HUFF-DUFF : AN INDOOR NAVIGATION SYSTEM

A PROJECT REPORT

Submitted by

NADHEEDHA S (311120205041)
SHARON NIVEDITA P (311120205051)
SHREE VARAA MANGAI V (311120205052)

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

MAY 2023

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project **"HUFF-DUFF: AN INDOOR NAVIGATION SYSTEM"** is the bonafide work of **NADHEEDHA S(311120205041), SHARON NIVEDITA P(311120205051),SHREE VARAA MANGAI V(311120205052)** 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

Dr. A Janani, M.S., Ph.D.,
SUPERVISOR
Associate 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 S Sebastian SJ** and the Dean of Students **Dr D. Caleb Chanthi Raj**.

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, M.E., Ph.D.,** for her comments and suggestions.

We also would like to thank the project coordinators **Dr. A Janani, M.S., Ph.D.,** and **Ms G Shobana, M.Tech.,** 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 our project guide **Dr. A Janani, M.S., Ph.D.,** 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.

TABLE OF CONTENTS

ABSTRACT

Indoor navigation systems play a crucial role in helping individuals navigate complex indoor environments. Our project, HUFF-DUFF, is an indoor navigation app that aims to simplify and enhance the way individuals navigate the indoor spaces of the campus, ultimately improving overall campus experience and efficiency . The app assists students, faculties and visitors in navigating seamlessly within the buildings and facilities of the campus. HUFF-DUFF app leverages advanced technologies such as Bluetooth Low Energy (BLE)& Wi-Fi positioning to provide accurate and intuitive navigation experiences. The app utilizes a combination of indoor maps, real-time location tracking, and personalized routing algorithms to guide users through the campus buildings efficiently.

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

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

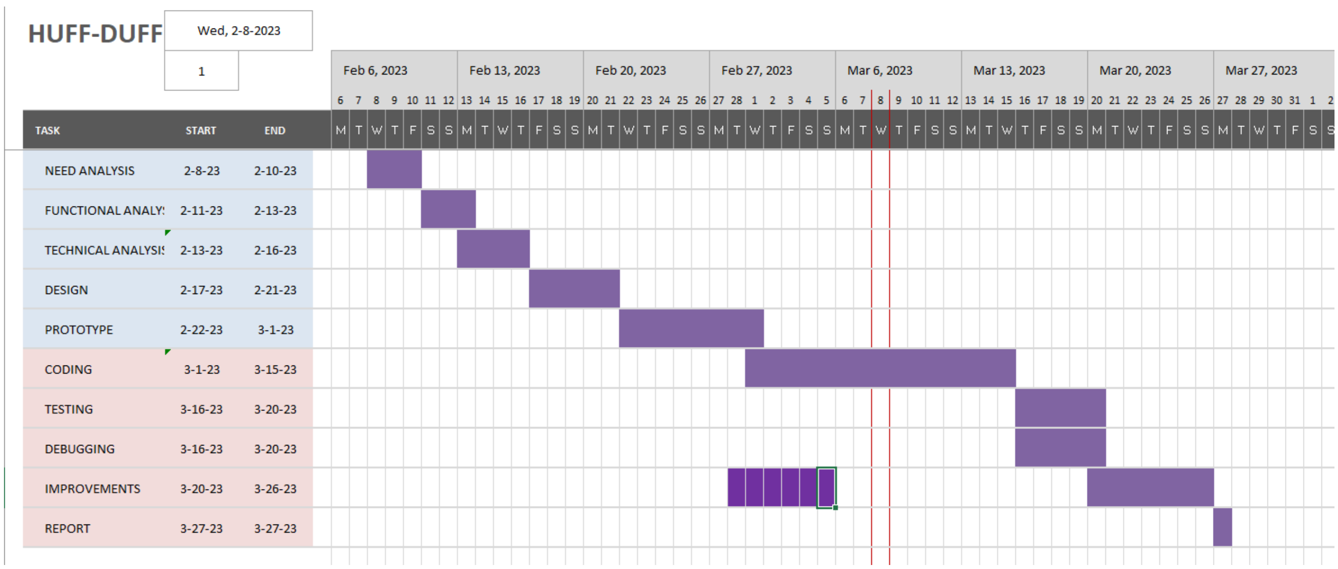


Figure 1 – Gantt chart

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

[1] Title: Indoor navigation using IndoorAtlas library

Author(s): Jan Hurtuk, Jakub Cervena, Martin Stancel, Michal Hulic, Peter Feciak

Abstract:

The aim of the paper was to design and implement an indoor navigation app using a highly efficient technology. The system consists of a localization and a navigation part. The designed app uses the IndoorAtlas library to locate devices (in our case a smart phone) inside of buildings. IndoorAtlas measures specific properties of a magnetic field of a building which is created by interaction of the Earth's magnetic field with ferromagnetic materials used in building constructions. In this paper there is At the end of the paper there were conducted two experimental measurements to verify the proposed solution.

Inference: Thus,this paper describes a design of a solution and its implementation in Java programming language for creating an indoor navigation app in android platform using indoor atlas .

[2] Title: Indoor positioning and wayfinding systems: a survey

Author(s): Jayakanth Kunhoth, AbdelGhani Karkar, Somaya Al-Maadeed & Abdulla Al-Ali

Abstract:

In indoor environments, lack of Global Positioning System (GPS) signals and line of sight with orbiting satellites makes navigation more challenging compared to outdoor environments. Radio frequency (RF) signals, computer vision, and sensor-based solutions are more suitable for tracking the users in indoor environments. This article provides a comprehensive summary of evolution in indoor navigation and indoor positioning technologies.the paper reviews different computer vision-based indoor navigation and positioning systems along with indoor scene recognition methods that can aid indoor navigation

Inference: Thus, the paper focuses on navigation and positioning systems that utilize pedestrian dead reckoning (PDR) methods and various communication technologies, such as Wi-Fi, Radio Frequency Identification (RFID) visible light, Bluetooth and ultra-wide band (UWB), as well.

[3] Title: Wi-Fi Based Indoor Navigation System For Campus Directions

Author(s): Saeed Ahmed Magsi; Nordin Saad; Mohd Haris Bin Md Khir; Gunawan Witjaksono; Muham Aadil Siddiqui,Linta Sameer

Abstract:

Car navigation systems have taken place everywhere around the globe. Outdoor navigation systems can be widely located in almost every geographical location that has access to the internet. However, indoor navigation systems are very important for pedestrians to find out their ways in complicated indoor areas such as shopping malls, subways, universities, exhibitions, underground areas, and tunnels etc. The recent rapid expansion of Wi-Fi zones and the increase of Wi-Fi equipped smartphones have enabled transformation of this idea to reality. This work presents a university indoor navigation system designed to help students, staff, and visitors to find their desired location.

Inference: Thus,this paper shows promising results when using distance measuring algorithm for indoor navigation systems.

[4] Title: Evolution of Indoor Positioning Technologies: A Survey

Author(s): Ramon F. Brena,Juan Pablo García-Vázquez,Carlos E. Galván-Tejada,David Muñoz-Rodriguez,Cesar Vargas-Rosales and James Fangmeyer

Abstract:

Indoor positioning systems (IPS) use sensors and communication technologies to locate objects in indoor environments. IPS are attracting scientific and enterprise interest because there is a big market opportunity for applying these technologies. There are many previous surveys on indoor positioning systems; however, most of them lack a solid classification scheme that would structurally map a wide field such as IPS, or omit several key technologies or have a limited perspective; finally, surveys rapidly become obsolete in an area as dynamic as IPS. The goal of this paper is to provide a technological perspective of indoor positioning systems, comprising a wide range of technologies and approaches.

Inference: Thus,this paper focuses on the evolution of the underlying technologies has had a very positive impact on the evolution of indoor positioning systems. We realized that changes in the subsequent versions of standards in a given technology can reduce some tasks in the positioning system or even solve some limitations.

[4] Title: Evolution of Indoor Positioning Technologies: A Survey

Author(s): Ramon F. Brena,Juan Pablo García-Vázquez,Carlos E. Galván-Tejada,David Muñoz-Rodriguez,Cesar Vargas-Rosales and James Fangmeyer

Abstract:

Indoor positioning systems (IPS) use sensors and communication technologies to locate objects in indoor environments. IPS are attracting scientific and enterprise interest because there is a big market opportunity for applying these technologies. There are many previous surveys on indoor positioning systems; however, most of them lack a solid classification scheme that would structurally map a wide field such as IPS, or omit several key technologies or have a limited

perspective; finally, surveys rapidly become obsolete in an area as dynamic as IPS. The goal of this paper is to provide a technological perspective of indoor positioning systems, comprising a wide range of technologies and approaches.

Inference: Thus,this paper focuses on the evolution of the underlying technologies has had a very positive impact on the evolution of indoor positioning systems. We realized that changes in the subsequent versions of standards in a given technology can reduce some tasks in the positioning system or even solve some limitations.

