INDOOR NAVIGATION SYSTEM USING AUGMENTED REALITY

Submitted in partial fulfillment of the requirements
for the degree of

B.E. Information Technology

By

Mr. Dikshant Jopat 43

Mr. Krutarth Makwana 41

Mr. Hardik Dhanmeher 39

Supervisor
Ms. Shraddha More
Assistant Professor

Department of Information Technology

St. John College of Engineering and Management

Vevoor Road, Palghar(E),401404

UNIVERSITY OF MUMBAI

2019-2020

CERTIFICATE

This is to certify that the project entitled **“Indoor Navigation System Using Augmented Reality”** is a bonafide work of **Dikshant Jopat (PID No. 1164011), Krutarth Makwana (PID No. 1164041) and Hardik Dhanmeher (PID No. 1164012)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of B.E. in Information Technology.

Ms. Joyce Lemos
Project Guide

Dr. Terence Johnsoon
Head of Department

Dr. G.V. Mulgund
Principal

Project Report Approval for B. E.

The project report entitled **Indoor Navigation System Using Augmented Reality** by **Dikshant Jopat, Krutarth Makwana, Hardik Dhanmeher** is approved for the degree of **Bachelor of Engineering** in **Information Technology.**

Examiners:

1.--

2.--

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Dikshant Jopat(1164011)

Krutarth Makwana (1164041)

Hardik Dhanmeher(1164012)

Date:

Abstract

We present an overall framework of services for indoor navigation, which includes Indoor Mapping, Indoor Positioning, Path Planning, and En-route Assistance. Within such framework we focus on an augmented reality (AR) solution for en-route assistance. AR assists the user walking in a multi-floor building by displaying a directional arrow under a camera view, thus freeing the user from knowing his/her position. Our AR solution relies on geomagnetic positioning and north-oriented space coordinates transformation. Therefore, it can work without infrastructure and without relying on GPS. The AR visual interface and the integration with magnetic positioning is the main novelty of our solution. Personal locations recognized by the proposed method in real-time. The results are overplayed on the user's view through AR technique. The ideas presented in this system are applicable to a navigation system.

Contents

List of Figures

List of tables

Chapter 1

Introduction

1.1 Introduction

Augmented reality combines real and computer-based scenes and images to deliver a unified but enhanced view of the world. Augmented Reality (AR) is a technology which utilizes computer visualization to superimpose virtual objects on real world images for user interaction. Its mode of operation is to calculate the position and angle of a video device through real-time positioning or image processing, and then superimpose a virtual model and information on the image of the real world. Its objective is to present virtual objects in the scene of the real world via a video device and to allow users to interact with it.

Hardware components for augmented reality are: processor, display, sensors and input devices. Modern mobile computing devices like smartphones and tablet computers contain these elements which often include a camera and MEMS sensors such as accelerometer, GPS, and solid state compass, making them suitable AR platforms. Augmentation techniques are typically performed in real time and in semantic context with environmental elements. Moreover, navigation is not about finding areas and paths, but also guiding and monitoring the movement of the user.

1.2 Problem Formulation

Most of the time we spend our time usually indoors. We hang out with our friends in Malls and various other buildings like University, Offices, Museums, Cinemas, Library, Airport, Railway Station and Supermarket. If we are not familiar with those buildings we get confused and get lost in those huge buildings. For navigation inside these buildings we don’t have any system or service with us. GPS can’t work for navigating inside the buildings. The purpose is to use all means of available information to make an as accurate estimation of the position of the phone as possible. The focus of the project is in estimating the position of the phone inside a building where the GPS signal is bad or unavailable.

1.3 Description

The aim of this project is to examine the level of accuracy that can be achieved in precision positioning by using built-in sensors in an Android smartphone. The main problem to be solved is the implementation of a system realized from the sensor fusion of the Android inertial sensors, such as the accelerometer, magnetometer, and the gyroscope. The system will guide the user by directed arrow to user's destination. Universities around the world are expanding structurally at a rapid rate especially when the number of students gets greater and greater with every enrollment period.

1.4 Motivation

In the modern world, there is tremendous increase in the new college campus then malls etc. But due to shortage of time, the work takes more time to find the place between them because we do not know about that places. This time shortage can be overcome with the help of Indoor navigation, to solve all the problems. Indoor navigation can help to find place more efficiently than human beings because they are so fast and precise. This app can used by the user at anytime when he want to used. The aim of the project is to reduce the time to finding the exact place into the unknown places and reach the user at there exact place.

1.5 Proposed Solution

Indoor navigation can help to find place more efficiently than human beings because they are so fast and precise. This app can used by the user at anytime when he want to used. The aim of the project is to reduce the time to finding the exact place into the unknown places and reach the user at there exact place.

1.6 Scope of the project

Augmented reality combines real and computer-based scenes and images to deliver a unified but enhanced view of the world. Augmented Reality (AR) is a technology which utilizes computer visualization to superimpose virtual objects on real world images for user interaction. Its mode of operation is to calculate the position and angle of a video device through real-time positioning or image processing, and then superimpose a virtual model and information on the image of the real world.

Chapter 2

Literature Review

In this section, we will look at several similar systems that are been researched and implemented by other researchers. for further understanding on their methods and techniques, refer to the references page at the end of this report.

2.1. Smartphone Indoor Localization with Accelerometer and Gyroscope

In this research paper, they propose to develop a smartphone indoor positioning application based on accelerometer and gyroscope data [1]. The PDR (Pedestrian Dead Reckoning) method is they used to build this application. Calibration points are marked both on the floor ground and on the map of the application. The user first finds a calibration mark, stand on it and face the right direction. He/she then place the android icon (representing the user) on top of the calibration point. When he/she starts to move, the android icon also moves on the map following the real-time estimates of step length and orientation change for each step from accelerometer and gyroscope data, respectively. Preliminary results in walking distance and orientation estimation show high accuracy. This application seems promising and useful as long as a map and calibration marks are built in advance.

2.2. Real-time Indoor Navigation using Smartphone Sensors

This paper presents an indoor navigation algorithm that uses multiple kinds of sensors and technologies, such as MEMS sensors (i.e., gyros, accelerometers, magnetometers, and a barometer), Wi-Fi, and magnetic matching [2]. The corresponding real-time software on smartphones includes modules such dead-reckoning, Wi-Fi positioning, and magnetic matching. DR is used for providing continuous position solutions and for the blunder detection of both Wi-Fi fingerprinting and magnetic matching. Finally, Wi-Fi and magnetic matching results are passed into the position-tracking module as updates. Meanwhile, a barometer is used to detect floor changes, so as to switch floors and the Wi-Fi and magnetic databases.

2.3. Multiple Sensors Integration for Pedestrian Indoor Navigation

This paper discusses the fusion schemes of multiple sensors including accelerometer, gyro, magnetometer and pressure sensor for improved indoor positioning accuracy [3]. The main focus is to make fully use of complementary features in sensors (such as gyro and magnetometer) and navigation structure (such as PDR and strapdown INS). A centralized Kalman filter is designed and implemented for this integration. Tests are conducted at indoor without using external wireless signal aiding. Positioning performance is evaluated among different integration strategies by different sensors and different complementary features.

2.4 Accurate and Reliable Real-Time Indoor Positioning on Commercial Smartphones

This paper outlines the software navigation engine that was developed by SPIRIT Navigation for indoor positioning on commercial smartphones [4]. A distinctive feature of our approach is concurrent use of multiple technologies for indoor positioning. Measurements from such smartphone sensors as IMU (3D accelerometer, gyroscope), a magnetic field sensor (magnetometer), Wi-Fi and BLE modules, together with the floor premises plan are used for hybrid indoor positioning in the navigation engine. Indoor navigation software uses such technologies as PDR, Wi-Fi fingerprinting, geomagnetic fingerprinting, and map matching.

2.5 Improved heading estimation for smartphone-based indoor positioning systems

In this paper, they present an indoor positioning scheme using recently introduced smartphones equipped with sensors such as accelerometer, magnetometer, and gyroscope [5]. They propose a practical indoor positioning method to handle complicated human movements and noisy inertial sensors in smartphones. They focus on providing indoor locations with accuracy by applying improved heading estimation solution. The key concept is to reduce errors caused by directional change since the critical component of positioning system is a heading orientation

Chapter 3

System Design

3.1 Functional Requirements

Start Camera:- accessing the app through the camera. The camera in work can be back camera as well as front camera according to the convenience of the user.

Detect Object:-The camera will point towards the object and detect it(the user doesn't have to take the picture, the picture will be taken automatically for the history log

Gather Information :- After detecting the object, all the related information should be loaded. Details of information is given in next section.

Collect Sensor Data:- When the object is getting detected, the mobile will store the sensory data from the mobile (sensors like gyroscope and accelerometer)

Create AR objects :- Based on the object detected it will create an AR object and will show on the screen

Place AR objects :- According to the collected sensor data, the object is shown besides the object after doing following check accordingly.

Check Obstruction (sub function) :- The information to be shown should not obstruct the user's view. If it does, the way of displaying information should adapt to the view.

Free View Mode :- Drag drop objects and information with added feature that - the user can rotate the object to see its 360 degree view and also the position of the AR object can be changed on the screen and can be zoomed to see the details of the object.

