End semester report on R & D Project (NU 302)

Academic Year- 2020-21

on

GUEST RECOGNITION FEATURE IN HOTELS

A dissertation
Submitted in partial fulfillment of the requirements for the award of the degree
Bachelor of Technology

by

1.	**Shrihari Shukla**	**BT18GCS121**	**CSE**

Under supervision of
Prof.Manish Hurkat

NU
NIIT UNIVERSITY
THE UNIVERSITY OF THE FUTURE
Industry-linked | Technology-based | Research-driven | Seamless

NIIT University, Neemrana, Rajasthan-301705
May 2021

NU

NIIT UNIVERSITY

THE UNIVERSITY OF THE FUTURE

Industry-linked | Technology-based | Research-driven | Seamless

DECLARATION BY STUDENT(S)

I/We hereby declare that the project report entitled "**GUEST RECOGNITION FEATURE IN HOTELS"** which is being submitted for the partial fulfilment of the Degree of Bachelor of Technology, at NIIT University, Neemrana, is an authentic record of my/our original work under the guidance of **Prof.Manish Hurkat**. Due acknowledgements have been given in the project report to all other related work used. This has previously not formed the basis for the award of any degree, diploma, associate/fellowship or any other similar title or recognition in NIIT University or elsewhere.

Place: NIIT University, Neemrana, Rajasthan. 301705.

Date: 14th May, 2021.

1. Shrihari Shukla	**BT18GCS121**	**CSE**

NU

NIIT UNIVERSITY

THE UNIVERSITY OF THE FUTURE

Industry-linked | Technology-based | Research-driven | Seamless

CERTIFICATE BY SUPERVISOR(S)

This is to certify that the present R&D project entitled “**GUEST RECOGNITION FEATURE IN HOTELS”** being submitted to NIIT University, Neemrana, in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology, in the area of CSE, embodies faithful record of original research carried out by **Prof.Manish Hurkat**. They have worked under my guidance and supervision and that this work has not been submitted, in part or full, for any other degree or diploma of NIIT or any other University.

Place: NIIT University, Neemrana, Rajasthan. 301705.

Name of the Supervisor(s) with signature

Prof.Manish Hurkat

Date: 14th May, 2021.

ACKNOWLEDGEMENT

We express our sincere gratitude to Dean Research for providing us an opportunity to be part of this learning process and letting us work on a Research based Project on ‘Guest Recognition Feature’. We would also like to thank Prof. Manish Hurkat for mentoring us all along and guiding us in the right direction. His guidance and encouragement in carrying out this project work has made this possible.

We would also like to thank all the developers of already existing similar products like that of ours and all the authors of published research papers that we referred to while trying to have a better understanding of this project.

1. INTRODUCTION

The purpose of this report is to outline both the functional and non-functional requirements of the ‘Hotel Customer Recognition Application’. In addition to this, the document also provides a detailed profile of the external interfaces, performance parameters and design constraints imposed on the implementation. Our intention is to present the requirements having qualities like correctness, unambiguity, completeness, consistency, verifiability, modifiability and traceability. This document can act as a foundation for efficient and well-managed project completion and further serve as an accurate reference in the future for further developments.

The Hotel Guest Recognition system project is intended for the recognition of guests using beacon technology. Our Guest recognition system will have one end user who is working at that particular hotel (hotel staff). The main goal of this project is to simplify and automate the everyday process at the hotel by recognizing the VIP and VVIP persons from normal persons by placing beacons inside their keycards and greeting them when they come near hotel staff who’ll be having our device. This automation will be able to replace the existing way of greeting customers.

The intended readers of this document are those working on Hotel management systems like who can use this for cross reference to improve on their hospitality features, project developers who can use this for the tracking and verifying of the functions implemented, project panel who can use this to verify the quantity and quality of the end product. Thus this document can help to bridge up the gaps between the project stakeholders i.e. the designer, developer, system users and owners by giving them an idea what functionalities this application is capable of and what not thus giving future considerations to the project.

2. PROBLEM STATEMENT

In hospitality, the aim is tackling the guest opportunity in the best way possible. Every Guest Opportunity is an opportunity to win a loyal customer. In the field of hospitality, Identifying and resolving 'little' issues thus preventing the small glitch from snowballing into a big complaint is essential. Grabbing such opportunities and improving the hospitality is a tough task.

3. LITERATURE REVIEW

Towards the Omni-Channel: Beacon-Based Services in Retail by **Anja Thamm** et al. First section of the research paper[1], discusses the potential that the beacon technology possesses which could be exploited extensively to improve consumer experience. It suggests how the prevalence of smartphones offers an opportunity to connect the physical and digital world. The article describes how beacons when detected by smartphones or other devices in proximity indicate the presence of a user within the area of the beacon. There are different types of bluetooth beacons that are currently available. The next section of this article elaborates on how exactly a type of beacons known as the Bluetooth Low Energy Beacons work to precisely locate a user.
Under the second section of the article, ‘Fundamentals of Beacon-Based Interactions’, the author talks about Bluetooth Low Energy Beacons. This part mentions how these beacons are advantageous in terms of being inexpensive and could be deployed without additional infrastructure. In addition to this, the technical basis of Bluetooth Low Energy (BLE) allows for very low energy consumption, hence a coin battery can power it up to almost two years. As far as the basic working is concerned, the communication is unidirectional (beacons cannot receive data), providing a location to phones while the users remain anonymous to the beacon. An application can determine the distance to the beacon under three proximity levels depending upon the signal strength, “immediate” (a few centimeters), “near” (a few meters) and “far” (greater than 10 meters) or “unknown”, if the distance could not be determined. If more than three beacons are in range, the position of the receiver can be determined using techniques like trilateration or triangulation.

The underlying Bluetooth communication protocol remains standardized but the message formats supported by beacons vary by manufacturer. Smartphones and other Bluetooth-enabled devices can receive these messages, connect to the beacon and process received information using dedicated apps.

Three-dimensional positioning system using Bluetooth low-energy beacons by **Hyunwook Park** et al. To begin with, the author discusses the importance of developing technologies like 3-D posi-tioning, 3-D routing, and 3-D handover in the introduction section. This article exclusively focuses on the positioning system and defines the term as the technology that determines the location of a user.

In two-dimensional (2-D) networks, the most research has been performed in the 2-D positioning system that calculates 2-D location information in the form of (x,y) coordinates. Triangulation is the most suited tracking technique for 2-D positioning systems and mainly used for location estimation. Triangulation calculates the intersection of common points of one set of coordinates using the characteristics of a triangle.

Unlike previous 2-D networks, there is a need for 3-D location coordinates in the form of (x,y,z) in 3-D networks. Therefore, a 3-D positioning system is important in 3-D networks to overcome the limitation of a 2-D positioning system that only returns (x,y) coordinates. The author proposes a 3-D positioning system that can return (x,y,z) 3-D coordinates.

4. PROPOSED METHODOLOGY

4.1 GUEST RECOGNITION

This feature enables the hotel staff to recognise their customer and send a notification on the staff’s mobile or if they have a bluetooth earpiece connected, the name of the customer is pronounced in their ear when they are in the radius of 3 meter .