[5] Title: Head First Android Development

Author(s): Dawn Griffiths and David Griffiths

Abstract:

Using the unique "Head First" approach, the book combines engaging visuals, interactive exercises, and real-world examples to make learning Android enjoyable and effective.The book starts with an introduction to the Android platform and guides readers through the process of setting up their development environment. It covers essential concepts such as building user interfaces with XML and Android Studio's visual editor, handling user input, and working with various types of views.

Inference: Thus,the book enables readers to gain a solid understanding of Android development fundamentals and apply them in real-world scenarios.

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 faculties who are unfamiliar with the building.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 PROBLEM STATEMENT

The problem is we have an 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 systems are required to save people's energy and time and direct them to their destination.

3.1 NEED ANALYSIS

3.1.1 BULL DIAGRAM

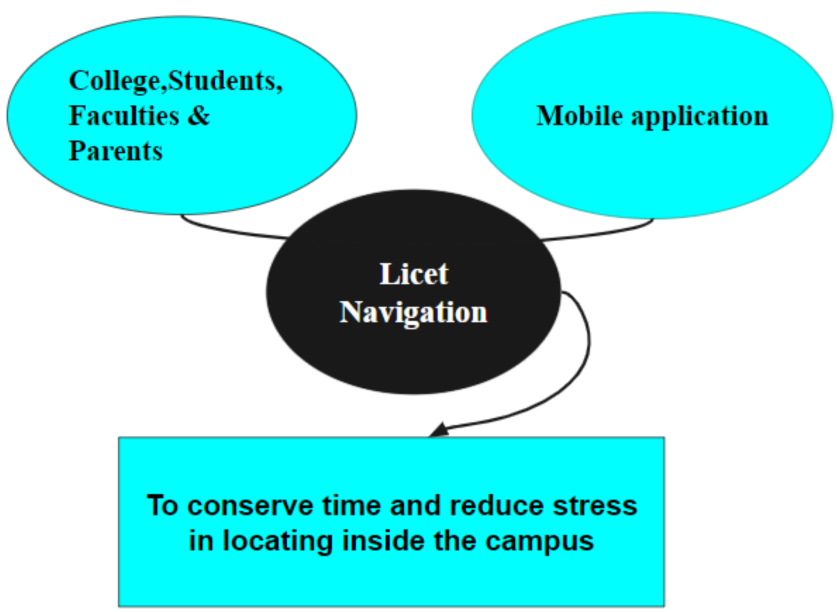


*Figure 1 – **Bull Diagram***

3.2 FUNCTIONAL ANALYSIS

3.2.1 OCTOPUS DIAGRAM

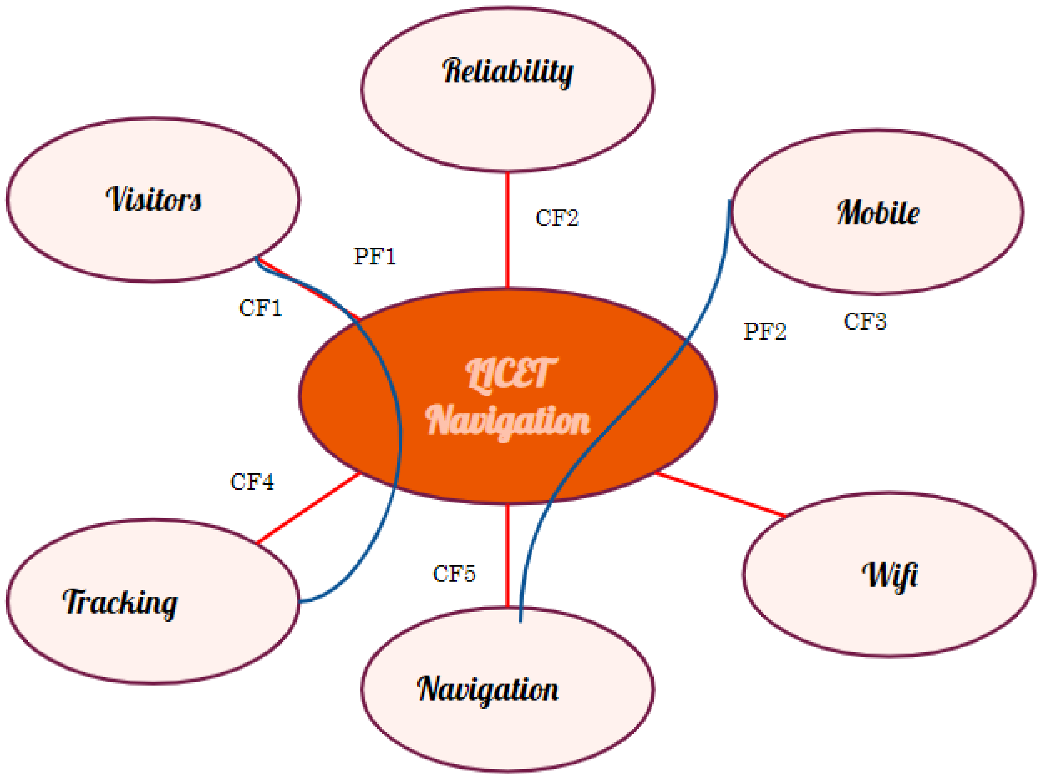


*Figure 2 – **Octopus diagram***

__**Principal Functions**__

PF1: Track current location

PF2: Navigate from current location to destination

__**Constraint Functions**__

CF1: To be user friendly

CF2: To avoid failure

CF3: To display path

CF4: To track user's current location

3.3 DATA FLOW DIAGRAM

3.3.1 DFD-0

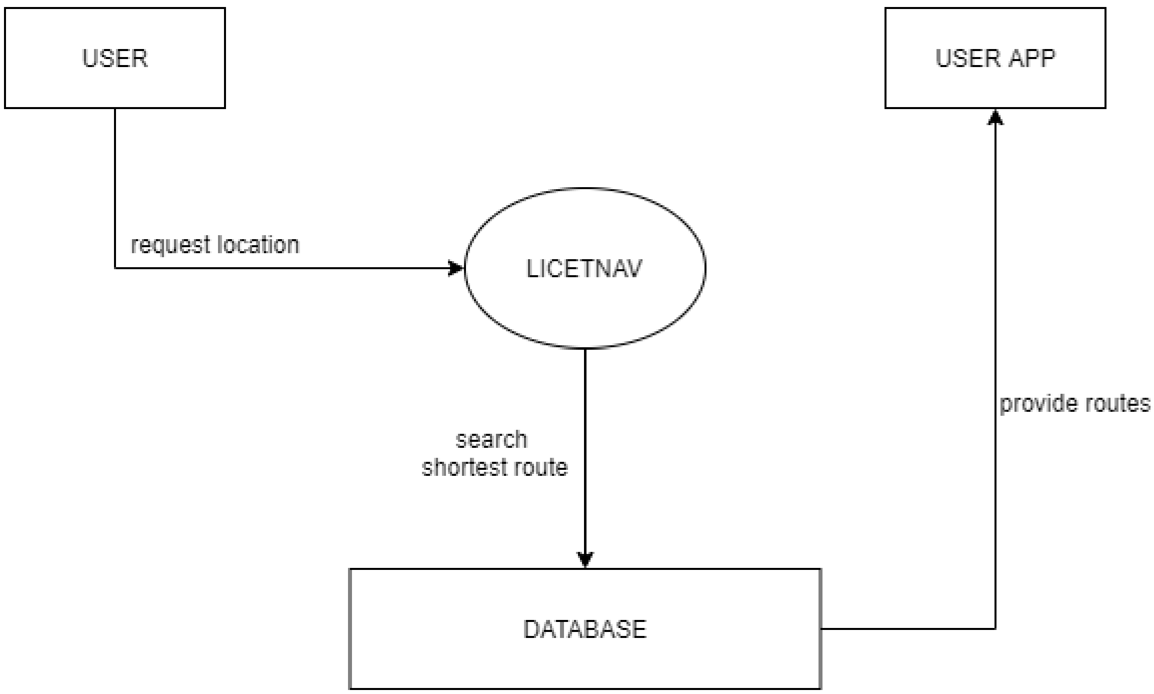


Figure 3

DFD 0 : the system as a single process with its relationship to external entities.

3.3.2 DFD-1

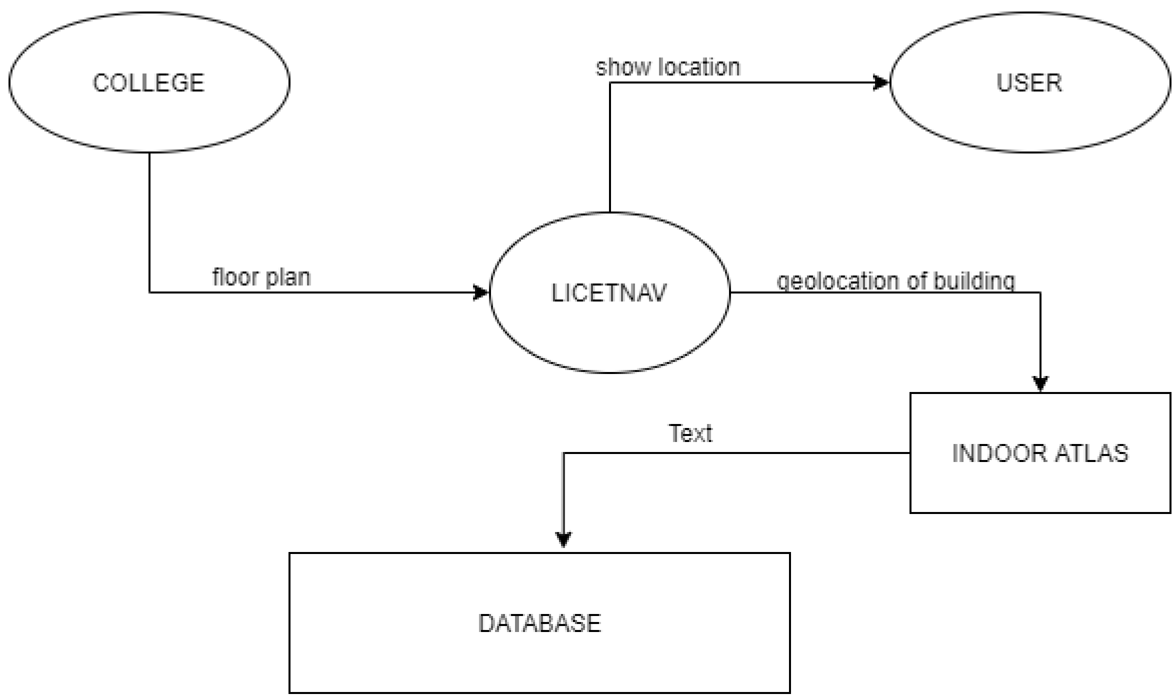


Figure 4

DFD 1 : main functions of the system

3.3.3 DFD-2

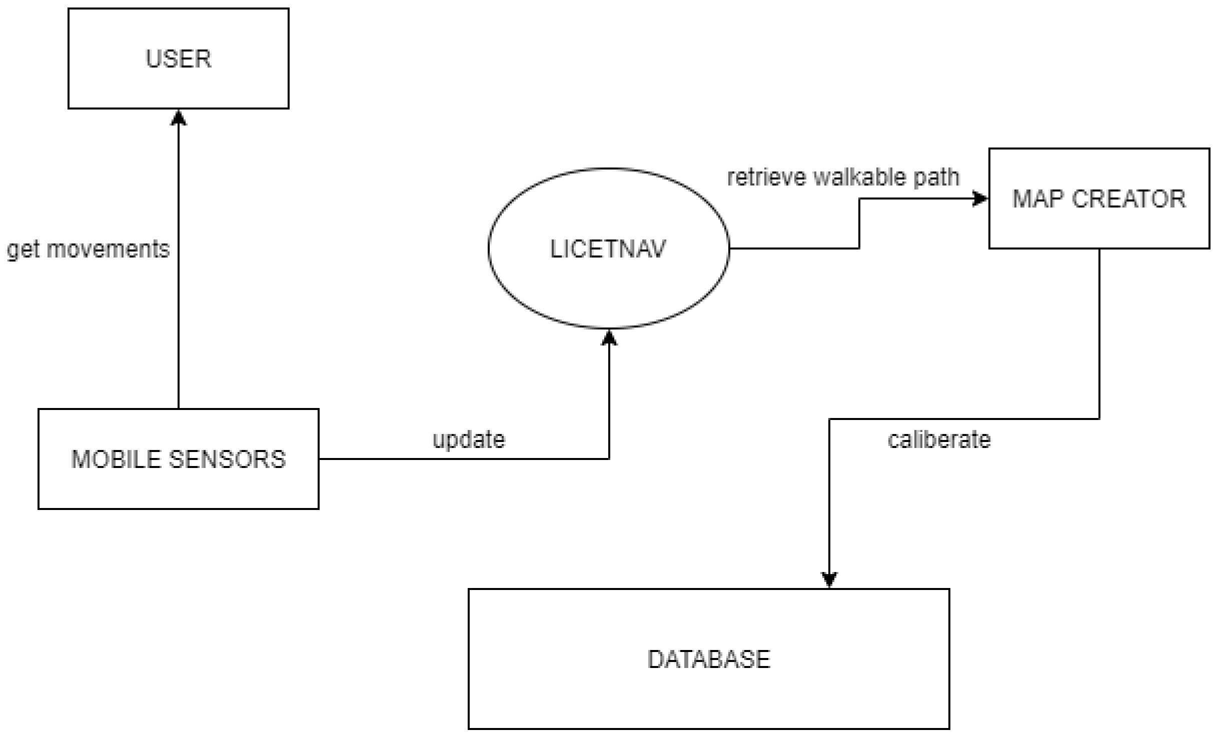


Figure 5

DFD 2 : system's functioning.

3.4 UML DIAGRAM

3.4.1 USE CASE DIAGRAM

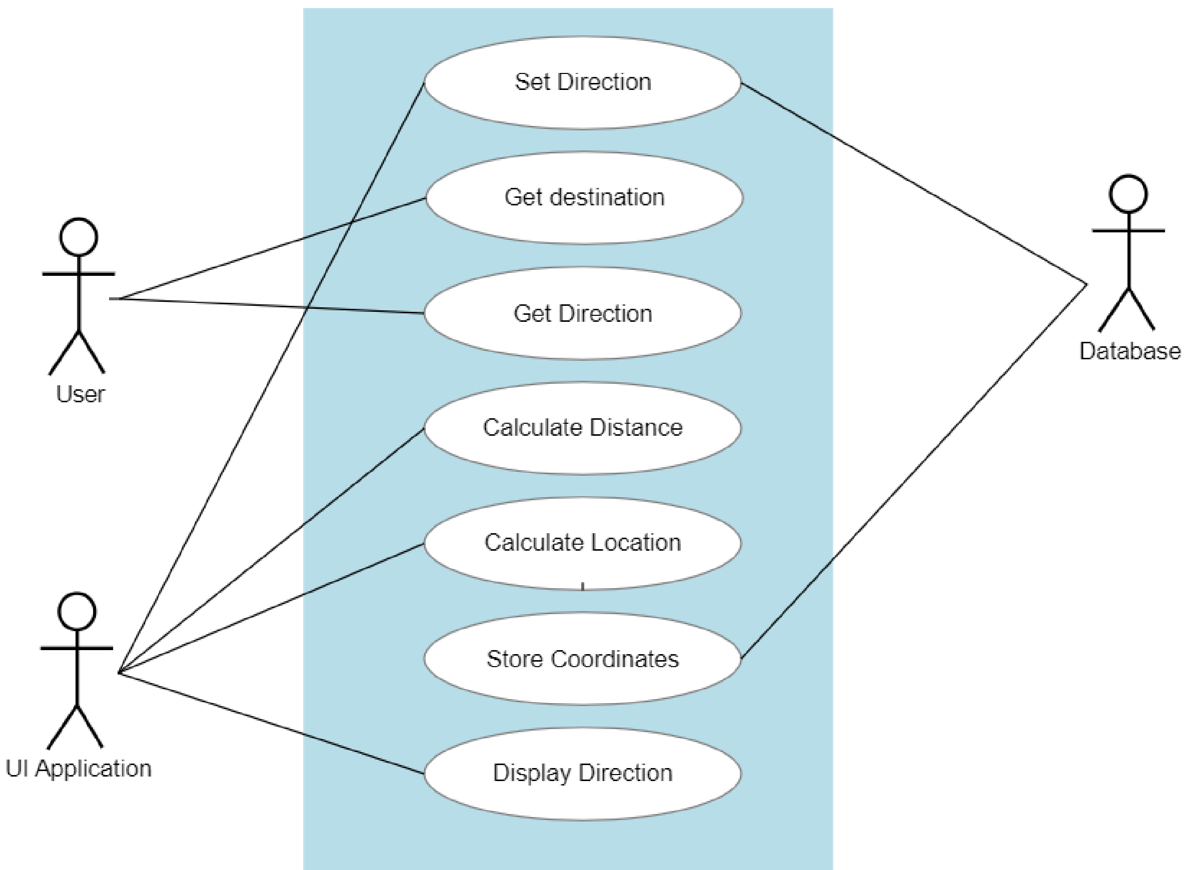


Figure 6

Representation of a user's interaction with the system

3.4.2 CLASS DIAGRAM

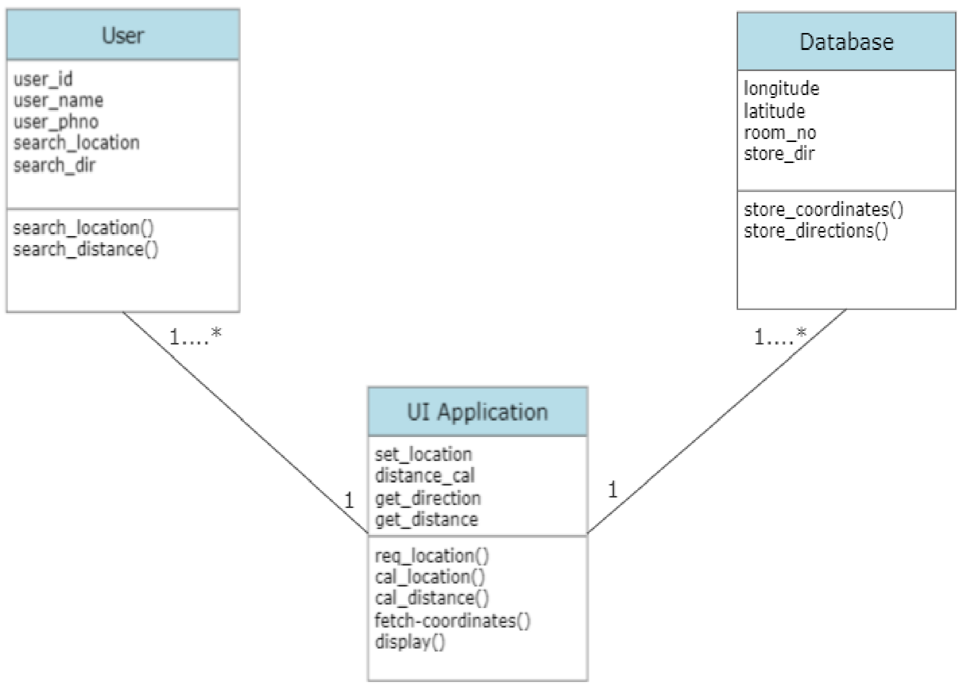


Figure 7

type of static structure diagram that describes the structure of a system

3.4.3 SEQUENCE DIAGRAM

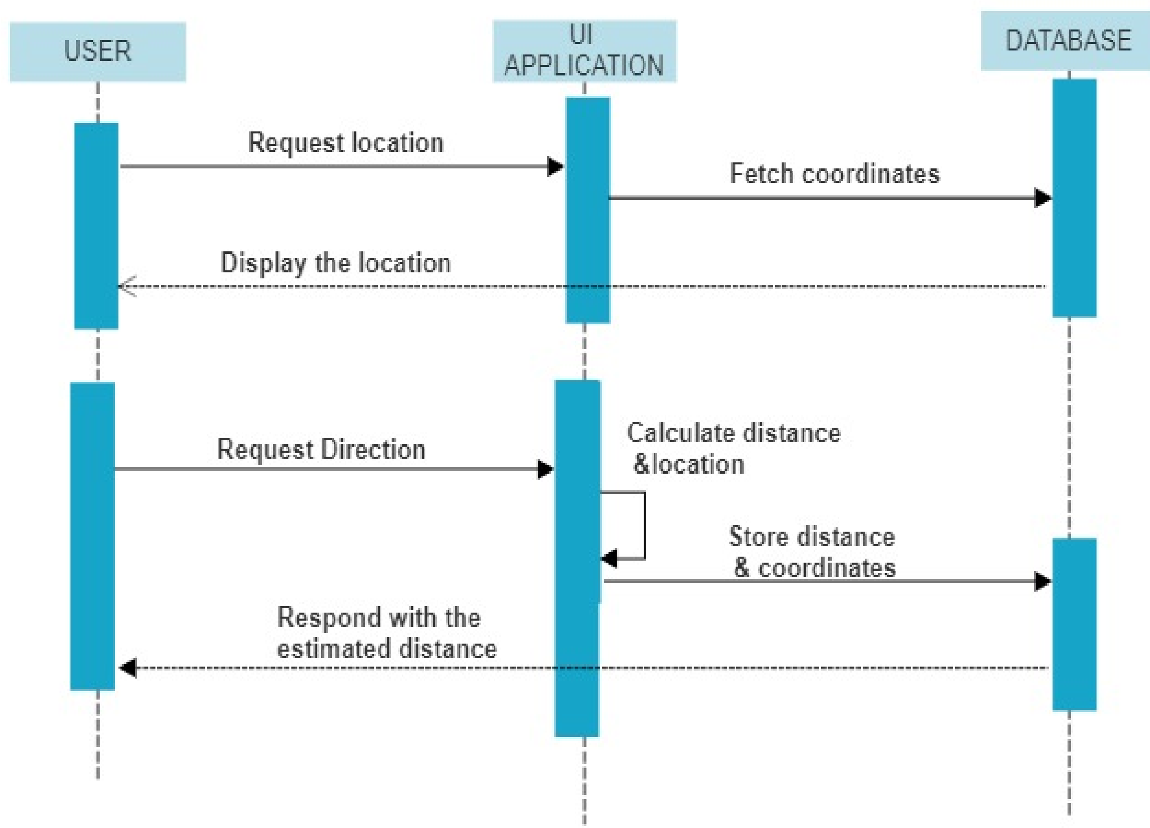


Figure 8

object interactions arranged in time sequence.

3.4.4 COLLABORATION DIAGRAM

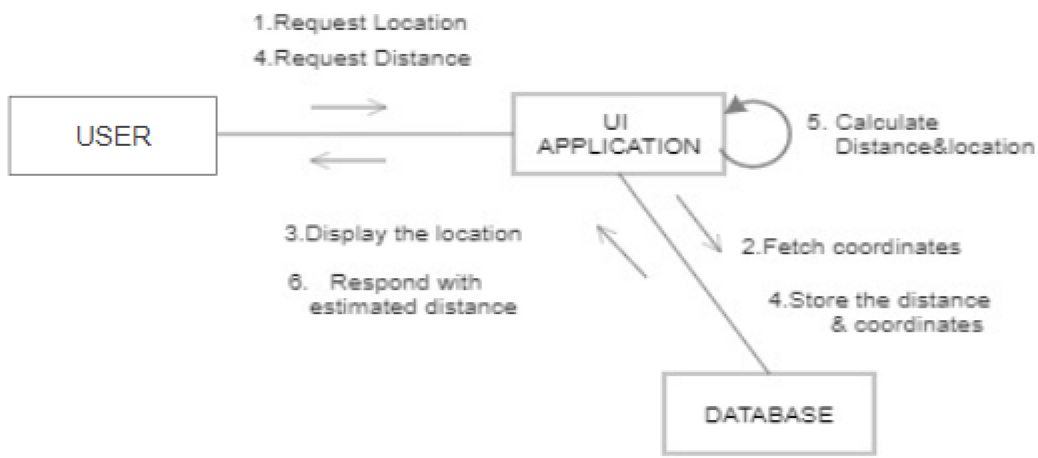


Figure 9

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

3.4.5 STATE CHART DIAGRAM

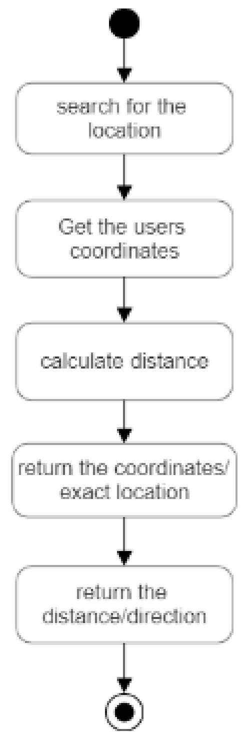


Figure 10
A state diagram shows the behavior of classes in response to external stimuli.

3.4.6 ACTIVITY DIAGRAM

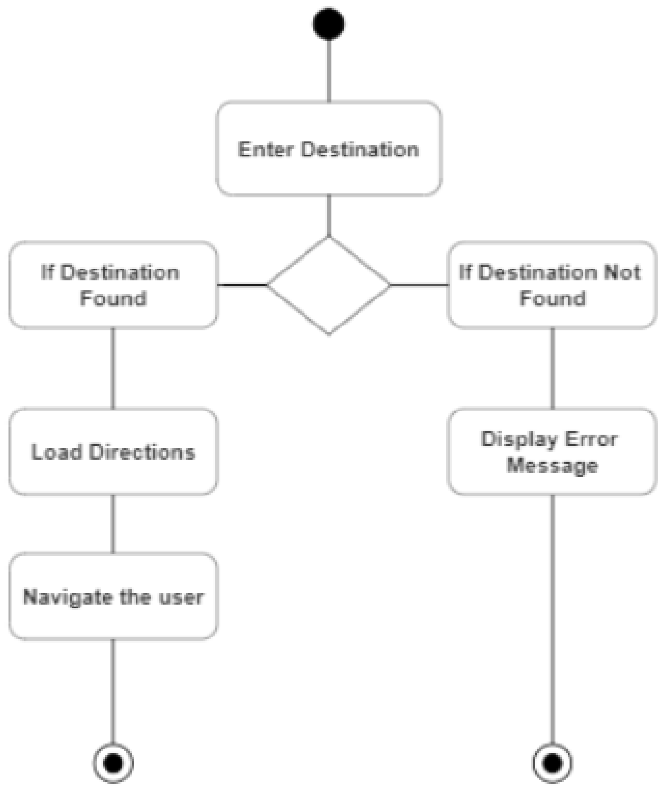


Figure 11
Activity diagrams are graphical representations of workflows of stepwise activities and actions