3.2 Non-Functional Requirements

1. Switch between the front camera and back camera
2. Auto rotation feature
3. List of Artifacts already covered
4. History log
5. Navigation system with footprints AR

3.3 Specific Requirements:

The very first phase in any system developing life cycle is preliminary investigation. The feasibility study is major part of this phase. A measure of how beneficial or practical the development of the system would be to the organization is feasibility study.

3.3.1 Hardware Requirements:

- iOS Mobile

3.3.2 Software Requirements:

- Visual Basic
- Unity 3D
- Android SDK
- AR Core SDK

3.4 Use-Case Diagram and Description

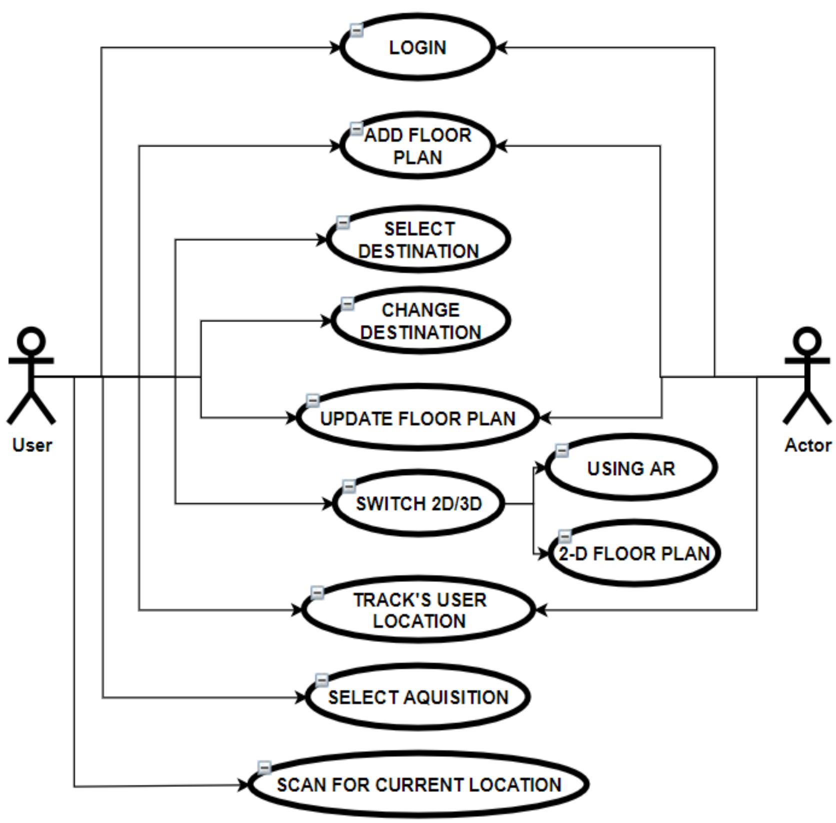

Fig 3.4 Use Case Diagram

Chapter 4

Analysis Modeling

4.1 Activity Diagram

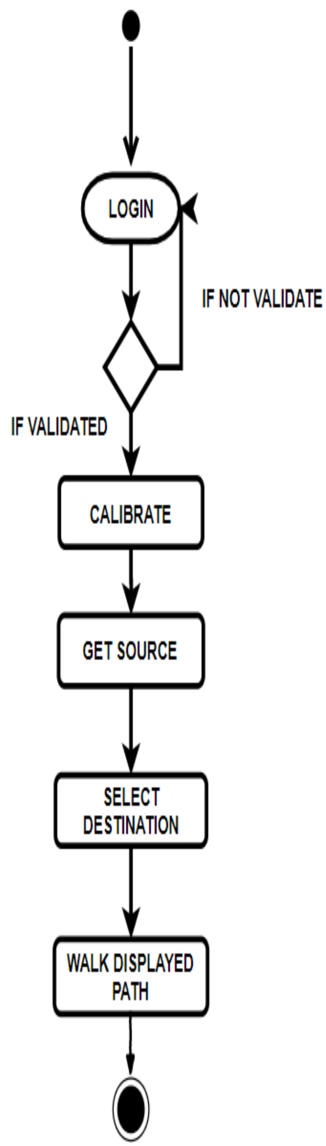

Fig 4.1 Activity Diagram

4.2 Sequence Diagram

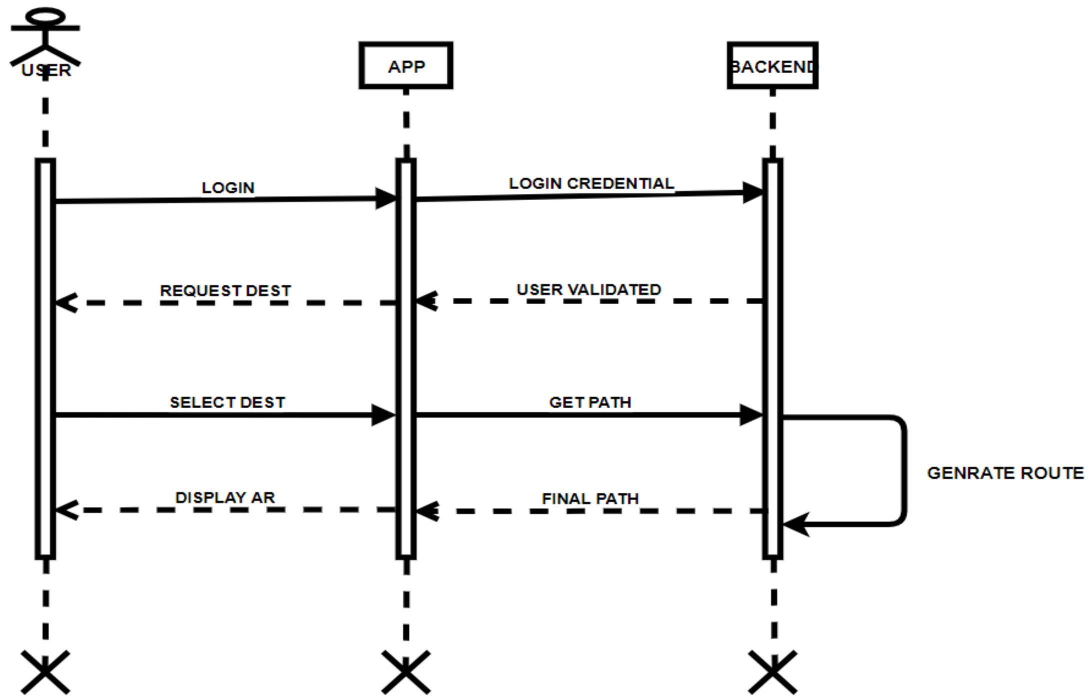

Fig 4.2 Sequence Diagram

4.3 Functional Modeling

4.3.1 DFD level 0