Proposed Methodology

Hotel Staff Recognises the guest when he is the radius of 3 meters through beacon and the mobile application of the staff. For this, a small bluetooth beacon is attached to the keycard given to the guest which helps in tracking the customer when he is in the radius of 3 meters.

When the customer is in the given range the mobile application of the staff detects the beacon and the unique id associated with the beacon to find the customer name and details.

4.2 WORKFLOW

4.2.1 Requirement Analysis

This was the most initial phase of the project where we analysed various situations and possibilities about our app idea and decided if it was practical and feasible to be weaved into an android app solution. We also established our purpose for building the app and the audience we wish to cater to our app. Based on this, the main functionality parameters of our app were decided and all the concept documents were collected for research and reference, to implement various functionalities as modules in our app.

4.2.2 Project Scope Finalization

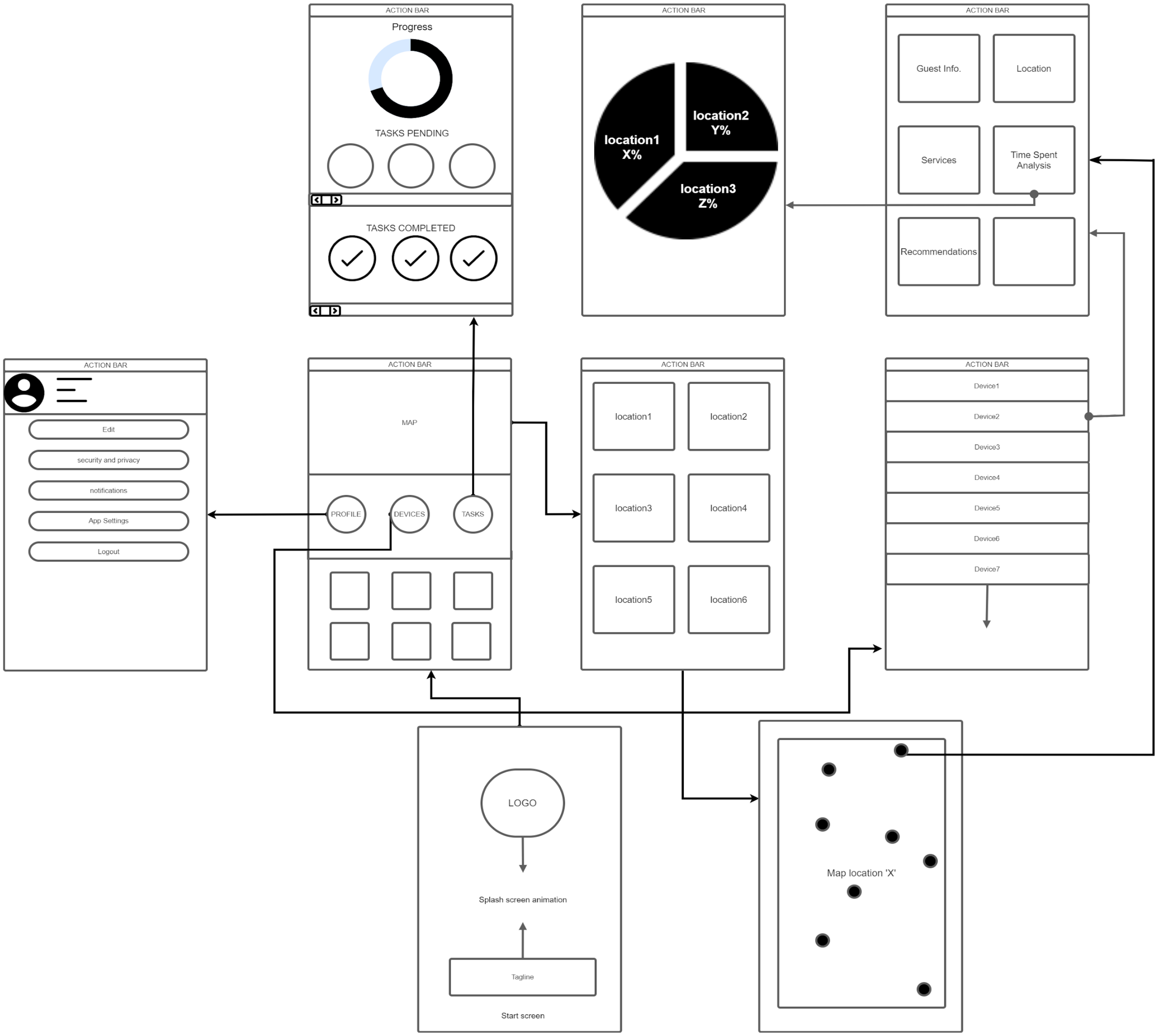
The project scope lists down the various functionalities of the app along with deciding the application feature list. Based on the total feature list, the project scope is finalised and the final project scope decides the technical and time requirements of the project.

For the scope of this project, we deduced that the Android Studio platform with dependencies maven, Firebase, Bluetooth, Dagger, butterknife and APIs for Eddystone and iBeacon would easily suffice to our requirements.

4.2.3 Wireframe Development

This is where the work first came in progress and we designed the wireframe of the app based on the project scope that was finalised. This wireframe decides the user journey. It includes all the basic functionalities that the user will get while he/she browses through the app, this will give a detailed idea of how the app will work and what would be the main functionalities of the app.

Fig4.1 Wireframe model of the application



4.2.4 Design Phase

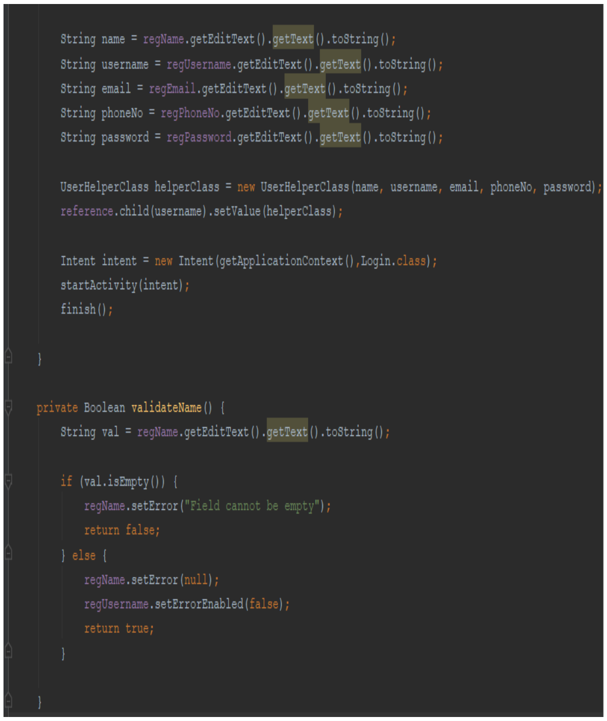
Referring to the wireframe design and considering the functionalities our application intends to offer, a user interface design that best suits our app was finalised. A suitable interface design is very essential to reach out to the intended audience and make the app look more engaging.

The process involved deciding the colour palette, the graphics, the animation, the pictures used, design elements, etc. Along with designing the user interface, user experience or UX was also designed in this phase. This included designing animations, motion graphics, the storytelling approach, the content placement, and the mouse-over elements that would make the app look more attractive.