3.4.7 COMPONENT DIAGRAM

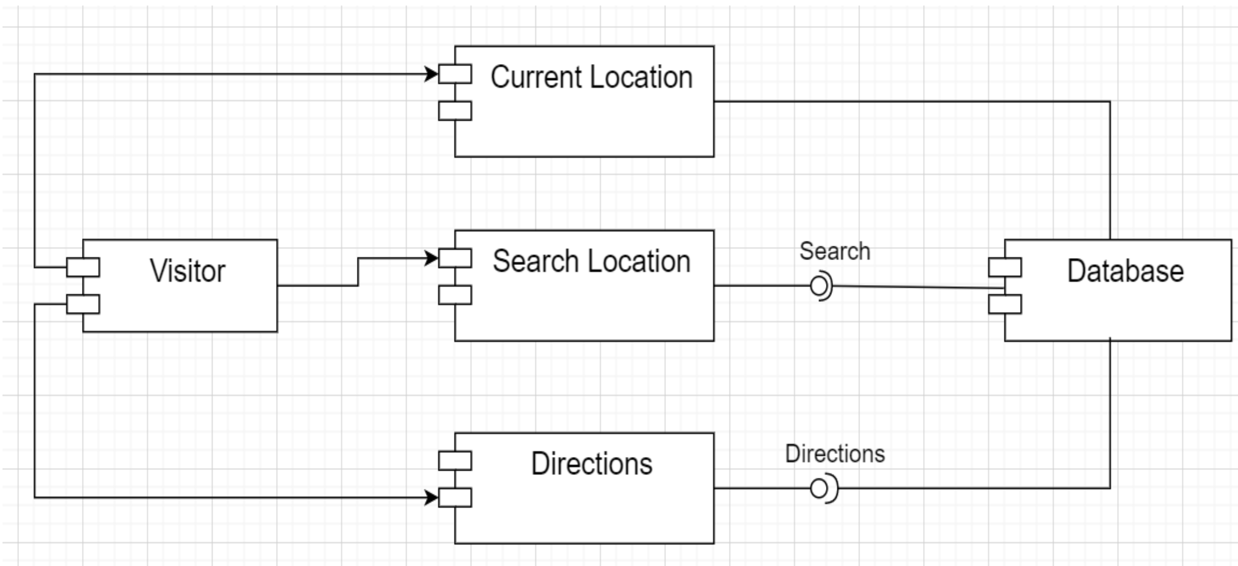


Figure 12
A component diagram depicts how components are wired together to form larger components or software systems.

3.4.8 DEPLOYMENT DIAGRAM

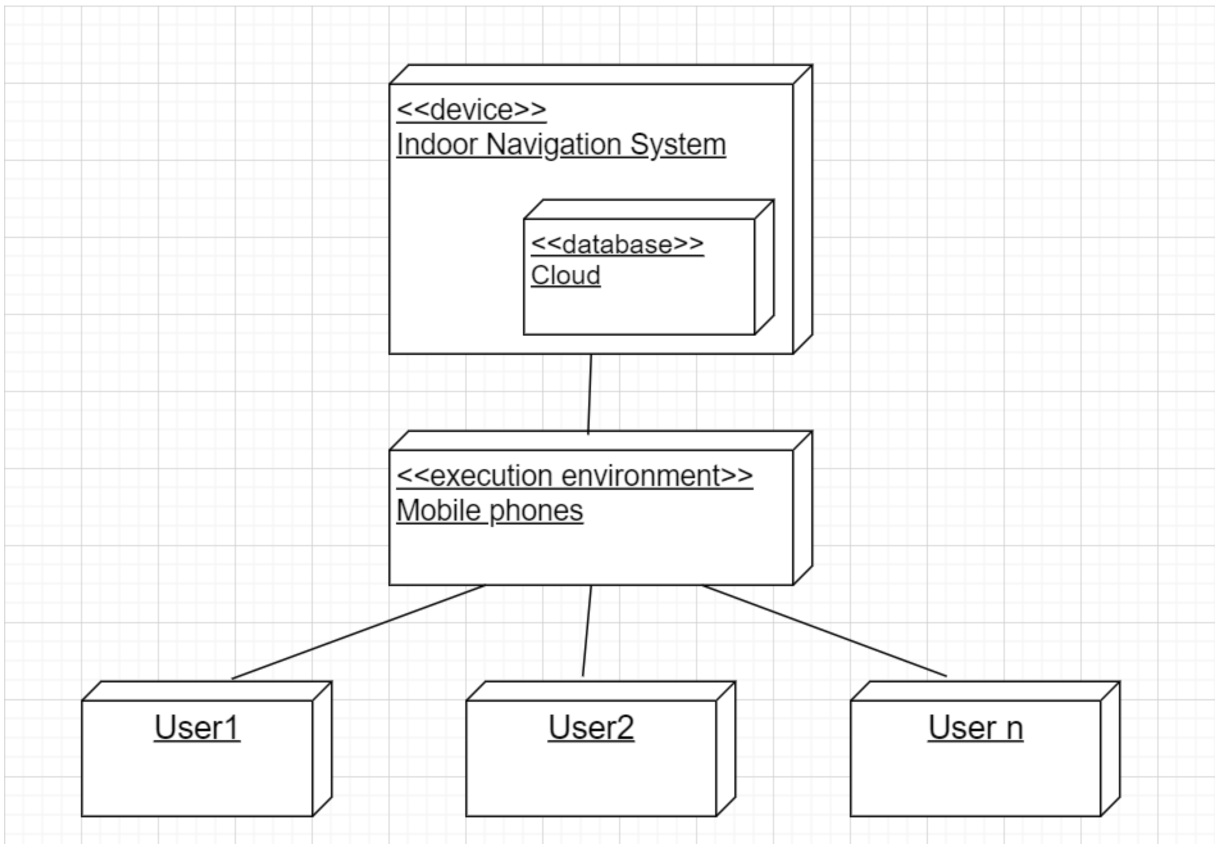


Figure 13

A deployment diagram models the physical deployment of artifacts on nodes.

3.4.9 PACKAGE DIAGRAM

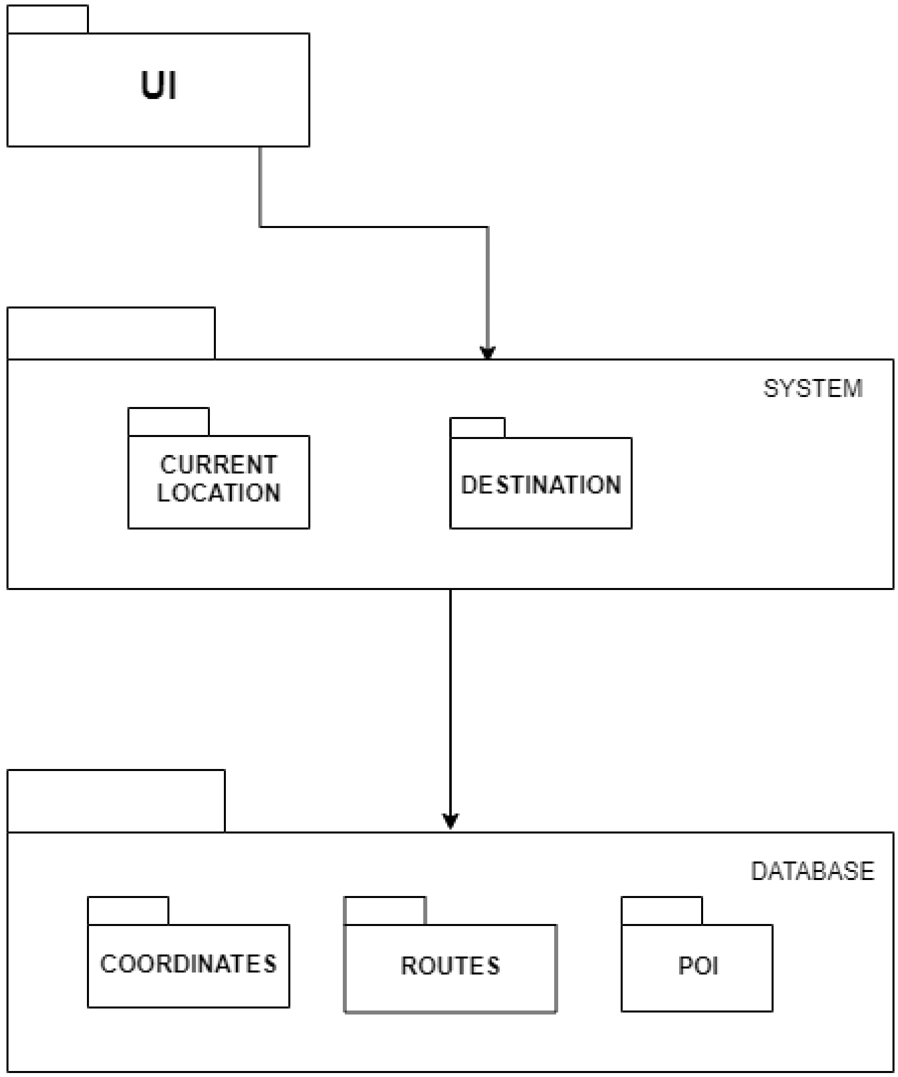


Figure 14
Package Diagram can be used to simplify complex class diagrams.

3.5 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) has 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.

4.1.1.3 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 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 of 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. Let 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 transportation

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

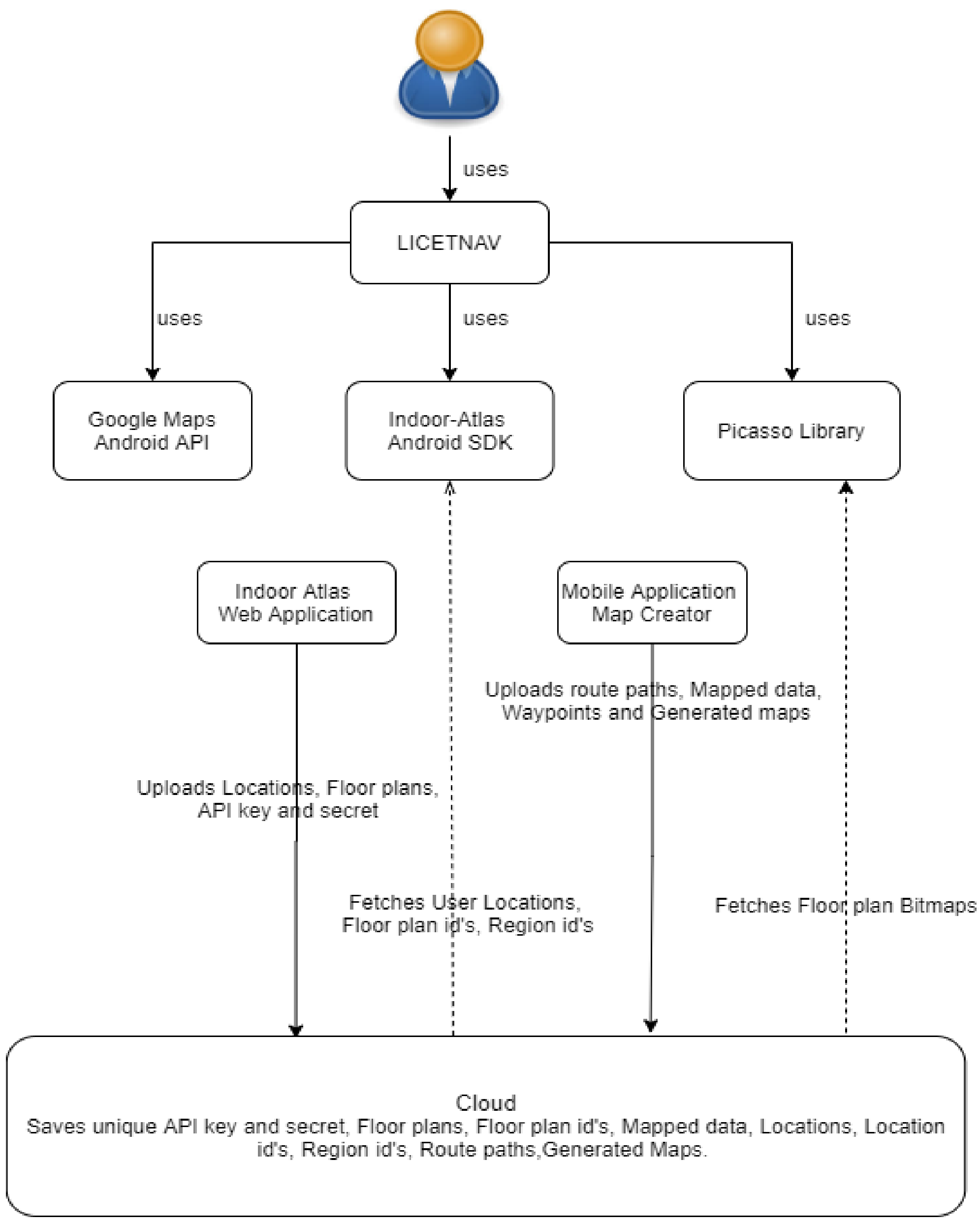


Figure 15 – System Architecture

HUFF-DUFF 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 HUFF-DUFF 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 how we locate the user inside a Building. Indoor Positioning Systems is a technology which works 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 smartphone. 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 Coordinates. These floor plans are saved in the cloud and can be fetched by the HUFF-DUFF 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.

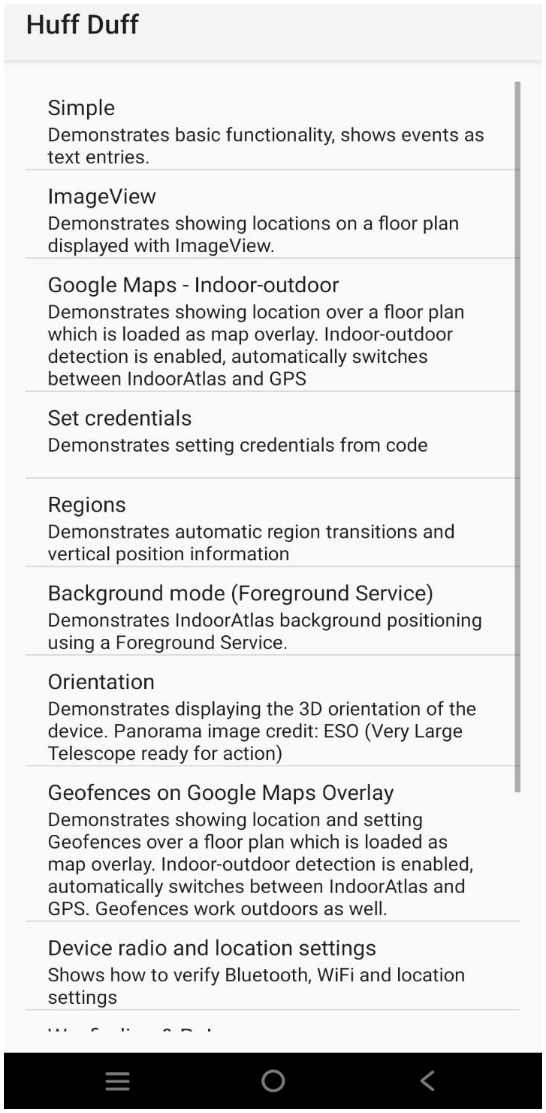
The main functions of Google Map API in this project are:

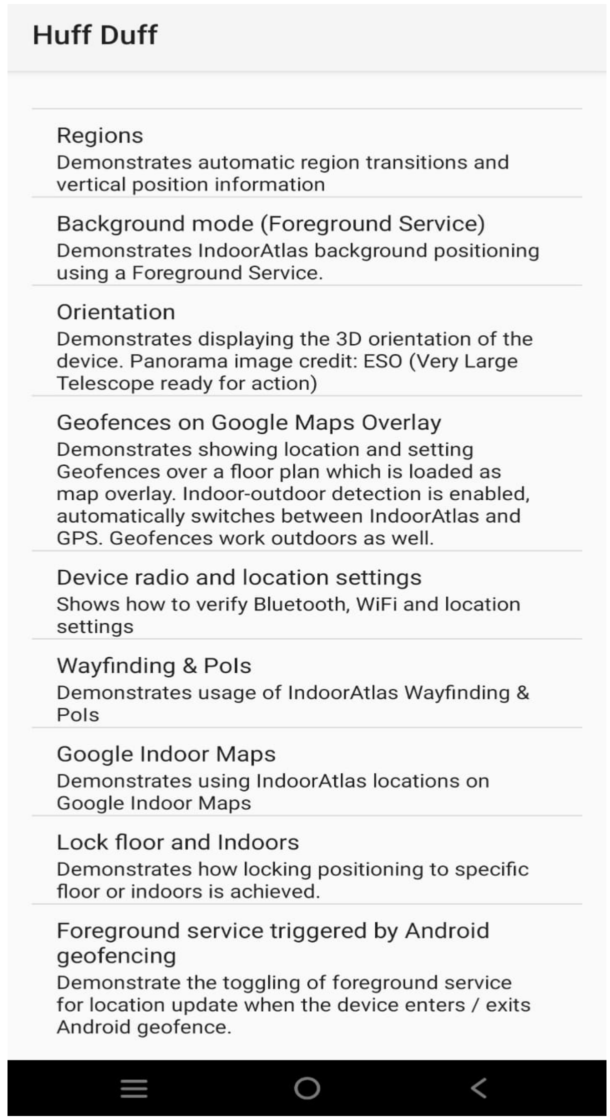
- Displaying maps on screen.
- Adding floor plans as Ground.

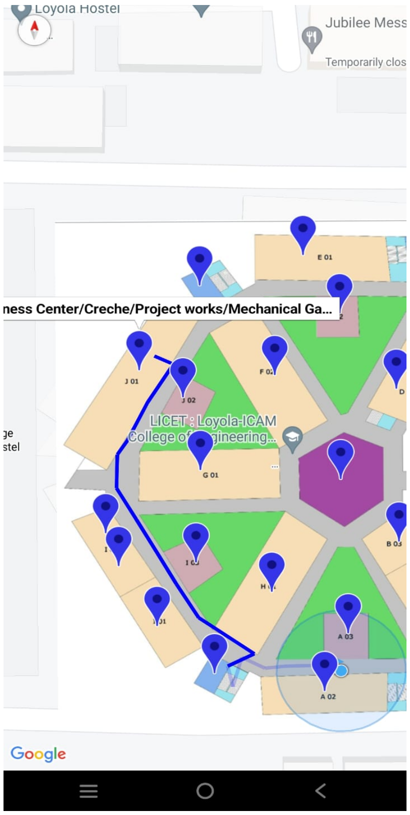
5.2.2 Adding Floor plans as overlay

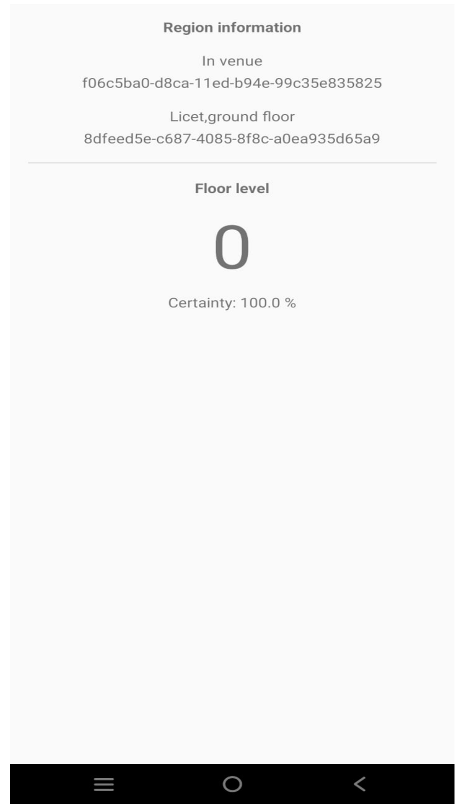
Google Maps API lets us add images on maps as overlays. This serves as an advantage to this project as we can fetch the floor plan from Cloud and add it as an overlay on Google Maps. To fetch floor plans ,we use Picasso library. Picasso allows us to easily download images from target URLs. To sum it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

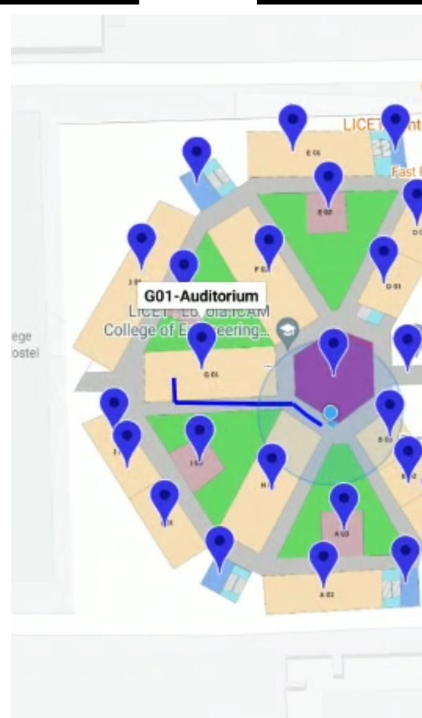
5.3 SCREENSHOTS











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 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

Activity-main.xml

```
<ListView xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@android:id/list"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".ListExamplesActivity"/>
```

Text-only.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin">

    <ScrollView
        android:id="@+id/scroller"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@id/buttons">

        <TextView
            android:id="@+id/text"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="top"
            android:textAppearance="?android:textAppearanceSmall"/>
    </ScrollView>

</FrameLayout>
```

MapsOverlayActivity.java

```
package com.example.mp.mapsoverlay;

import android.Manifest;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.drawable.Drawable;
import android.location.Location;
import android.location.LocationListener;
import android.os.Bundle;
import com.google.android.material.snackbar.Snackbar;
import android.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;
import android.util.Log;
import android.view.View;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptor;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.Circle;
import com.google.android.gms.maps.model.CircleOptions;
import com.google.android.gms.maps.model.GroundOverlay;
import com.google.android.gms.maps.model.GroundOverlayOptions;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.indooratlas.android.sdk.IALocation;
import com.indooratlas.android.sdk.IALocationListener;
import com.indooratlas.android.sdk.IALocationManager;
import com.indooratlas.android.sdk.IALocationRequest;
```

```
import com.indooratlas.android.sdk.IARegion;
import com.example.mp.R;
import com.example.mp.SdkExample;
import com.example.mp.utils.ExampleUtils;
import com.indooratlas.android.sdk.resources.IAFloorPlan;
import com.indooratlas.android.sdk.resources.IALatLng;
import com.indooratlas.android.sdk.resources.IALocationListenerSupport;
import com.squareup.picasso.Picasso;
import com.squareup.picasso.RequestCreator;
import com.squareup.picasso.Target;

@SdkExample(description = R.string.example_googlemaps_overlay_description)
public class MapsOverlayActivity extends FragmentActivity implements LocationListener,
OnMapReadyCallback {

    private static final String TAG = "IndoorAtlasExample";
    /* used to decide when bitmap should be downscaled */
    private static final int MAX_DIMENSION = 2048;
    private GoogleMap mMap; // Might be null if Google Play services APK is not available.
    private Circle mCircle;
    private Marker mMarker;
    private IARegion mOverlayFloorPlan = null;
    private GroundOverlay mGroundOverlay = null;
    private IALocationManager mIALocationManager;
    private boolean mCameraPositionNeedsUpdating = true; // update on first location
    private boolean mShowIndoorLocation = false;

    private void showBlueDot(LatLng center, double accuracyRadius, double bearing) {
        if (mCircle == null) {
            // location can received before map is initialized, ignoring those updates
            if (mMap != null) {
                mCircle = mMap.addCircle(new CircleOptions()
                        .center(center)
                        .radius(accuracyRadius)
                        .fillColor(0x201681FB)
                        .strokeColor(0x500A78DD)
                        .zIndex(1.0f)
                        .visible(true)
                        .strokeWidth(5.0f));
                mMarker = mMap.addMarker(new MarkerOptions()
                        .position(center)
                        .icon(BitmapDescriptorFactory.fromResource(R.drawable.map_blue_dot))
                        .anchor(0.5f, 0.5f)
```

```
                .rotation((float) bearing)
                .flat(true));
        }
    } else {
        // move existing markers position to received location
        mCircle.setCenter(center);
        mCircle.setRadius(accuracyRadius);
        mMarker.setPosition(center);
        mMarker.setRotation((float) bearing);
    }
}

/**
        @Override
    public void onLocationChanged(IALocation location) {

        Log.d(TAG, "new location received with coordinates: " + location.getLatitude()
                + "," + location.getLongitude());

        if (mMap == null) {
            // location received before map is initialized, ignoring update here
            return;
        }

        final LatLng center = new LatLng(location.getLatitude(), location.getLongitude());

        if (mShowIndoorLocation) {
            showBlueDot(center, location.getAccuracy(), location.getBearing());
        }

        // our camera position needs updating if location has significantly changed
        if (mCameraPositionNeedsUpdating) {
            mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(center, 17.5f));
            mCameraPositionNeedsUpdating = false;
        }
    }
};

/**
 * Listener that changes overlay if needed
 */
private IARegion.Listener mRegionListener = new IARegion.Listener() {
    @Override
    public void onEnterRegion(IARegion region) {
        if (region.getType() == IARegion.TYPE_FLOOR_PLAN) {
            final String newId = region.getId();
            // Are we entering a new floor plan or coming back the floor plan we just left?
            if (mGroundOverlay == null || !region.equals(mOverlayFloorPlan)) {
                mCameraPositionNeedsUpdating = true; // entering new fp, need to move camera
```

```
                if (mGroundOverlay != null) {
                    mGroundOverlay.remove();
                    mGroundOverlay = null;
                }
                mOverlayFloorPlan = region; // overlay will be this (unless error in loading)
                fetchFloorPlanBitmap(region.getFloorPlan());
            } else {
                mGroundOverlay.setTransparency(0.0f);
            }
            mShowIndoorLocation = true;
            showInfo("Showing IndoorAtlas SDK\'s location output");
        }
        showInfo("Enter " + (region.getType() == IARegion.TYPE_VENUE

                ? "VENUE "

                : "FLOOR_PLAN ") + region.getId());

    }

    @Override
    public void onExitRegion(IARegion region) {
        if (mGroundOverlay != null) {
            // Indicate we left this floor plan but leave it there for reference
            // If we enter another floor plan, this one will be removed and another one loaded
            mGroundOverlay.setTransparency(0.5f);
        }

        mShowIndoorLocation = false;
        showInfo("Exit " + (region.getType() == IARegion.TYPE_VENUE
                ? "VENUE "
                : "FLOOR_PLAN ") + region.getId());
    }

};

@Override
public void onLocationChanged(Location location) {
    if (!mShowIndoorLocation) {
        Log.d(TAG, "new LocationService location received with coordinates: " +
location.getLatitude()
                + "," + location.getLongitude());

        showBlueDot(
                new LatLng(location.getLatitude(), location.getLongitude()),
                location.getAccuracy(),
                location.getBearing());
    }
}
```

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

         // Try to obtain the map from the SupportMapFragment.
        ((SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map))
                .getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        // do not show Google's outdoor location
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        mMap.setMyLocationEnabled(false);
        // Setup long click to share the traceId
        mMap.setOnMapLongClickListener(new GoogleMap.OnMapLongClickListener() {
            @Override
            public void onMapLongClick(LatLng latLng) {
                ExampleUtils.shareText(MapsOverlayActivity.this,
                        mIALocationManager.getExtraInfo().traceId, "traceId");
            }
        });
    }

    /**
     * Sets bitmap of floor plan as ground overlay on Google Maps
     */
    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);
        }
    }

    /**
     * Download floor plan using Picasso library.
     */
    private void fetchFloorPlanBitmap(final IAFloorPlan floorPlan) {

        final String url = floorPlan.getUrl();
        // N/A
        Target mLoadTarget = new Target() {

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

            @Override
            public void onBitmapFailed(Exception e, Drawable errorDrawable) {
                showInfo("Failed to load bitmap");
                mOverlayFloorPlan = null;
            }

        final int bitmapWidth = floorPlan.getBitmapWidth();
        final int bitmapHeight = floorPlan.getBitmapHeight();
      Picasso.get().load("https://ida-fp-images-prod.s3.eu-west-1.amazonaws.com/82082bd5-4a8d-4
      563-9b1f-4d8acbd59296/c363d7e7-8fa6-44e2-bb9f-f080e5ba28c6").into(mLoadTarget);
    }

    private void showInfo(String text) {
        final Snackbar snackbar = Snackbar.make(findViewById(android.R.id.content), text,
                Snackbar.LENGTH_INDEFINITE);
        snackbar.setAction(R.string.button_close, new View.OnClickListener() {
```

```
            @Override
            public void onClick(View view) {
                snackbar.dismiss();
            }
        });
        snackbar.show();
    }
}
```

ImageViewActivity.java

```
package com.example.mp.imageview;

import android.graphics.PointF;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import androidx.fragment.app.FragmentActivity;
import android.util.Log;
import android.widget.Toast;
import com.davemorrissey.labs.subscaleview.ImageSource;
import com.indooratlas.android.sdk.IALocation;
import com.indooratlas.android.sdk.IALocationListener;
import com.indooratlas.android.sdk.IALocationManager;
import com.indooratlas.android.sdk.IALocationRequest;
import com.indooratlas.android.sdk.IAOrientationListener;
import com.indooratlas.android.sdk.IAOrientationRequest;
import com.indooratlas.android.sdk.IARegion;
import com.example.mp.R;
import com.example.mp.SdkExample;
import com.example.mp.utils.ExampleUtils;
import com.indooratlas.android.sdk.resources.IAFloorPlan;
import com.indooratlas.android.sdk.resources.IALatLng;
import com.indooratlas.android.sdk.resources.IALocationListenerSupport;
import com.squareup.picasso.Picasso;
import com.squareup.picasso.RequestCreator;
import com.squareup.picasso.Target;


@SdkExample(description = R.string.example_imageview_description)
public class ImageViewActivity extends FragmentActivity {
```

```
private static final String TAG = "IndoorAtlasExample";
// blue dot radius in meters
private static final float dotRadius = 1.0f;
private IALocationManager mIALocationManager;
private IAFloorPlan mFloorPlan;
private BlueDotView mImageView;
private Target mLoadTarget;
private IALocationListener mLocationListener = new IALocationListenerSupport() {
   @Override
   public void onLocationChanged(IALocation location) {
        Log.d(TAG, "location is: " + location.getLatitude() + "," +
       location.getLongitude());
      if (mImageView != null && mImageView.isReady()) {
           IALatLng latLng = new IALatLng(location.getLatitude(),
           location.getLongitude());
         PointF point = mFloorPlan.coordinateToPoint(latLng);
         mImageView.setDotCenter(point);
         mImageView.setUncertaintyRadius(
              mFloorPlan.getMetersToPixels() * location.getAccuracy());
         mImageView.postInvalidate();
      }
   }
};

private IAOrientationListener mOrientationListener = new IAOrientationListener() {
   @Override
   public void onHeadingChanged(long timestamp, double heading) {
      if (mFloorPlan != null) {
         mImageView.setHeading(heading - mFloorPlan.getBearing());
      }
   }
};

private final IARegion.Listener mRegionListener = new IARegion.Listener() {

   @Override
   public void onEnterRegion(IARegion region) {
```

```
        if (region.getType() == IARegion.TYPE_FLOOR_PLAN) {
            String id = region.getId();
            Log.d(TAG, "floorPlan changed to " + id);
            Toast.makeText(ImageViewActivity.this, id, Toast.LENGTH_SHORT).show();
            fetchFloorPlanBitmap(region.getFloorPlan());
        }
    }

    @Override
    public void onExitRegion(IARegion region) {
        // leaving a previously entered region
    }

};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_image_view);
    // prevent the screen going to sleep while app is on foreground
    findViewById(android.R.id.content).setKeepScreenOn(true);

    mImageView = findViewById(R.id.imageView);

    mIALocationManager = IALocationManager.create(this);

    // Setup long click listener for sharing traceId
    ExampleUtils.shareTraceId(findViewById(R.id.imageView), ImageViewActivity.this,
            mIALocationManager);

}
@Override
protected void onPause() {
    super.onPause();
    mIALocationManager.removeLocationUpdates(mLocationListener);
    mIALocationManager.unregisterRegionListener(mRegionListener);
    mIALocationManager.unregisterOrientationListener(mOrientationListener);
}

/**
 * Methods for fetching bitmap image.
 */

private void showFloorPlanImage(Bitmap bitmap) {
```

```
        mImageView.setDotRadius(mFloorPlan.getMetersToPixels() * dotRadius);
        mImageView.setImage(ImageSource.bitmap(bitmap));
    }

    /**
     * Download floor plan using Picasso library.
     */
    private void fetchFloorPlanBitmap(final IAFloorPlan floorPlan) {
        mFloorPlan = floorPlan;
        final String url = floorPlan.getUrl();
        mLoadTarget = new Target() {

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

            @Override
            public void onBitmapFailed(Exception e, Drawable errorDrawable) {
                Toast.makeText(ImageViewActivity.this, "Failed to load bitmap",
Toast.LENGTH_SHORT).show();
                mFloorPlan = null;
            }

            @Override
            public void onPrepareLoad(Drawable placeHolderDrawable) {
                // N/A
            }
        };

        Picasso.get().load(url).into(mLoadTarget);
    }
```

REFERENCE

[1] Jan Hurtuk, Jakub Cervena, Martin Stancel, Michal Hulic, Peter Feciak, “Indoor navigation using IndoorAtlas library”, *IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY),2019*

[2] Jayakanth Kunhoth, AbdelGhani Karkar, Somaya Al-Maadeed & Abdulla Al-Ali , “Indoor positioning and wayfinding systems: a survey”, *Journal of Springeropen ,2020*

[3] Saeed Ahmed Magsi; Nordin Saad; Mohd Haris Bin Md Khir; Gunawan Witjaksono; Muham Aadil Siddiqui,Linta Sameer, “Wi-Fi Based Indoor Navigation System For Campus Directions”,8th International Conference on Intelligent and Advanced Systems (ICIAS),2018

[4] Saeed Ahmed Magsi; Nordin Saad; Mohd Haris Bin Md Khir; Gunawan Witjaksono; Muham Aadil Siddiqui,Linta Sameer, “Evolution of Indoor Positioning Technologies: A Survey”, Journal of Sensors,2017

[5] Dawn Griffiths and David Griffiths, ”Head First Android Development”, 2nd edition,pp 339-380,2017