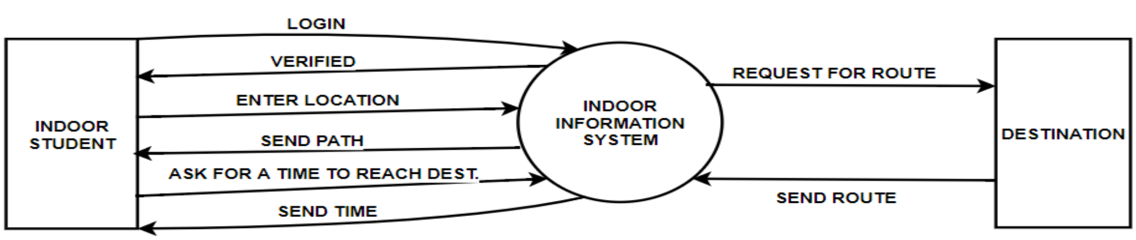

Fig 4.3.1 DFD Level 0

4.3.2 DFD level 1

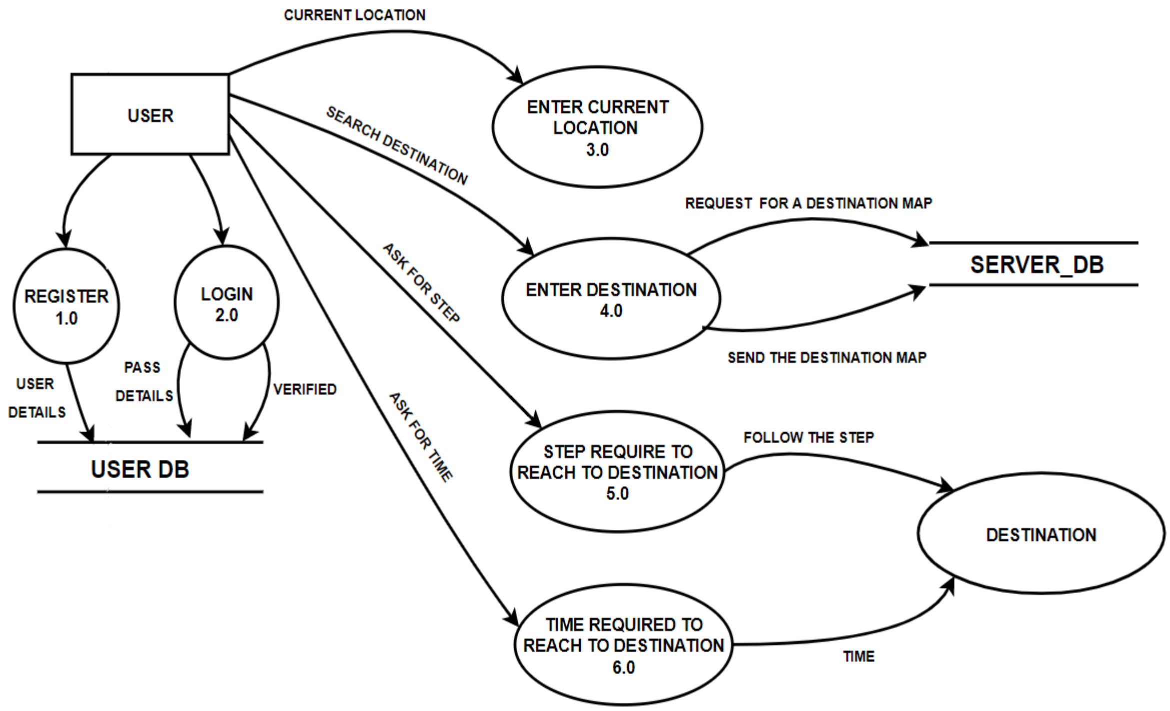

Fig 4.3.2 DFD Level 1

4.3.3 DFD level 2

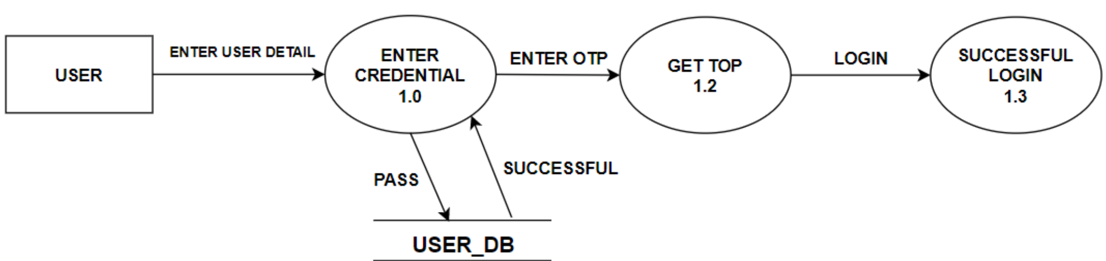

Fig 4.3.3 DFD Level 2

4.4 Time Line Chart

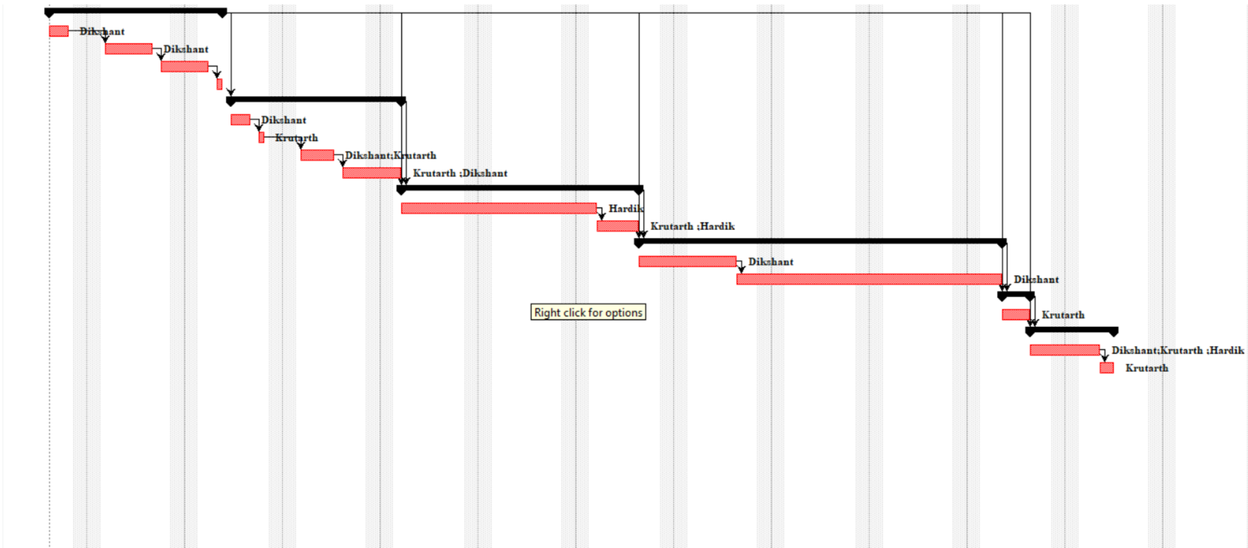

Fig 4.4 Time Line Chart

4.5 WBS Chart

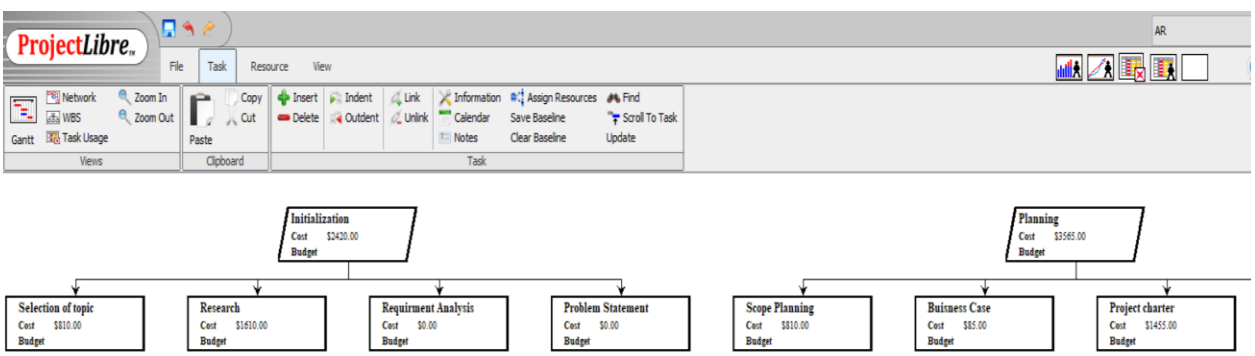

Fig 4.5(a) W.B.S Chart

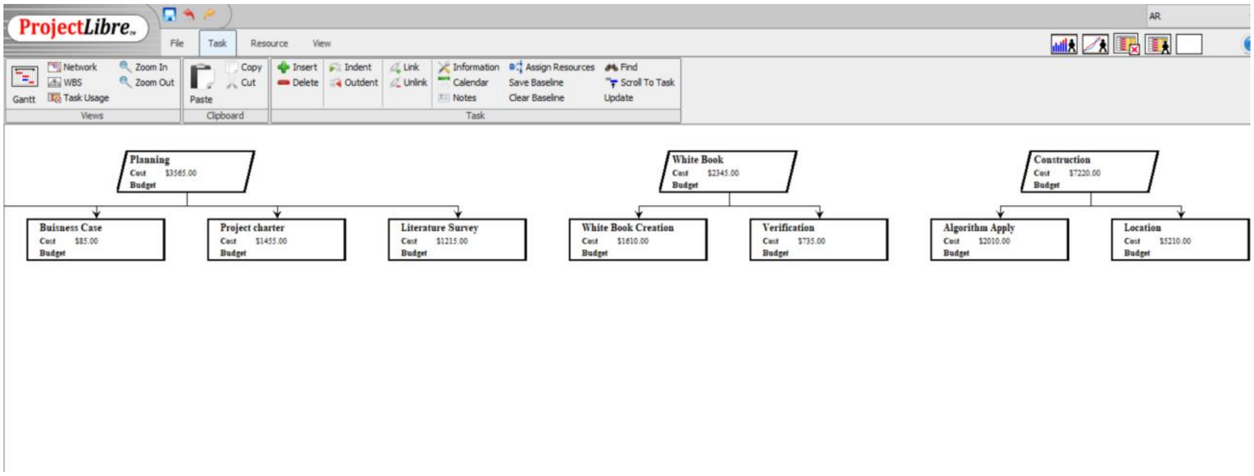

Fig 4.5(b) W.B.S Chart

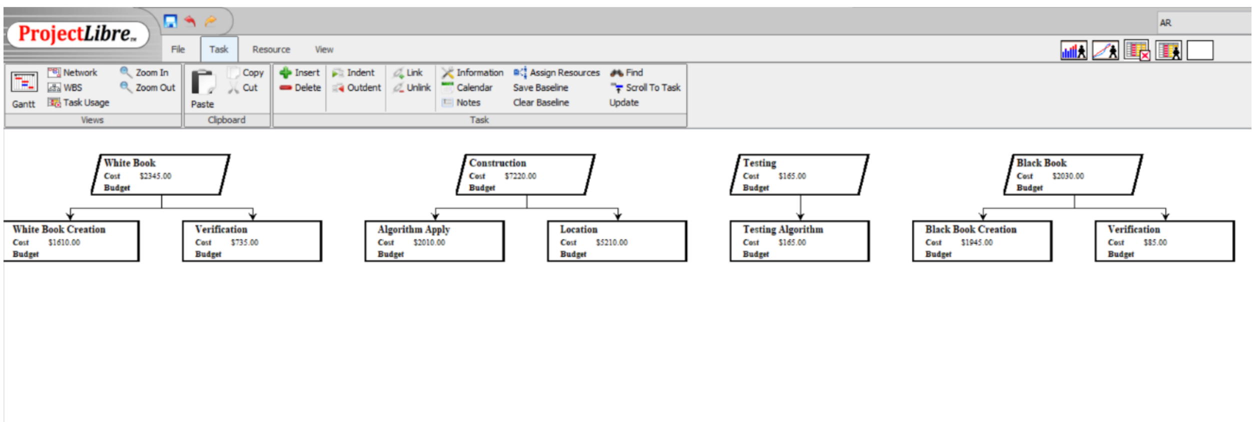

Fig 4.5(c) W.B.S Chart

Chapter 5

Proposed Systems

5.1 Scope

1. It is an application for smart phones that works on Android Operating System.
2. This application has a real time feedback by providing lots of information related to visual, location, and heading and acceleration data.
3. This application will help the users to navigate inside a building were GPS service is not available.
4. The application has various feature like voice assistance, path finding, locating person etc.

