INDOOR NAVIGATION SYSTEM

A PROJECT REPORT

Submitted by

ARSHIYA FAMINA (311117205012)
MARIA JOVITA VAZ (311117205036)
RAKSHINI (311117205046)

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

LOYOLA-ICAM

COLLEGE OF ENGINEERING AND TECHNOLOGY

CHENNAI-600034

ANNA UNIVERSITY: CHENNAI - 600025

APRIL 2020

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project **"INDOOR NAVIGATION SYSTEM"** is the bonafide work of **ARSHIYA FAMINA(311117205012), MARIA JOVITA VAZ(311117205036), RAKSHINI(311117205046)** who carried out the project work under my supervision.

SIGNATURE

Dr. R JULIANA, M.E, Ph.D
HEAD OF THE DEPARTMENT
Professor
Information Technology
Loyola-ICAM college of engineering and technology Loyola campus, Nungambakkam, Chennai-34

SIGNATURE

MS. B PRABHA M.E
SUPERVISOR
Assistant professor
Information Technology
Loyola-ICAM college of engineering and technology Loyola campus, Nungambakkam, Chennai-34

Submitted for the Project Viva voce examination held on______________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First of all, we are grateful to God for granting us this opportunity and capability to proceed successfully. Working on this project has been a rewarding experience.

We would like to express our sincere thanks and gratitude to our Director, **Rev. Dr Maria Wenisch SJ**, the Dean of Students **Rev. Dr Justine SJ**, and the Dean of Engineering Education, **Mr. Nicholas Juhel** for their constant support and encouragement; We thank them for making the right resources available at the right time.

We would like to extend my gratitude to our Principal, **Dr L Antony Michael Raj, M.E., Ph.D.,** for his motivating, constructive criticism and valuable guidance during the course of the project. We would like to express my sincere thanks to the Head of the Department **Dr R Juliana, B.E., M.E., Ph.D.**, for her comments and suggestions.

We also would like to thank the project coordinator **Dr. A Janani, M.S., Ph.D.**, the faculty and technical staff of our department for their critical advice and guidance which was helpful in completion of the project.

We extend our sincere gratitude to my project guide **Ms. B PRABHA, M.E.,** for providing valuable insights and resources leading to the successful completion of our project. We would also like to thank the faculty members of our department for their critical advice and guidance which was indispensable for our work. Last but not the least, We place a deep sense of gratitude to our friends and families who have been a constant source of inspiration during the preparation of this project.

ABSTRACT

Navigation entails the continuous tracking of the user's position and his surroundings for the purpose of dynamically planning and following a route to the user's intended destination. The Global Positioning System (GPS) made the task of navigating outdoors relatively straightforward, but due to the lack of signal reception inside buildings, navigating indoors has become a very challenging task. However, increasing smartphone capabilities have now given rise to a variety of new techniques that can be harnessed to solve this problem of indoor navigation. In this report, we propose a navigation system for smartphones capable of guiding users accurately to their destinations in an unfamiliar indoor environment providing turn-by-turn directions, without requiring any expensive alterations to the infrastructure or any prior knowledge of the site's layout. The system uses the floor map of the building and it directs the user with help of a route. Indoor navigation relies on different technologies like wifi, geomagnetic signal data, gyroscope and accelerometer (IMU sensors) for deploying indoor navigation. Using a mobile app, students, teachers and visitors can start a guided tour from their current location. Lecture halls, offices and rooms as well as other POIs (e.g. libraries) can be selected as navigation destinations.

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. POI – POINT OF INTEREST
2. IPS – INDOOR POSITIONING SYSTEM
3. GPS – GLOBAL POSITIONING SYSTEM
4. SDK – SOFTWARE DEVELOPMENT KIT
5. API – APPLICATION PROGRAM INTERFACE

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION

1.1 INTRODUCTION TO THE DOMAIN

Navigation is the process of accurately establishing the user's position and then displaying directions to guide them through feasible paths to their desired destination. The Global Positioning System (GPS) is the most common and the most utilised satellite navigation system. Almost every aircraft and ship in the world employs some form of GPS technology. In the past few years, smartphones have evolved to contain a GPS unit, and this has given rise to location-based mobile applications such as geofencing and automotive navigation for the common user. However, GPS has its limitations. In particular we are concerned with the lack of GPS signal reception in indoor environments. GPS satellites fail to deliver a signal to a device if there is a direct obstruction on its path. Therefore we have to consider alternate methods of achieving indoor navigation on a smartphone.

1.2 OVERVIEW OF THE PROJECT

A Smartphone collects radio signals, geomagnetic fields, inertial sensor data, barometric pressure, camera data and other sensory information to provide navigation inside a building. This application makes use of Indoor Positioning System which aims at navigating and tracking objects inside the building using the IndoorAtlas SDK, which works on the theory of Hybrid Indoor Positioning Technology. Google Maps plays a very crucial role in this project. Its Android API helps us display maps in the screen. We are able to align the floor plan to the exact coordinates of the building and is fetched from cloud and overlayed on the google maps.

1.3 PURPOSE OF THE PROJECT

In most of the Universities, there are many buildings with various departments, if someone are new to the university there is a possibility that they may be lost easily inside the vast campus. Not only in universities but people in most of the shopping malls face the same problem, they are unaware about the locations of the shops. In Universities, the people who come for cultural, functions or any other purpose for the first time and the students in the beginning of each scholar year have problem in locating the rooms. To avoid this situation, the indoor navigation system can be developed and can used as a mobile application. This will help the students to locate their classes and the people who are new to the campus can use this to find the places where they have to reach.

1.4 PROJECT PLAN

	18/12-1/1	1/1-8/1	8/1-15/1	15/1-22/1	22/1-2/2	2/2-12/2	12/2-27/2	27/2-4/3	4/3-12/3	12/3-14/3	14/3-15/3
Need Analysis	7										
Functional Analysis		7									
Specification			7								
Selection of Technology				7							
Venue Mapping					11						
Calibration of positioning						10					
Coding							15				
Testing								6			
Further Enhancement									8		
Evaluation										2	
Report											1

0^{th} Review 1^{st} Review 2^{nd} Review

Figure 1 – Gantt chart

The project planning stage includes gathering the floor plan of the building and geolocating it to the world map, adding waypoints and collecting the signal data to track the user's current location and walkable paths of the building and add nodes to each rooms which are connected by edges to display shortest route to the user. An Android code is developed to fetch the above information and to provide turn-by-turn directions to the user.

1.5 SCOPE OF THE PROJECT

Maps which uses GPS for navigating in outdoor cannot be used for indoors as GPS satellites fail to deliver a signal to a device if there is a direct obstruction on its path. Therefore we have to consider alternate methods of achieving indoor navigation on a smartphone. The indoor navigation system makes use of sensor signal data like geomagnetic signals, gyroscope and accelerometer(IMU sensor) and other data for tracking the user's current location and navigating them to their destinations. This is useful for people to find their destination inside an unfamilier building quickly which saves people's time and energy.

1.6 SUMMARY

The introductory part of the report provides the overview, purpose, and scope of the project. This application can be used by the visitors, students and staffs who are unfamilier with routes of the building.

CHAPTER 2

2.1 LITERATURE SURVEY

With the rapid development of the mobile Internet, data and multimedia services are increased rapidly. The demand of positioning and navigation is increasing day by day. Especially in complex indoor environments, the indoor position of the mobile terminal i.e. its owner, facilities and goods, is generally needed to be determined in real-time. These complex indoor environments include airport halls, exhibition halls, Universities, warehouses, supermarkets, libraries, underground car parks and other environments. The indoor positioning system is to locate and track a person within buildings or closed environment, and these systems are based on the principle of radio waves, optical tracking, magnetic field or ultrasonic technology etc. The technology of positioning and tracking the object is the basis for many applications on monitoring and activity recognition. For example, in the supermarket, through obtaining the relevant position information of the consumers and target commodities, the supermarket is able to provide the services of the route guidance and the intelligent shopping guide.

The indoor positioning system can be specially employed for guiding the rescuers to search the building and rescue the trapped personnel quickly when a sudden disaster happens. In addition, the position technology can be used in hospital, i.e. the patient’s monitoring and the management of the medical equipment. So, it is necessary and important to apply the indoor positioning technology and system into people’s lives. It is well known that global positioning system (GPS) is the most widely used satellite-based positioning system and it provides location-based services, such as navigation, tourism and so on. However, GPS is not suitable in indoor positioning due to a variety of obstacles. For solving the indoor positioning problems, many solution technologies are proposed in literatures. In this technology, inertial navigation system (INS) [9], magnetic field positioning technology, etc.

2.2 SUMMARY

This application is specifically designed for Loyola ICAM College of Engineering and Technology(LICET) to provide a turn-by-turn directions for visitors, students and staffs who are unfamiliar with the building.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 PROBLEM STATEMENT

The problem is we have application like Google Maps for navigating outdoors which uses GPS technology but it cannot be used for navigating indoors. Places like Universities, shopping malls in which people may get lost easily inside the vast campus therefore indoor navigation system is required to save people's energy and time and direct them to their destination.

