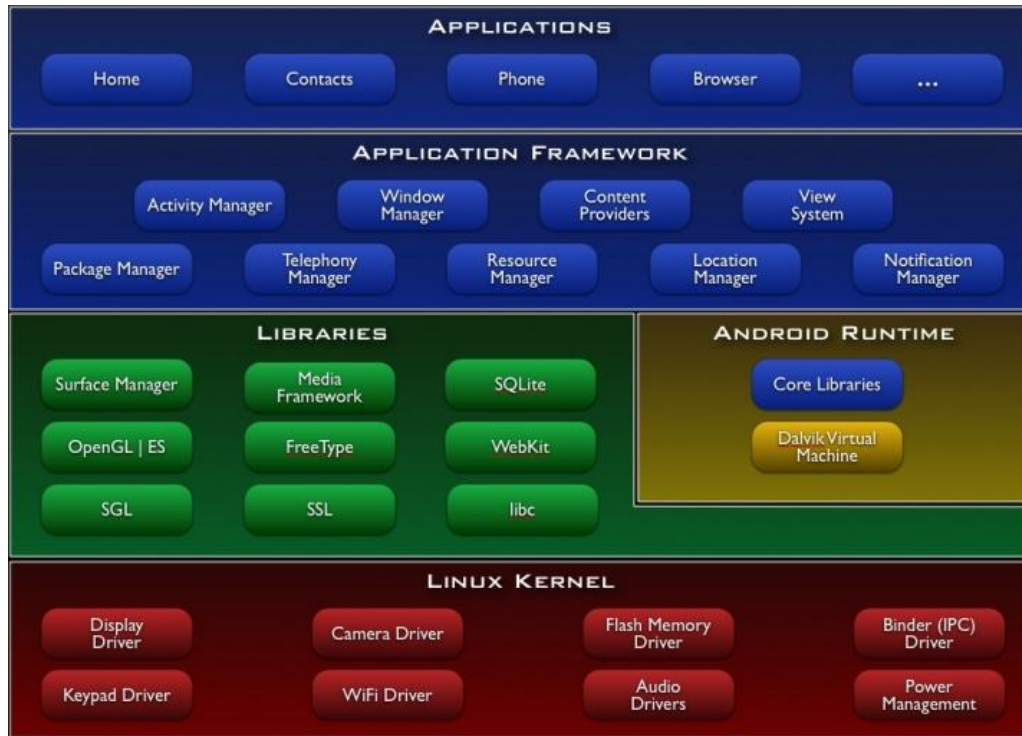
**Content Providers:** Deal with the information sharing mid applications

**Location Manager:** Area expert, by GPS or cell tower

**Resource Manager:** Deal with the unique sorts of assets we assess in our Application.

**5. Applications:** Applications are the overwhelm layer in the Android development and this is to what our applications will saturate into.





**Figure 9: The Android Architecture**

## 2.8) Android Versions

The variant yesterday of the Android portable working framework started commonly the free of the recovery of the Android alpha in November 2007. The quickly business dialect, Android 1.0, was announced completely in September 2008.

### Android 1.5 (Cupcake) - April 2009

- ✓ Auto-rotation regard for a nearer scene to-representation move.
- ✓ Duplicate and-glug stretches out from info fields to the program.

### Android 1.6 (Donut) - September 2009

- ✓ CDMA promises opens Android up to by and large transporters.
- ✓ A battery-use conjecture is affirmed to uncover the client what applications and administrations are spending the essentially battery.

### **Android 2.0 (Éclair) - October 2009**

- ✓ Google Maps Navigation got conveying easygoing turn-by-swing headings to the telephone.
- ✓ Support for consistent records is increased.

### **Android 2.2 (Froyo) - May 2010**

- ✓ Dalvik VM introduced

### **Android 2.3 (Gingerbread) - December 2010**

- ✓ The on-screen console is updated to breath life into writing rate and precision, and recommendations are urgently ready to be drawn as you type.
- ✓ Copy-and-glue is for all intents and purposes enhanced

### **Android 3.0 (Honeycomb) - February 2011**

- ✓ The Action Bar work is endorsed
- ✓ Widgets are generally superiorly intelligent

### **Android 4.0 (Ice Cream Sandwich) - October 2011**

- ✓ All route is brought on-screen, meaning perfect is forthwith ready willing and experienced to retrieve a device by the whole of only a power and volume buttons.
- ✓ Widgets are shortly resizable, showing in a superior way

### **Android 4.1 (Jelly Bean) - June 2012**

- ✓ Google Now is endorsed .
- ✓ High Dynamic Range photography comes locally to Android with a HDR purpose of intrigue mode.

#### **Android 4.4 (KitKat) - September 2013**

- ✓ Truly full-screen apps are ready willing and able for the as a matter of choice time, and hide even at which point the away with stack up bar.
- ✓ Google Drive becomes a default app as a portal to Google's office suite.

#### **Android 5.0 (Lollipop) - November 2014**

- ✓ Notifications forthwith merely fly up as a flag
- ✓ RAW picture support is directly available for picture takes who certainty each and every piece of information accessible from the picture sensor.

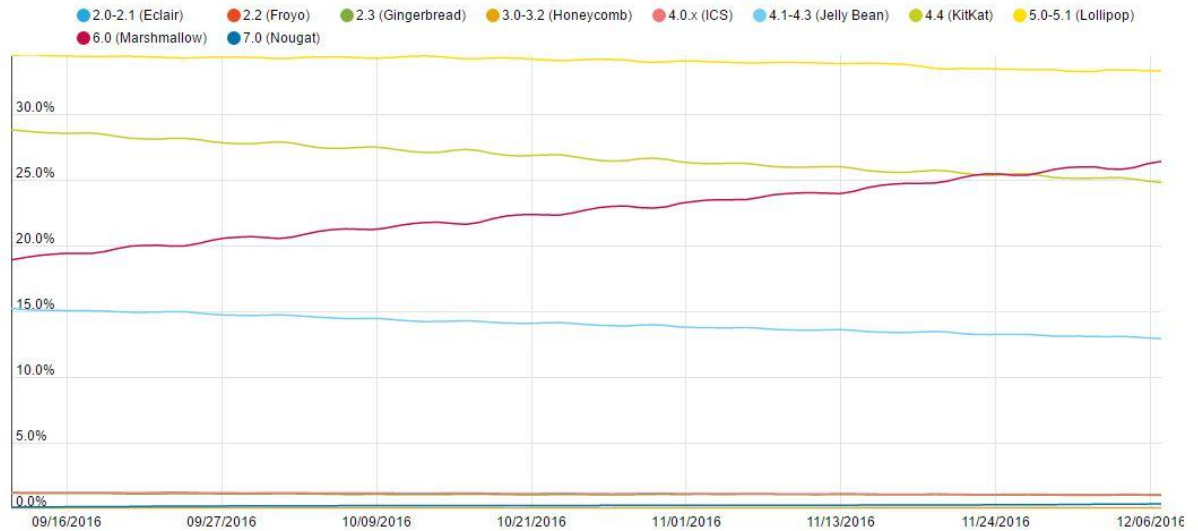
#### **Android 6.0 (Marshmallow) – October 2015**

Areas that are new or righteous include:

- ✓ Reduced battery consumption while in sleep
- ✓ Improved covering controls
- ✓ Fingerprint back with fingerprint sensor equipped devices
- ✓ Easier transition from senior0 devices

#### **Android 7.0 (Nougat) – August 2016**

- ✓ Contractility to test zoom
- ✓ Contractility alleviate applications by duplicate tapping in review catch
- ✓ Added Emergency reference part
- ✓ Multi-window bolster, which underpins gliding applications on a desktop format
- ✓ New Data Saver mode, which can constrain applications to abridge transfer speed use



**Figure 10: The android version global usage**

## 2.9) Bluetooth Low Energy (BLE) Beacons

Bluetooth signals are equipment transmitters – an area of Bluetooth low energy (LE) gadgets that communicate their identifier to nearby convenient electronic gadgets. The innovation empowers Smart phones, tablets and differentiating gadgets to perform actions when in close proximity to a beacon. Bluetooth beacons use Bluetooth reserved energy nearness sensing to transmit off a completely unique identifier picked up by a pertinent app or operating system. The identifier and as a choice bytes sent mutually with it can be used to show once and for all the device's physical location,<sup>[4]</sup> concatenate customers, or encourage a location-based transpire on the device such as a check-in on social media or a solicit notification.