5.2 Architecture Diagram

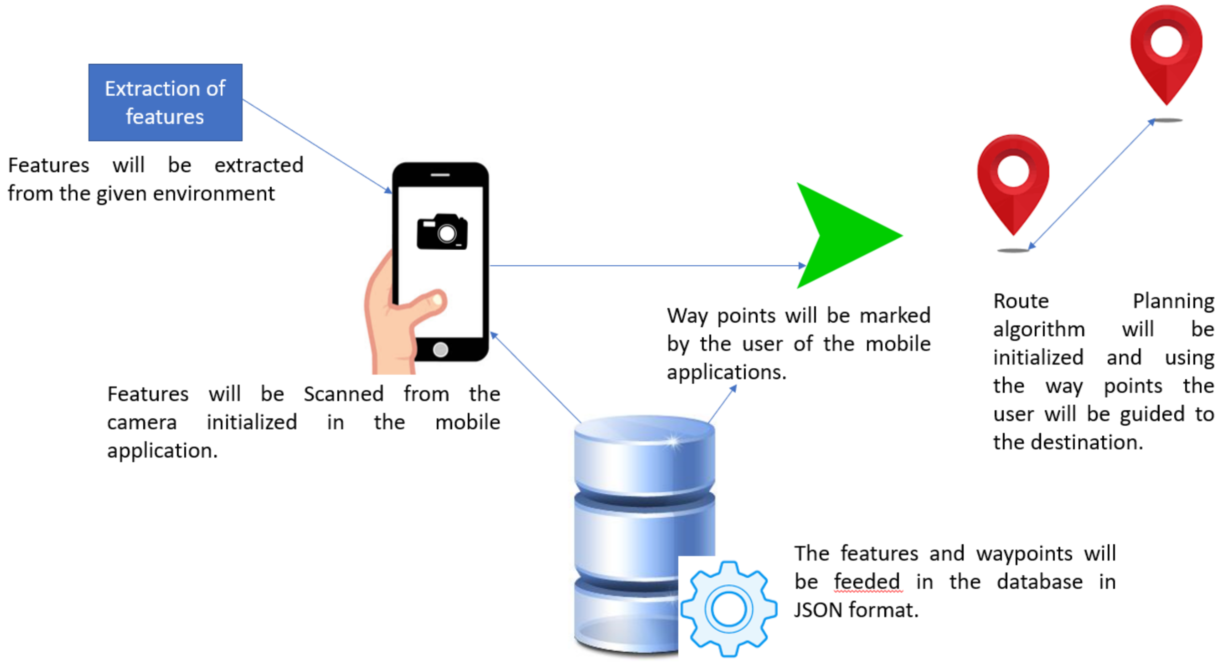

Fig 5.1 System Architecture

DESCRIPTION:

The proposed system is divided into following phases:

Phase 1:-

Mapping Phase: - In this phase, the real-time mapping of the route occurs; we map the features present on the route and place them in a JSON format in the database.

Below given is the implementation level milestones for this phase: -

A. Extraction of features

B. The initialization of the camera for real-time scanning of objects

C. Features will be uploaded in the database in JSON format.

D. Marking of waypoints .

Phase 2:-

Testing Phase: - In this phase, we will take the user input from the user for the destination, and then the route planning algorithm will map the most suitable route from the user's current position to the destination. Below given is the implementation level milestones for this phase: - (a)User input will be taken. (b)The camera will be initialized for showing the route to the users. (c)The route planning algorithm will find the best route possible from the current position. (d)The marked waypoints will be shown on the mobile screen using arrows.

So this is how our proposed system works:

A. Firstly, the user will open an application based on AR.

B. This app will intimate the user either to create his own map/route or will ask him/her if the user needs assistance with navigation.

C. If the user clicks on the create map, then the user automatically will be asked the name of the source and destination and will be asked to start his/her camera.

D. After the camera is initialized, the application using ARCore will start extracting features from the Indoor Navigation System Using Augmented Reality camera input, and the user will be asked for marking waypoints at the place where feature extraction has been completed.

E. This will complete the mapping phase.

F. If the user clicks on „assist me" for indoor navigation, the user will be next asked for a destination.

G. Once the destination has been taken, the camera will be initialized, and features from the current location of the user will be extracted.

H. These features will be matched with the various features already present in the database, and if a match is found, the system will get the user's current location.

I. After this, the system will start the route planning algorithm and will find the best route from various routes that have already been mapped.

J. If no route is to be found to the destination, the user will get a prompt message that either the entered destination is wrong or no route has been found for the entered destination.

K. This route will lead the user to the destination.

Chapter 6

Technologies Used

6.1 GPS For Indoor Navigation:

GPS is one of the existing technologies in the navigation space; however, it cannot provide accurate positioning inside a building. It's relatively accurate in large and low-rise buildings, such as airports; but it can't determine finer details such as the floor number, and the only way to achieve this is to do it manually, like the way it's done with Apple & Google Maps. The takeaway here is GPS works but it's not a viable solution.

6.2 Visual Positioning Syatem (VPS):

Visual Positioning System (VPS) – is advanced compared to the two and holds so much promise. Google, for example, uses Street View data to clarify a user's position in AR-based Outdoor Navigation, using surrounding buildings as reference points.

Furthermore, ARKit 2 introduced the ARWorld Map class, which can also serve a similar purpose. The ARWorld Map is essentially a set of feature points around a user, like "the world's fingerprint", which can be recognized.

6.3 Beacon Based Indoor Navigation:

Beacon technology is another buzzword when it comes to indoor navigation. The most common example is the 2,000 battery-powered Bluetooth Low Energy beacons installed at Gatwick airport in the UK which gives, as they claim, +/-3m accuracy. But according to Apple documentation, beacons give only an approximate distance value hence we can't rely on the signal strength while trying to calculate this data manually.

And when you consider other factors such as cost ($10-20 per item), battery replacement (once every 1-2 years) and the working range (10-100 metres), it's clear that the use of beacons for indoor navigation is effective only under certain conditions.

But all in all, beacons can still do a good job as most of the current indoor navigation technologies on the market are beacon-based and use AR for route visualization (only). However it's possible to leverage ARKit and ARCore SDKs' features to solve the problem of user positioning.

6.4 SLAM:

SLAM (Simultaneous Localization and Mapping) is the most effective way to render virtual images over real-world objects. SLAM simultaneously localizes sensors with respect to their surroundings, while at the same time mapping the structure of the environment.

SLAM is an approach to solve complex AR simulation problems and is not any specific algorithm or software. The SLAM system is, in fact, a set of algorithms aimed at solving simultaneous localization and mapping problem. This can be done in multiple ways and now every augmented reality development kit has its upon to providing SLAM functionality.

6.5 Wi-Fi Positioning System (WPS):

Wi-Fi positioning system (WPS), WiPS or WFPS is a geolocation system that uses the characteristics of nearby Wi-Fi hotspots and other wireless access points to discover where a device is located. It is used where satellite navigation such as GPS is inadequate due to various causes including multipath and signal blockage indoors, or where acquiring a satellite fix would take too long. Such systems include assisted GPS, urban positioning services through hotspot databases, and indoor positioning systems. Wi-Fi positioning takes advantage of the rapid growth in the early 21st century of wireless access points in urban areas.

6.6 Radio-Frequency Identification (RFID):

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID tag consists of a tiny radio transponder; a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader. This number can be used to inventory goods. There are two types. Passive tags are powered by energy from the RFID reader's interrogating radio waves. Active tags are powered by a battery and thus can be read at a greater range from the RFID reader; up to hundreds of meters. Unlike a barcode, the tag doesn't need to be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method of automatic identification and data capture.

Chapter 7

Implementation

7.1 Algorithms used:-

A * Algorithm:

A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.

Informally speaking, A* Search algorithms, unlike other traversal techniques, it has “brains”. What it means is that it is really a smart algorithm which separates it from the other conventional algorithms.And it is also worth mentioning that many games and web-based maps use this algorithm to find the shortest path very efficiently (approximation).