3.2 NEED ANALYSIS

3.2.1 BULL DIAGRAM

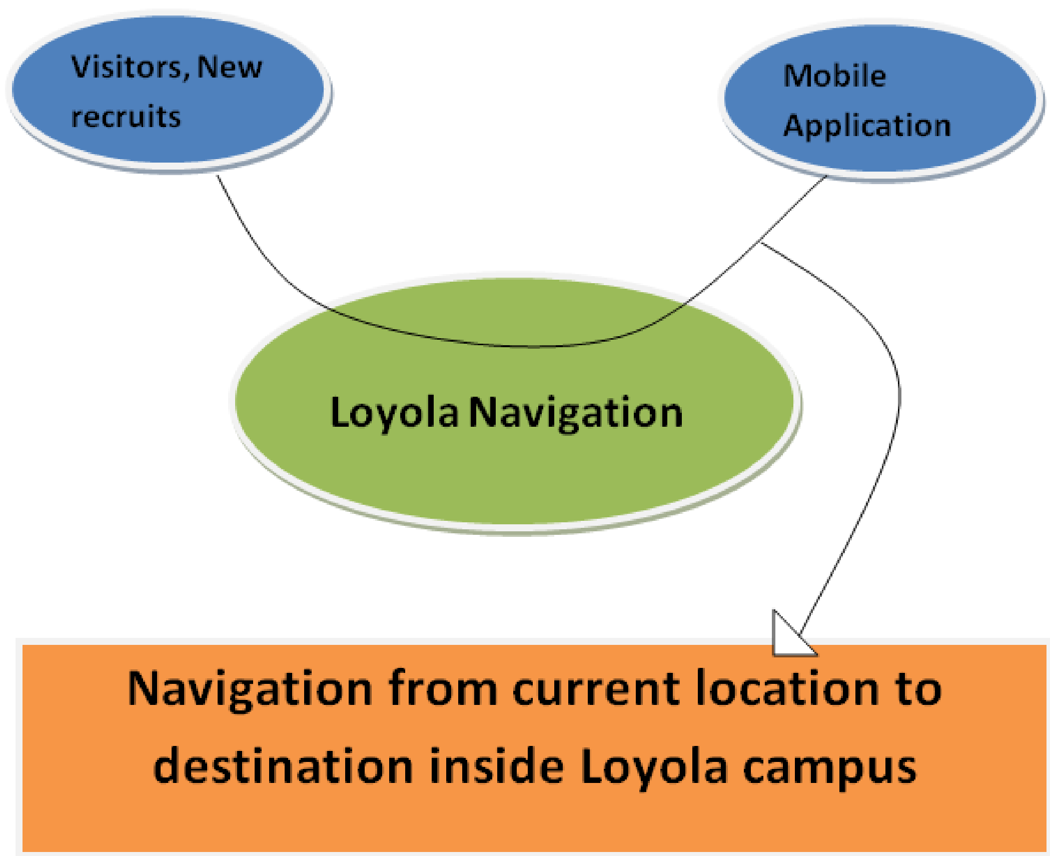


*Figure 1 – **Bull Diagram***

3.3 FUNCTIONAL ANALYSIS

3.3.1 OCTOPUS DIAGRAM

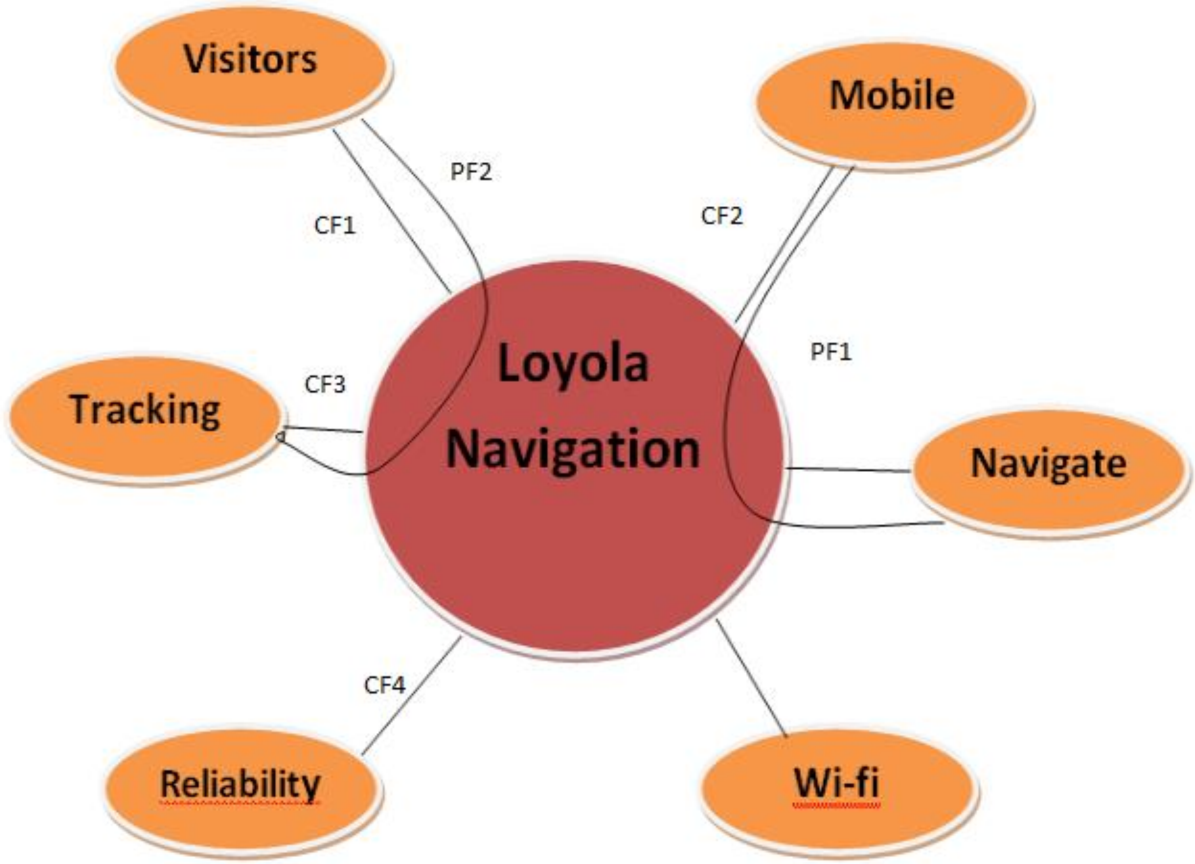


*Figure 2 – **Octopus diagram***

Principal Functions

PF1: Navigate from current location to destination

PF2: Track current location

Constraint Functions

CF1: To be user friendly

CF2: To display path

CF3: To track user's current location

CF4: To avoid failure

3.4 DATA FLOW DIAGRAM

3.4.1 DFD-0

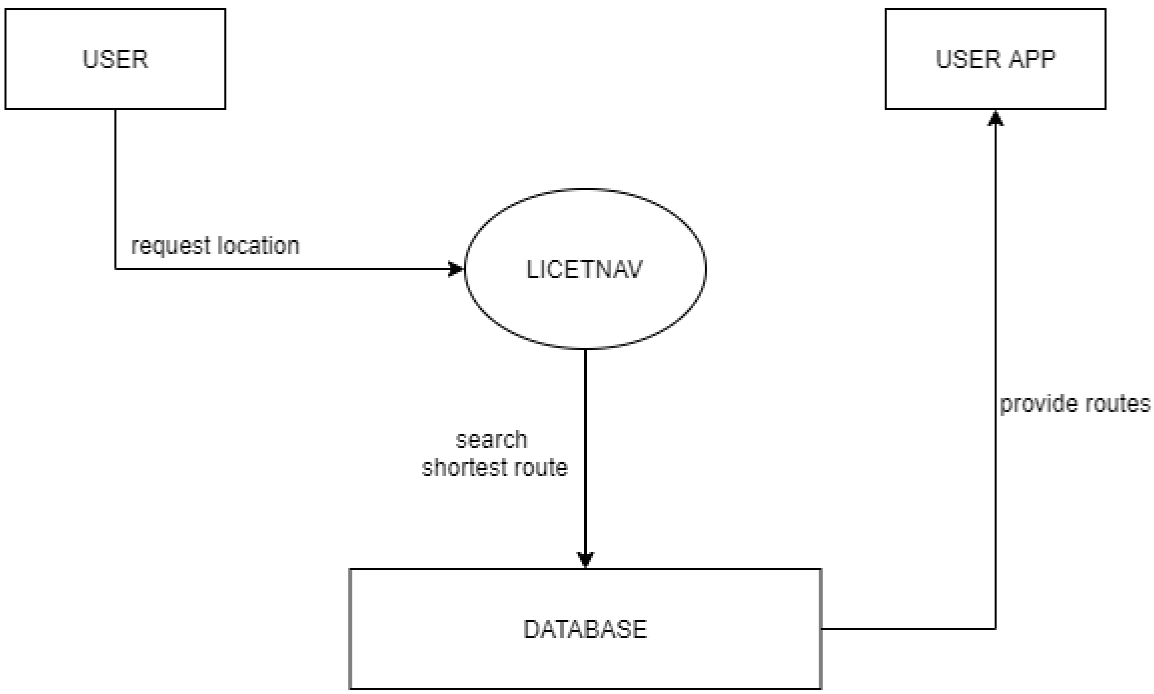


Figure 3

DFD 0 is designed to be an abstraction view, showing the system as a single process with its relationship to external entities.

3.4.2 DFD-1

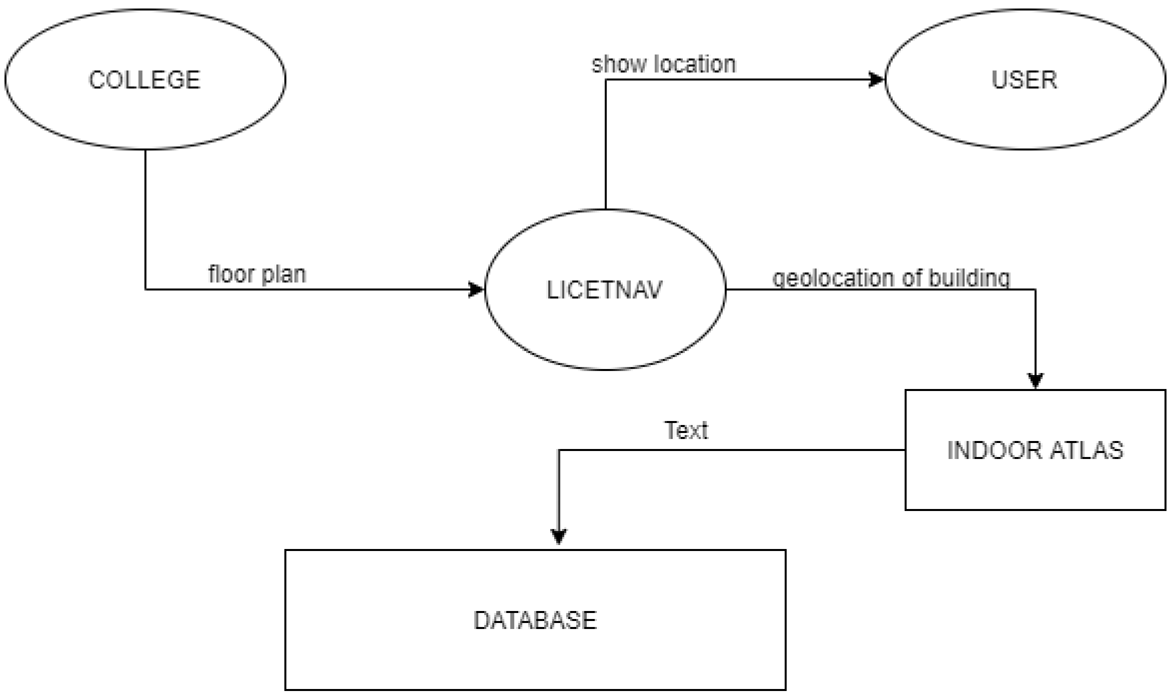


Figure 4

DFD 1 is used to highlight the main functions of the system and breakdown the high level process of 0-level DFD into subprocesses.

3.4.3 DFD-2

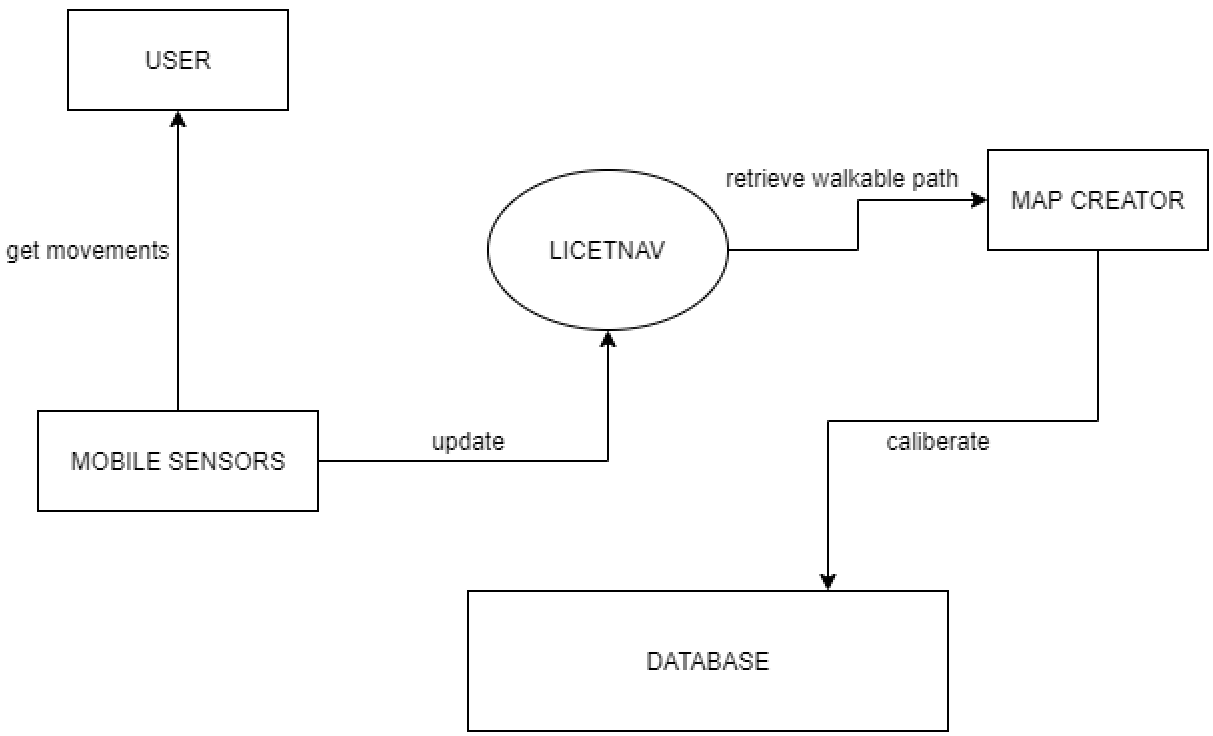


Figure 5

DFD 2 is used to plan or record the specific/necessary detail about the system's functioning.

3.5 UML DIAGRAM

3.5.1 USE CASE DIAGRAM

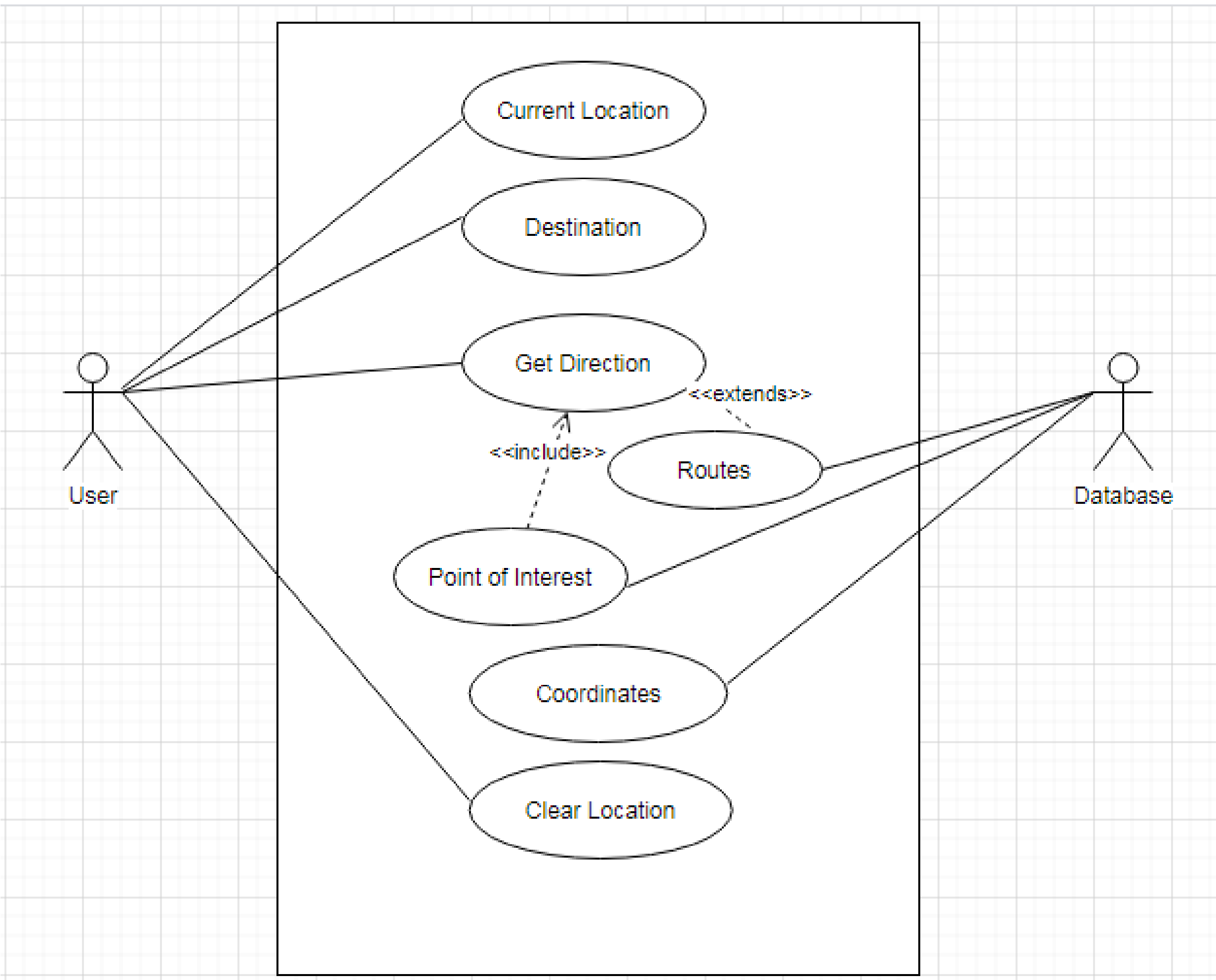


Figure 6

A use case diagram is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

3.5.2 CLASS DIAGRAM

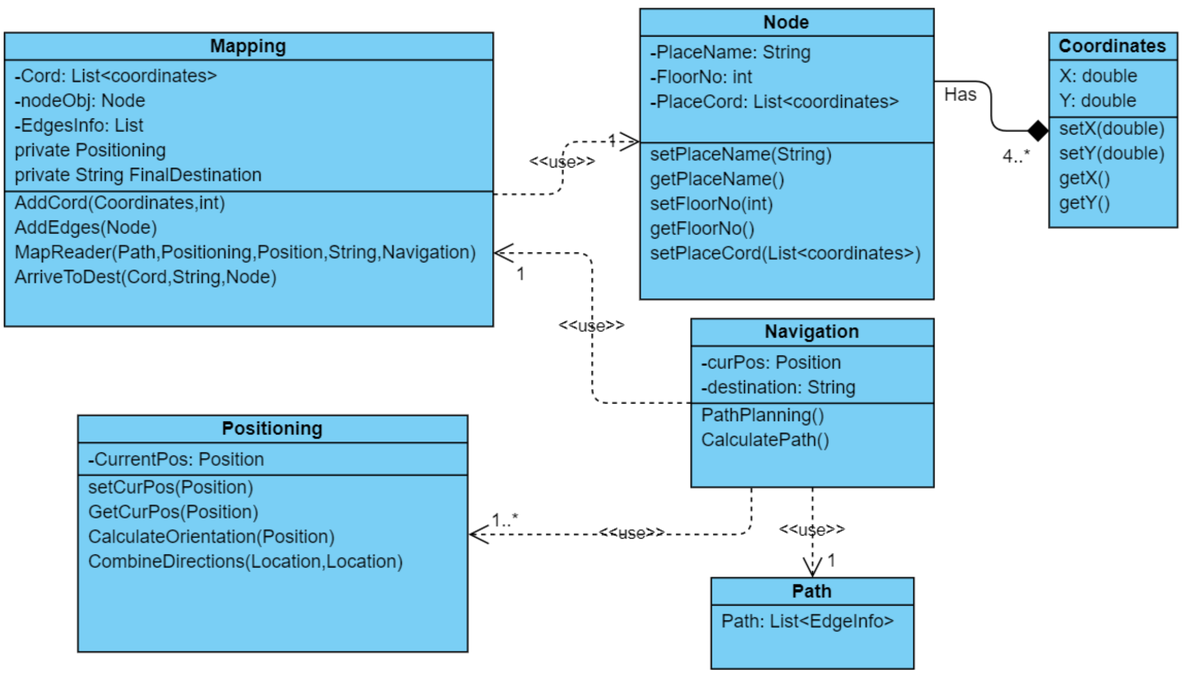


Figure 7

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

3.5.3 SEQUENCE DIAGRAM

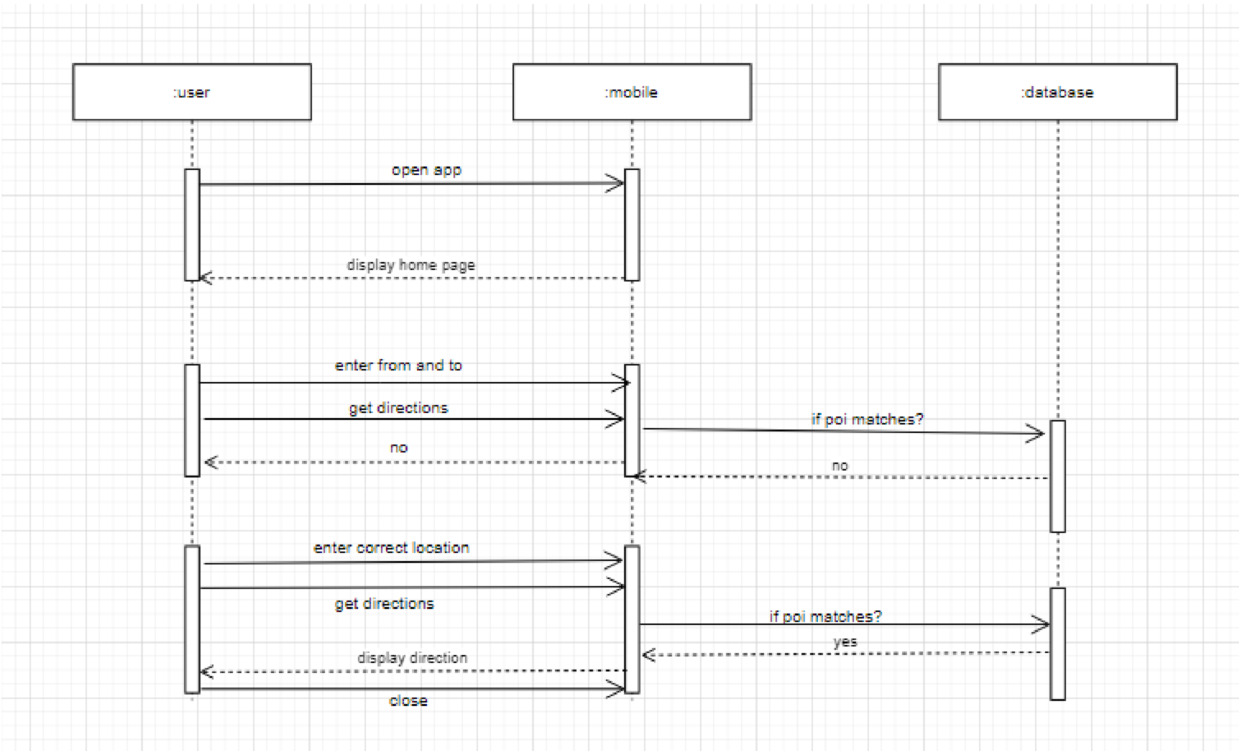


Figure 8

A sequence diagram shows object interactions arranged in time sequence. It depicts the sequence of messages exchanged between the objects to carry out the functionality of the scenario.

3.5.4 COLLABORATION DIAGRAM

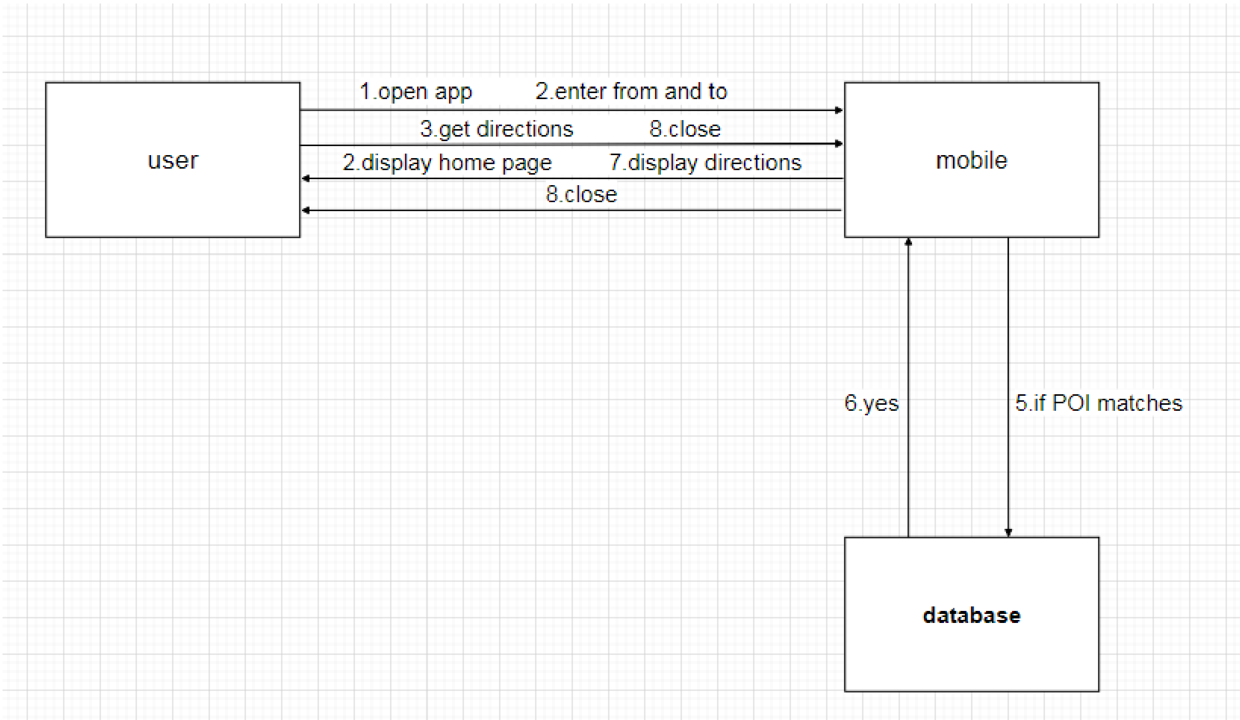


Figure 9

A Collaboration diagram models the interactions between objects or parts in terms of sequenced messages.

3.5.5 STATE CHART DIAGRAM

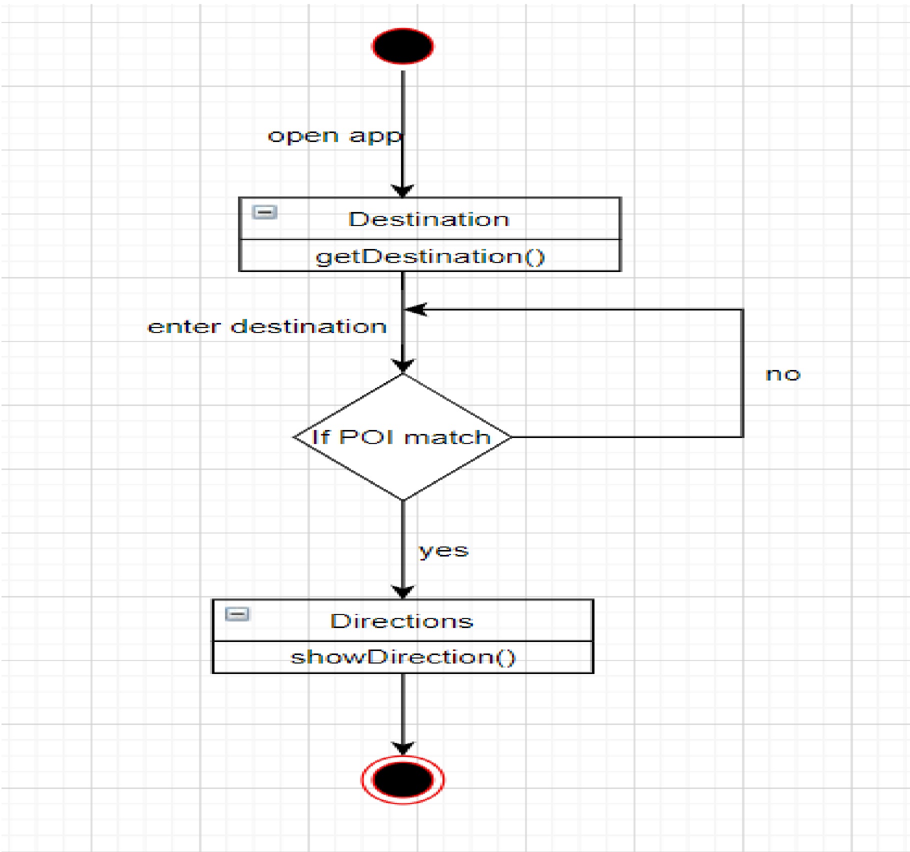


Figure 10

A state diagram shows the behaviour of classes in response to external stimuli. Specifically, it describes the behaviour of a single object in response to a series of events in a system.

3.5.6 ACTIVITY DIAGRAM

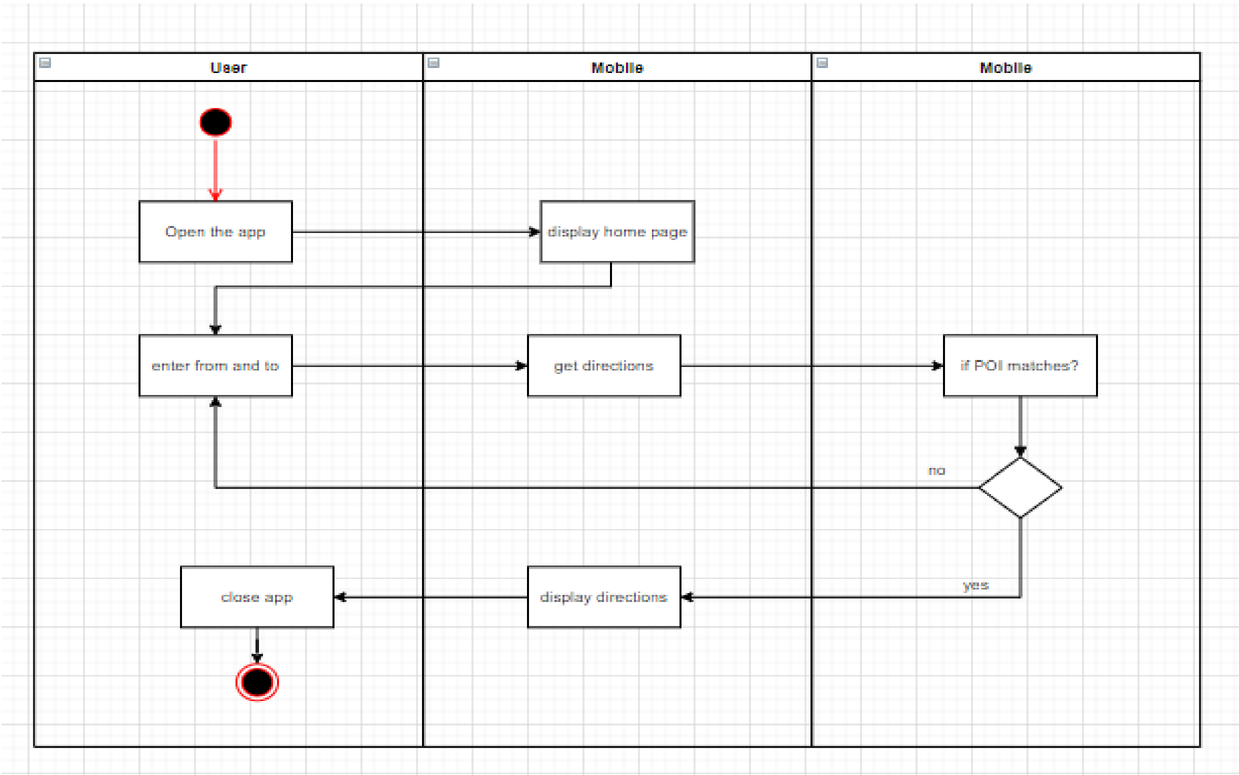


Figure 11

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

3.5.7 COMPONENT DIAGRAM

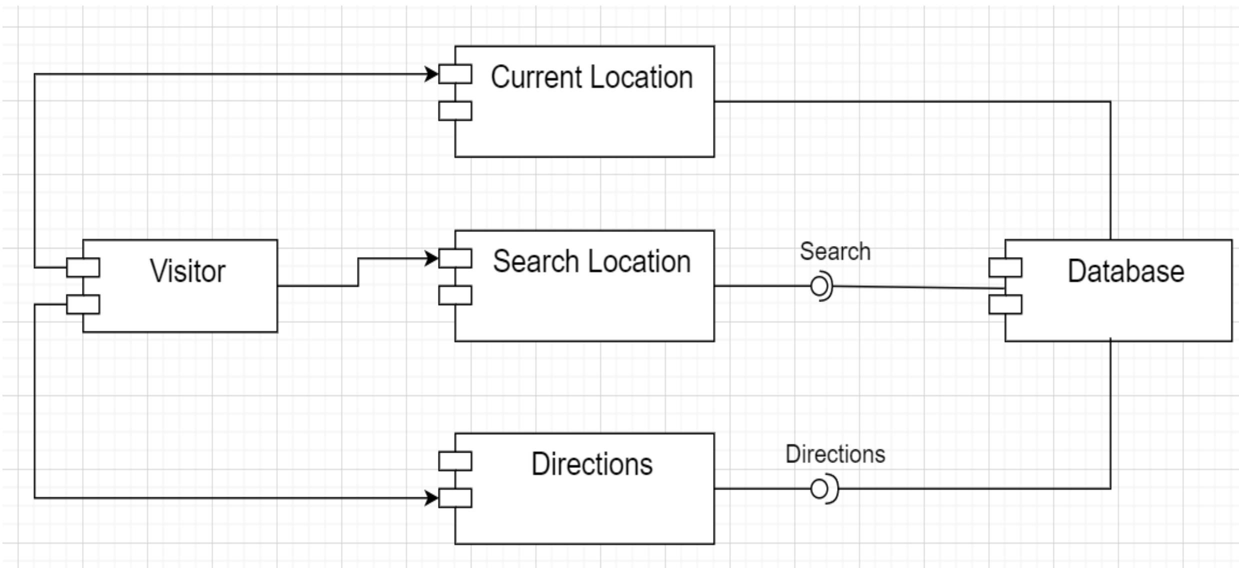


Figure 12

A component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

3.5.8 DEPLOYMENT DIAGRAM

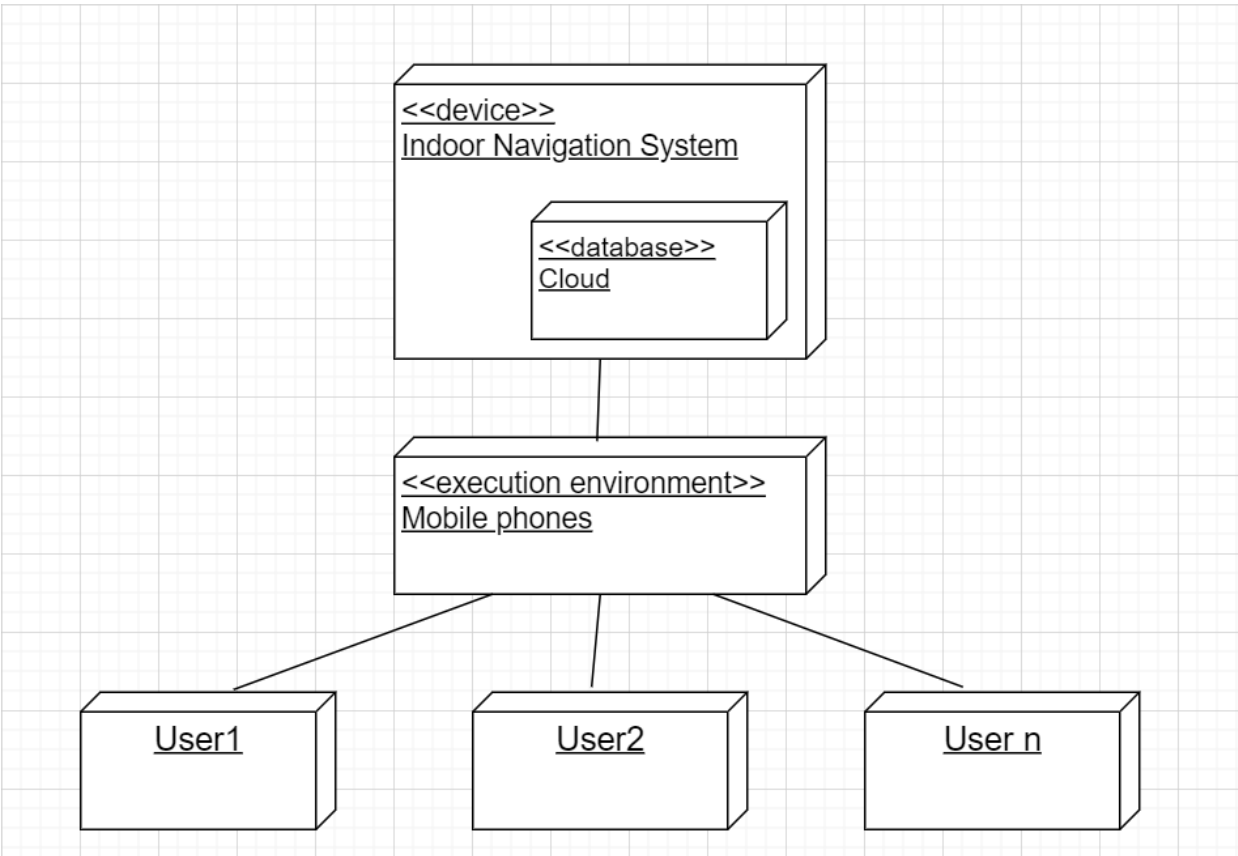


Figure 13

A deployment diagram models the physical deployment of artifacts on nodes. It shows what hardware components exist, what software components run on each node, and how the different pieces are connected.

3.5.9 PACKAGE DIAGRAM

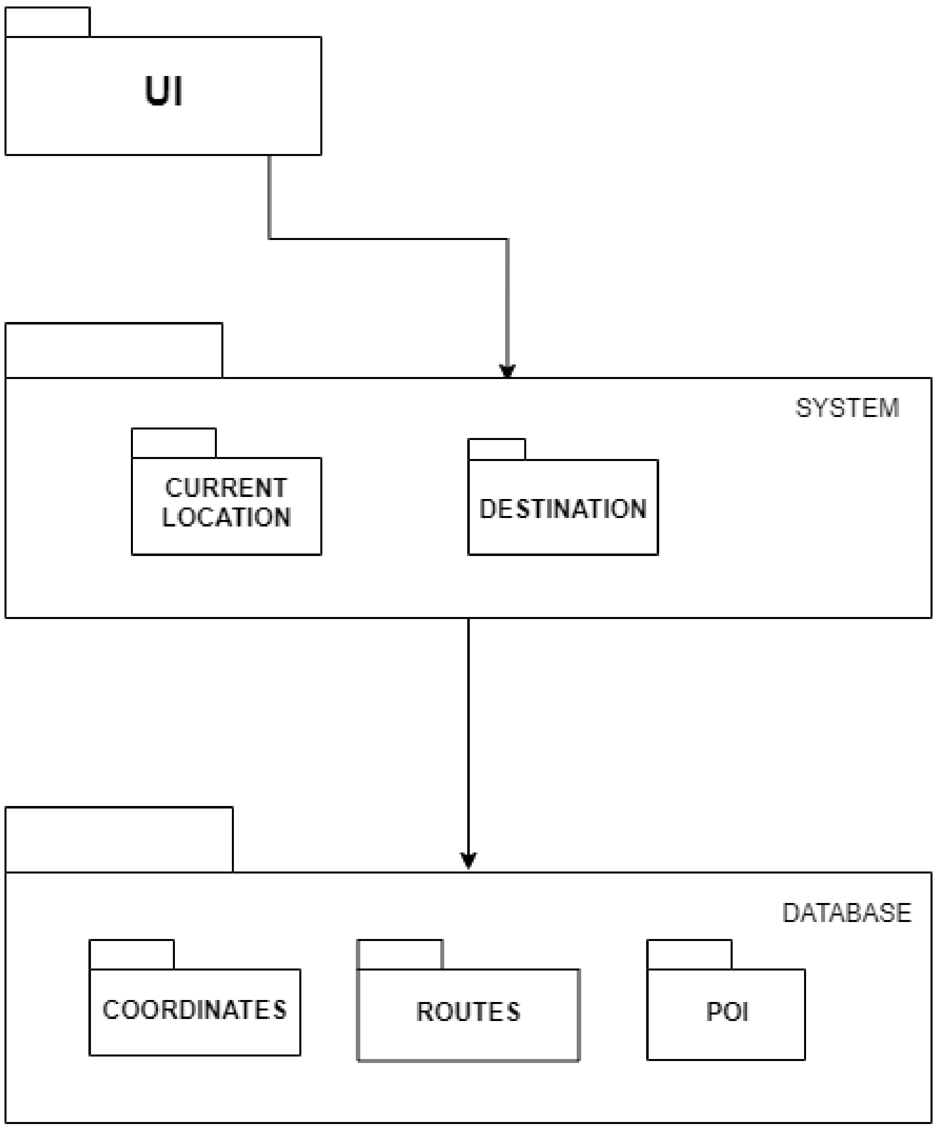


Figure 14

Package Diagram can be used to simplify complex class diagrams, it can group classes into packages. A package is a collection of logically related UML elements.

3.6 SUMMARY

All the UML diagrams for the system are drawn displaying various objects, functions and classes of the system.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

4.1.1 Software Requirements

4.1.1.1 Android Studio

For Google's Android operating system, Android Studio is the official integrated development environment (IDE). It is built on Jetbrains' IntelliJ IDEA software and specifically designed for Android development. It is available for download on all Operating systems like Windows, macOS and Linux. Eclipse Android Development Tools (ADT) had been the primary IDE for Native Android applications. Android studio has been a replacement for Eclipse.

4.1.1.1.2 Features of Android

The following features are provided in the current stable version:

- Gradle-based build support.
- Android-specific refactoring and quick fixes Lint tools to catch performance, usability, version compatibility and other problems.
- ProGuard integration and app-signing capabilities.

4.1.1.1.3 Advantages of Android Studio

1. Instant Run helps Faster Deployment Bringing incremental changes to an existing app code or resource is now easier and faster.
2. Accurate and Easier Programming Android Studio makes code writing faster and easier.
3. Comparatively Faster Testing and Programming.
4. Inclusive App Development using Cloud Test Lab Integration.

4.1.1.2 Google Maps

Google offers a web mapping Service called the Google Maps. It offers maps in different set of views like the satellite view, panoramic view, street view for better User Interface. Google Maps provides us with an API which allows the users to embed maps into any website or mobile application. Route Planner is a main feature Google Maps which provides us four modes of transportation - Car, train, cycle or walk.

4.1.1.2.1 Features of Google Maps

i. Figuring out ways through big and confusing buildings can be a confusing task. Google Maps provides navigation through these buildings.

ii. It allows users to Download maps such that you can access them in situations where you are not connected to the Internet.

iii. It helps you keep track of what places you have visited in the last year.

iv. Also helps users to book flight tickets.

v. Lets us see confusing road trips at a glance.

vi. It saves the home or work addresses.

vii. Get tickets for concerts and shows.

viii. Get directions using Route planner. Get Directions to any target or destination.

ix. Gives turn by turn navigations to any destination quickly. You can choose any mode of transportation.

x. Explore all the wonders of the world.

4.1.1.2.2 Advantages:

- Full of information.
- Sharing benefits.
- Multiple Modes of Transport.

4.1.1.3 Detect Position

Should be able to detect continuously the current position of the user inside the building.

4.1.1.4 Generate Route

According to the detected position of the user inside the hallway and the specified end destination by the user the system generates a route between the 2 points according to the shortest path.

4.2 NON FUNCTIONAL REQUIREMENTS

4.2.1 Supporting Technologies

The system implementation should be feasible using technologies that are accessible to the end-users.

Area : Usability

4.2.2 Device Software Compatibility

The mobile interfaces must be compatible with Android**. Area : Portability**

4.2.3 Language

The language should be localized to the preference of the user. **Area : Delivery**

4.2.4 Time Response

The system must perform in a proper time constraint that reflects average walking speed, motion and obstacles in the environment. **Area : Performance Efficiency**

4.2.5 Multi User System

The system is able to consider the presence of more than one user in the same environment. All the features of the system should operate properly for all users.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 SYSTEM ARCHITECTURE

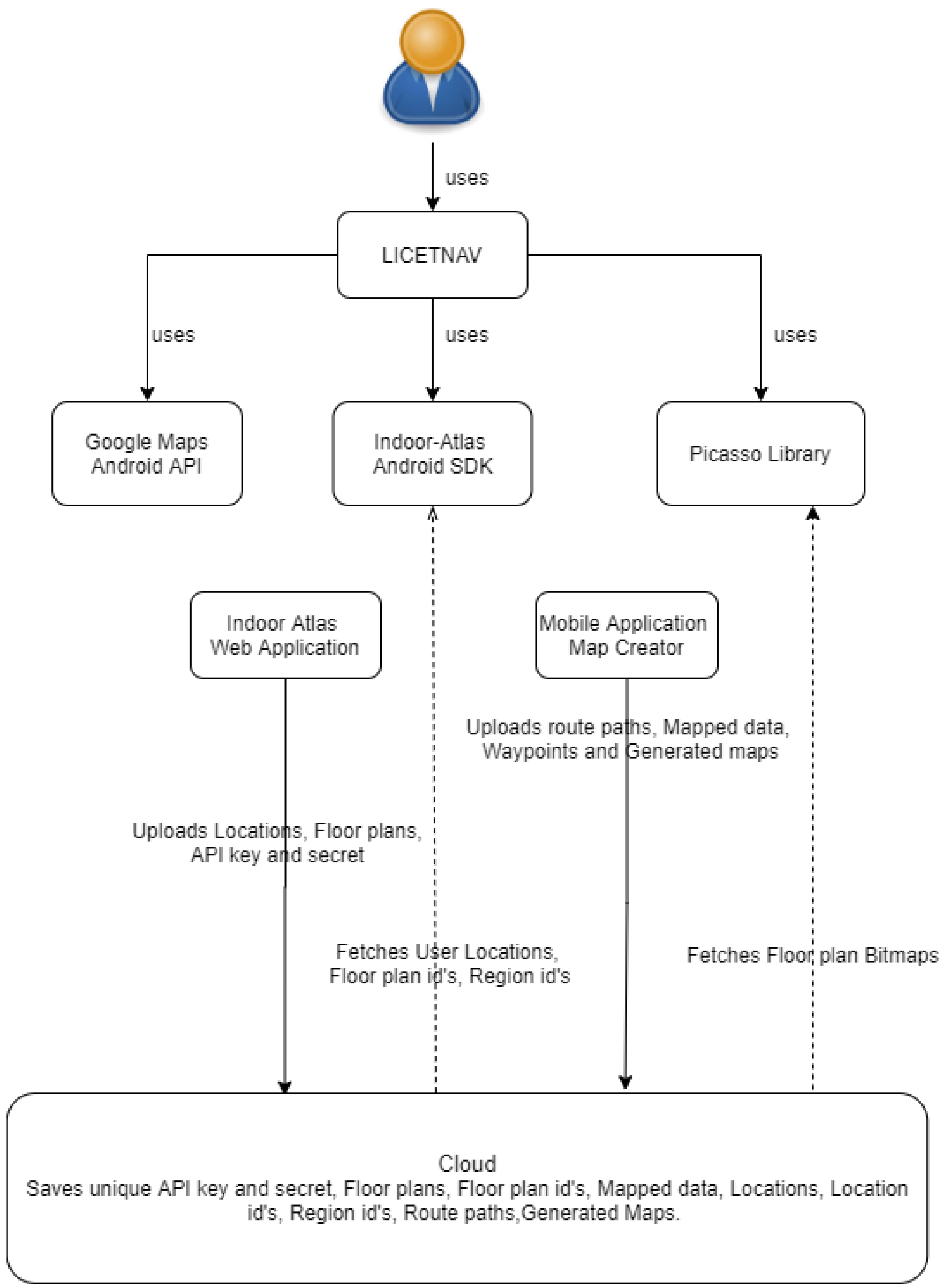


Figure 15 – System Architecture

LICETNAV is an Android mobile application that aims at directing the user to their destination inside the building. This application is designed especially for Loyola ICAM College of Engineering and Technology(LICET), Chennai. Searching for rooms can be a challenging task. Fortunately, this application makes our task easier. As shown in the above figure, the major software components in LICETNAV are IndoorAtlas and Google Maps. This mobile application majorly focuses on locating the user inside a Building and directing them to their destination.

Firstly, we will discuss on how we locate the user inside a Building. Indoor Positioning Systems is a technology which work on locating objects indoors. As the microwave signals do not pass inside buildings, GPS can only detect the location outdoors. The Indoor Positioning Systems have been working on detecting objects inside buildings using the radio signals and magnetic waves collected by a smart phone. IndoorAtlas is a technology that works on the theory of Indoor Positioning system. They make use of the Wi-Fi and Bluetooth inside a building and calculate the position of the user inside a building. Each Application has a unique API key and secret. This key distinguishes one mobile application to another. After creating the Location and generating the API key and secret, we upload floor plans. Each floor plan needs to be aligned carefully with the Co-ordinates. These floor plans are saved in the cloud and can be fetched by the LICETNAV using the Picasso Library.

MapCreator is a Mobile application which maps the locations uploaded by the user. Mapping or Fingerprinting is the process of gathering signal data from a venue. This application must be installed from Google Play on an Android phone. This application helps us map the area of the location that has been added. You map the area by walking around and collecting data. This data is collected using Wi-Fi and Bluetooth signals in the smartphone. This data is sent automatically to the cloud. The cloud saves all the data uploaded by the user, which distinguishes the data of all the users by its API key and id.

IndoorAtlas SDK uses the features of IndoorAtlas in the mobile applications. They have a cross platform SDK both for Android and iOS. In this project Android SDK has been used. The Main Features of this SDK include fetching the location data from Cloud. Fetching the information of the region and fetching the information of the floor. The Package of IndoorAtlas SDK has a set of Interfaces and Classes

5.2 MODULE DESCRIPTION

5.2.1 User Application

Google Map API plays a major role in this project. The user location that is collected from the Cloud needs to be displayed on Google Maps as the SDK retrieves Location objects from the Cloud. The Location objects are a set of coordinates having latitudes and longitude data. Google Maps API is known for displaying location of the coordinates on the map.

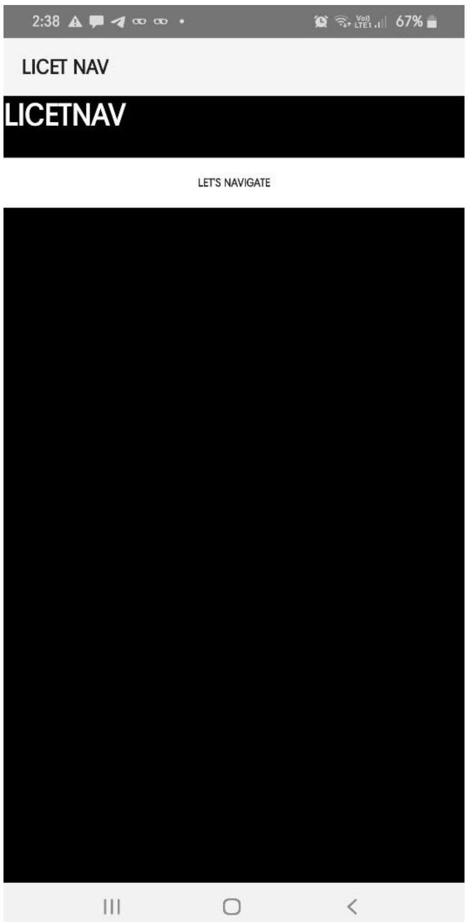
The main functions of Google Map API in this project are:

- Displaying maps on screen.
- Adding floor plans as Ground.

5.2.2 Adding Floor plans as overlay

Google Maps API lets us add images on maps as overlays. This serves as an advantage to this project as we can fetch the floor plan from Cloud and add it as an overlay on Google Maps. To fetch floor plan we use Picasso library. Picasso allows us to easily download images from target URLs. To sum it up, we fetch the floor plans from Cloud using Picasso, add them as a Ground Overlay on Google Map.

5.3 SCREENSHOTS



5.4 SUMMARY

The proposed application is to provide a turn-by-turn directions to the users who are not familiar with the interiors of the building, this saves time and energy of the user.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In this modern world, every individual has a mobile device. These devices have technologies like Wi-Fi, Bluetooth and GPS. The main aim is to use these technologies to track an individual inside a building. Indoor positioning is one of the most challenging things to achieve these days. Using indoor atlas SDK the inside of a building is mapped and the position of the position of the individual is obtained using the Wi-Fi, Bluetooth and GPS. Indoor positioning systems (IPS) locate people inside a building using radio signals, geomagnetic fields, inertial sensor data, barometric pressure, camera data or other sensory information collected by a smartphone device or tablet. Thus it is able to provide directions from the user's current location to the destination.

6.2 FUTURE ENHANCEMENT

In future, a QR code could be used to load the map of a particular building, the user can simply scan the QR code available in the entrance of a building to get the floor map in which he can get the directions to the destination.

SAMPLE CODE

UI DESIGN

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="59dp"
        android:text="LICETNAV"
        android:textColor="#FFF"
        android:textSize="30dp"
        android:textStyle="bold"/>

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="LET'S NAVIGATE"
        android:background="#FFF"/>
</LinearLayout>
```

WAYFINDING

```
        public void onWayfindingUpdate(IARoute route) {
            mCurrentRoute = route;
            if (hasArrivedToDestination(route)) {
                showInfo("You're there!");
                mCurrentRoute = null;
                mWayfindingDestination = null;
                mIALocationManager.removeWayfindingUpdates();
            }
            updateRouteVisualization();
        }
    };

    private IAOrientationListener mOrientationListener = new IAOrientationListener() {
        @Override
        public void onHeadingChanged(long timestamp, double heading) {
            updateHeading(heading);
        }
```

```
    @Override
    public void onOrientationChange(long timestamp, double[] quaternion) {
    }
};

private int mFloor;

private void showLocationCircle(LatLng center, double accuracyRadius) {
    if (mCircle == null) {
            if (mMap != null) {
        mCircle = mMap.addCircle(new CircleOptions()
                .center(center)
                .radius(accuracyRadius)
                .fillColor(0x201681FB)
                .strokeColor(0x500A78DD)
                .zIndex(1.0f)
                .visible(true)
                .strokeWidth(5.0f));
        mHeadingMarker = mMap.addMarker(new MarkerOptions()
                .position(center)
                .icon(BitmapDescriptorFactory.fromResource(R.drawable.map_blue_dot))
                .anchor(0.5f, 0.5f)
                .flat(true));
      }
    } else {
        mCircle.setCenter(center);
        mHeadingMarker.setPosition(center);
        mCircle.setRadius(accuracyRadius);
    }
}



@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    findViewById(android.R.id.content).setKeepScreenOn(true);
    mIALocationManager = IALocationManager.create(this);

    ((SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map))
            .getMapAsync(this);
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setMyLocationEnabled(false);
    mMap.setOnMapClickListener(this);
    mMap.getUiSettings().setMapToolbarEnabled(false);

    mMap.setOnMarkerClickListener(new GoogleMap.OnMarkerClickListener() {
        @Override
        public boolean onMarkerClick(Marker marker) {
            if (marker == mDestinationMarker) return false;
```

```
            setWayfindingTarget(marker.getPosition(), false);
            return false;
        }
    });
}

private void setupPoIs(List<IAPOI> pois, int currentFloorLevel) {
    Log.d(TAG, pois.size() + " PoI(s)");
    for (Marker m : mPoIMarkers) {
        m.remove();
    }
    mPoIMarkers.clear();
    for (IAPOI poi : pois) {
        if (poi.getFloor() == currentFloorLevel) {
            mPoIMarkers.add(mMap.addMarker(new MarkerOptions()
                    .title(poi.getName())
                    .position(new LatLng(poi.getLocation().latitude, poi.getLocation().longitude))
                    .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE))));
        }
    }
}

private void setupGroundOverlay(IAFloorPlan floorPlan, Bitmap bitmap) {

    if (mGroundOverlay != null) {
        mGroundOverlay.remove();
    }

    if (mMap != null) {
        BitmapDescriptor bitmapDescriptor = BitmapDescriptorFactory.fromBitmap(bitmap);
        IALatLng iaLatLng = floorPlan.getCenter();
        LatLng center = new LatLng(iaLatLng.latitude, iaLatLng.longitude);
        GroundOverlayOptions fpOverlay = new GroundOverlayOptions()
                .image(bitmapDescriptor)
                .zIndex(0.0f)
                .position(center, floorPlan.getWidthMeters(), floorPlan.getHeightMeters())
                .bearing(floorPlan.getBearing());

        mGroundOverlay = mMap.addGroundOverlay(fpOverlay);
    }
}
private void fetchFloorPlanBitmap(final IAFloorPlan floorPlan) {

    if (floorPlan == null) {
        Log.e(TAG, "null floor plan in fetchFloorPlanBitmap");
        return;
    }

    final String url = floorPlan.getUrl();
    Log.d(TAG, "loading floor plan bitmap from "+url);

    mLoadTarget = new Target() {

        @Override
        public void onBitmapLoaded(Bitmap bitmap, Picasso.LoadedFrom from) {
            Log.d(TAG, "onBitmap loaded with dimensions: " + bitmap.getWidth() + "x"
                    + bitmap.getHeight());
            if (mOverlayFloorPlan != null && floorPlan.getId().equals(mOverlayFloorPlan.getId())) {
                Log.d(TAG, "showing overlay");
```

```
                setupGroundOverlay(floorPlan, bitmap);
            }
        }

    RequestCreator request = Picasso.with(this).load(url);

    final int bitmapWidth = floorPlan.getBitmapWidth();
    final int bitmapHeight = floorPlan.getBitmapHeight();

    if (bitmapHeight > MAX_DIMENSION) {
        request.resize(0, MAX_DIMENSION);
    } else if (bitmapWidth > MAX_DIMENSION) {
        request.resize(MAX_DIMENSION, 0);
    }

    request.into(mLoadTarget);
}


@Override
public void onMapClick(LatLng point) {
    if (mPoIMarkers.isEmpty()) {
        setWayfindingTarget(point, true);
    }
}

private void setWayfindingTarget(LatLng point, boolean addMarker) {
    if (mMap == null) {
        Log.w(TAG, "map not loaded yet");
        return;
    }

    mWayfindingDestination = new IAWayfindingRequest.Builder()
            .withFloor(mFloor)
            .withLatitude(point.latitude)
            .withLongitude(point.longitude)
            .build();

    mIALocationManager.requestWayfindingUpdates(mWayfindingDestination, mWayfindingListener);

    if (mDestinationMarker != null) {
        mDestinationMarker.remove();
        mDestinationMarker = null;
    }

    if (addMarker) {
        mDestinationMarker = mMap.addMarker(new MarkerOptions()
                .position(point)
                .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE)));
    }
    Log.d(TAG, "Set destination: (" + mWayfindingDestination.getLatitude() + ", " +
            mWayfindingDestination.getLongitude() + "), floor=" +
            mWayfindingDestination.getFloor());

}

private boolean hasArrivedToDestination(IARoute route) {
    if (route.getLegs().size() == 0) {
        return false;
```

```
        }

        final double FINISH_THRESHOLD_METERS = 8.0;
        double routeLength = 0;
        for (IARoute.Leg leg : route.getLegs()) routeLength += leg.getLength();
        return routeLength < FINISH_THRESHOLD_METERS;
    }
    private void clearRouteVisualization() {
        for (Polyline pl : mPolylines) {
            pl.remove();
        }
        mPolylines.clear();
    }

    private void updateRouteVisualization() {

        clearRouteVisualization();

        if (mCurrentRoute == null) {
            return;
        }

        for (IARoute.Leg leg : mCurrentRoute.getLegs()) {

            if (leg.getEdgeIndex() == null) {

                continue;
            }

            PolylineOptions opt = new PolylineOptions();
            opt.add(new LatLng(leg.getBegin().getLatitude(), leg.getBegin().getLongitude()));
            opt.add(new LatLng(leg.getEnd().getLatitude(), leg.getEnd().getLongitude()));

            if (leg.getBegin().getFloor() == mFloor && leg.getEnd().getFloor() == mFloor) {
                opt.color(0xFF0000FF);
            } else {
                opt.color(0x300000FF);
            }

            mPolylines.add(mMap.addPolyline(opt));
        }
    }
}
```

REFERENCE

[1] Sand, S., Dammann, A., & Mensing, C. (2014). Positioning in Wireless Communications Systems. Hoboken: Wiley

[2] Karimi, H. (2015). Indoor Wayfinding and Navigation. Hoboken: CRC Press.

[3] Horton, J., Vasconcelos, H., & Portales, R. (2016). Android: Programming for Developers. Packt Publishing.

[4] Ruiz, A. (2015). Mastering Android Application Development.Birmingham: Packt Publishing.

[5] https://www.infsoft.com/Portals/0/Images/use-cases/2018/infsoft-Use-Case-Indoor-Navigation-and-Tracking-on-Campus-Areas.pdf

[6] Alberto J. Palma (2017). Evolution of Indoor Positioning Technology