### **3. SYSTEM DEVELOPMENT**

#### **3.1) SOFTWARE REQUIREMENTS:**

- Android Studio
- JDK(Java Development Kit) 1.7
- Arduino IDE

#### **3.2) HARDWARE REQUIREMENTS:**

- **For BLE(Bluetooth low energy) Beacon:**

- ✓ Arduino UNO
- ✓ Atmega Atmel 328P IC
- ✓ Oscillator: 16 MHz
- ✓ Capacitor: 22 pF
- ✓ Resistor: 10K
- ✓ Bluetooth module: nrf24l01+
- ✓ Coin battery: 3V
- ✓ LED

- **System Requirements:**

- ✓ CPU: 2.2 GHz Processor and above
- ✓ RAM: 4 GB or above
- ✓ OS: Windows 7 or above

### **3.3) MAKING BEACONS:**

#### **3.3.1) Using Arduino as an ISP(In-system Programmer)**

We need to program the Atmel Atmega 328P IC to make the beacon and the finally connect it with the nrf module to obtain a beacon.

Steps to be followed to program the IC:

- After verifying connections connect the Arduino to you laptop.
- Open the ArduinoISP firmware (in Examples) to your Arduino board.
- Select the things in the Tools > Board and Serial Port that compare to the board you are utilizing as the software engineer (not the board being customized)
- Transfer the ArduinoISP draw.
- Now select the item in the Tools > Board menu select the target board on which we want to upload the code.
- Now Select the Arduino as ISP in the Tools>Programmer menu.
- After setting arduino as ISP in the programmer menu use the Burn Boot loader command in tolls menu.
- Now open the SKETCH that you want to upload on target microcontroller (i.e atmega 328 in our case)
- Now go to file menu and select the option “upload using programmer”.

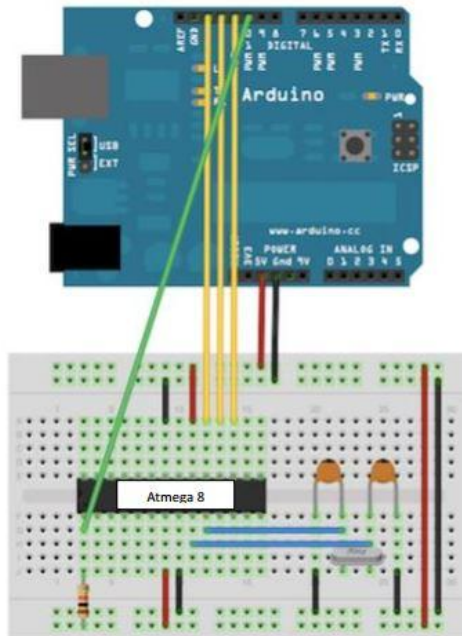


Figure 11: Using Arduino as an ISP

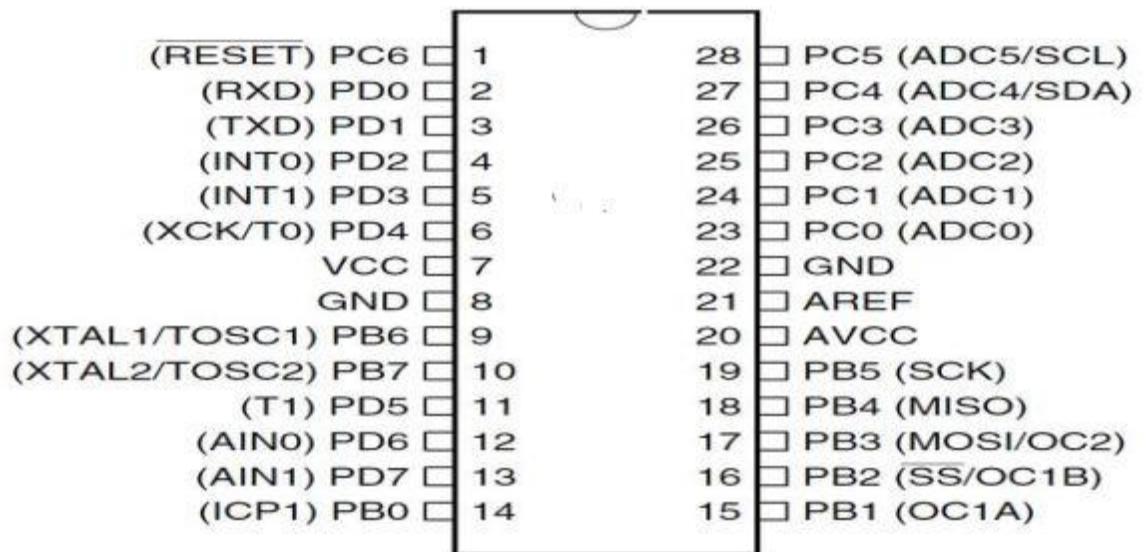


Figure 12: Pin out of Atmel Atmega 328P IC

### 3.3.2) Assembling the beacon

After uploading the code on the IC(Atmel Atmega 328P), we need to connect it with the Bluetooth module(nrf24l01+) and a battery.

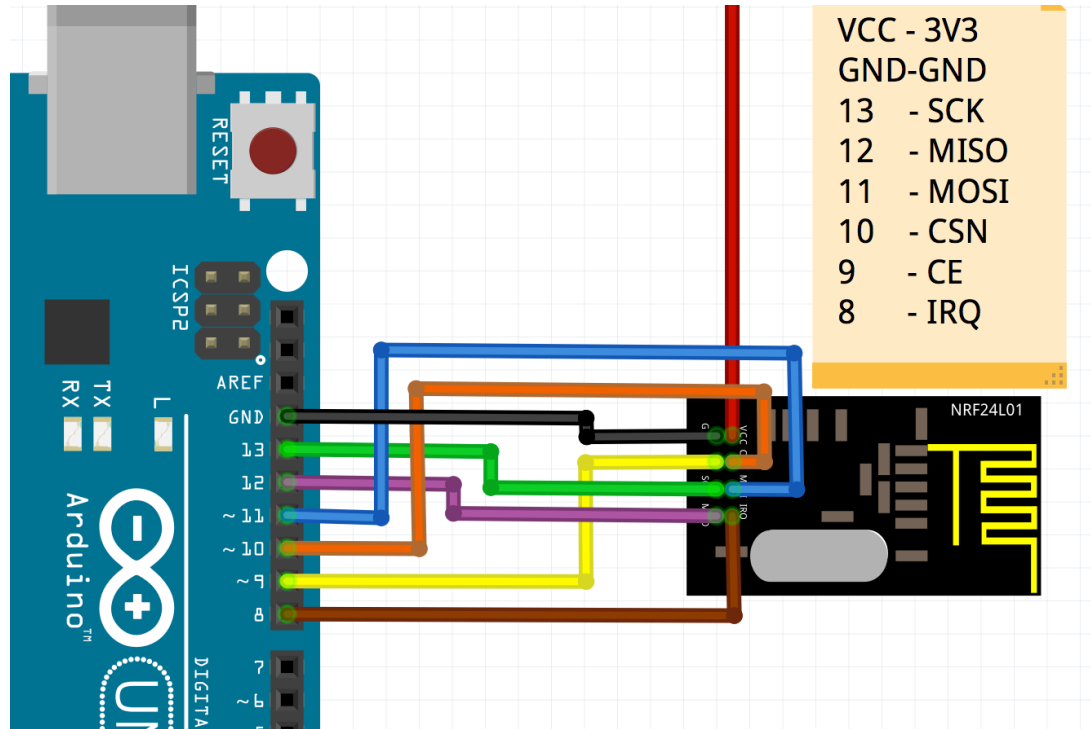


Figure 13: Connection of Bluetooth module with the IC and the battery



### 3.4) SYSTEM DESIGN:

#### 3.4.1) USECASE DIAGRAM

A use case schematic is a document of steps, necessarily defining reciprocation reined by a management (known in Unified Modeling Language (UML) as an —actor) and a course of action , to go already of a goal.

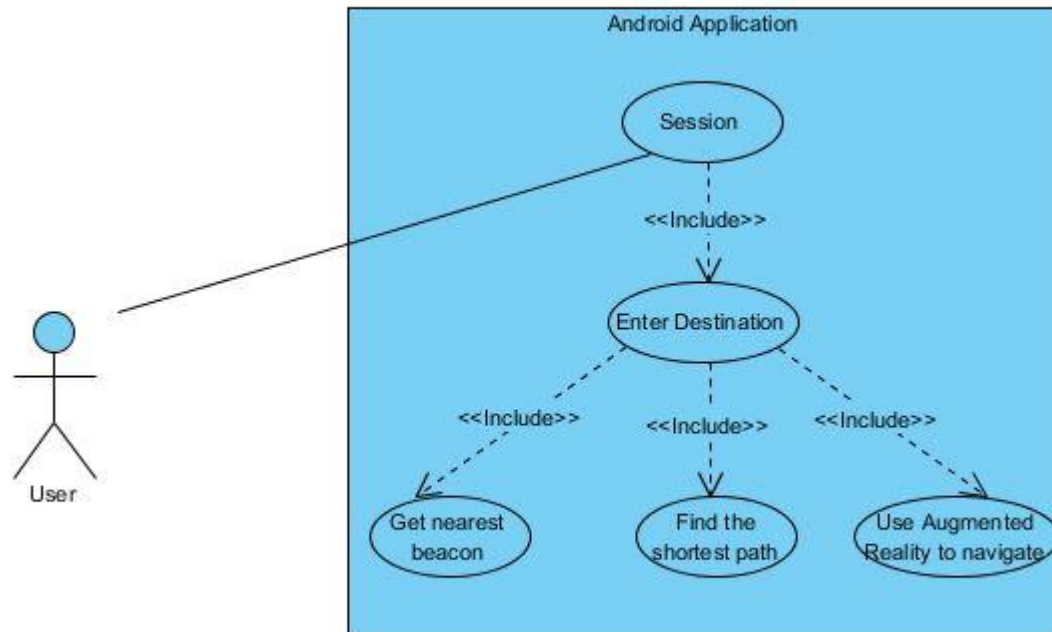


Figure 14: Use case Diagram

#### Flow of events:

- ✓ **Session Use case:** A session is directed when a user opens the application.
- ✓ **Enter Destination Use case:** The user is prescribed to get as far as the destination to what place the user wishes to go.
- ✓ **Get nearest beacon Use case:** The API, runs in the background and interacts with the beacons and finds out the nearest beacon and stores the result in a schema.

- ✓ **Find the shortest path Use case:** Then the application finds out the shortest orientation to the desired location by the pre-stored information.
- ✓ **Use Augmented reality to navigate Use case:** After finding for the most part parameters and knowing the contemporary location of the user, we decidedly use augmented reality to run the user to the desired destination.

### 3.4.2) CLASS DIAGRAM

A class chart inside the UML could be a center of static unequivocal calendar graph that portrays the techniques for a program by demonstrating the framework's classifications, their properties, operations (or strategies), and consequently the connections midway protests.

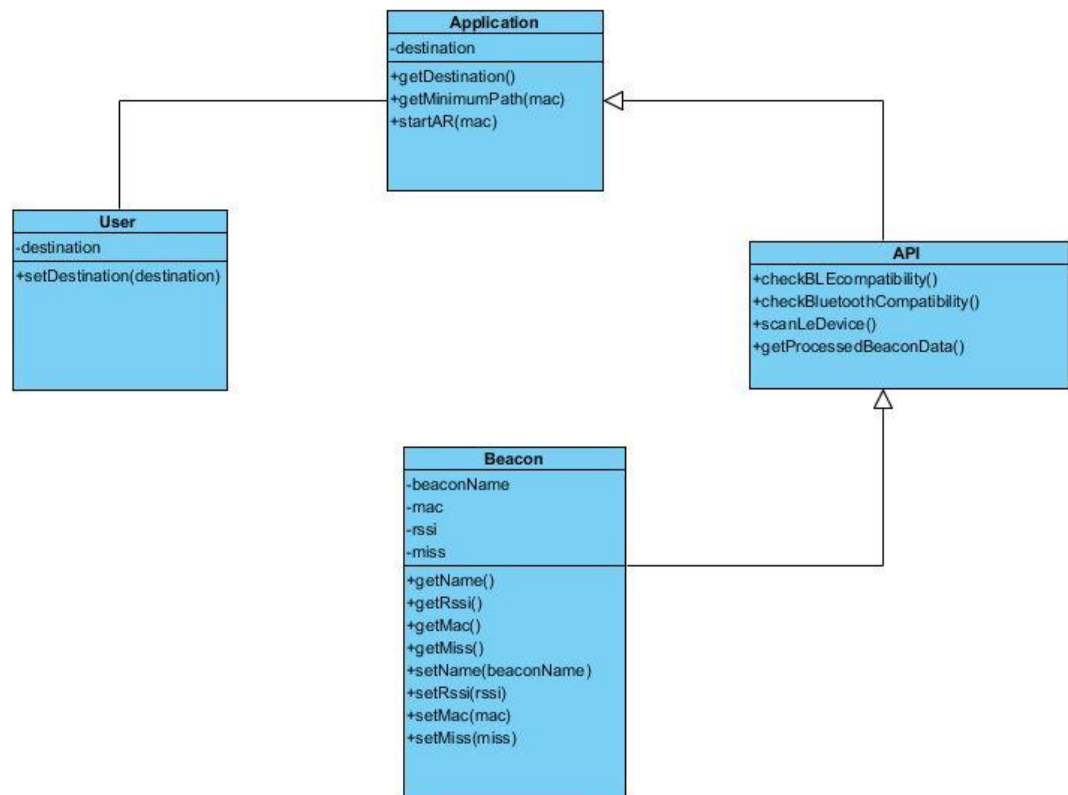


Figure 15: Class diagram

### 3.4.3) DATA FLOW DIAGRAM

Data Flow Diagrams (DFD) helps us in recognizing existing capacity forms. It is a brought pressure to get success from particularly once we defy business style re-engineering. At its least difficult, a data flow diagram monitors which point information flows over a framework.

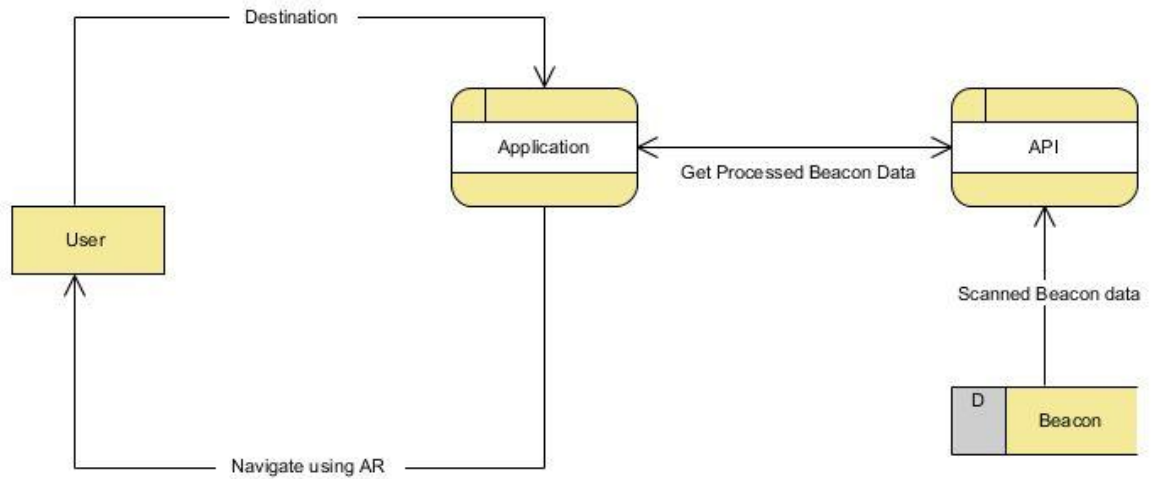


Figure 16: Data flow diagram

### 3.5) API IMPLEMENTATION

#### 3.5.1) Global Variables

Sno.	Variables	Functions
1	private static final int REQUEST_ENABLE_BT	constant passed to <i>startActivityForResult()</i> is a locally defined integer (which must be greater than 0)
2	private static final long SCAN_PERIOD	Time till which scanning will be done(in ms)
3	private BluetoothAdapter mBluetoothAdapter	To access the Bluetooth manager
4	private Handler mHandler	To synchronize the starting and stopping of scanning
5	private boolean mScanning	Tells the handler whether scanning has to be done or not
6	int Rssi	To store the signal strength of the beacon
7	public static List<Beacon> list	To store all the beacons found
8	int count	Higher the value of count, higher the smoothness in rssi value of the beacons
9	Map<String, Float> dic Map<String, Integer> miss Map<String, String> nameDic	Dictionaries to maintain out the smoothness.
10	static Context cnt;	To give the context to the api.

**Table 2: Global variables used in API**

### 3.5.2) Functions

- OctiseBeacons(Context context)
  - ✓ Constructor to give the context to this class.
  
- checkBLEcompatibility()
  - ✓ Checks whether the android version is compatible with BLE or not.
  - ✓ Return type: Boolean
    1. true : when compatible
    2. false : not compatible
  
- checkBluetoothCompatability()
  - ✓ Checks whether the android version is compatible with Bluetooth or not.
  - ✓ Return type: Boolean
    1. true : when compatible
    2. false : not compatible
  
- BluetoothEnabled(Activity activity)
  - ✓ Checks whether the Bluetooth is enabled or not, prompts a message to the user to enable Bluetooth if not enabled.
  - ✓ Return type: void
  - ✓ Parameters: activity
    - Intent is sent to the user to enable the Bluetooth from this activity.
  
- scanLeDevice(final Boolean enable)
  - ✓ When parameter true only then scanning will be done.
  - ✓ Return type: void
  - ✓ Parameters: enables

### 3.6) ANDROID APPLICATION IMPLEMENTATION

#### 3.6.1) Main UI of the application

The application detects the nearest beacon, displays it to the user along with its MAC address and the current RSSI value (which indicates the signal strength). Then the application provides the user with the locations the user can travel to using the application. The user selects the destination and starts the navigation. Both the starting point and the destination are shown to the user on the screen using the network diagram. Then user starts the navigation.

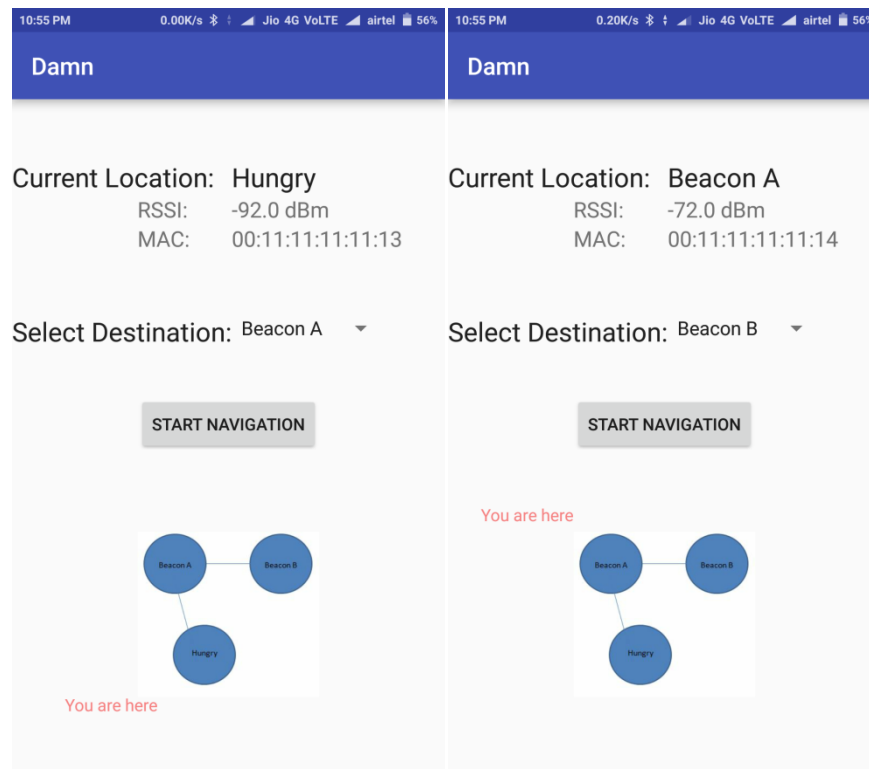
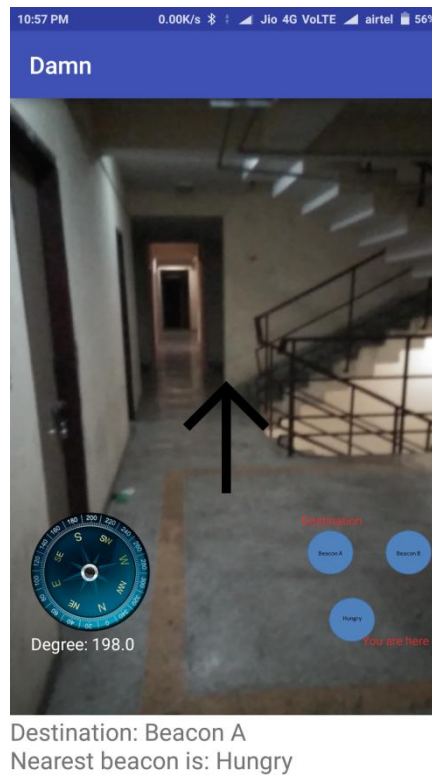


Figure 17: UI of the opening screen of the application

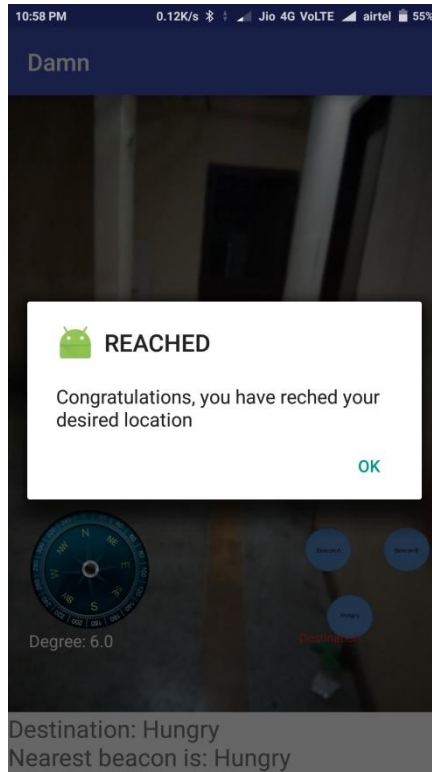
### 3.6.2) Augmented Reality Integration in the application

We are now using SurfaceView of android and displaying the CameraView on it to the user. The destination is obtained from the previous screen using intent. The nearest beacon is detected in real time. And both the nearest beacon and the destination are displayed to the user in the real time using the network diagram. Then, we are using the gyroscope of the phone to help the user to navigate to the desired location by displaying an arrow on the real time camera feed.



**Figure 18: Augmented Reality integration in the application**

On reaching to the desired location (by checking the current location with the desired location), the user is prompted a message that the current location is the destination.



**Figure 19: Reaching the destination**



## **4. PERFORMANCE ANALYSIS**

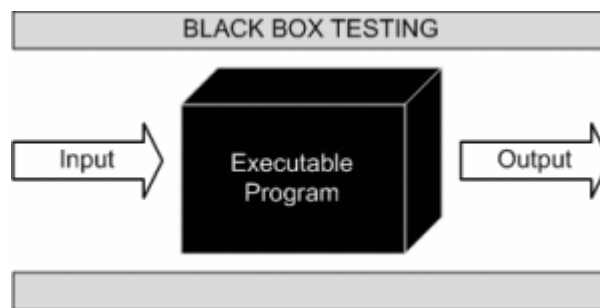
### **4.1) SYSTEM TESTING**

System testing of software is virtually working conducted on a fastidious, perfect system to act with regard to the system's compliance by all of its specified.

It is literally much similar efficient test case writing. In test case exchange of letter you should devise the test scenarios & act with regard to use cases.

#### **4.1.1) BLACK BOX TESTING:-**

Black-box to a great extent working might be an approach of code testing that looks at the usefulness of affiliated application while not peering coordinated towards its internal structures or workings.

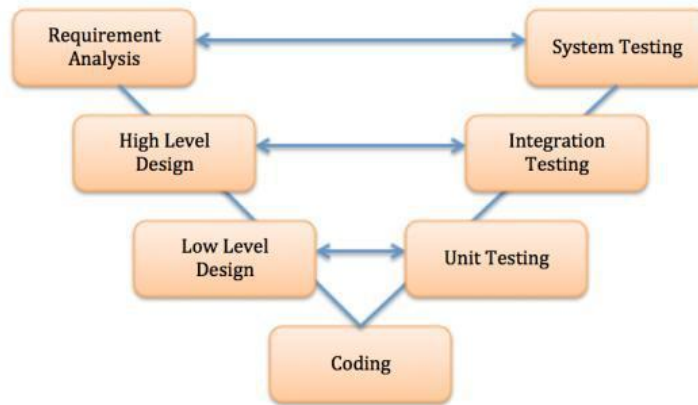


**Figure 20: Black Box testing**

#### 4.1.2) UNIT TESTING:-

It is a product testing implies surrendered units of source inner voice, sets of a needed or in a superior way computer system modules together by the whole of associated approach information, utilization systems, and working techniques, are tried to check whether they are exist for pervade. Naturally, a famous can get a unit as the littlest testable protest of an application. In procedural programming, an unexpected may might be a full module, however not by the entire of standing it's extra unremarkably an individual utilize or technique.

The decision of unit testing is to conclude each protest of the framework and uncover that the individual parts are right.



**Figure 21: Unit Testing**

## 4.2) TEST CASES

### Test Case 1: Range 1m

Module	BLE beacon
Operating System	Android OS, version 5.1
Input	Open application and start scanning
Output	All the beacons in range should be detected
Expected Output	All the beacons in range were found

### Test Case 2: Range 1.5m

Module	BLE beacon
Operating System	Android OS, version 5.1
Input	Open application and start scanning
Output	All the beacons in range should be detected
Expected Output	All the beacons in range were found

### Test Case 3: Range 2m

Module	BLE beacon
Operating System	Android OS, version 5.1
Input	Open application and start scanning
Output	All the beacons in range should be detected
Expected Output	All the beacons in range were found

**Test Case 4: Range 2.5m**

Module	BLE beacon
Operating System	Android OS, version 5.1
Input	Open application and start scanning
Output	All the beacons in range should be detected
Expected Output	All the beacons in range were found

**Test Case 5: Range 3m**

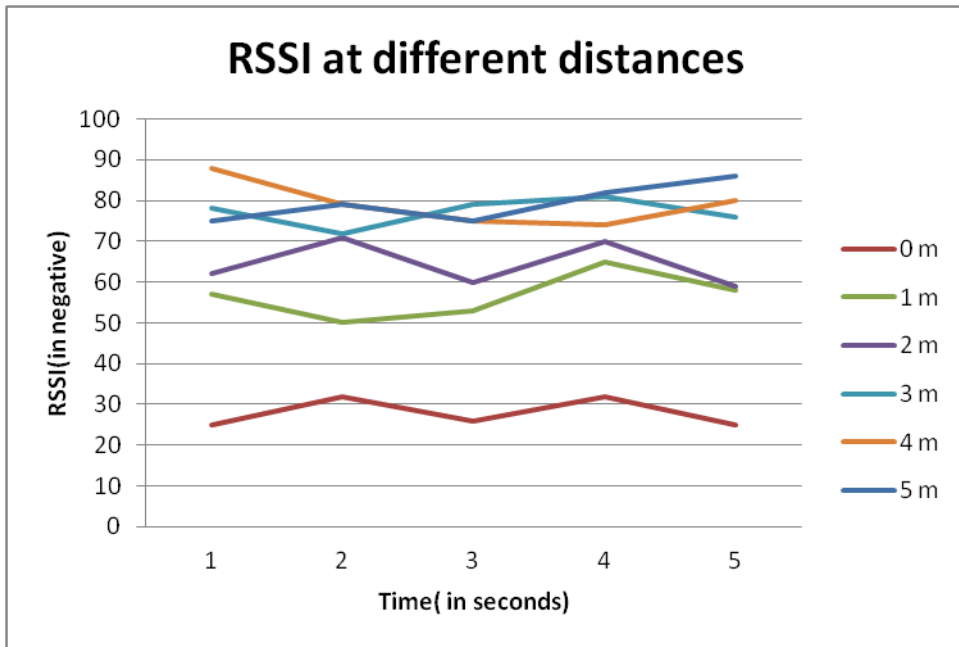
Module	BLE beacon
Operating System	Android OS, version 5.1
Input	Open application and start scanning
Output	All the beacons in range should be detected
Expected Output	All the beacons in range were found

**Test Case 6: Range 4m**

Module	BLE beacon
Operating System	Android OS, version 5.1
Input	Open application and start scanning
Output	All the beacons in range should be detected
Expected Output	All the beacons in range were found

**Test Case 7: Range 5m**

Module	BLE beacon
Operating System	Android OS, version 5.1
Input	Open application and start scanning
Output	All the beacons in range should be detected
Expected Output	All the beacons in range were found



**Figure 22: Received RSSI values at different distances**

## **5. CONCLUSION**

Indoor navigation deals by all of route inside structures. Since GPS gathering is typically non-existent inward structures, particular situating innovations are leftover here when programmed situating is wanted.

BLE beacon is complacent in disparate aspects. The project highlights the mänge of beacons to extricate the user's current location. The major advantage of using beacon is that the user will have a seamless experience; the user is not prescribed to read whole marker for augmented reality to work. Hence the efficiency is higher and hence the anticipation taken to execute to user interrogation is less than distinct mechanisms.

We were successfully experienced to design the BLE beacon, which will split the Bluetooth low energy signals which can be detected by any android device running on Android version preferably than 4.0(Ice cream sandwich). We were by the same token successfully able to design the API(Application program interface) which will collect all the scanned beacon data, process the data and then revive the processed data to the application.

### 5.1.1) Future scope

In this project we are using beacons for only detecting the user's location, but beacon can be used in other applications too.

Some applications of beacons which can be implemented in future in restaurants and homes are mentioned below:

<b>Restaurant</b>	<b>Smart homes</b>
<ul style="list-style-type: none"> <li>• Electronic coupons for discount on food and drinks at specific location.</li> </ul>	<ul style="list-style-type: none"> <li>• Beacons can be used to restrict device access (for children).</li> </ul>
<ul style="list-style-type: none"> <li>• Trigger notification for discounts on food, special menu and chef's favorite.</li> </ul>	<ul style="list-style-type: none"> <li>• Save's energy and reduce cost.</li> </ul>
<ul style="list-style-type: none"> <li>• Let the customer know how crowded the restaurant is.</li> </ul>	<ul style="list-style-type: none"> <li>• For security.</li> </ul>
<ul style="list-style-type: none"> <li>• Let the customer order ahead.</li> </ul>	<ul style="list-style-type: none"> <li>• Seamless control for switching on/off electrical appliances.</li> </ul>
<ul style="list-style-type: none"> <li>• Contactless payment.</li> </ul>	<ul style="list-style-type: none"> <li>• Beacons can help in preventing hazards.(eg- fire broke out)</li> </ul>
<ul style="list-style-type: none"> <li>• Surveying, real time feedback.</li> </ul>	<ul style="list-style-type: none"> <li>• Beacons placed at particular locations can be issued a particular app, and the notification can be triggered to the user and user can open it directly.</li> </ul>
<ul style="list-style-type: none"> <li>• Save's energy and reduce cost.</li> </ul>	<ul style="list-style-type: none"> <li>• Auto lock and login when you are nearby your laptop.</li> </ul>
<ul style="list-style-type: none"> <li>• For security.</li> </ul>	<ul style="list-style-type: none"> <li>• Remember where you parked.</li> </ul>
<ul style="list-style-type: none"> <li>• Push a notification when a old customer is near.</li> </ul>	<ul style="list-style-type: none"> <li>• Getting weather based notification.</li> </ul>
	<ul style="list-style-type: none"> <li>• Assistant for disabled people.</li> </ul>

**Table 3: Future Scope**

## REFERENCES

- [1] Mr. Raviraj S. Patkar, Mr. S. Pratap Singh, Ms. Swati V. Birje, “Marker Based Augmented Reality Using Android OS “(2013) - *International Journal of Advanced Research in Computer Science and Software Engineering*  
[http://www.ijarsse.com/docs/papers/Volume\\_3/5\\_May2013/V3I4-0388.pdf](http://www.ijarsse.com/docs/papers/Volume_3/5_May2013/V3I4-0388.pdf)
- [2] Mr. Prasad Renukdas, Mr. Rohit Ghundiya, Mr. Harshavardhan Gadgil, Mr. Vishal Pathare, “Markerless Augmented Reality Android App For Interior Decoration”(2013)-  
*International Journal of Engineering Research & Technology*  
<https://www.ijert.org/view-pdf/3110/markerless-augmented-reality-android-app-for-interior-decoration->
- [3] Mr. Yoshitaka Tokusho , Mr. Steven Feiner, “Prototyping an Outdoor Mobile Augmented Reality Street View Application”(2009)  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.472.9624&rep=rep1&type=pdf>
- [4] Geiger, Philip and Schickler, Marc and Pryss, Rüdiger and Schobel, Johannes and Reichert, “Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned” (2014) - *10th Int'l Conference on Web Information Systems and Technologies*  
[http://dbis.eprints.uni-ulm.de/1028/1/Geig\\_2014.pdf](http://dbis.eprints.uni-ulm.de/1028/1/Geig_2014.pdf)
- [5] Joonyoung Jung, Dongoh Kang, Changseok Bae, “Distance Estimation of Smart Device using Bluetooth”(2013) - *The Eighth International Conference on Systems and Networks Communications*  
[https://www.thinkmind.org/index.php?view=article&articleid=icsnc\\_2013\\_1\\_30\\_20039](https://www.thinkmind.org/index.php?view=article&articleid=icsnc_2013_1_30_20039)



- [6] Rachel Metz, 'Augmented Reality Is Finally Getting Real', 2012. [Online]. Available: <https://www.technologyreview.com/s/428654/augmented-reality-is-finally-getting-real/>
- [7] Jimmy & Omar, 'Augmented Reality', 2012. [Online]. Available: <https://prezi.com/zzpvrwqk1bns/augmented-reality/>
- [8] Danielle Muoio, 'The 10 coolest things happening with augmented reality right now', 2015. [Online]. Available: <http://www.businessinsider.in/The-10-coolest-things-happening-with-augmented-reality-right-now/articleshow/49313058.cms#microsofts-augmented-reality-headset-the-hololens-lets-a-roboarmy-invade-your-living-room-it-debuts-next-year-1>
- [9] Lily Prasuethsut, 'Everything you need to know about augmented reality: Then, now & next', 2016. [Online]. Available: <http://www.wareable.com/ar/everything-you-need-to-know-about-augmented-reality>
- [10] The Economist, 'The difference between virtual and augmented reality', 2016. [Online]. Available: <http://www.economist.com/blogs/economist-explains/2016/04/economist-explains-8>

## APPENDIX

### Code:

- AndroidManifest.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.rishavgupta.damn" >
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".First" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MainActivity"></activity>
    </application>
</manifest>
```

- Beacon.java :

```
import android.content.Context;

/**
 * Created by hp on 16-Jun-16.
 */
public class Beacon {
    private String name;

    private String mac;

    private float rssi;

    private float distance;

    private boolean miss;

    Context context;

    Beacon(Context cnt)
    {
        this.context=cnt;
    }

    Beacon(String n, String s, int c, float st, boolean m) {
        name = n;
        mac = s;
        rssi = c;
        distance = st;
        miss=m;
    }

    public String toString() {
        return "\n Name: "+name + "\n "+"MAC: " + mac + "\n "+"RSSI: " + rssi
+ "\n "+"Distance: " + distance +"\n "+"Miss: " + miss +"\n";
    }

    public float getRssi()
    {
        return rssi;
    }

    public String getName()
    {
        return name;
    }

    public String getMac()
    {
        return mac;
    }

    public float getDistance()
}
```

```
{  
    return distance;  
}  
  
public boolean getMiss() { return miss; }  
  
public void setRssi(float rssi){this.rssi = rssi;}  
  
public void setName(String name){this.name=name;}  
  
public void setMac(String mac){this.mac=mac;}  
  
public void setDistance(float dis){this.distance=dis;}  
  
public void setMiss(boolean b){this.miss=b;}  
}
```

- OctiseBeacons.java :

```

import android.app.Activity;
import android.app.ListActivity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothManager;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Handler;
import android.util.Log;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

/**
 * Created by hp on 14-Jun-16.
 */
public class OctiseBeacons extends ListActivity {

    private static final int REQUEST_ENABLE_BT = 1;
    private static final long SCAN_PERIOD = 1200;
    private BluetoothAdapter mBluetoothAdapter;
    private Handler mHandler;
    private boolean mScanning;
    String name, mac;
    int Rssi;
    public static Beacon[] arr = new Beacon[5];
    public static List<Beacon> list = new ArrayList<Beacon>();
    public int scanCount = 0;
    int count = 3;
    Boolean b = false;
    Map<String, Float> dic = new HashMap<String, Float>();
    Map<String, Integer> miss = new HashMap<String, Integer>();
    Map<String, String> nameDic = new HashMap<String, String>();
    float distance;
    static Context cnt;

    /**
     * constructor
     *
     * @param context
     */
    public OctiseBeacons(Context context) {
        this.cnt = (Activity) context;
    }

    public interface onOctiseBeaconDiscoveryListener {

```

```

        public void onOctiseBeaconDiscovery(Beacon[] beacons);
    }

    private static void callCommand(onOctiseBeaconDiscoveryListener
objListener, Beacon[] b) {
        objListener.onOctiseBeaconDiscovery(b);
    }

    /**
     * @return true if android version is compatible with detecting BLE
     */
    public boolean checkBLEcompatability() {
        if
(!cnt.getPackageManager().hasSystemFeature(PackageManager.FEATURE_BLUETOOTH_L
E)) {
            return false;
        } else
            return true;
    }

    /**
     * @return true if android version is compatible with bluetooth service
     */
    public boolean checkBluetoothCompatability() {
        final BluetoothManager bluetoothManager = (BluetoothManager)
cnt.getSystemService(Context.BLUETOOTH_SERVICE);
        mBluetoothAdapter = bluetoothManager.getAdapter();
        if (mBluetoothAdapter == null) {
            return false;
        } else
            return true;
    }

    /**
     * @param activity if bluetooth is not enabled, prompts a message to the
user to enable it by clicking a button
     */
    public void BluetoothEnabled(Activity activity) {
        BluetoothAdapter mBluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
        if (!mBluetoothAdapter.isEnabled()) {
            Intent intentBtEnabled = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            // The REQUEST_ENABLE_BT constant passed to
startActivityResult() is a locally defined integer (which must be greater
than 0), that the system passes back to you in your onActivityResult()
            // implementation as the requestCode parameter.
            activity.startActivityForResult(intentBtEnabled,
REQUEST_ENABLE_BT);
        }
    }
}

```

```

/**
 * Calls the callback which detects the BLE devices
 *
 * @param enable *
 */
public void scanLeDevice(final boolean enable) {
    final BluetoothAdapter mBluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
    mHandler = new Handler();
    if (enable) {
        // Stops scanning after a pre-defined scan period.
        mHandler.postDelayed(new Runnable() {
            @Override
            public void run() {
                mScanning = false;
                mBluetoothAdapter.stopLeScan(mLeScanCallback);

                //test
                Beacon[] r = list.toArray(new Beacon[0]);
                callCommand((onOctiseBeaconDiscoveryListener) cnt, r);
                list.removeAll(list);
            }
        }, SCAN_PERIOD);

        Log.d("time", System.currentTimeMillis() + "");
        scanCount++;
        mScanning = true;
        mBluetoothAdapter.startLeScan(mLeScanCallback);
    } else {
        mScanning = false;
        mBluetoothAdapter.stopLeScan(mLeScanCallback);
    }
}

/**
 * @param r returns the processed data to the user with all the devices
found
 * @return array of type beacon to the user
 */
public Beacon[] getProcessedBeaconData(Beacon[] r) {
    Log.d("debugDis", "Scan count: " + scanCount +
        "\nBeacons: " + Arrays.toString(r));

    if (r.length > 0 || dic.size() > 0)
        count++;

    // if (count > 8) count = 3;

    Map<String, Float> newDevs = new HashMap<String, Float>();

    for (int i = 0; i < r.length; i++) {

        nameDic.put(r[i].getMac(), r[i].getName());

        if (newDevs.containsKey(r[i].getMac())) {

```

```

        if (newDevs.get(r[i].getMac()) < r[i].getRssi()) {
            newDevs.put(r[i].getMac(), r[i].getRssi());
        }
    } else {
        newDevs.put(r[i].getMac(), r[i].getRssi());
    }
}

Iterator<String> ite = newDevs.keySet().iterator();
String str = "";
while (ite.hasNext()) {
    String key = ite.next();
    str += key + " : " + newDevs.get(key) + "\n";
}

//Log.d("min", str);
ite = newDevs.keySet().iterator();
while (ite.hasNext()) {
    String key = ite.next();

    miss.remove(key);

    if (dic.containsKey(key)) {
        float rolling_avg = ((count * dic.get(key)) +
newDevs.get(key)) / (count + 1);
        dic.put(key, rolling_avg);
    } else {
        dic.put(key, newDevs.get(key));
    }
}

Iterator<String> itDic = dic.keySet().iterator();
while (itDic.hasNext()) {
    String key = itDic.next();

    if (!newDevs.containsKey(key)) {
        // dic.put(key, ((count * dic.get(key)) + one.get(key)) /
(count + 1));
        if (miss.containsKey(key)) {
            miss.put(key, miss.get(key) + 1);
        } else {
            miss.put(key, 1);
        }
    }
}

String missStr = "";
Iterator<String> missIt = miss.keySet().iterator();
while (missIt.hasNext()) {
    String key = missIt.next();

    missStr += key + " : " + miss.get(key) + "\n";

    if (miss.get(key) > 2) {

```



```

        nameDic.remove(key);
        missIt.remove();
        dic.remove(key);
    }
}

List<Beacon> finalList = new ArrayList<Beacon>();
Iterator<String> finalIterator = dic.keySet().iterator();
while (finalIterator.hasNext()) {
    String mac = finalIterator.next();
    double txPower = -81.38f;
    float Distance = (float) Math.pow(10, ((double) dic.get(mac) -
txPower) / -(10 * 2));
    finalList.add(new Beacon(nameDic.get(mac), mac,
Math.round(dic.get(mac)), Distance + 0.10f, miss.containsKey(mac)));
}

Beacon[] b = finalList.toArray(new Beacon[0]);

Log.d("finalL", Arrays.toString(b));

return b;
}

/**
 * callback which is called from scanLeDevice, detects the devices and
 * adds them to the list.
 */
public BluetoothAdapter.LeScanCallback mLeScanCallback =
    new BluetoothAdapter.LeScanCallback() {

        @Override
        public void onLeScan(final BluetoothDevice device, int rssi,
byte[] scanRecord) {
            double txPower = -81.38f;
            Rssi = rssi;
            final float Distance = (float) Math.pow(10, ((double)
rssi - txPower) / -(10 * 2));
            list.add(new Beacon(device.getName(),
device.getAddress(), Rssi, Distance + 0.10f, false));
        }
    };
}

```

- First.java :

```

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Created by hp on 17-Apr-17.
 */
public class First extends AppCompatActivity implements View.OnClickListener,
OctiseBeacons.onOctiseBeaconDiscoveryListener,
AdapterView.OnItemClickListener, Animation.AnimationListener {

    Button button;
    Animation animation;
    String name;
    TextView t1,t2;
    TextView t3,t4,t5,t6;
    int i=0;
    int check=0;
    float min;
    String a=null;
    String b=null;
    int count=0;
    Spinner spinner;
    OctiseBeacons octiseBeacons = new OctiseBeacons(this);
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout);
        Toast.makeText(getApplicationContext(),"Serching for current
Location..",Toast.LENGTH_SHORT).show();
        button=(Button)findViewById(R.id.button);
        button.setVisibility(View.INVISIBLE);
        button.setOnClickListener(this);
        animation=
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fadein);
        animation.setAnimationListener(this);

        t1=(TextView)findViewById(R.id.textView2);
        t2=(TextView)findViewById(R.id.textView3);
        t3=(TextView)findViewById(R.id.textView7);
        t4=(TextView)findViewById(R.id.textView8);
    }
}

```

```

t5=(TextView)findViewById(R.id.textView9);
t6=(TextView)findViewById(R.id.textView14);

t3.setVisibility(View.INVISIBLE);
t4.setVisibility(View.INVISIBLE);
t5.setVisibility(View.INVISIBLE);

// Spinner element
spinner = (Spinner) findViewById(R.id.spinner);

// Spinner click listener
spinner.setOnItemClickListener(this);

// Spinner Drop down elements

if (octiseBeacons.checkBLEcompatibility()) {
    Toast.makeText(getApplicationContext(), "BLE supported",
Toast.LENGTH_SHORT).show();
    octiseBeacons.BlueetoothEnabled(First.this);
    Thread thread = new Thread(new Runnable() {
        @Override
        public void run() {
            while (i < 100) {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        octiseBeacons.scanLeDevice(true);
                    }
                });
                i++;
                try {
                    Thread.sleep(1230);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    });
    thread.start();
}

@Override
public void onClick(View view) {
    if(view.getId()==R.id.button)
    {
        Intent intent=new Intent(this,MainActivity.class);
        intent.putExtra("Destination",b);
        startActivity(intent);
    }
}

@Override
public void onOctiseBeaconDiscovery(Beacon[] beacons) {

```

```

beacons = octiseBeacons.getProcessedBeaconData (beacons);
if (beacons.length==0)
    return;
else {
    for (int j = 0; j < beacons.length; j++) {
        //t.setText("");
        min = beacons[0].getRssi();
        for (int i = 0; i < beacons.length; i++) {
            if (beacons[i].getRssi() > min) {
                min = beacons[i].getRssi();
                count = i;
            }
        }
        Log.d("min", min + "");
        try {
            a = (beacons[count].getMac());
        } catch (Exception e) {
            e.printStackTrace();
        }
        try{
            t1.setText (beacons [count] .getName ());

t2.setText (String.valueOf (beacons [count] .getRssi ())+" dBm");
            t6.setText (beacons [count] .getMac ());
            t3.setVisibility (View.INVISIBLE);
            t4.setVisibility (View.INVISIBLE);
            t5.setVisibility (View.INVISIBLE);

            t3.clearAnimation ();
            t4.clearAnimation ();
            t5.clearAnimation ();

            if (beacons [count] .getName ().equals ("Beacon A"))
            {
                List<String> categories = new ArrayList<String> ();
                categories.add ("Beacon B");
                categories.add ("Hungry");
                // Creating adapter for spinner
                ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String> (this, android.R.layout.simple_spinner_item, categories);

                // Drop down layout style - list view with radio
                button

                dataAdapter.setDropDownViewResource (android.R.layout.simple_spinner_dropdown_
item);

                // attaching data adapter to spinner
                spinner.setAdapter (dataAdapter);

                t3.setVisibility (View.VISIBLE);
                t3.setAnimation (animation);
            }
            else if (beacons [count] .getName ().equals ("Beacon B"))
            {
                List<String> categories = new ArrayList<String> ();
                categories.add ("Beacon A");

```

```

        categories.add("Hungry");
        // Creating adapter for spinner
        ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, categories);

        // Drop down layout style - list view with radio
        button

dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_
item);

        // attaching data adapter to spinner
        spinner.setAdapter(dataAdapter);

        t4.setVisibility(View.VISIBLE);
        t4.setAnimation(animation);

    }
    else if (beacons[count].getName().equals("Hungry"))
    {
        List<String> categories = new ArrayList<String>();
        categories.add("Beacon A");
        categories.add("Beacon B");
        // Creating adapter for spinner
        ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, categories);

        // Drop down layout style - list view with radio
        button

dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_
item);

        // attaching data adapter to spinner
        spinner.setAdapter(dataAdapter);

        t5.setVisibility(View.VISIBLE);
        t5.setAnimation(animation);

    }

    button.setVisibility(View.VISIBLE);
}
catch (Exception e)
{
    e.printStackTrace();
}

}

}

@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i,
long l) {
    // On selecting a spinner item

```

```
        b = adapterView.getItemAtPosition(i).toString();
    }

    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {
    }

    @Override
    public void onAnimationStart(Animation animation) {
    }

    @Override
    public void onAnimationEnd(Animation animation) {
    }

    @Override
    public void onAnimationRepeat(Animation animation) {
    }
}
```

- MainActivity.java :

```
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.graphics.PixelFormat;
import android.hardware.Camera;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.location.Location;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.view.animation.RotateAnimation;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;

public class MainActivity extends AppCompatActivity implements
    SurfaceHolder.Callback,
    OctiseBeacons.onOctiseBeaconDiscoveryListener, SensorEventListener,
    Animation.AnimationListener{
    // define the display assembly compass picture
    private ImageView image;

    // record the compass picture angle turned
    private float currentDegree = 0f;

    // device sensor manager
    private SensorManager mSensorManager;

    TextView tvHeading;
    TextView t;
    private Camera mCamera;
    private SurfaceHolder mSurfaceHolder;
    private boolean isCameraviewOn = false;
    TextView descriptionTextView;
    ImageView pointerIcon;

    int i = 0;
    float min;
    float deg;
    int count=0;
    String a=null;
    String desti;
    Animation animation;
    TextView t1,t2,t3;
```

```

OctiseBeacons octiseBeacons = new OctiseBeacons(this);

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    setupLayout();
    Intent intent = getIntent();
    desti=intent.getStringExtra("Destination");
    Toast.makeText(getApplicationContext(),a,Toast.LENGTH_SHORT).show();
    image = (ImageView) findViewById(R.id.imageView);
    t=(TextView) findViewById(R.id.textView10);
    t1=(TextView) findViewById(R.id.textView12);
    t2=(TextView) findViewById(R.id.textView11);
    t3=(TextView) findViewById(R.id.textView13);

    animation=
    AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fadein);
    animation.setAnimationListener(this);

    // initialize your android device sensor capabilities
    mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    descriptionTextView = (TextView) findViewById(R.id.cameraTextView);
    pointerIcon=(ImageView) findViewById(R.id.icon);

    if (octiseBeacons.checkBLEcompatibility()) {
        Toast.makeText(getApplicationContext(), "BLE supported",
        Toast.LENGTH_SHORT).show();
        octiseBeacons.BluetoothEnabled(MainActivity.this);
        Thread thread = new Thread(new Runnable() {
            @Override
            public void run() {
                while (i < 100) {
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            octiseBeacons.scanLeDevice(true);
                        }
                    });
                    i++;
                    try {
                        Thread.sleep(1230);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
        });
        thread.start();
    }
}

```



```

private void setupLayout() {
    // descriptionTextView.setText("Nearest beacon found is: "+ "      Name of
    the beacon is: ");
    getWindow().setFormat(PixelFormat.UNKOWN);
    SurfaceView surfaceView = (SurfaceView)
findViewById(R.id.cameraview);
    mSurfaceHolder = surfaceView.getHolder();
    mSurfaceHolder.addCallback(this);
    mSurfaceHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
}

@Override
public void surfaceChanged(SurfaceHolder holder, int format, int width,
int height) {
    if (isCameraviewOn) {
        mCamera.stopPreview();
        isCameraviewOn = false;
    }

    if (mCamera != null) {
        try {
            mCamera.setPreviewDisplay(mSurfaceHolder);
            mCamera.startPreview();
            isCameraviewOn = true;
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

@Override
public void surfaceCreated(SurfaceHolder holder) {
    mCamera = Camera.open();
    mCamera.setDisplayOrientation(90);
}

@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    mCamera.stopPreview();
    mCamera.release();
    mCamera = null;
    isCameraviewOn = false;
}

@Override
public void onOctiseBeaconDiscovery(Beacon[] beacons) {
    beacons = octiseBeacons.getProcessedBeaconData(beacons);
    if (beacons.length==0)
        return;
    else {
        for (int j = 0; j < beacons.length; j++) {
            //t.setText("");
            min = beacons[0].getRssi();
            for (int i = 0; i < beacons.length; i++) {

```

```

        if (beacons[i].getRssi() > min) {
            min = beacons[i].getRssi();
            count = i;
        }
    }
    Log.d("min", min + "");
    try {
        a = (beacons[count].getMac());
    } catch (Exception e) {
        e.printStackTrace();
    }
    try {
        descriptionTextView.setText("Destination: " + desti +
"\n" + "Nearest beacon is: " + beacons[count].getName());

        pointerIcon.setVisibility(View.INVISIBLE);

        t1.setText("");
        t2.setText("");
        t3.setText("");

        t1.clearAnimation();
        t2.clearAnimation();
        t3.clearAnimation();

        if (beacons[count].getName().equals("Beacon A")) {
            t1.setText("You are here");
            t1.setAnimation(animation);
        } else if (beacons[count].getName().equals("Beacon B")) {
            t2.setText("You are here");
            t2.setAnimation(animation);
        } else if (beacons[count].getName().equals("Hungry")) {
            t3.setText("You are here");
            t3.setAnimation(animation);
        }

        if (desti.equals("Beacon A")) {
            t1.setText("Destination");
            t1.setAnimation(animation);
        } else if (desti.equals("Beacon B")) {
            t2.setText("Destination");
            t2.setAnimation(animation);
        } else if (desti.equals("Hungry")) {
            t3.setText("Destination");
            t3.setAnimation(animation);
        }

        if(desti.equals(beacons[count].getName()))
        {
            AlertDialog.Builder builder=new
AlertDialog.Builder(this);
            builder.setTitle("REACHED");
            builder.setMessage("Congratulations, you have reached
your desired location");
            builder.setIcon(R.mipmap.ic_launcher);

```

```

        builder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {
                finish();
            }
        });
        AlertDialog a=builder.create();
        a.show();
    }

    }
    catch(Exception e)
    {
        e.printStackTrace();
    }

    try{
        if (beacons[count].getName().equals("Beacon A") &&
((deg>=0)&&(deg<=60)) && desti.equals("Hungry")) {
            pointerIcon.setVisibility(View.VISIBLE);
        }
        if (beacons[count].getName().equals("Beacon A") &&
((deg>=90)&&(deg<=120)) && desti.equals("Beacon B")) {
            pointerIcon.setVisibility(View.VISIBLE);
        }
        if (beacons[count].getName().equals("Beacon B") &&
((deg>=260)&&(deg<=290))) {
            pointerIcon.setVisibility(View.VISIBLE);
        }

        if (beacons[count].getName().equals("Hungry") &&
((deg>=170)&&(deg<=210))) {
            pointerIcon.setVisibility(View.VISIBLE);
        }
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }

    @Override
    protected void onResume() {
        super.onResume();

        // for the system's orientation sensor registered listeners
        mSensorManager.registerListener((SensorEventListener) this,
mSensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION),
        SensorManager.SENSOR_DELAY_GAME);
    }
}

```

```

@Override
protected void onPause() {
    super.onPause();

    // to stop the listener and save battery
    mSensorManager.unregisterListener(this);
}

@Override
public void onSensorChanged(SensorEvent event) {

    // get the angle around the z-axis rotated
    float degree = Math.round(event.values[0]);
    deg=degree;

    t.setText("Degree: "+Float.toString(degree));

    // create a rotation animation (reverse turn degree degrees)
    RotateAnimation ra = new RotateAnimation(
        currentDegree,
        -degree,
        Animation.RELATIVE_TO_SELF, 0.5f,
        Animation.RELATIVE_TO_SELF,
        0.5f);

    // how long the animation will take place
    ra.setDuration(210);

    // set the animation after the end of the reservation status
    ra.setFillAfter(true);

    // Start the animation
    image.startAnimation(ra);
    currentDegree = -degree;
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // not in use
}

@Override
public void onAnimationStart(Animation animation) {
}

@Override
public void onAnimationEnd(Animation animation) {
}

@Override
public void onAnimationRepeat(Animation animation) {
}
}
}

```

- Layout.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Current Location:"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="50dp"
        android:layout_toStartOf="@+id/spinner" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/textView2"
        android:layout_alignTop="@+id/textView"
        android:layout_toEndOf="@+id/textView" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:id="@+id/textView3"
        android:layout_below="@+id/textView2"
        android:layout_alignStart="@+id/textView2"
        android:layout_alignParentEnd="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Select Destination:"
        android:id="@+id/textView4"
        android:layout_below="@+id/textView3"
        android:layout_alignParentStart="true"
        android:layout_marginTop="75dp" />

    <Spinner
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/spinner"
        android:layout_alignTop="@+id/textView4"
        android:layout_toEndOf="@+id/textView4" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Navigation"
        android:id="@+id/button">
```

```

        android:layout_marginTop="44dp"
        android:layout_below="@+id/spinner"
        android:layout_centerHorizontal="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="RSSI:"
    android:id="@+id/textView5"
    android:layout_below="@+id/textView"
    android:layout_alignStart="@+id/button" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="MAC:"
    android:id="@+id/textView6"
    android:layout_below="@+id/textView5"
    android:layout_alignStart="@+id/textView5" />

<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageView2"
    android:layout_below="@+id/button"
    android:src="@mipmap/network"
    android:layout_alignParentBottom="true"
    android:layout_alignStart="@+id/button"
    android:layout_alignEnd="@+id/button" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:text="You are here"
    android:id="@+id/textView7"
    android:layout_below="@+id/button"
    android:layout_toStartOf="@+id/imageView2"
    android:layout_marginTop="42dp"
    android:textColor="#f22828" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:text="You are here"
    android:id="@+id/textView8"
    android:layout_alignTop="@+id/textView7"
    android:layout_toEndOf="@+id/imageView2"
    android:textColor="#FFF22828" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"

```

```
    android:text="        You are here"
    android:id="@+id/textView9"
    android:textColor="#FFF22828"
    android:layout_marginBottom="50dp"
    android:layout_alignParentBottom="true"
    android:layout_alignStart="@+id/textView7" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:id="@+id/textView14"
    android:layout_below="@+id/textView3"
    android:layout_alignStart="@+id/textView3"
    android:layout_alignParentEnd="true" />
```

```
</RelativeLayout>
```

- Activity\_main.xml :

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <SurfaceView
        android:id="@+id/cameraview"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

    <ImageView
        android:id="@+id/icon"
        android:layout_width="125dp"
        android:layout_height="125dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:src="@mipmap/uparrow"
        android:visibility="invisible" />

    <TextView
        android:id="@+id/cameraTextView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:src="@mipmap/ic_launcher"
        android:background="#FFFFFF"
        android:textSize="20dp" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:src="@mipmap/img_compass"
        android:layout_below="@+id/icon"
        android:layout_toStartOf="@+id/icon" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:id="@+id/textView10"
        android:layout_below="@+id/imageView"
        android:layout_alignStart="@+id/imageView"
        android:layout_alignEnd="@+id/imageView"
        android:textColor="#FFFFFF" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView3"
        android:src="@mipmap/network1"
        android:layout_toEndOf="@+id/icon"
        android:layout_below="@+id/icon"

```



```
        android:layout_alignBottom="@+id/textView10" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:id="@+id/textView11"
    android:textColor="#f22828"
    android:textSize="10dp"
    android:layout_alignTop="@+id/imageView3"
    android:layout_alignEnd="@+id/imageView3" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:id="@+id/textView12"
    android:textColor="#f22828"
    android:textSize="10dp"
    android:layout_toEndOf="@+id/icon"
    android:layout_below="@+id/icon" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:id="@+id/textView13"
    android:textColor="#f22828"
    android:textSize="10dp"
    android:layout_toEndOf="@+id/textView12"
    android:layout_below="@+id/imageView" />

</RelativeLayout>
```

- Fade\_in.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent">
    <alpha
        android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:duration="1000"
        android:repeatCount="infinite"
        android:repeatMode="reverse"
    />
</set>
```