Code:-

```
using System.Collections.Generic;
using UnityEngine;

public class AStar : MonoBehaviour {

   public List<Node> FindPath(Node startNode, Node targetNode, Node[] allNodes) {
      List<Node> openSet = new List<Node>();
      openSet.Add(startNode);

      List<Node> closedSet = new List<Node>();

      while (openSet.Count > 0) {
         Node currentNode = openSet[0];
         for (int i = 1; i < openSet.Count; i++) {
            if (openSet[i].FCost < currentNode.FCost
               || (openSet[i].FCost.Equals(currentNode.FCost)
                  && openSet[i].HCost < currentNode.HCost)) {
               currentNode = openSet[i];
            }
         }

         openSet.Remove(currentNode);
         closedSet.Add(currentNode);

         if (currentNode == targetNode) {
            Debug.Log("RETURNING CORRECT NODE!!");
            return RetracePath(startNode, targetNode);
         }

         foreach (Node connection in currentNode.neighbors) {
```

```
            if (!closedSet.Contains(connection)) {
                float costToConnection = currentNode.GCost + GetEstimate(currentNode,
connection) + connection.Cost;

                if (costToConnection < connection.GCost || !openSet.Contains(connection)) {
                    connection.GCost = costToConnection;
                    connection.HCost = GetEstimate(connection, targetNode);
                    connection.Parent = currentNode;

                    if (!openSet.Contains(connection)) {
                        openSet.Add(connection);
                    }
                }
            }
        }
    }
    Debug.Log("RETURNING NULL");
    return null;
}

private static List<Node> RetracePath(Node startNode, Node endNode) {
    List<Node> path = new List<Node>();

    Node currentNode = endNode;

    while (currentNode != startNode) {
        path.Add(currentNode);
        currentNode = currentNode.Parent;
    }

    path.Reverse();

    return path;
}

private float GetEstimate(Node first, Node second) {
    float distance;

    float xDistance = Mathf.Abs(first.pos.x - first.pos.x);
    float yDistance = Mathf.Abs(second.pos.z - second.pos.z);

    if (xDistance > yDistance) {
        distance = 14 * yDistance + 10 * (xDistance - yDistance);
    } else {
        distance = 14 * xDistance + 10 * (yDistance - xDistance);
    }
```

```
        return distance;
    }
}
```

7.2 Code:-

7.2.1 FOR CREATE MAP:-

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using UnityEngine.UI;
using UnityEngine.XR.iOS;
using System.Runtime.InteropServices;
using System.IO;
using Newtonsoft.Json.Linq;
using Newtonsoft.Json;

[RequireComponent(typeof(CustomShapeManager))]
public class CreateMap : MonoBehaviour, PlacenoteListener {

    public Text debugText;

    private const string MAP_NAME = "GenericMap";

    private CustomShapeManager shapeManager;

    private bool shouldRecordWaypoints = false;
    private bool shouldSaveMap = true;
    private bool mARInit = false;

    private UnityARSessionNativeInterface mSession;

    private LibPlacenote.MapMetadataSettable mCurrMapDetails;

    private BoxCollider mBoxColliderDummy;
    private SphereCollider mSphereColliderDummy;
    private CapsuleCollider mCapColliderDummy;

    // Use this for initialization
    void Start() {

        shapeManager = GetComponent<CustomShapeManager>();

        Input.location.Start();

        mSession = UnityARSessionNativeInterface.GetARSessionNativeInterface();
        StartARKit();
        FeaturesVisualizer.EnablePointcloud();
        LibPlacenote.Instance.RegisterListener(this);
```

```
    }

    void OnDisable() {
    }

    // Update is called once per frame
    void Update() {
        if (!mARInit && LibPlacenote.Instance.Initialized())
        {
            Debug.Log("Ready To Start!");
            mARInit = true;

            return;
        }

        if (shouldRecordWaypoints) {
            Transform player = Camera.main.transform;
            //create waypoints if there are none around
            Collider[] hitColliders = Physics.OverlapSphere(player.position, 1f);
            int i = 0;
            while (i < hitColliders.Length) {
                if (hitColliders[i].CompareTag("waypoint")) {
                    return;
                }
                i++;
            }
            Vector3 pos = player.position;
            Debug.Log(player.position);
            pos.y = -.5f;
            shapeManager.AddShape(pos, Quaternion.Euler(Vector3.zero), false);
        }
    }

    public void CreateDestination() {
        shapeManager.AddDestinationShape();
    }

    private void StartARKit() {
        Debug.Log("Initializing ARKit");
        Application.targetFrameRate = 60;
        ConfigureSession();
    }


    private void ConfigureSession() {
#if !UNITY_EDITOR
```

```
                ARKitWorldTrackingSessionConfiguration config = new
ARKitWorldTrackingSessionConfiguration ();

                if (UnityARSessionNativeInterface.IsARKit_1_5_Supported ()) {
                        config.planeDetection = UnityARPlaneDetection.HorizontalAndVertical;
                } else {
                        config.planeDetection = UnityARPlaneDetection.Horizontal;
                }

                config.alignment = UnityARAlignment.UnityARAlignmentGravity;
                config.getPointCloudData = true;
                config.enableLightEstimation = true;
                mSession.RunWithConfig (config);
#endif
  }

  public void OnStartNewClick()
  {
    ConfigureSession();

    if (!LibPlacenote.Instance.Initialized())
    {
      Debug.Log("SDK not yet initialized");
      return;
    }

    Debug.Log("Started Session");
    LibPlacenote.Instance.StartSession();

    //start drop waypoints
    Debug.Log("Dropping Waypoints!!");
    shouldRecordWaypoints = true;
  }

  public void OnSaveMapClick() {
    OverwriteExistingMap();
  }

  void OverwriteExistingMap() {
    if (!LibPlacenote.Instance.Initialized()) {
      Debug.Log("SDK not yet initialized");
      return;
    }

    // Overwrite map if it exists.
    LibPlacenote.Instance.SearchMaps(MAP_NAME, (LibPlacenote.MapInfo[] obj) => {
```

```
            bool foundMap = false;
            foreach (LibPlacenote.MapInfo map in obj) {
                if (map.metadata.name == MAP_NAME) {
                    foundMap = true;
                    LibPlacenote.Instance.DeleteMap(map.placeId, (deleted, errMsg) => {
                        if (deleted) {
                            Debug.Log("Deleted ID: " + map.placeId);
                            SaveCurrentMap();
                        } else {
                            Debug.Log("Failed to delete ID: " + map.placeId);
                        }
                    });
                }
            }

            if (!foundMap) {
                SaveCurrentMap();
            }
        });
    }

    void SaveCurrentMap() {
        if (shouldSaveMap) {
            shouldSaveMap = false;

            if (!LibPlacenote.Instance.Initialized()) {
                Debug.Log("SDK not yet initialized");
                return;
            }

            bool useLocation = Input.location.status == LocationServiceStatus.Running;
            LocationInfo locationInfo = Input.location.lastData;

            Debug.Log("Saving...");
            debugText.text = "uploading...";
            LibPlacenote.Instance.SaveMap(
                (mapId) => {
                    LibPlacenote.Instance.StopSession();

                    LibPlacenote.MapMetadataSettable metadata = new
LibPlacenote.MapMetadataSettable();
                    metadata.name = MAP_NAME;
                    Debug.Log("Saved Map Name: " + metadata.name);

                    JObject userdata = new JObject();
                    metadata.userdata = userdata;
```

```
                JObject shapeList = GetComponent<CustomShapeManager>().Shapes2JSON();

                userdata["shapeList"] = shapeList;

                if (useLocation) {
                    metadata.location = new LibPlacenote.MapLocation();
                    metadata.location.latitude = locationInfo.latitude;
                    metadata.location.longitude = locationInfo.longitude;
                    metadata.location.altitude = locationInfo.altitude;
                }
                LibPlacenote.Instance.SetMetadata(mapId, metadata);
                mCurrMapDetails = metadata;
            },
            (completed, faulted, percentage) => {
                if (completed) {
                    Debug.Log("Upload Complete:" + mCurrMapDetails.name);
                    debugText.text = "upload complete!!";
                } else if (faulted) {
                    Debug.Log("Upload of Map Named: " + mCurrMapDetails.name + "faulted");
                } else {
                    Debug.Log("Uploading Map Named: " + mCurrMapDetails.name + "(" +
percentage.ToString("F2") + "/1.0)");
                }
            }
        );
    }
}

public void OnPose(Matrix4x4 outputPose, Matrix4x4 arkitPose) { }

public void OnStatusChange(LibPlacenote.MappingStatus prevStatus,
LibPlacenote.MappingStatus currStatus) {
        Debug.Log("prevStatus: " + prevStatus.ToString() + " currStatus: " +
currStatus.ToString());
        if (currStatus == LibPlacenote.MappingStatus.RUNNING && prevStatus ==
LibPlacenote.MappingStatus.LOST) {
            Debug.Log("Localized");
            //                          GetComponent<ShapeManager> ().LoadShapesJSON
(mSelectedMapInfo.metadata.userdata);
        } else if (currStatus == LibPlacenote.MappingStatus.RUNNING && prevStatus ==
LibPlacenote.MappingStatus.WAITING) {
            Debug.Log("Mapping");
        } else if (currStatus == LibPlacenote.MappingStatus.LOST) {
            Debug.Log("Searching for position lock");
        } else if (currStatus == LibPlacenote.MappingStatus.WAITING) {
```