4.2.5 Development Phase

This phase was carried out entirely on Android Studio (v3.5.1) platform. According to the wireframe design finalised and keeping the functional requirements in mind, the development process followed the following flow.

- **Registration:** For registering to this application, a new user has to “Sign up” and fill out the necessary details in the registration form. The data collected through this form for every user has been stored under unique references in the Firebase database (which is the primary and only database used in this project). The activity ‘SignUp.java’ and ‘activity_sign_up.xml’ deals with the logic and UI part of the registration feature respectively.



```
        String name = regName.getEditText().getText().toString();
        String username = regUsername.getEditText().getText().toString();
        String email = regEmail.getEditText().getText().toString();
        String phoneNo = regPhoneNo.getEditText().getText().toString();
        String password = regPassword.getEditText().getText().toString();

        UserHelperClass helperClass = new UserHelperClass(name, username, email, phoneNo, password);
        reference.child(username).setValue(helperClass);

        Intent intent = new Intent(getApplicationContext(),Login.class);
        startActivity(intent);
        finish();

    }

    private Boolean validateName() {
        String val = regName.getEditText().getText().toString();

        if (val.isEmpty()) {
            regName.setError("Field cannot be empty");
            return false;
        } else {
            regName.setError(null);
            regUsername.setErrorEnabled(false);
            return true;
        }

    }
```

Fig4.2 code snippet from SignUp.java

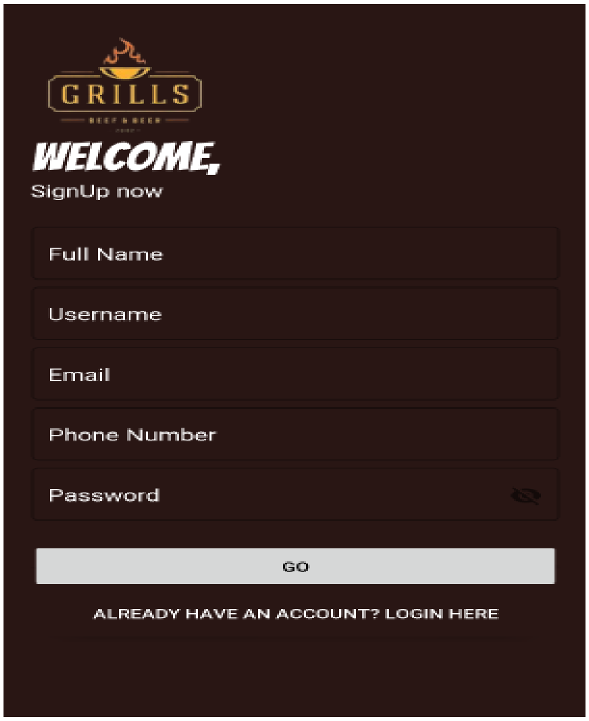
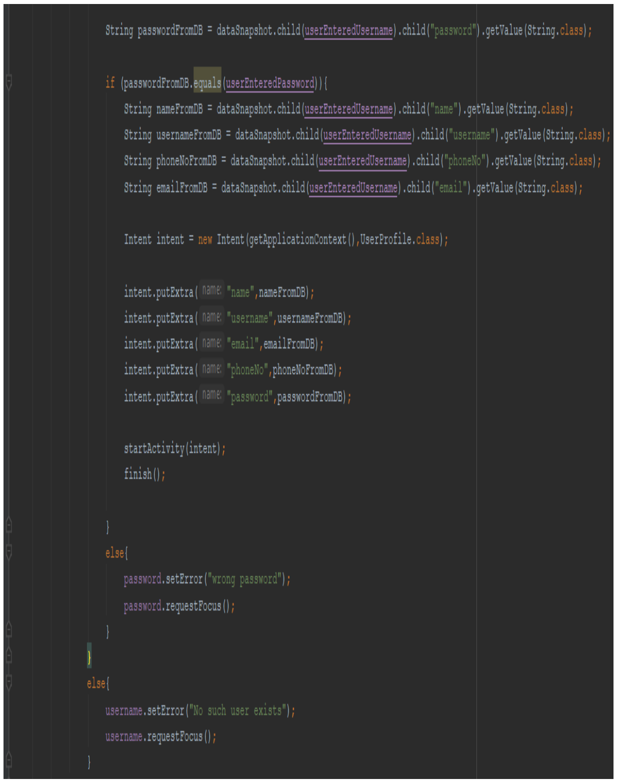


Fig4.3 Sign up interface

- **Login:** After signing up for this application, the user is directed to the login screen which ensures that only authorized users have access to the app. The user can login to the app using their credentials which they specified during the registration process. The activity responsible for handling the logic and ux part of the login feature are ‘Login.java’ and ‘activity_login.xml’ respectively.



```java
    String passwordFromDB = dataSnapshot.child(userEnteredUsername).child("password").getValue(String.class);

    if (passwordFromDB.equals(userEnteredPassword)){
        String nameFromDB = dataSnapshot.child(userEnteredUsername).child("name").getValue(String.class);
        String usernameFromDB = dataSnapshot.child(userEnteredUsername).child("username").getValue(String.class);
        String phoneNoFromDB = dataSnapshot.child(userEnteredUsername).child("phoneNo").getValue(String.class);
        String emailFromDB = dataSnapshot.child(userEnteredUsername).child("email").getValue(String.class);

        Intent intent = new Intent(getApplicationContext(),UserProfile.class);

        intent.putExtra( name: "name",nameFromDB);
        intent.putExtra( name: "username",usernameFromDB);
        intent.putExtra( name: "email",emailFromDB);
        intent.putExtra( name: "phoneNo",phoneNoFromDB);
        intent.putExtra( name: "password",passwordFromDB);

        startActivity(intent);
        finish();

    }
    else{
        password.setError("wrong password");
        password.requestFocus();
    }
}
else{
    username.setError("No such user exists");
    username.requestFocus();
}
```

Fig4.4 code snippet from Login.java

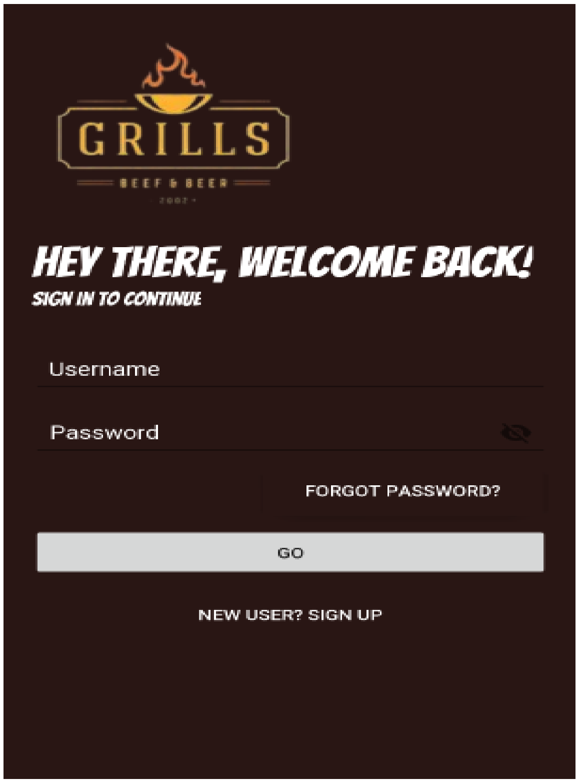
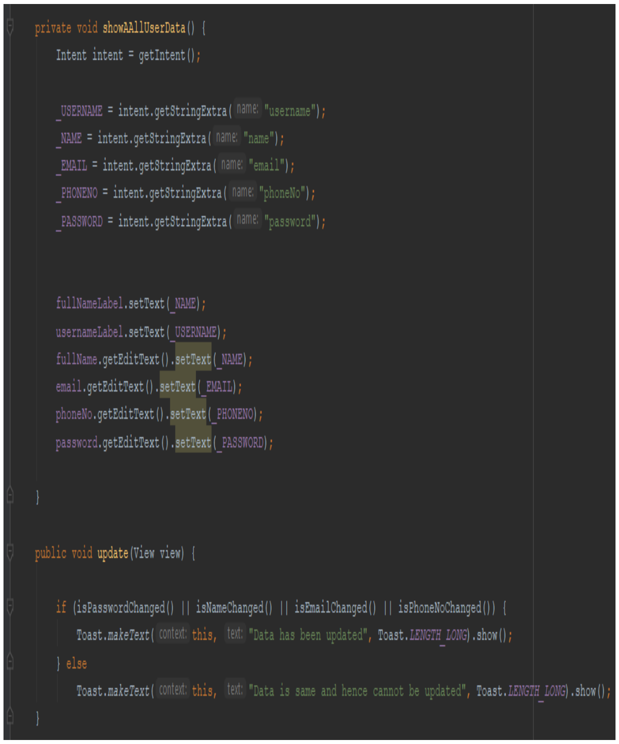


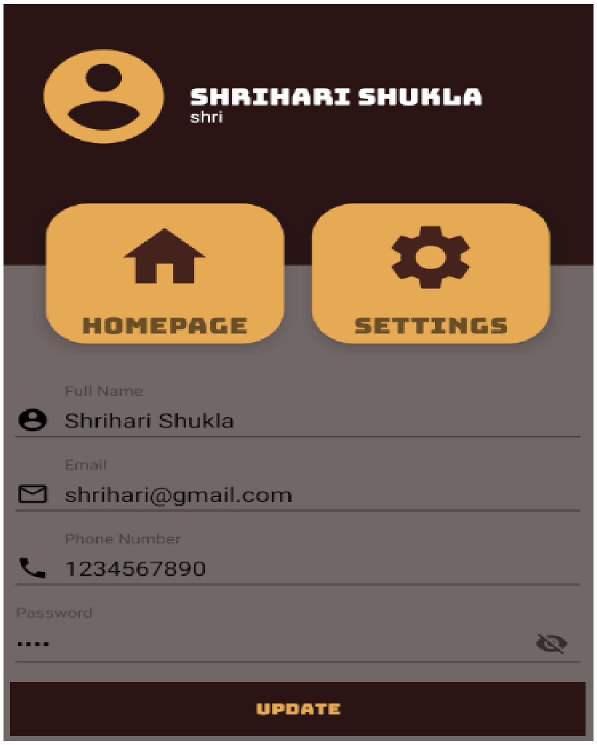
Fig4.5 Login Interface

- **User Profile:** On a successful login, the first screen a user sees would be his/her profile where they can edit and update their personal information concerned with the app. The ‘UserProfile.java’ and ‘activity_user_profile.xml’ activity handle the code and ux part of the user profile page. From here, the user has a choice to either go to the Homepage or to the App Settings page.



```java
private void showAAllUserData() {
    Intent intent = getIntent();

    _USERNAME = intent.getStringExtra( name: "username");
    _NAME = intent.getStringExtra( name: "name");
    _EMAIL = intent.getStringExtra( name: "email");
    _PHONENO = intent.getStringExtra( name: "phoneNo");
    _PASSWORD = intent.getStringExtra( name: "password");


    fullNameLabel.setText(_NAME);
    usernameLabel.setText(_USERNAME);
    fullName.getEditText().setText(_NAME);
    email.getEditText().setText(_EMAIL);
    phoneNo.getEditText().setText(_PHONENO);
    password.getEditText().setText(_PASSWORD);

}

public void update(View view) {

    if (isPasswordChanged() || isNameChanged() || isEmailChanged() || isPhoneNoChanged()) {
        Toast.makeText( context: this,  text: "Data has been updated", Toast.LENGTH_LONG).show();
    } else
        Toast.makeText( context: this,  text: "Data is same and hence cannot be updated", Toast.LENGTH_LONG).show();
}
```

Fig4.6 code snippet from UserProfile.java



SHRIHARI SHUKLA
shri

Full Name
Shrihari Shukla

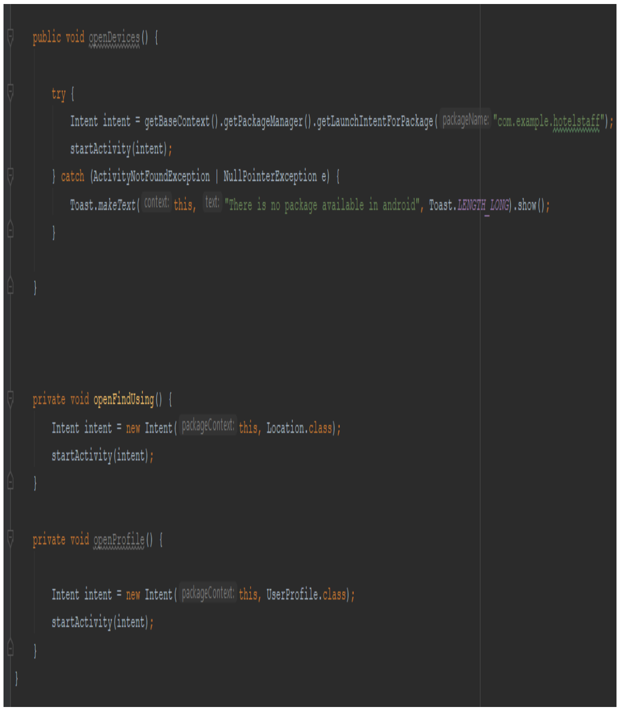
Email
shrihari@gmail.com

Phone Number
1234567890

Password
••••

Fig4.7 User Profile Interface

- **Homepage:** The Homepage is where the users are presented with the navigation bar to make use of different app functions like finding registered devices, retrieving guest information, tracking the customer, duties of the staff member etc. (some of these functionalities currently lie in the future scope of this project).



```java
public void openDevices() {


    try {
        Intent intent = getBaseContext().getPackageManager().getLaunchIntentForPackage( packageName: "com.example.hotelstaff");
        startActivity(intent);
    } catch (ActivityNotFoundException | NullPointerException e) {
        Toast.makeText( context: this,  text: "There is no package available in android", Toast.LENGTH_LONG).show();
    }

}



private void openFindUsing() {
    Intent intent = new Intent( packageContext: this, Location.class);
    startActivity(intent);
}

private void openProfile() {

    Intent intent = new Intent( packageContext: this, UserProfile.class);
    startActivity(intent);
}
}
```

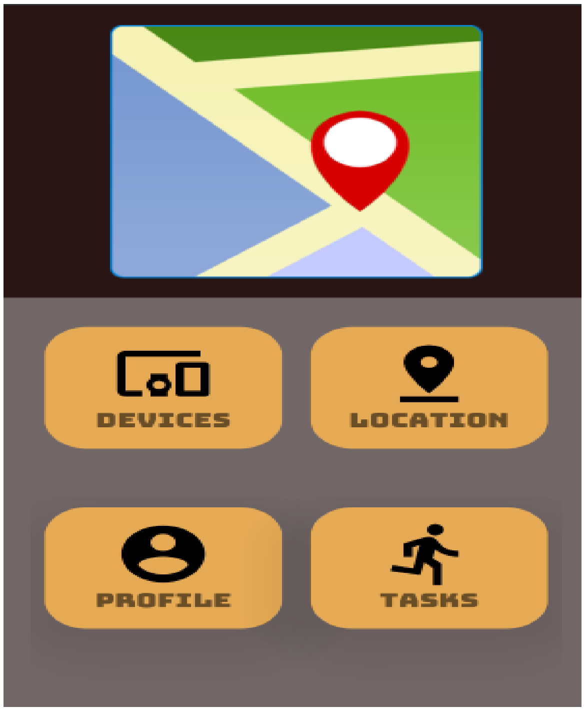
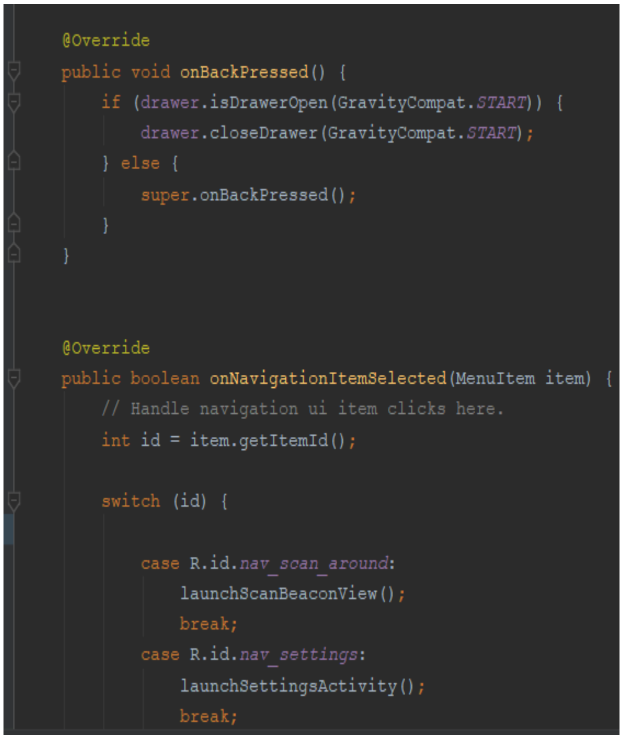


Fig4.8 code snippet from Homepage.java

Fig4.9 Homepage Interface

- **Devices**: This section of the app can be accessed from the Homepage and acts as a gateway to a menu which holds sections for every operation that is to be performed on the Bluetooth low energy devices (which are either currently active or were in the past) in the proximity.



```
@Override
public void onBackPressed() {
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
}


@Override
public boolean onNavigationItemSelected(MenuItem item) {
    // Handle navigation ui item clicks here.
    int id = item.getItemId();

    switch (id) {

        case R.id.nav_scan_around:
            launchScanBeaconView();
            break;
        case R.id.nav_settings:
            launchSettingsActivity();
            break;
```

Fig4.10 code snippet from MainNavigationActivity.java

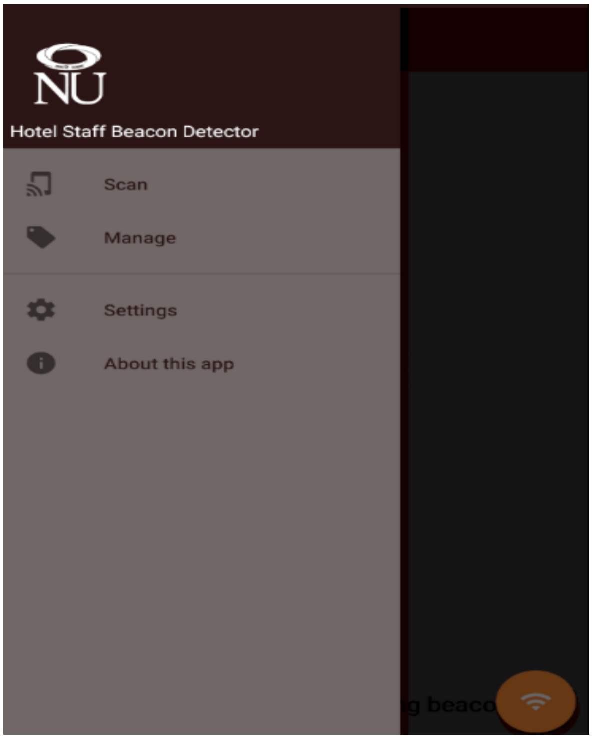


Fig4.11 Devices section navigation Interface

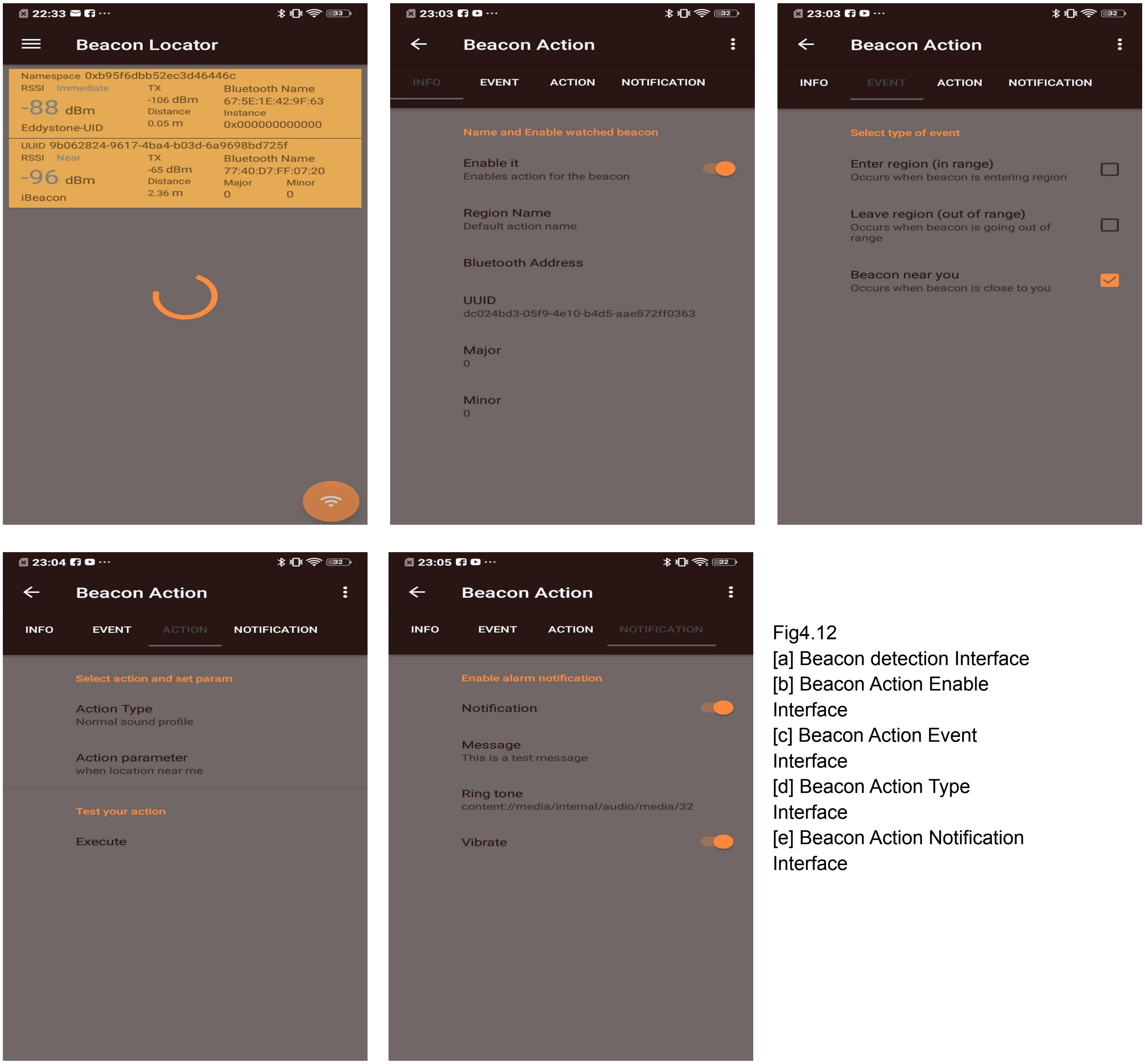
  - **Scan**: This feature is responsible for launching the scan for any available Bluetooth low energy beacons in the proximity. For any tracked BLE device in the proximity (which is still active), the app is capable of identifying its UUID, RSSI, Major, Minor, Beacon format, Bluetooth id and its distance from the scanner.

  - **Manage** : Under the devices section the user can also choose to look at the previously detected beacons which are not active currently using the

‘Manage’ functionality available in the menu bar on the ‘Devices’ page. The user can also see when the particular saved beacon was last active.

○ **Settings** : The menu bar also offers the ‘Settings’ option where the user can choose to decide if the scanning of beacons nearby should be enabled even when the app is closed (i.e. in background) or not.

○ **Action**: For any detected beacon, either active currently or in the past, the user can specify a set of triggers from the given options which takes place on the occurrence of defined events by the user under the ‘Action’ functionality.



22:33

Beacon Locator

Namespace 0xb95f6dbb52ec3d46446c
RSSI Immediate
-88 dBm
Eddystone-UID
TX -106 dBm
Distance 0.05 m
Bluetooth Name 67:5E:1E:42:9F:63
Instance 0x000000000000

UUID 9b062824-9617-4ba4-b03d-6a9698bd725f
RSSI Near
-96 dBm
iBeacon
TX -65 dBm
Distance 2.36 m
Bluetooth Name 77:40:D7:FF:07:20
Major 0
Minor 0

23:03

Beacon Action

INFO EVENT ACTION NOTIFICATION

Name and Enable watched beacon

Enable it
Enables action for the beacon

Region Name
Default action name

Bluetooth Address

UUID
dc024bd3-05f9-4e10-b4d5-aae872ff0363

Major
0

Minor
0

23:03

Beacon Action

INFO EVENT ACTION NOTIFICATION

Select type of event

Enter region (in range) □
Occurs when beacon is entering region

Leave region (out of range) □
Occurs when beacon is going out of range

Beacon near you ☑
Occurs when beacon is close to you

23:04

Beacon Action

INFO EVENT ACTION NOTIFICATION

Select action and set param

Action Type
Normal sound profile

Action parameter
when location near me

Test your action

Execute

23:05

Beacon Action

INFO EVENT ACTION NOTIFICATION

Enable alarm notification

Notification

Message
This is a test message

Ring tone
content://media/internal/audio/media/32

Vibrate

Fig4.12
[a] Beacon detection Interface
[b] Beacon Action Enable Interface
[c] Beacon Action Event Interface
[d] Beacon Action Type Interface
[e] Beacon Action Notification Interface

4.3 TECHNOLOGY USED

4.3.1 Development Platform

Java, an object-oriented programming language, is an official Android development language. Since it is a language that is easy to handle and we had prior experience working with java on Android Studio platform so we chose to go with it. It has open-source libraries that are readily available for users to choose from. Java is a technology that offers the best documentation and community support. Also, For the scope of this project, we deduced that the Android Studio platform with dependencies like maven, Firebase, Bluetooth, Dagger, butterknife and APIs for Eddystone and iBeacon would easily suffice to our requirements.

The following project structure shows the primary modules which hold various components that control the organisation of different aspects of the application.

The Android Project View has the default ‘app’ module which consists of components:

app/build : which contains build outputs.

app/libs : which contains libraries.

app/build.gradle(module specific) : for defining module-specific build configurations.

app/src : which contains all code and resource files for the module in the following subdirectories:

app/src/androidTest : which contains code for instrumentation tests that run on an Android device.

app/src/main : Contains the "main" sourceset files: the Android code and resources shared by all build variants.

app/src/main/java: consists of java source code files.

apps/src/main/AndroidManifest.xml: To describe the nature of the application and each of its components. It validates all java activities present in the project and specifies the launching activity as well.

apps/src/main/res: Contains application resources, such as drawable files, layout files, and UI string.

apps/src/main/test: Contains code for local tests that run on your host JVM.

Another module ‘gradle’ comprises of:
gradle/build.gradle(project level) : for defining build configurations that apply to all modules. This file is integral to the project.

4.3.2 Bluetooth Low Energy

Bluetooth Low Energy (BLE) is a low power wireless communication technology that can be used over a short distance to enable smart devices to communicate. Some of the devices we interact with every day such as our smartphones, smart watch, fitness tracker, wireless headphones and computer use BLE to create a seamless experience between devices. BLE has a wide range of possibilities and is implemented in a broad set of fields such as health, fitness, security, home automation, home entertainment, smart industry, Hospitality and IoT (Internet of Things).

BLE is a relatively new Bluetooth standard defined by the Bluetooth Special Interest Group (SIG) with a focus on low energy. It has enabled device manufacturers to add a low power communications interface on existing solutions. It has also been used to create new low power devices such as beacons that can be powered by a small coin cell battery for months or even years.

Beacons are small, wireless transmitters that use low-energy Bluetooth technology to send signals to other smart devices nearby. They connect and transmit information to smart devices making location-based searching and interaction easier and more accurate.

Each device contains a CPU, radio, and batteries, and it works by repeatedly broadcasting out an identifier. This identifier is picked up by a smart device, usually a mobile, and marks out an important place in that environment.

The identifier is a unique ID number that the smartphone recognizes as unique to the beacon. Once connected, the beacon will carry out whatever function it has been programmed to perform.

Google entered the beacon world when it launched Eddystone in 2015, two years after Apple introduced iBeacon. We assessed the differences between the two beacon protocols to see what they offered.

- **Compatibility** : both are compatible with iOS and Android, but Eddystone is also compatible with any platform which supports BLE.
- **Privacy and Security** : The signal transmitted by a beacon is a public signal and can be detected by any, while Eddystone has a built-in feature called EIDs, which allow the signal to be identified by only authorized users.
- **Broadcast packets** : iBeacon broadcasts only one packet which contains all of the important information, while Eddystone broadcasts three different packets. Unique ID number packet, URL packet, and sensor telemetry packet.
- **Implementation** : iBeacon is easy to use. Eddystone requires more complicated integration because it sends more packets.
- **Open-source** : the specification of iBeacon is controlled by Apple, Eddystone is open-source and its specification is published on Github.

For the scope of this project, we have designed the application to be able to detect both these formats (Eddystone and iBeacon) of beacons.

4.3.3 Operating Environment

Software Requirements

Operating system	**Android 4.2, Android 4.4.2, or Android 4.4.4 or higher.**

Hardware Requirements

Processor	Intel Atom® **Processor** Z2520 1.2 GHz, or faster **processor.**
Storage	Between 850 MB and 1.2 GB, depending on the language version.
RAM	**Minimum** of 512 MB, 2 GB is recommended.

5. RESULT AND ANALYSIS:

5.1 RESULT

Beacon Detection

The Primary functionality of our application we aimed to achieve was to detect the VIP and VVIP customers using the bluetooth low energy emitted from the bluetooth device (beacon) attached to their key card. Our application is capable of calculating the distance of the guest from the Hotel staff member (i.e distance between bluetooth signal emitter and the application) with the help of RSSI(Received Signal Strength Indicator) value and notify the staff member with the details of the customer approaching him/her so that the staff member can greet the customer (or perform any other expected service) with the notified details. The application is also capable of deducing other side details like the beacon format and the time it was last active.

Beacon Simulation

Since the procurement of real beacons could not be made possible due to reasonable circumstances, we decided to have beacon simulation technique as an alternative. We tried out some existing applications which catered to this need of ours. The essence of most of these was more or less along the same line as far as the process of transmission of signal is concerned. These apps ask you to create your virtual beacon specifying parameters like ‘beacon format’, ‘transmission power’ etc. and then activate it for a beacon scanner to be able to detect it. But the result obtained as far as accuracy is concerned for parameters like ‘distance of beacon from the scanner’ were mostly different for all these beacon simulation applications. We made a detailed comparative study between these applications and found out error margins by considering differences between expected and achieved values and chose the application with minimum error margin.

	Beacon Scope	Locate Beacon	Beacon Simulator
Range(max)	25	25	25
Error margin	+-1.5m	+-1m	+-0.4m
No of beacons	Depends on the mobile	Depends on the mobile	Depends on the mobile
Transmission power	Adjustable	Adjustable	Non-Adjustable
Beacon Format	Adjustable	Adjustable	Adjustable
Calibration	Not allowed	allowed	Not allowed
Compatible devices	Android	Android	Android

Table-1 : Comparison of different beacon simulators

With the help of this comparison we concluded to use ‘Beacon Simulator’ as the beacon simulator to transmit the beacon signal as it is having less error margin when compared to others.

5.2 ANALYSIS

The scanned RSSI values of the beacon that is being transmitted through the beacon simulator appear to fluctuate for a while before getting stable because of the noise interference from the surrounding conditions. This in turn decreases the accuracy of the distance calculated. The accuracy of the distance is dependent on factors like transmission power and obstacles in between the beacon simulator and the scanner. So, it takes time for the scanner to calculate the distance and obtain a stable value. These values and their accuracy might vary when using an actual beacon instead of a simulator. For now we were bound to use a simulator as procurement of real beacons could not be made possible due to reasonable circumstances .

For this exact reason of not using real beacons but simulators, the notification feature is also currently not functioning the way it is intended to.

Another important point of observation we recorded was that the application currently is competent in detecting limited formats of beacons only and does not identify the format correctly in other cases like that of an AltBeacon.

6. CONCLUSION AND FUTURE SCOPE:

6.1 CONCLUSION

We have developed an Android Application which connects the hotel management industry to the IOT technology using beacons. The application uniquely identifies any beacon present in the vicinity and can calculate the distance to it as well. Moving forward as developers, if we collaborate with any hotel fascinated with the idea of providing better customer experience to their guests through the beacon technology, then it will be easy to just perform a unique matching for every guest name in the hotel's database with a unique beacon id. This will enable the Hotel staff to deliver quality services to their customers and ensure that the guest has a nice time staying at the hotel. To take hospitality to the next level and to make this happen in the real world we definitely need mobile applications that connect IOT and Hotel Management.

6.2 SHORTCOMINGS

1. The application does not accurately calculate the distance of the detected beacon (and hence the customer eventually) and has some significant error margin.
2. The notification feature which is meant to get active on the occurrence of a specified event concerned with the movement of beacon does not work as intended currently since we are actually using a beacon simulator rather than an actual beacon device. Shifting the project to using an actual beacon would resolve this issue.
3. As of now we're able to identify only limited registered formats of the beacon through our application.

6.3 FUTURE SCOPE

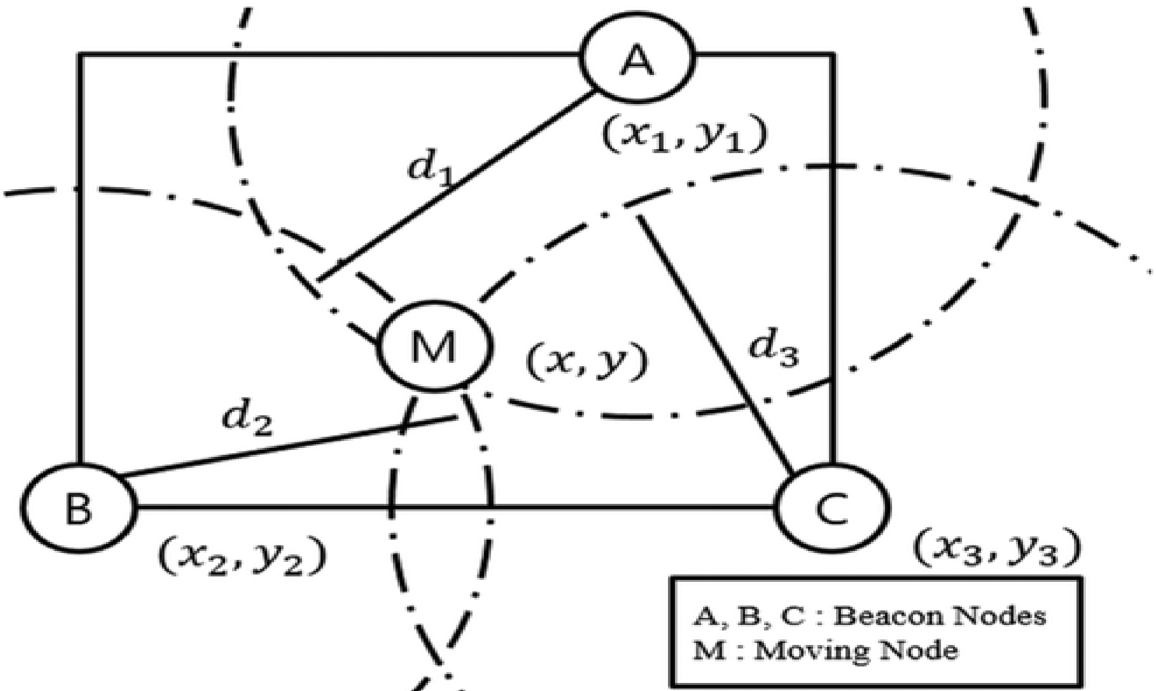
6.3.1 Location Tracking

In future, we plan to extend our application to precisely point out the location of a particular customer inside the hotel premises with the help of the UUID (Universally Unique Identifier) belonging to their beacon and using the triangulation method.

2-Dimensional triangulation

2-D triangulation method uses relative location information to find out the location of a desired user. 2-D triangulation is an important technology in 2-D positioning systems as it calculates the location based on (x, y) coordinates which are the key factor in 2-D positioning systems. To mark a location using 2-D triangulation requires three beacon nodes along with a desired node (this is the node that we want to pinpoint) in motion. This method finds the point of intersection of these three circles, whose radii are the distance among a desired node (M), and three beacon nodes, as shown in the figure below. It is a classic method to estimate the real-time location of an object to move on a 2-D plane. In 2-D triangulation, the distance between a desired node and a target point is calculated with the help of Pythagorean theorem. The following equation derives the distance that is calculated using the Pythagorean theorem, that will then be applied to each of three circles to find the distances.

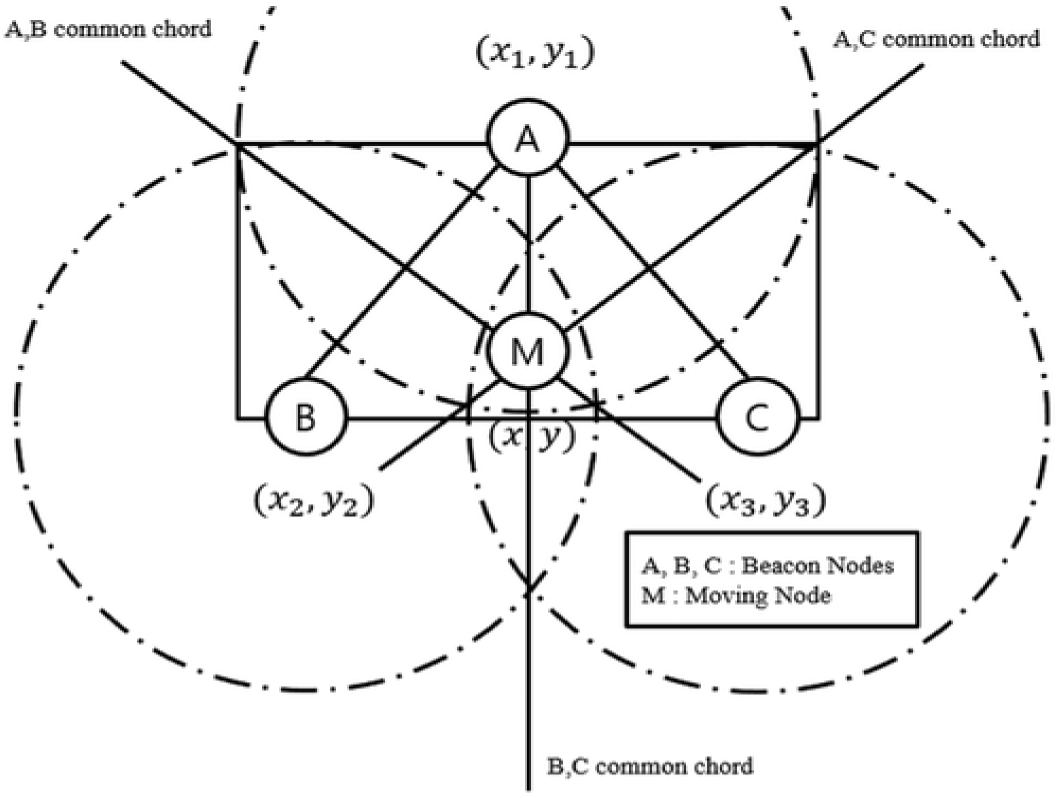
$$(x-x_n)^2 + (y-y_n)^2 = d_n^2$$



However, there are clear constraints in this strategy. It must be applied to the situation where a crossing point for every one of the three circles exists. On the other side, if the zone covered by three circles is excessively huge, it is hard to pinpoint the specific area of a moving hub.

Triangulation using points crossed by common chords

This method describes a triangulation method using the points crossed by common chords as shown in the figure below. It improves(i.e decreases) the distance error margin when compared to the 2-D triangulation method. In the conventional triangulation method, it is hard to estimate the exact user location when beacon nodes and a desired node(node that we want to mark out) are not located on the same plane with equal height. The conventional triangulation method cannot generate a standard crossing point for three circles if those are not located on an equivalent height. To sort out this problem, the enhanced method suggests a triangulation to use chords that is common to the circles.



When two circles, which are given by $x^2 + y^2 + ax + by + c = 0$, $x^2 + y^2 + a'x + a'y + c' = 0$ meet at two points, an equation for a common chord is derived as $(a-a')x + (b-b')y + (c-c') = 0$. Using the derived equation we can find equations for three chords, which are common to the circles A and B, A and C, and B and C, respectively. Through the derived equations for three common chords, a position value for their crossing point can finally be calculated as (x, y) coordinate of a target location.

However, even the enhanced method considers the difference in height, it still yields a 2-D coordinated position. It can derive the 3-D location but the derived result produces a significant estimation error margin.

6.3.2 Time Spent Analysis

We expect to eventually integrate further side functionalities to our application as well which involve ‘Determining a guest’s interests’ by analysing the amount of time spent at various locations inside the hotel. The time spent by the guest at any location in the hotel is recorded using the location tracking feature and eventually a dataset is generated which suggests the percentage time the guest spent at specific locations in the hotel. Using a sort algorithm the items (which essentially are percentages) of this dataset can now be arranged in a descending order so as to understand and analyse the priorities (in terms of interests) for the guest.

6.3.3 Recommendation System

Using this information we can develop a recommendation system for the application which provides services related to the guest’s priorities thus delivering what the guest desires without even him/her asking for it. The recommendation system would follow the approach of collaborative methods which involves the user to rate the services provided to them following which the system learns from these ratings and improves. The system might however also follow a content based approach if the need arises.

Apart from this, we also plan to have a separate section for the user (which is a hotel staff member) where he or she will be able to check their ‘completed’ and ‘left’ tasks for the day.

7. REFERENCES:

[1] Thamm, Anja & Anke, Jürgen & Haugk, Sebastian & Radic, Dubravko. (2016). Towards the Omni-Channel: Beacon-Based Services in Retail. 255. 181-192. 10.1007/978-3-319-39426-8_15.

[2] Park, Hyunwook & Noh, Jaewon & Cho, Sunghyun. (2016). Three-dimensional positioning system using Bluetooth low-energy beacons. International Journal of Distributed Sensor Networks. 12. 10.1177/1550147716671720.

8. SOURCE CODE

https://github.com/shrreeeehari/guestRecognition