```
            if (GetComponent<CustomShapeManager>().shapeObjList.Count != 0) {
                GetComponent<CustomShapeManager>().ClearShapes();
            }
        }
    }
}
```

7.2.2 FOR READ MAP:-

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using UnityEngine.UI;
using UnityEngine.XR.iOS;
using System.Runtime.InteropServices;
using System.IO;
using Newtonsoft.Json.Linq;
using Newtonsoft.Json;

public class ReadMap : MonoBehaviour, PlacenoteListener {

    private const string MAP_NAME = "GenericMap";

    private UnityARSessionNativeInterface mSession;
    private bool mARInit = false;

    private LibPlacenote.MapMetadataSettable mCurrMapDetails;

    string currMapID = String.Empty;

    private LibPlacenote.MapInfo mSelectedMapInfo;
    private string mSelectedMapId {
        get {
            return mSelectedMapInfo != null ? mSelectedMapInfo.placeId : null;
        }
    }

    // Use this for initialization
    void Start() {
        Input.location.Start();

        mSession = UnityARSessionNativeInterface.GetARSessionNativeInterface();
        StartARKit();
```

```
        FeaturesVisualizer.EnablePointcloud();
        LibPlacenote.Instance.RegisterListener(this);
    }

    void OnDisable() {
    }

    // Update is called once per frame
    void Update() {
        if (!mARInit && LibPlacenote.Instance.Initialized())
        {
            Debug.Log("Ready to Start!");
            mARInit = true;

            // Load Map
            FindMap();
        }
    }

    void FindMap() {
        //get metadata
        LibPlacenote.Instance.SearchMaps(MAP_NAME, (LibPlacenote.MapInfo[] obj) => {
            foreach (LibPlacenote.MapInfo map in obj) {
                if (map.metadata.name == MAP_NAME) {
                    mSelectedMapInfo = map;
                    Debug.Log("FOUND MAP: " + mSelectedMapInfo.placeId);
                    LoadMap();
                    return;
                }
            }
        });
    }

    void LoadMap() {
        ConfigureSession(false);

        LibPlacenote.Instance.LoadMap(mSelectedMapInfo.placeId,
            (completed, faulted, percentage) => {
                if (completed) {
                    Debug.Log("Loaded ID: " + mSelectedMapInfo.placeId + "...Starting session");

                    LibPlacenote.Instance.StartSession();

                } else if (faulted) {
                    Debug.Log("Failed to load ID: " + mSelectedMapInfo.placeId);
                } else {
```

```
                Debug.Log("Map Download: " + percentage.ToString("F2") + "/1.0");
            }
        }
    );
  }
private void StartARKit() {
    Debug.Log("Initializing ARKit");
    Application.targetFrameRate = 60;
    ConfigureSession(false);
  }
private void ConfigureSession(bool clearPlanes) {
#if !UNITY_EDITOR
            ARKitWorldTrackingSessionConfiguration config = new
ARKitWorldTrackingSessionConfiguration ();
            config.planeDetection = UnityARPlaneDetection.None;
            config.alignment = UnityARAlignment.UnityARAlignmentGravity;
            config.getPointCloudData = true;
            config.enableLightEstimation = true;
            mSession.RunWithConfig (config);
#endif
  }
  public void OnPose(Matrix4x4 outputPose, Matrix4x4 arkitPose) { }

  public void OnStatusChange(LibPlacenote.MappingStatus prevStatus,
LibPlacenote.MappingStatus currStatus) {
    Debug.Log("prevStatus: " + prevStatus.ToString() + " currStatus: " +
currStatus.ToString());
    if (currStatus == LibPlacenote.MappingStatus.RUNNING && prevStatus ==
LibPlacenote.MappingStatus.LOST) {
      Debug.Log("Localized: " + mSelectedMapInfo.metadata.name);

GetComponent<CustomShapeManager>().LoadShapesJSON(mSelectedMapInfo.metadata.userd
ata);
      FeaturesVisualizer.DisablePointcloud();
    } else if (currStatus == LibPlacenote.MappingStatus.RUNNING && prevStatus ==
LibPlacenote.MappingStatus.WAITING) {
      Debug.Log("Mapping");
    } else if (currStatus == LibPlacenote.MappingStatus.LOST) {
      Debug.Log("Searching for position lock");
    } else if (currStatus == LibPlacenote.MappingStatus.WAITING) {
      if (GetComponent<CustomShapeManager>().shapeObjList.Count != 0) {
        //GetComponent<CustomShapeManager>().ClearShapes();
      }
    }
  }
}
```

Chapter 8

Testing

8.1 Test Cases

Table 3.1 shows various test cases which were considered while implementing this project.

Considering table 8.1 from that first user have to scan the location which he/she want. After that user have start capturing images and video through his phone. So total area can be scanned. After that through database map will be provide to the user whether that scanned map is saved in database or not. And that location of the user can saved in database successfully. If user want to change the location then he can update the location. After that database will check that map-id, map-name, latitude, longitude, altitude of user and save that metadata in database. After that completion process can shown to the user in the form of percentage. And that can shown in GUI. After all the processes are completed the placenote SDK can be activated. It will check the map that previously saved and according to that show the direction arrow to the user. User can follow that arrow to reach his destination. In this way user can successfully reach to his/her destination.

Table 8.1: Test Cases for User

Sr No.	Test Module	Test Case description	Input data	Expected Output	Actual Output	Test Status
1	Check Scanning	To Check whether user can scan the entire location he/she want	The image and video capture using camera.	Scanned area.	Scanned area.	Pass
2	Saving the map	To check whether the scanned map gets saved.	The image and video capture using camera.	Saving the map successfully.	Save map successfully.	Pass
3	Update the scanned map	To check whether the saved map can be updated	The image and video capture using camera.	Update the previous map.	Updating the map successfully.	Pass
4	Storing metadata	To check that map-id, map-name, latitude, longitude, altitude is saved.	The image and video capture using camera.	Saving each metadata.	Metadata saved.	Pass

5	Giving percentage of completion	To check how much the data of map is updated.	The image and video capture using camera.	Shows percentage in GUI	Gives percentage on GUI.	Pass
6	Placenote SDK activated.	To check whether placenote library is initialized.	Scanning of map and uploading it.	Successfully scan of map without crash.	Successfully initialized SDK.	Pass
7	Reading the map	To check whether the scanned map is read	Previously scanned path/map.	Showing the directional arrow.	Gives the arrow to user for destination	Pass

Chapter 9

Results and Discussion

9.1 Result 1:

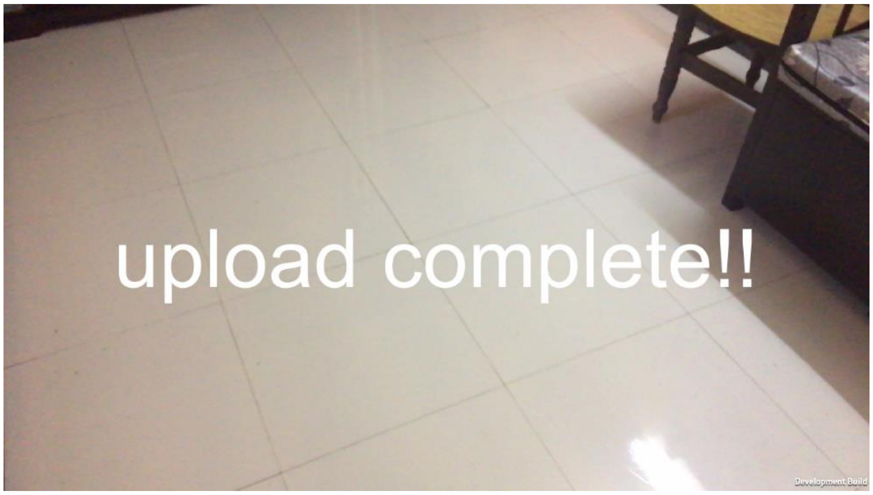

Fig 9.1 Complete Data Uploading

The Figure 9.1 shows that user uploaded scanned map successfully. After that it will check in system database that particular destination is present in that or not according to that it will show the route to the user to reach till that user destination..

9.2 Result 2:

Fig 9.2 Path Detection

The Figure 9.2 shows that the arrow shows path to the user to reach that user destination according to user gave destination to the system. That arrow navigate user in real world.

9.3 Result 3:

Fig 9.3 Reach Destination

The Figure 9.3 shows that when user reached to the destination the point we put on that particular destination will shown on the screen so get confirmation that he reached on his destination.

Chapter 10

Conclusion and Future Scope

10.1 Conclusion

In this project we have developed a system for indoor navigation based on augmented reality which is supported on android platform. We have kept in mind the tiresome work that goes into individuals mind for looking through the map to navigate themselves in indoor environments. We have also kept in mind the complex scenarios which can be found in various indoor environments where one cannot find a real world object which acts as a feature for the navigation hence we have introduced the Rode based feature which helps users in scenarios such as the above mentioned

10.2 Future Scope

As a future work we are planning to make this application cross platform by using native environments for development this would be helpful not only for android users but also for users who use ios and other major os. We would also work on how the system will be able to differentiate between similar objects placed at different places in the same indoor environment. We would also try to integrate outdoor environment navigation techniques with our system so as to navigate from one room in one part of the campus to the other room present at some other part of the same campus.

Appendix

AR	Augmented Reality
GPS	Global Positioning System
MEME	Micro-Electro-Mechanical Systems
PDR	Pedestrian Dead Reckoning
INS	Inertial Navigation System
IMU	Inertial Measurement Unit
UML	Unified Modeling Language
SDK	Software Development Kit
WBS	Work Breakdown Structure
VPS	Virtual Private Server
SLAM	Simultaneous Localization and Mapping
RFID	Radio Frequency Identification

Bibliography and Reference

[1]. O. Köroğlu, "En Yaygın Yeni İletişim Ortamı Mobil İletişim Ortamında İçerik ve Yayıncılık", Civilacademy Journal of Social Sciences, vol. 8, no. 2, pp. 55-86, 2010.

[2]. Z. T´oth, “Ilona: indoor localization and navigation system,” Journal of Location Based Services, vol. 10, no. 4, pp. 285–302, 2016.

[3]. A. Serra, D. Carboni, and V. Marotto, "Indoor Pedestrian Navigation System Using a Modern Smartphone ", in Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services, pp. 397-398, (2010).

[4]. C. G. Low and Y. L. Lee, "SunMap+: An intelligent location-based virtual indoor navigation system using augmented reality," International Conference on Frontiers of Communications, Networks and Applications (ICFCNA 2014 - Malaysia), Kuala Lumpur, 2014, pp. 1-6

[5]. Z. T´oth and J. Tam´as, “Miskolc iis hybrid ips: Dataset for hybrid indoor positioning,” in Radio elektronika (RADIOELEKTRONIKA), 2016 26th International Conference. IEEE, 2016, pp. 408–412

.

[6]. Harter and A. Hopper, “A Distributed Location System for the Active Office,” IEEE Network, January 1994.

[7]. N. Z. Naqvi, A. Kumar,A. Chauhan and K. Sahni, “Step Counting Using Smartphone-Based Accelerometer,” International Journal on Computer Science and Engineering, vol 4, no 5, pp. 675-681, May, 2012.

[8]. B.Ozdenizci, V.Coskun, andK. Ok,“Nfc internal: An indoor navigation system,” Sensors, vol. 15, no. 4, pp. 7571–7595, 2015.

[9]. Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, Micaela Wiseman, “Indoor location sensing using geomagnetism”, MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services.

[10]. A.R. Pratama, Widyawan, and R. Hidayat,"Smartphonebased Pedestrian Dead Reckoning as an Indoor Positioning System", International Conference on System Engineering and Technology (ICSET), pp.1-6, (2012).

[11]. Verma, S., Omanwar, R., Sreejith, V., & Meera, G. S. (2016). A smartphone based indoor navigation system. 2016 28th International Conference on Microelectronics (ICM). doi:10.1109/icm.2016.7847886

[12]. Yadav, R., Chugh, H., Jain, V., & Baneriee, P. (2018). Indoor Navigation System Using Visual Positioning System with Augmented Reality. 2018 International Conference on Automation and Computational Engineering (ICACE). doi:10.1109/icace.2018.8687111

[13]. Bin Abdul Malek, M. F., Sebastian, P., & Frieberg, M. (2017). Augmented reality assisted localization for indoor navigation on embedded computing platform. 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA). doi:10.1109/icsipa.2017.8120589

[14]. Jong-Bae Kim, Jeong-Mi Lee, & Hee-Sung Jun. (n.d.). AR-Based Indoor Navigation System for Personal Locating. 2006 Digest of Technical Papers International Conference on Consumer Electronics. doi:10.1109/icce.2006.1598388

[15]. Li, Y.-Y., Chiu, P.-H., Yeh, S.-C., & Zhou, C. (2017). Effects of Virtual Reality and Augmented Reality on Induced Anxiety. 2017 5th International Conference on Enterprise Systems (ES). doi:10.1109/es.2017.28

Publication

[1] Ms. Joyce Lemos, Mr. Dikshant Jopat, Mr. Krutarth Makwana, Mr. Hardik Dhanmeher, “Indoor Navigation System Using Augmented Reality”, *Journal of International organization of Scientific Research, Vol. 10, Issue 3, March 2020, ||Series -II|| PP 14-18.*

IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 10, Issue 3, March 2020, ||Series -II|| PP 14-18

Indoor Navigation System Using Augmented Reality

Dikshant Jopat[1], Krutarth Makwana[2], Hardik Dhanmeher[3], Joyce Lemos[4]
[1](St. John College of Engineering & Management, India)
[2](St. John College of Engineering & Management, India)
[3](St. John College of Engineering & Management, India)
[4](St. John College of Engineering & Management, India)
Corresponding Author: Joyce Lemos
Received 01 March 2020; Accepted 16 March 2020

Abstract: In today's life, when the cities are rising, usage areas of mobile phones have increased in the last ten years. Although there have been significant improvements in many areas, most of the developments are in the field of positioning systems. Although the people's lives continue in indoor circumstances, location-based information system receives details from the satellites, which can detect a person's location in the outdoor region alone. However, for indoor areas, no efficient and perfect technology has been developed for navigation or positioning. In this paper we have come up with an exceptional solution that is indoor navigation mobile application which works with augmented reality, in our proposed solution, we will be using the mobile camera as the scanner for getting the path and extracting the features from various objects in the pathway. We use ARCore SDK, which is the heart of this project, which has an inbuilt property called area learning, which helps the system to extract and learn about the features present in a particular outline using Machine Learning.
Keywords: Mobile application, augmented reality, ARCore SDK, indoor positioning, and navigation.

I. INTRODUCTION

The booming enhancement of mobile technology in today's 21st century is one of the prime factors of the growing technological environment in every country. One of the critical concerns has always been the development of mobile applications, and researchers have put in a lot of effort into compatible operating systems for supporting these applications[3]. Thanks to modern-day technologies, android has emerged as one of the top-notch mobile operating systems, which even has a more straightforward development interface for application developers as compared to other OS. This paved the path for the more effortless development of complex applications for developers [4]. One of the many essential requirements in our day to day life is positioning and navigation of oneself, i.e., where the person wants to go and his actual position. With the advancement in mobile technologies, there has been an urge for the improvement of navigation-based technologies, in which mobile applications can play a crucial role. With the advancements in above-said technologies, outdoor navigation has enhanced ten-folds. Nevertheless, with the lives of people who are indoor oriented, a complex indoor scenario like office space, IT-campus, hospitals, malls navigation becomes a significant concern [5]. Going to a new campus and getting used to the campus environment takes time for a fresher. In thesetypes of conditions, an indoor navigation system is a must and becomes a friend in need.

A. Bluetooth Beacons:

Bluetooth Beacons broadcast their identifier to nearby portable electronic devices [6].

Disadvantage: But this technology enables smartphones/tablets and other devices to perform actions and navigation only when close to a beacon

B. Wi-fi positioning system (WPS):

WPS uses the characteristics of nearby wi-fi hotspots and other wireless access points to find where a device is located. It is the most conventional and considerable localization technique used for positioning is based on measuring the power of the received signal and the procedure of fingerprinting [7].

Disadvantage:But the accuracy depends on the number of access points whose positions have been entered into the database[8].

C. Radio Frequency Identification Tags (RFID):

Radio Frequency Identification uses electromagnetic fields to recognize and track tags fixed to things instinctively.

Disadvantage: But a large amount of data may be generated that is not useful for managing indoor navigation or other applications and makes the system inefficient [1]

II. LITERATURE REVIEW

In 2016, Verma, S., Omanwar [11] represented a system for indoor navigation using off-the-shelf mobile phones. In the first phase, a user creates an inside map of the world by linking panoramic images using our web application, Map Maker. This indoor map is then employed by our smartphone-based Navigation application to estimate a user's location, calculate the shortest path and help in navigating the user to a destination. In future work, we decide to extend our smartphone application to automate the method of step counting when capturing panoramas. We also decide to explore real-time image matching techniques using smartphones to extend the accuracy of our indoor navigation system.

In 2018, Jain, V.[12] introduced the concept of this approach shows the potential for future upgradations. It can be a revolutionary step in an indoor navigation system. This work incorporates visual technology, which is proliferating. The accuracy of the AR toolkit in detecting a marker depends on lighting conditions. If the lighting condition is almost the same as the lighting condition in which the marker is developed, then the AR toolkit will work correctly. In the future, there can be various modifications made to the navigation system, such as using an audio module for helping in navigating, and also, the processing and calibration of the camera can be improved. AR toolkit can be trained in detecting colored markers too. As a part of future tasks, we would like to work on building a system for the preprocessing of the indoor floor plan of the building. This will directly work towards the idea proposed in this work.

In 2017, Bin Abdul Malek[13]paper represented an alternative navigation tool that can be used in an indoor environment. This is due to restrictions on Global Positioning System signals that cannot be detected in indoor locations. The work presented here shows the event of an interactive indoor localization system that uses live input video capture and may identify location markers to point its current location. Besides, augmented reality also can superimpose augmented reality objects above the situation markers to point the direction to be taken by the user, which assists the user in navigating to the chosen destination. The developed system was implemented on a Raspberry Pi, an embedded computing platform, with a USB camera and display glasses for live video capture and display devices, respectively. It was tested in Universiti Teknologi PETRONAS' Information Resource Center, across multiple locations and different floors of the middle.

In 2006, Jong-Bae Kim[14]described the personal location method in image order using a wearable computer for an Augmented Reality-based indoor navigation system, which allows the user to navigate an unfamiliar and unknown spot in an indoor environment. The system uses a color histogram matching technique to recognize the personal location in the image sequence. Individual locations are recognized by the suggested method in real-time. The results are overplayed on the user's view through the Augmented Reality technique. The plan presented in this system applies to a navigation system.

In 2017, Li's[15]study indicated that the proposed Augmented Reality and Virtual Reality systems effectively induced anxiety in subjects. Moreover, by measuring the subjective scale scores, skin conductance, and heart rate, no significant difference was found between the proposed Augmented Reality and Virtual Reality systems in terms of inducing anxiety. Future research will further quantify the contribution of each type of stimulus from human perceptions (visual, auditory, haptic, or smell) and investigate whether Augmented Reality or Virtual Reality system is more effective in inducing anxiety.

III. RESEARCH METHODOLOGY

In our proposed system, the heart of the project goes with ARCore SDK, which takes the real-world objects as input through the camera and converts them into logical and unique features. ARCore SDK has an exclusive property called Places learning model, which is based on Machine Learning that helps it to extract the features from real-world objects. It uses motion tracking that is useful for feature extraction following the movement and orientation of the phone. So this is how our proposed system works:

A. Firstly, the user will open an application based on AR.

B. This app will intimate the user either to create his own map/route or will ask him/her if the user needs assistancewith navigation.

C. If the user clicks on the create map, then the user automatically will be asked the name of the source and destination and will be asked to start his/her camera.

D. After the camera is initialized, the application using ARCore will start extracting features from the camera input, and the user will be asked for marking waypoints at the place where feature extraction has been completed.

E. This will complete the mapping phase.

F. If the user clicks on 'assist me' for indoor navigation, the user will be next asked for a destination.

G. Once the destination has been taken, the camera will be initialized, and features from the current location of the user will be extracted.

H. These features will be matched with the various features already present in the database, and if a match is found, the system will get the user's current location.

I. After this, the system will start the route planning algorithm and will find the best route from various routes that have already been mapped.

J. If no route is to be found to the destination, the user will get a prompt message that either the entered destination is wrong or no route has been found for the entered destination.

K. This route will lead the user to the destination.

The most exceptional advantage that our system has is that it is adaptive because the user input will be taken as a video feed other than AR camera feed and will be analyzed if the place has added some new objects or features, if yes then those features will also be added to the database. Furthermore, another major problem that exists is if there is an environment where no noticeable/unique objects are present, for example, stairs of a building, then we will be taking the help of QR codes, which will act as features for these kinds of environments.

s

In our proposed system we have two phases:

Mapping Phase: - In this phase, the real-time mapping of the route occurs; we map the features present on the route and place them in a JSON format in the database.

Below given is the implementation level milestones for this phase: -

A. Extraction of features

B. The initialization of the camera for real-time scanning of objects

C. Features will be uploaded in the database in JSON format.

D. Marking of waypoints

Testing Phase: - In this phase, we will take the user input from the user for the destination, and then the route planning algorithm will map the most suitable route from the user's current position to the destination. Below given is the implementation level milestones for this phase: -

(a)User input will be taken.

(b)The camera will be initialized for showing the route to the users.

(c)The route planning algorithm will find the best route possible from the current position.

(d)The marked waypoints will be shown on the mobile screen using arrows.

Below given is the architectural diagram of our proposed system: -

Fig .1 Architecture Diagram

IV. EXPERIMENTAL RESULTS

ARCore SDK normalized detection parameters

These parameters symbolize the average distance required for object detection. The object recognition accuracy of our system is affected by the camera-object separation. The minimum distance recommended by the ARCore SDK manufacturers is 30 cm. As the separation increases, the quality of the depth data deteriorates 1¼ times, and beyond 1 m, texture details of target objects are hard to capture [9]. The SDK capitalizes on any error with the help of unique features like machine learning models and motion testing from various angles. From reliable observations, to achieve reliable sensing and good quality visualization, we set an acceptable score of 0.5-1 for both the metrics [9].

Fig .2 Real-time indoor navigation

Fig.3 Feature point extracted Fig .4 Visible AR object

The above figure shows the feature extraction from the location of theLabs/Classroom/Staffrooms by indicating AR object on the ground using plane detection. It superimposes a AR object in real world with the help of ARCore SDK and Unity.

V. CONCLUSION AND FUTURE SCOPE

This propose system, we have developed a system for indoor navigation based on augmented reality, which is supported on the android platform. We have kept in mind the tiresome work that goes into individuals' mind for looking through the map to navigate themselves in indoor environments. We have also kept in mind the complex scenarios that can be found in various indoor environments where one cannot find a real-world object which acts as a feature for the navigation; hence we have introduced the QRcode based feature which helps users in scenarios such as the above mentioned. As future work, we are planning to make this application cross-platform by using native environments for development. This would be helpful not only for android users but also for users who use iOS and other OSes. We would also work on how the system will be able to differentiate between similar objects placed at different places in the same indoor environment. We would also try to integrate outdoor environment navigation techniques with our system to navigate from one room in one part of the campus to the other room present at some other part of the same campus.

REFERENCES

[1]. O. Köroğlu, "En Yaygın Yeni İletişim Ortamı Mobil İletişim Ortamında İçerik ve Yayıncılık", Civilacademy Journal of Social Sciences, vol. 8, no. 2, pp. 55-86, 2010.

[2]. Z. T´oth, "Ilona: indoor localization and navigation system," Journal of Location Based Services, vol. 10, no. 4, pp. 285–302, 2016.

[3]. A. Serra, D. Carboni, and V. Marotto, "Indoor Pedestrian Navigation System Using a Modern Smartphone ", in Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services, pp. 397-398, (2010).

[4]. C. G. Low and Y. L. Lee, "SunMap+: An intelligent location-based virtual indoor navigation system using augmented reality," International Conference on Frontiers of Communications, Networks and Applications (ICFCNA 2014 - Malaysia), Kuala Lumpur, 2014, pp. 1-6

[5]. Z. T´oth and J. Tam´as, "Miskolc iis hybrid ips: Dataset for hybrid indoor positioning," in Radio elektronika (RADIOELEKTRONIKA), 2016 26th International Conference. IEEE, 2016, pp. 408–412.

[6]. Harter and A. Hopper, "A Distributed Location System for the Active Office," IEEE Network, January 1994.

[7]. N. Z. Naqvi, A. Kumar,A. Chauhan and K. Sahni, "Step Counting Using Smartphone-Based Accelerometer," International Journal on Computer Science and Engineering, vol 4, no 5, pp. 675-681, May, 2012.

[8]. B.Ozdenizci, V.Coskun, andK. Ok,"Nfc internal: An indoor navigation system," Sensors, vol. 15, no. 4, pp. 7571–7595, 2015.

[9]. Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, Micaela Wiseman, "Indoor location sensing using geomagnetism", MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services.

[10]. A.R. Pratama, Widyawan, and R. Hidayat,"Smartphonebased Pedestrian Dead Reckoning as an Indoor Positioning System", International Conference on System Engineering and Technology (ICSET), pp.1-6, (2012).

[11]. Verma, S., Omanwar, R., Sreejith, V., & Meera, G. S. (2016). A smartphone based indoor navigation system. 2016 28th International Conference on Microelectronics (ICM). doi:10.1109/icm.2016.7847886

[12]. Yadav, R., Chugh, H., Jain, V., & Baneriee, P. (2018). Indoor Navigation System Using Visual Positioning System with Augmented Reality. 2018 International Conference on Automation and Computational Engineering (ICACE). doi:10.1109/icace.2018.8687111

[13]. Bin Abdul Malek, M. F., Sebastian, P., & Drieberg, M. (2017). Augmented reality assisted localization for indoor navigation on embedded computing platform. 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA). doi:10.1109/icsipa.2017.8120589

[14]. Jong-Bae Kim, Jeong-Mi Lee, & Hee-Sung Jun. (n.d.). AR-Based Indoor Navigation System for Personal Locating. 2006 Digest of Technical Papers International Conference on Consumer Electronics. doi:10.1109/icce.2006.1598388

[15]. Li, Y.-Y., Chiu, P.-H., Yeh, S.-C., & Zhou, C. (2017). Effects of Virtual Reality and Augmented Reality on Induced Anxiety. 2017 5th International Conference on Enterprise Systems (ES). doi:10.1109/es.2017.28

Acknowledgement

We are highly grateful to our institute St. John College of Engineering and management and the Information Technology Department for providing us with this learning opportunity with the required resources to accomplish our task.

We wish to express our deep gratitude to our project guide **Ms. Joyce Lemos**, Assistant Professor in the Department of Information Technology for all the advice, encouragement and constant supervision for the betterment of the project. This work would not have been possible without her intellectual and valuable suggestions.

We would also like to extend our special thanks to **Dr. Terence Johnson**, head of Information Technology Department and **Ms. Shraddha More**, Project Co-ordinator for their encouragement, wholehearted cooperation and support.

We express our sincere gratitude to our respected principal Dr. G.V. Mulgund for encouragement and facilities provided to us.

We would also like to acknowledge the patience that our ever-beloved parents have shown during our efforts and the encouragement we have received from them.

Dikshant Jopat	PID No. 1164011
Krutarth Makwana	PID No. 1164041
Hardik Dhanmeher	PID No. 1164012

Date: