



Syed Shahzaib Sajjad

**01-134182-062**

Amtul Wakeel Saleha

**01-134182-011**

# GuideBOT

**Bachelors Of Science in Computer Science**

Supervisor: Ms. Maryam Bibi

Department of Computer Science

Bahria University, Islamabad

**June 2022**



# Certificate

We accept the work contained in the report titled “**GuideBot**”, written by **Syed Shahzaib Sajjad** and **Amtul Wakeel Saleha** as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

**Approved By:**

**Supervisor:** Maryam Bibi

---

**Internal Examiner:**

---

**External Examiner:**

---

**Project Cordinator:** Dr. Moazam Ali

---

**May 27<sup>th</sup>, 2022**



# Abstract

In this era of the modern world, we are advancing in the field of technology. We are focusing on how to do our work more efficiently, more precisely, and with less human effort. Augmented Reality (AR) is an interactive experience of the physical world that is created by using digital visual elements and computer-generated perceptual information. It helps in experiencing an enhanced version of the physical world without experiencing it in real. AR is assisting us in different industries and fields of life. Our project is based on an application that is using AR in helping users to guide them to different locations within an indoor environment. Navigation using AR is more innovative and interactive. Such types of applications are being used in developed countries. Hence, we want to bring such advancement to our country as well. Our application, GuideBOT, is an Augmented Reality based navigation application that is helping its users navigate to their desired locations without any human assistance in an indoor/closed environment. Hence, AR has made navigation accurate, interactive, and innovative in the indoor environment for our application.

# Acknowledgement

In the name of Allah, the Most Gracious and the Most Merciful. It is just because of his blessings, that we were able to finish this Project. We want to convey our most sincere thanks and appreciation towards **Ms. Maryam Bibi**, for putting her un-tiring efforts and time in this project. Her involvement in the project is remarkable and the way she guides her students has been an incredible wellspring of motivation to us to work hard with our full dedication and motivation.

SYED SHAHZAIB SAJJAD, AMTUL WAKEEL SALEHA  
Islamabad, Pakistan

June 2022

*“You are the artist of your own life. Don’t hand the paintbrush to anyone else.”*

Satya Nani



# Contents

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.1.1 Open/Outdoor Environment . . . . .	1
1.1.2 Closed/Indoor Environment . . . . .	1
1.2 Background . . . . .	2
1.3 Problem Description . . . . .	3
1.4 Project Objective . . . . .	3
1.5 Project Scope . . . . .	3
<b>2 Literature Review</b>	<b>4</b>
2.1 Augmented Reality . . . . .	4
2.1.1 Augmented Reality vs Virtual Reality . . . . .	5
2.2 Applications of Augmented Reality . . . . .	5
2.3 Related Researches . . . . .	6
2.4 Navigation . . . . .	7
2.4.1 Outdoor Navigation . . . . .	7
2.4.2 Indoor Navigation . . . . .	7
2.4.3 Methods For Navigation . . . . .	8
2.4.4 Maps . . . . .	8
2.4.5 Global Positioning System . . . . .	8
2.4.6 Augmented Reality . . . . .	9
2.4.7 WiFi Positioning System . . . . .	9
2.4.8 Vision-Based Localization System . . . . .	10
2.4.9 Comparison Between These Methods . . . . .	10
2.5 AR-Based Navigation . . . . .	10
2.5.1 Marker-Based Methods . . . . .	11
2.5.2 Method based on 2D image recognition . . . . .	11
2.5.3 Method based on 3D image recognition . . . . .	11
<b>3 Software Requirement Specification</b>	<b>13</b>
3.1 Existing System . . . . .	13
3.2 Proposed System . . . . .	14
3.3 Work Flow . . . . .	14
3.4 System Requirements . . . . .	15
3.4.1 Software Requirement . . . . .	15
3.4.2 Hardware Requirement . . . . .	15

3.5	Software Requirements . . . . .	16
3.5.1	Functional requirements . . . . .	16
3.5.2	Non-Functional requirements . . . . .	16
3.6	Use Cases . . . . .	17
3.6.1	Overall Use case . . . . .	18
3.6.2	Interacting with the Application . . . . .	18
3.6.3	Selecting the desired location . . . . .	19
3.6.4	Toggling the navigation line visibility . . . . .	20
3.6.5	Scanning the QR code . . . . .	21
3.6.6	Following the AR based navigation arrows . . . . .	22
3.6.7	Reaching the destination . . . . .	23
<b>4</b>	<b>System Design</b>	<b>25</b>
4.1	System Architecture . . . . .	25
4.2	Design methodology . . . . .	26
4.3	Design Constraints . . . . .	26
4.4	Context Diagram . . . . .	26
4.5	Sequence Diagram . . . . .	27
4.5.1	Interaction with the application . . . . .	27
4.5.2	Selection of the desired location . . . . .	28
4.5.3	Toggling the visibility of navigation line . . . . .	29
4.5.4	Following Navigation Line . . . . .	29
4.5.5	Scanning the QR Code . . . . .	30
4.5.6	Reaching the Destination . . . . .	30
4.5.7	AR based Navigating Application . . . . .	31
4.6	Activity Diagram . . . . .	32
4.7	GUI Design . . . . .	33
4.7.1	Front Screen Layout . . . . .	33
4.7.2	Map Screen . . . . .	34
4.7.3	Scanning the QR code . . . . .	35
4.7.4	Navigation Screen Layout . . . . .	36
<b>5</b>	<b>System Implementation</b>	<b>37</b>
5.1	System Architecture . . . . .	37
5.2	System Components . . . . .	37
5.2.1	Analyzing the environment . . . . .	38
5.2.2	Creating a 2D map . . . . .	38
5.2.3	Recreating 2D map into the 3D map . . . . .	38
5.2.4	Placing the targets . . . . .	39
5.2.5	Target Indicator . . . . .	39
5.2.6	Determining the shortest path . . . . .	40
5.2.7	Generating the AR based indicator . . . . .	43
5.2.8	Resetting the session . . . . .	44
5.3	Tools and Technology Used . . . . .	44
5.3.1	AR Core . . . . .	44
5.3.2	AR Foundation . . . . .	44
5.3.3	Unity 3D . . . . .	45
5.3.4	ZXing .NET . . . . .	45

5.3.5	Visual Studio . . . . .	45
5.3.6	Inkscape . . . . .	45
5.4	Language Used . . . . .	45
<b>6</b>	<b>System Testing and Evaluation</b>	<b>46</b>
6.1	Graphical User Interface Testing . . . . .	46
6.2	Usability Testing . . . . .	46
6.3	Software Performance Testing . . . . .	46
6.4	Exception Handling . . . . .	47
6.5	Installation Testing . . . . .	47
6.6	Test Cases . . . . .	47
6.6.1	Launching Application Test Case . . . . .	47
6.6.2	AR Camera Working Test . . . . .	48
6.6.3	Position on Mini Map . . . . .	48
6.6.4	Navigation Line visibility on Mini Map . . . . .	49
6.6.5	AR based navigation line visibility in application's camera in- terface . . . . .	49
6.6.6	Reaching the destination selected . . . . .	50
6.6.7	Visibility of the AR based targets on the destination . . . . .	51
6.6.8	Toggling navigation line visibility . . . . .	52
6.6.9	Selecting the destination from the list provided . . . . .	52
6.6.10	Scanning the QR Code . . . . .	53
6.6.11	Scanning QR Code resets the session . . . . .	53
6.7	Initial Build Test of the Application . . . . .	54
6.8	Final Build Test of the Application . . . . .	55
<b>7</b>	<b>Conclusions</b>	<b>58</b>
7.1	Future Work . . . . .	58
7.2	Conclusion . . . . .	59
<b>A</b>	<b>User Manual</b>	<b>60</b>
A.1	Introduction . . . . .	60
A.2	Loading Screen . . . . .	61
A.3	Front Screen . . . . .	62
A.4	Map Selection screen . . . . .	63
A.5	Live Camera Screen . . . . .	64
A.6	Destination Selection Screen . . . . .	65
A.7	Navigation Line Visibility . . . . .	66
A.7.1	AR based navigation line toggled on . . . . .	66
A.7.2	AR based navigation line toggled off . . . . .	67
A.8	Destination indication . . . . .	68
	<b>References</b>	<b>69</b>

# List of Figures

1.1	Visual presentation of the AI based AR chat bot [1] . . . . .	2
1.2	Visual presentation of AR based navigation system [2] . . . . .	2
2.1	Milgram Reality Virtuality Continuum [3] . . . . .	5
3.1	Image illustrating various navigation methods [4] . . . . .	13
3.2	Image illustrating work flow of the system . . . . .	14
3.3	Use case for overall working of application . . . . .	18
3.4	Use case for opening the application . . . . .	19
3.5	Use Case for choosing the desired location . . . . .	19
3.6	Use Case for toggling the navigation line . . . . .	20
3.7	Use case for scanning the QR Code . . . . .	21
3.8	Use case to follow digital markers . . . . .	22
3.9	Use case for user reaching the desired location . . . . .	23
4.1	System Architecture design of the system . . . . .	26
4.2	Contextual design of the system . . . . .	27
4.3	Sequence diagram for interaction with the application . . . . .	28
4.4	Sequence diagram for selection of the destination . . . . .	28
4.5	Sequence diagram for toggling the visibility of navigation line . . . . .	29
4.6	Sequence diagram for following navigation line . . . . .	29
4.7	Sequence diagram for scanning the QR code . . . . .	30
4.8	Sequence diagram for arriving on the destination . . . . .	30
4.9	Sequence diagram for Overall system design . . . . .	31
4.10	Activity Diagram for the system of the system . . . . .	32
4.11	Front Screen Layout of the application . . . . .	33
4.12	Map choosing screen of the application . . . . .	34
4.13	QR code scanning screen . . . . .	35
4.14	Navigation Guiding Screen . . . . .	36
5.1	Applying the NavMesh on testing map . . . . .	39
5.2	3D model of the balloon . . . . .	40
5.3	Set of nodes connected [5] . . . . .	41
5.4	Root node A [5] . . . . .	41
5.5	Traversing from root node to goal node [5] . . . . .	42
5.6	Traversing from my room to the main entrance . . . . .	42
5.7	Traversing from my room to the guest room . . . . .	43
5.8	3D model of the bot . . . . .	44

6.1	Screenshots from the initial builds of the application . . . . .	54
6.2	Application is launched and running . . . . .	55
6.3	Maps can be selected successfully . . . . .	56
6.4	AR camera view and AR based bot visible . . . . .	56
6.5	Successfully visible AR based line and target . . . . .	57
A.1	Loading screen of the application . . . . .	61
A.2	Front screen of the application . . . . .	62
A.3	Map selection screen of the application . . . . .	63
A.4	Live AR camera screen of the application . . . . .	64
A.5	Destination selection screen of the application . . . . .	65
A.6	Navigation line visibility turned on . . . . .	66
A.7	Navigation line visibility turned off . . . . .	67
A.8	3D model of hot air balloon while approaching destination . . . . .	68

# List of Tables

2.1	Comparison table between navigation methods . . . . .	10
3.1	Description Table for use case 1 . . . . .	19
3.2	Description Table for use case 2 . . . . .	20
3.3	Description Table for use case 3 . . . . .	21
3.4	Description Table for use case 4 . . . . .	22
3.5	Description Table for use case 5 . . . . .	23
3.6	Description Table for use case 6 . . . . .	24
6.1	Table for Test Case 1 . . . . .	47
6.2	Table for Test Case 2 . . . . .	48
6.3	Table for Test Case 3 . . . . .	48
6.4	Table for Test Case 4 . . . . .	49
6.5	Table for Test Case 5 . . . . .	49
6.6	Table for Test Case 6 . . . . .	50
6.7	Testing the application for destination 1 . . . . .	50
6.8	Testing the application for destination 2 . . . . .	50
6.9	Testing the application for destination 3 . . . . .	51
6.10	Table for Test Case 7 . . . . .	51
6.11	Table for Test Case 8 . . . . .	52
6.12	Table for Test Case 9 . . . . .	52
6.13	Table for Test Case 10 . . . . .	53
6.14	Table for Test Case 11 . . . . .	53

# Acronyms and Abbreviations

<b>AP</b>	Access Point
<b>API</b>	Application Programming Interface
<b>AR</b>	Augmented Reality
<b>AV</b>	Augmented Virtuality
<b>BSSI</b>	Basic Services Set Identification
<b>GPS</b>	Global Positioning System
<b>HMD</b>	Head Mounted Display
<b>IDE</b>	Integrated Development Environment
<b>MARTA</b>	Mobile Augmented Reality Technical Assistant
<b>SDK</b>	Software Development Kit
<b>UUID</b>	Universally Unique Identifier
<b>VR</b>	Virtual Reality

# Chapter 1

## Introduction

### 1.1 Overview

In this era of the modern world, we are advancing in the field of technology. We are focusing on how to do our work more efficiently, more precisely and with less human effort. Augmented Reality (AR) is an interactive experience of the physical world that is created by using digital visual elements and computer-generated perceptual information. It helps in experiencing an enhanced version of the physical world without experiencing it in real [6].

Nowadays, AR is helping us in various ways. It is assisting us in different industries and fields of life. Our project is based on designing an application that is using AR in helping users to guide them to different locations within a building. AR has made navigation interactive and innovative.

While keeping all this in mind we are introducing an application named as **GuideBOT**. GuideBOT is an **Augmented Reality** based navigation application that is helping its users to navigate to their desired locations without any human assistance in an indoor/closed environment.

#### 1.1.1 Open/Outdoor Environment

An open/outdoor environment is an environment in which various components of an organization will communicate and share information with each other and with others outside that organization. Hence, such type of environment will have the influence of the environment outside that organization.

#### 1.1.2 Closed/Indoor Environment

On the other hand, a closed/indoor environment is an environment which has no influence of the environment outside that particular organization. So, according to it the environment in that organization will remain same despite the external factors.

Hence, such type of environment having least concern with the external environment is serving as the best and ideal location for demonstrating this final year project.

## 1.2 Background

Different research papers have been published by different universities and respective publications that have discussed about or have mentioned similar applications [7]. For example, there is an AR bot developed by **ViewAR**, an American Company that serves the same functionality and much more by using Augmented Reality [8], and this AR bot has been mentioned in several publications and thesis [9]. Another application of this idea is implemented by **5GPPP-5G Coral**, an European company which is using this idea and helping users in the shopping malls to navigate to different shops [10]. Such type of applications are being used in developed countries hence, we want to bring such advancement in our country as well.

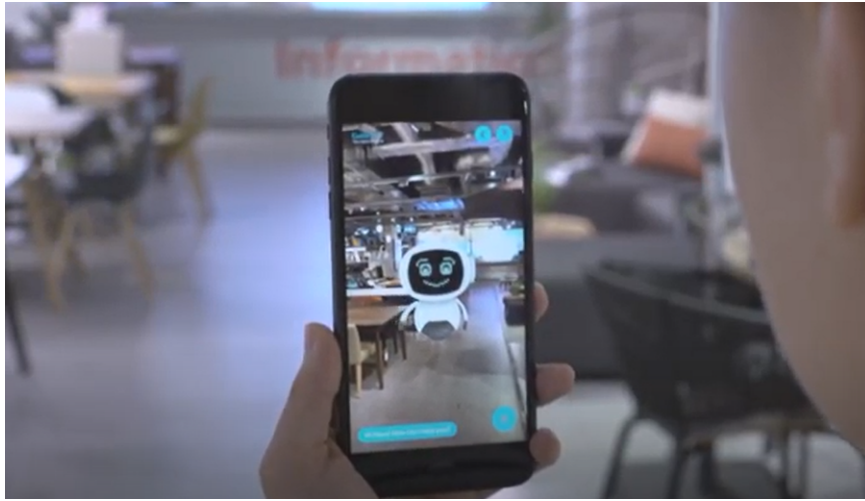


Figure 1.1: Visual presentation of the AI based AR chat bot [1]

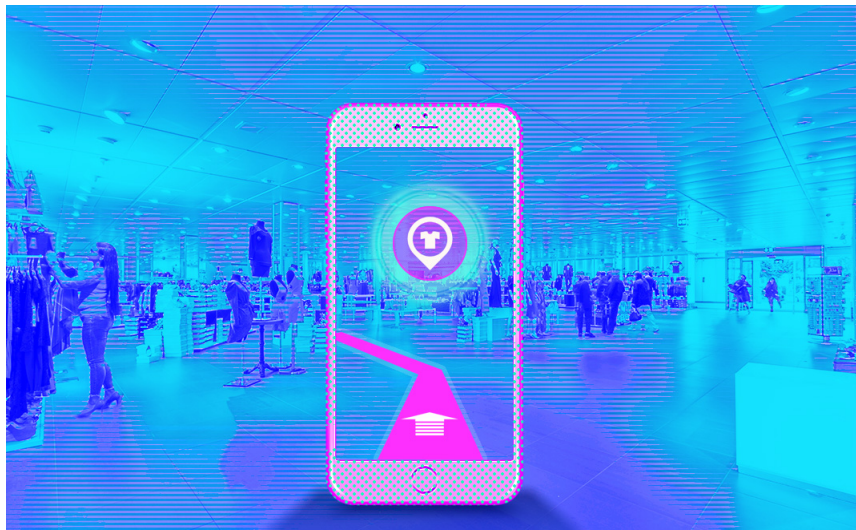


Figure 1.2: Visual presentation of AR based navigation system [2]

## 1.3 Problem Description

In this modern era of technology, augmented reality is in a surge and developers are utilizing this technology to make various applications that provide real helpful solutions for the world. We should have observed that whenever we visit a shopping mall, a new office, a new university, the first thing we do is to look for someone or some assistance to reach our desired location. It is not always easy to reach our desired location on our own. Sometimes, we get misguided and seeking assistance in order to know the right way can be time consuming.

However, our application aims to guide the new visitors without others guidance. With the help of this application we will be able to make human interaction for that purpose as low as possible because anyone with a smartphone can use this application to navigate through different locations in a closed environment. There are some other projects which are being processed on. But our idea is to reduce the use of hardware because many of these projects work on the basis of GPS hardware and sensors. We are using AR and digital pointers (anchors) in order to cater this issue.

## 1.4 Project Objective

The objective of this project is to develop an interactive application that will help its users to navigate through different locations within a building or small closed environment using augmented reality. We are using home and university as the indoor environment in this case.

- There is an AR bot based on AI that will communicate with the user while using the smartphone. This AR based application will be developed using AR Core and Unity 3D.
- There are digital anchors placed in different places in that indoor environment.
- The AR based navigation lines will guide the customer or user to his/her desired location with the help of these spatial anchors.
- When the user will be following the AR based bot. He will see an AR based object as soon as he is approaching his destination. It will indicate him that it's his destination .

## 1.5 Project Scope

This project aims to make the life of users easy. We are using a indoor environment for the project. The Spatial anchors will serve as the basis of navigation. This idea will reduce the human interaction in order to get guidance to go to a location within that environment. But in future if we want to use an open environment, we will have to make some amendments in the project. We might require to use the GPS or any other method in order to work perfectly in an open environment. Moreover, this application will initially be available for android users.

## Chapter 2

# Literature Review

For the purpose of providing an interactive and simple interface to the users, several researchers have utilized AR technology in developing an application that offers indoor navigation. However, this technology has been used with several methods to maximize its advantage and convenience.

The existing applications and research papers suggests that Augmented Reality has been integrated in various applications previously, but it has never used been in the same way. However, in order to achieve this integration, we first must know what it actually means and provide a detailed study of individual components.

### 2.1 Augmented Reality

A real-time direct or indirect view of a real-world environment that has been enhanced by the addition of virtual computer-generated information is characterised as Augmented Reality (AR). [3]

According to Borko Furht and Julie Carmigniani, AR is defined as a system that combines these three features:

1. A combination of real and virtual worlds
2. Real-time interaction
3. Accurate 3D registration of virtual and real objects [11]

Augmented Reality is not to be confused with Augmented Virtuality which is a completely different concept. However, combining these two concepts give birth to Mixed Reality.

Augmented Virtuality is a situation in which a virtual world is augmented with real-world objects.

Milgram's Reality-Virtuality Continuum is characterised by Paul Milgram and Fumio Kishino as a line that connects the real world and the virtual world, with Augmented Reality (AR) and Augmented Virtuality (AV) in between shown in figure 2.1, where AR is closer to the real world and AV is closer to a pure virtual world. [12]

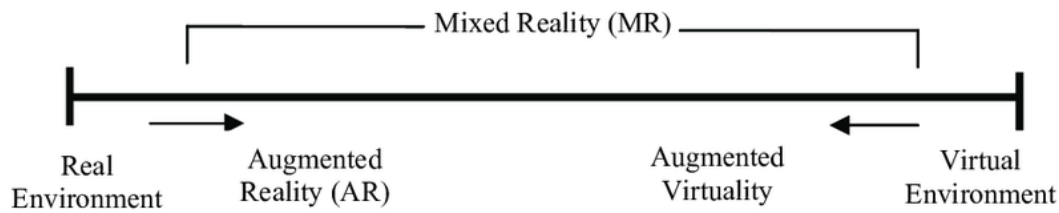


Figure 2.1: Milgram Reality Virtuality Continuum [3]

### 2.1.1 Augmented Reality vs Virtual Reality

The purpose of augmented reality is to make the user's life easier by integrating virtual information into his or her physical vicinity as well as any indirect perspective of the real-world environment. The user's perception of and interaction with the real environment is improved through augmented reality.

While Virtual Reality (VR) technology, or Virtual Environment as Milgram refers to it, completely immerses users in a synthetic world without allowing them to see the real world [12], AR technology improves the perception of realism by projecting virtual items and cues onto the real world in real time.

## 2.2 Applications of Augmented Reality

Considering this technology, we must know it is not only restricted to a certain type of display technologies such as head-mounted display (HMD), nor it is considered to be limited to the sense of sight.

Augmented Reality has vast areas of application and this technology can be used for several purposes.

As Azuma et al. [13] stated, AR can be applied to all senses, augmenting smell, touch and hearing as well. AR can also be used to enhance or substitute users' missing senses through perceptual substitution, such as using audio cues to augment the sight of blind or low-vision users, or using visual elements to augment the hearing of deaf users.

Azuma et al. [13] further considered AR applications that necessitate the removal of real elements from the environment in addition to the insertion of virtual objects which are known as *mediated* or *diminished reality*.

Borko Furht explained in his book this concept by Azuma as [11], taking objects out of the actual world entails covering them with virtual information that blends in with the background to provide the user the sense that they aren't there. Virtual items placed in the actual world provide information to the user that he or she cannot detect directly with his or her senses. The data carried on by the virtual item can assist the user in doing daily chores, such as aiding employees through electrical lines in an aeroplane by portraying digital data through a display head gears. [13]

The contents can also be used for entertainment purposes, as in Wikitude or other mobile AR applications [14].

The other applications of AR are classified as:

- Visualization in medical
- Entertainment
- Advertisement
- Repairing and maintenance
- Annotation
- Path Finding Robots and much more.

### 2.3 Related Researches

When Morton Heilig, a cinematographer, thought of film as an activity that might bring the viewer into the onscreen action by efficiently integrating all of the senses, that was when the AR featured for the first time in history in the 1950s. In 1962, Heilig built a Sensorama prototype that preceded digital computing and was based on his vision, which he described in 1955 in "The Cinema of the Future." [15]. Ivan Sutherland invented the head mounted display in 1966. In 1968, Sutherland was the first to design an augmented reality system using an optical see-through head-mounted display.

In 1975, Myron Krueger designed the Videoplace, a room that allows viewers to engage with virtual items. Later, Boeing's Tom Caudell and David Mizell introduced the term Augmented Reality, when assisting employees with the assembly of cables and cabling for an aircraft [3].

In the same year, L.B Rosenberg developed Virtual Fixtures, one of the first operational AR systems, and demonstrated its impact on human performance. Steven Feiner, Blair MacIntyre, and Doree Seligmann, on the other hand, presented the first comprehensive study on the KARMA, which is an AR system prototype. [3]. Paul Milgram and Fumio Kishino described the reality virtuality continuum as a continuum that covers the real and virtual worlds in 1994. [12]. In 1997, Ronald Azuma published the first survey in the field of AR, characterising it as a hybrid of real and virtual worlds that are both registered in 3D and are interactive in real time. [16].

Bruce Thomas developed the first outdoor mobile augmented reality game in 2000, ARQuake, which he presented at the International Symposium on Wearable Computers. The Horizon Report 2005 [17] predicted that AR technologies will mature in the next 4–5 years. Camera systems that can evaluate surroundings in live time and associate positions between items and the surroundings were developed in the same year to validate the prediction. This type of camera technology has formed the cornerstone for combining virtual objects with reality in AR systems.

Moreover, several other AR applications were produced in the upcoming years, particularly with mobile applications, such as Wikitude AR Travel Guide [14], which was introduced in 2008, however with the creation of medical applications in 2007.

Automobile manufacturers also introduced the concept of AR in 2013, developing the MARTA app (Mobile Augmented Reality Technical Assistant). Google released Google Glasses in 2014. Then in 2016, Microsoft released the HoloLens.

Google and Apple released AR core and AR kit in 2017 to help developers construct Augmented Reality apps. Google announced the addition of augmented reality (AR) technology to Google Maps in their 2018 Keynote [18].

Today, AR developers have a variety of software platforms to select from, but these model systems remain key research directions.

## 2.4 Navigation

According to Daniel R Montello [19], navigation is the coordinated and targeted movement of individuals or intelligent machines through their surroundings. It entails both movement planning and execution.

Navigation can be divided into two components:

1. Locomotion
2. Wayfinding

Locomotion is the movement of individuals directed towards the local surroundings; whereas, wayfinding is the process of planning and making decisions about one's surroundings, both close by and far away.

There are several navigation systems developed for these two surroundings:

1. Outdoor navigation
2. Indoor navigation

### 2.4.1 Outdoor Navigation

Outdoor navigation entails navigating in natural settings such as hiking routes. Hikers, backpackers, geocache enthusiasts, mountaineers, and bikers, as well as outdoor workers such as emergency responders, wildlife preservationists, forestry workers, farmers, and miners, employ outdoor navigation [20].

### 2.4.2 Indoor Navigation

An Indoor Navigation System is a system that assists users in navigating in an indoor setting by displaying a path to their goal from the user's current location [21]. Our research mainly focused on indoor navigation.

### 2.4.3 Methods For Navigation

As discussed above surroundings, following methods are used for indoor and outdoor navigation:

- Conventional maps
- Global positioning system (GPS)
- Augmented Reality (AR)
- WiFi positioning system
- Vision-based localization system

### 2.4.4 Maps

A map is a representation of space that emphasises connections between objects, places, or themes. A map does not accurately portray reality. It's a symbolic representation of a location that highlights the relationships between visible and unseen things in environment. A printed map is known to as a traditional map..

According to K. Salishev [22], a map consists of three basic elements. The map's essential components are:

1. Content
2. Mathematical foundation
3. Legend

The information about various natural and man-made objects in the map is displayed in the content. The mathematical foundation displays the coordinates of a given object or other data. Aside from that, the map also defines the spacing between blocks. The legend of the map, which is the third component, usually displays more detailed information about the objects on the map.

A map does not and cannot depict all that exists in a location. For the map to make sense, things must be omitted, simplified, and so on. This is how a map achieves clarity and utility: it removes world elements to allow the map's purpose to come through.

### 2.4.5 Global Positioning System

The global positioning system (GPS) is a a positioning approach that incorporates the utilisation of a satellite as a transmitter and GPS as a receiver. Although the GPS does not require the user to send any data and operates independently of telephonic or Internet reception, both technologies can help enhance the precision of GPS positioning data. Military, civic, and users in the commercial sector all across the globe rely on the GPS for crucial positioning.

According to Dennis Roddy [23], the system operates with at least 4 satellites orbiting at 13,000 miles (20,000 kilometres) above Earth and travelling at an average

speed of 8,700 miles per hour. The satellite transmits a signal to the receiver, and the time it takes for the signal to transit from transmitter to receiver is measured, allowing the position in three dimensions to be computed: east, north, and altitude. Aside from that, the GPS receiver requires two sorts of data: almanack and ephemeris. These two sorts of information ensure that the transmitter and receiver are in communication in order to obtain an accurate location result.

### 2.4.6 Augmented Reality

Augmented reality is a technique that uses a live direct or indirect view of a physical, real-world environment, with sensory input generated by a computer, such as music, video, visuals, or GPS data to enhance the experience.

Mark Altaweel [24] discussed the use of AR for navigation as a technology for understanding the actual world and integrating virtual components that adhere to physical limitations so that items presented on a screen are visually evident.

In AR navigation, there are frequently two steps to complete. To begin, ascertain where someone is in relation to the Earth's location, which requires using GPS and sensor data to apply IMUs to determine the location and trajectory of a moving object. Additionally, that information is used to determine where specific components should be placed in relation to a navigation user.

W. Narzt [25] also discussed the concept of AR in navigation by stating that this approach necessitates the use of GPRS, which obtains information about a specific geographic position that can be layered with tags and other information. This place can have images, videos, and other media overlay on it. A virtual path will be displayed by the navigation system that correlates to the real-world environment.

### 2.4.7 WiFi Positioning System

A WiFi positioning system is a positioning system that determines a device's location by analysing the characteristics of neighbouring WiFi hotspots and other wireless access points. When satellite navigation, such as GPS, is insufficient because to a variety of factors such as multipath and signal blocking indoors, or when obtaining a satellite fix, it is utilised. [26]

According to the research paper by N. Kawaguchi [27], and the following facts, which can be used to estimate WiFi location for both indoor and outdoor navigation:

1. Every WiFi access point has a unique identifier (ID) called Basic Services Set Identification (e.g. MAC address).
2. The Basic Services Set Identification (BSSI) of each WiFi access point (AP) is broadcasted on a regular basis.
3. The WiFi APs can broadcast Basic Services Set Identification to any WiFi client.

WiFi AP location or Data Slot Set Identification can be used to estimate a user's location. There are various ways for estimating position, however they are all generally classified into three types: proximity, triangulation, and scene analysis.

### 2.4.8 Vision-Based Localization System

For relative or global posture estimation, the goal of vision-based positioning is to match current images with prerecorded frames or models. According on the recorded data for matching, such algorithms can be divided into three groups: series of frames, image sets, and 3D models.

According to Joseph J [28], it is method for indoor navigation which marks the location with QR codes, bar codes, and other images.

The marker may retrieve the data specified by the developer via a scanner scan, and it will reveal the user's location in an indoor setting. The user can choose where they want to go and it will begin at the marker.

The advantage is that it does not necessitate a lot of gear to link the transmitter and receiver. Whereas, it does not feature real-time navigation, which is a disadvantage.

### 2.4.9 Comparison Between These Methods

Method	Outdoor	Indoor
Conventional map	Yes	Yes
Global positioning system	Yes	No
Augmented reality	Yes	Yes
WiFi Positioning System	Yes	Yes
Vision-Based localization system	No	Yes

Table 2.1: Comparison table between navigation methods

In the above table 2.1, the different methods of navigation has been compared and it has been shown that which of the methods provide support to indoor and outdoor navigation.

## 2.5 AR-Based Navigation

Our research was focused on AR-based indoor navigation. AR-based navigation can be categorized into three groups depending on different positioning technologies utilised in research papers:

1. Marker-based method
2. Method based on 2D image recognition
3. Method based on 3D image recognition

### 2.5.1 Marker-Based Methods

Markers are used as location anchors in marker-based approaches and are first deployed in an indoor environment. QR-Code or a customized pattern can be used as a marker. Each marker is universally unique identifier (UUID) and coordinates are pre-stored for future queries in either the smartphone's internal storage or a distant database. When looking for a marker, the user just puts his or her smartphone camera at it and scanning is done. The position of user and AR based model is then determined using the scanned image. A. Mulloni [29], G. Reitmayr [30], and C. Feng [31] utilized clearly recognized images as markers.

Every marker serves as a node on the navigation path. An AR based arrow model displays on the smartphone's screen when a user approaches a marker and directs the camera at it, directing the user to the other node. Although markers are simple to use, further user training is needed to assure that a marker-based navigation system is successful.

let us suppose, before acquiring position information and moving onward, users should be allowed to recognise landmarks in the surrounding area. It's especially tough for someone who has never seen a marker before. Moreover, The marker and the camera must be aligned to achieve accurate marker recognition, which might be inconvenient for the user.

### 2.5.2 Method based on 2D image recognition

Unlike marker-based approaches, 2D image recognition-based systems establishes the user's location by looking for pre-annotated things in the environment, like trademarks, billboards and signs. With the advancement of image recognition technology, the world is going more towards image recognition-based navigation and replacing marker-based navigation. Even if image recognition-based methods may not require maintenance, they may struggle when two items are too similar to identify.

In a large shopping mall, for example, chairs, signboards, and decorations are typically constructed in a similar manner, making image recognition much more difficult. Many other parameters might affect the accuracy of identification of images such as:

- Viewing angle of camera
- Distance of the user and the items
- The number of moving things in the immediate vicinity etc

Both G. H. Nam et al. [32] and J. Wu et al. [33] utilized the technology of recognizing the image for indoor navigation.

### 2.5.3 Method based on 3D image recognition

Method based on 3D image recognition capture properties of the entire space rather than the characteristics of 2D objects, as opposed to Method based on 2D image

recognition. A feature matching system is employed to establish a location of the user during runtime, and then an AR engine is used to generate a 3D model for direction indication.

G. Gerstweiler [34] and U. Rehman [35] employed Metaio SDK in order to create a 3D point cloud for indoor localization in their application.

T. Rustagi [36] collected vector data from the indoor environment and created a 3D based model using MapBox API. Their system includes the Unity game engine to display AR based models.

Despite 3D models may provide improved precision for indoor localization, they are expensive and time demanding to create, especially for large buildings. Furthermore, when the inside plan of a shopping mall or an exhibition hall is changed more often, therefore re-modeling of the system is required. In addition to that, moving objects inside the environment may diminish location accuracy dramatically.

## Chapter 3

# Software Requirement Specification

In this chapter we will discuss the requirements for our proposed project. We will talk about the already existing systems related to our project and how will our project be different from them. In the previous chapter, various methods that can help in navigation have already been discussed.

### 3.1 Existing System

The basic method that is being used in navigation is conventional maps. Other existing system that are already being used in modern navigation are, GPS and WiFi positioning system. Both the methods, using GPS or WiFi based positioning system uses the hardware components in order to work sufficiently. GPS based navigation system uses a satellite in order to help in navigation where satellite is a transmitter and GPS is a receiver. Whereas, WiFi position system uses neighbouring WiFi hotspots and other wireless access point in order to achieve authentic navigation. Moreover, use of hardware makes these systems bit costly as compared to a software based systems.



Figure 3.1: Image illustrating various navigation methods [4]

In this figure 3.1, various navigation methods are presented including the conventional one and the ones we use currently i.e. GPS and WiFi Positioning System.

### 3.2 Proposed System

Our system is based on **AR Core** for its functionality. AR Core is a software development kit by Google that allows the development of AR-based applications [37]. The application serves by helping users to navigate and 3D models making the experience interactive as they engage the while using the application for navigation. Spatial anchors that are actually digital pins, can be placed in any physical location. The application is able to use these digital pins to navigate users to their desired location. These digital pins are placed in various location in the building layout. The shortest distance of the various well known locations in that environment will be determined using the shortest path search algorithm. User will use his/her smartphone camera to view the AR based arrows and other application elements. Users will be interacting with an eye-catching application using Augmented Reality resulting in a fun filled guiding experience.

### 3.3 Work Flow

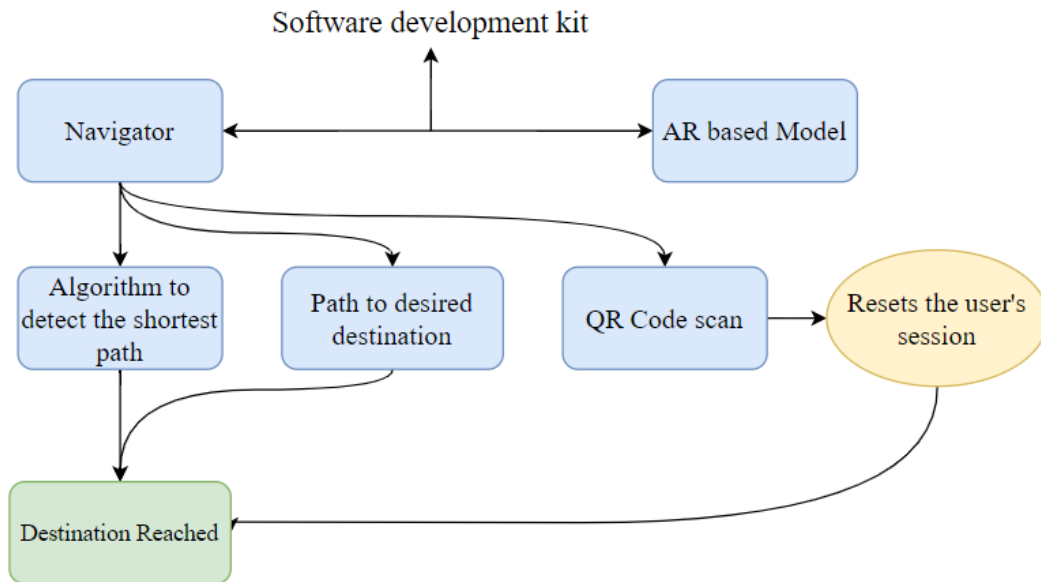


Figure 3.2: Image illustrating work flow of the system

The figure 3.2 illustrated above shows the work flow of how our FYP is working. Software Development Kit (SDK) is integrating both navigator and AR based model that are the main modules of the system. Navigator further has further jobs done that are calculating the shortest path for the destination, guiding the user about the path and QR code scan that resets the session of the user.

## 3.4 System Requirements

The requirements based on system are of 2 types,

1. Software Requirement
2. Hardware Requirement

### 3.4.1 Software Requirement

The AR based Navigation system will require,

- **AR Core**

AR Core is a software development kit by Google that allows the development of AR-based applications for android OS [37]. Our AR based application will be developed using AR Core.

- **AR Foundation**

AR foundation is also a software development kit that enables us to develop AR based application for both iOS and android.

- **Unity 3D**

Unity is basically a gaming engine, but it allows in the development of 3D models. We will use unity in order to develop the main functionality of the application.

- **Visual Studio**

Visual studio is an Integrated Development Environment by Microsoft that is used to write programs in different languages. We will need it in order to write scripts and programs in C# for our application.

- **Inkscape**

Inkscape is a free and open-source vector graphics editor used to create vector images, primarily in Scalable Vector Graphics format. We will be using it for creating 3D buttons for our application.

### 3.4.2 Hardware Requirement

- 4GB RAM smartphone
- Screen Resolution (1024 x 768 minimum)
- Minimum Storage 32GB

- Mobile phone supporting AR
- Smartphone should be running on Android OS

### 3.5 Software Requirements

We will be discussing software requirement in this section. Functional and non functional requirements of the project are mentioned below,

#### 3.5.1 Functional requirements

Functional Requirements specify what system should do. It's a mandatory phase of requirement specification. As, our system main aim is to navigate the user, so the functional requirements will be according to it. The functional requirements of our project are as follow:

- System should be able to start the navigation from the starting point.
- System should be able to find the shortest distance between the initial source and the destination of the user.
- System should be able to detect the digital targets placed in the environment.
- System should be able to see the AR based navigation path.
- System should be able to navigate user to the multiple destinations within the environment.
- System should be able to reset the session from where the QR code is scanned despite of the user being to the assumed starting point.
- System should be able to notify user if he/she is going off the track.
- System should be able to navigate user correctly to its desired location.
- The surrounding light should be good in order to navigate the user.

#### 3.5.2 Non-Functional requirements

Non functional requirements specify how the functional requirements of the system are fulfilled. The non-functional requirements of our project are as follow:

- **Performance**

System should be able to respond the user timely. Hence making the experience smooth.

- **Reliability**

The system should be reliable in order to navigate and respond the users. It should be able to tackle basic failures related to the system.

- **Maintainability**

The system should be convenient to maintain by the developers and ready to receive future system updates.

- **Usability**

The system should be easy to use and interactive in order to enhance its usability. The interface and the application itself should be user friendly.

- **Serviceability**

The service offered by the system i.e. navigation, should be available any time when the user of the system needs it.

### 3.6 Use Cases

Our application is intended for the users from various fields of life. All they need is a smartphone and they will be able to use our application. The user aims to navigate in the closed/indoor environment they are present without any human interaction. The user has to scan the initial marker in order to proceed further. After scanning the first marker, user can view the first digital pin and the AR based arrows in order to start his/her journey towards the desired location. Approaching the first anchor will indicate the next anchor and this process will continue until the user reaches its desired location specified earlier in the application.

Some of the use cases of the system are shown below:

### 3.6.1 Overall Use case

The following figure 3.3 shows the use case for overall working of the application.

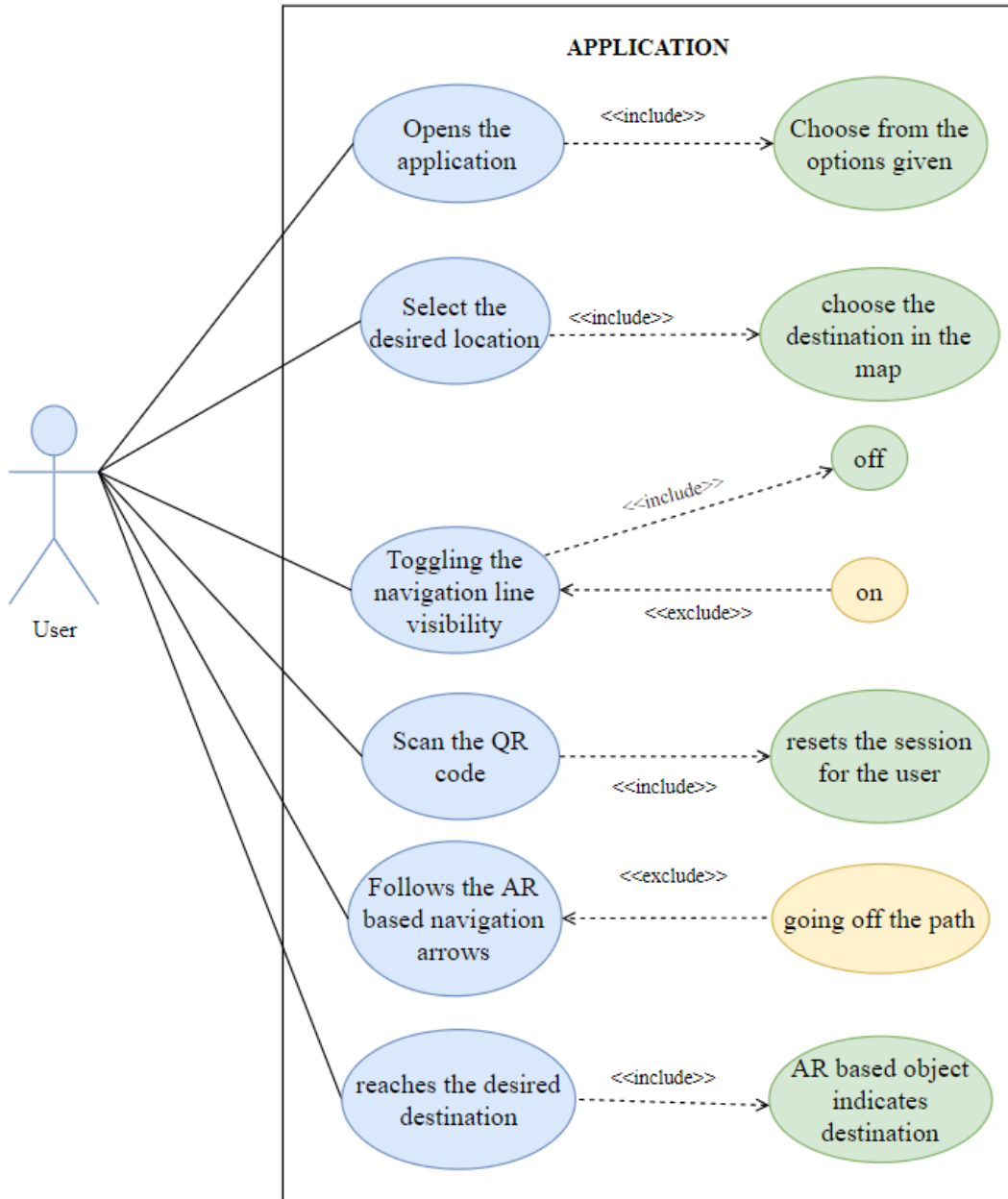


Figure 3.3: Use case for overall working of application

### 3.6.2 Interacting with the Application

The following figure 3.4 and table 3.1 shows the use case of how the user will launch the application and interact with it. There will be options provided from which the user will choose.

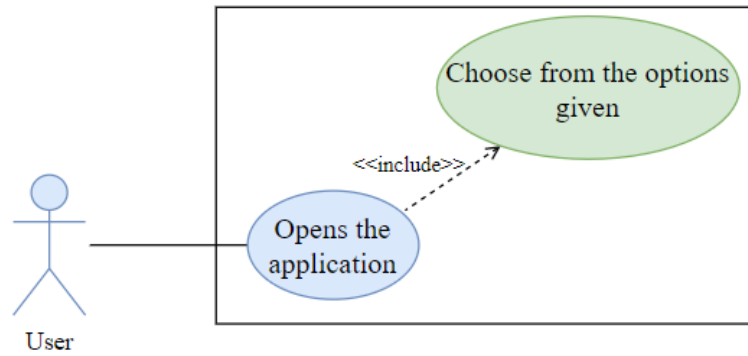


Figure 3.4: Use case for opening the application

**Tabular Description**

Use case ID	UC 1
Name	Interacting with the application
Description	User will launch the application from his/her smartphone. Application works finely and user starts using it.
Actor	User
Pre condition	User should be having smartphone
Post condition	Application has been launched
Success Scenario	Application launched successfully and user is interacting

Table 3.1: Description Table for use case 1

**3.6.3 Selecting the desired location**

The following figure 3.5 and table 3.2 shows the use case for the user selecting the map and choosing the location with in the selected map.

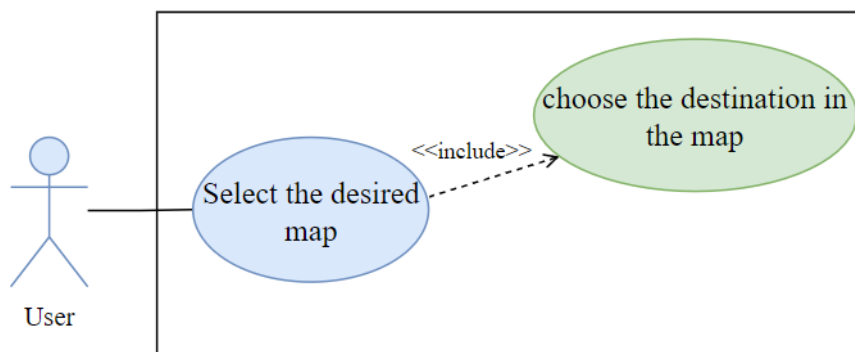


Figure 3.5: Use Case for choosing the desired location

### Tabular Description

Use case ID	UC 2
Name	Selecting the desired location from the menu
Description	After application was launched user will proceed further and select the map in which he/she wants to navigate. After selecting the map they will choose the destination with in the map from the list.
Actor	User
Pre condition	Application must be opened
Post condition	User has selected the map and then the destination with in the map from the list
Success Scenario	Location selected by user has been successfully recognized

Table 3.2: Description Table for use case 2

#### 3.6.4 Toggling the navigation line visibility

The following figure 3.6 and table 3.3 shows the use case of user toggling the visibility of the navigation line. By default the visibility is turned off and it can be turned on.

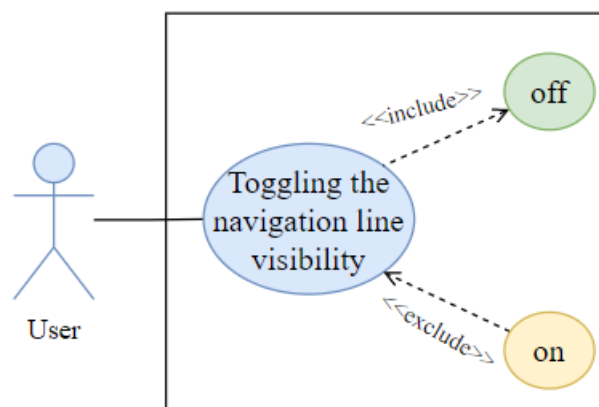


Figure 3.6: Use Case for toggling the navigation line

### Tabular Description

Use case ID	UC 3
Name	Toggling the visibility of navigation line
Description	When the user has selected the destination with in the map, he can use a toggle button to switch the visibility of navigation line on or off.
Actor	User
Pre condition	Destination must be selected
Post condition	AR based navigation line is visible
Success Scenario	Visibility of the navigation line can be turned on or off

Table 3.3: Description Table for use case 3

### 3.6.5 Scanning the QR code

The following figure 3.7 and table 3.4 shows the use case of user scanning QR code in order to make that point as the starting point and resetting the session for the user.

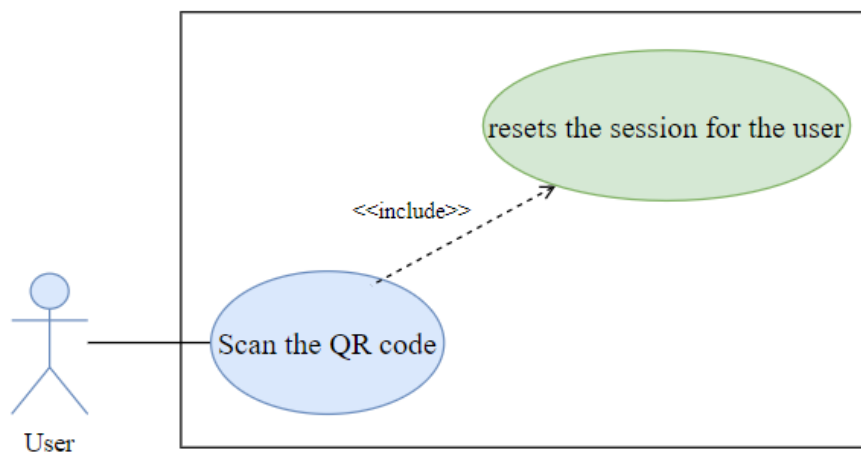


Figure 3.7: Use case for scanning the QR Code

### Tabular Description

Use case ID	UC 4
Name	Scanning the QR Code
Description	If the user hasn't start the navigation from the assumed starting point, he can scan the QR code. It will make that point starting point and reset the session for the user.
Actor	User
Pre condition	User has selected his/her desired location
Post condition	User has scanned the placed physical QR code
Success Scenario	The session has been reset for the user

Table 3.4: Description Table for use case 4

### 3.6.6 Following the AR based navigation arrows

The following figure 3.8 and table 3.5 shows the user following the AR based navigation arrows in order to get to his/her desired location.

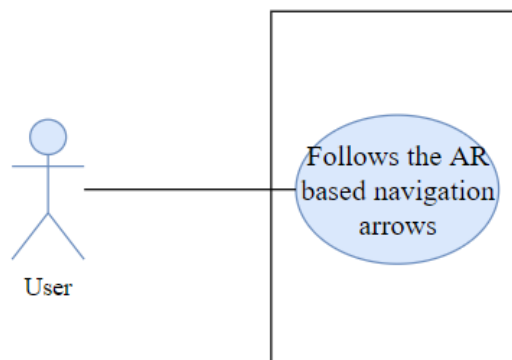


Figure 3.8: Use case to follow digital markers

**Tabular description**

Use case ID	UC 5
Name	Following the AR based navigation arrows
Description	The toggling of the navigation line will make a AR based navigation line to appear. Following this line will help user to reach the destination.
Actor	User
Pre condition	User has selected the desired location successfully
Post condition	User is using his smartphone to view digital navigation arrows
Success Scenario	User is following the digital arrows

Table 3.5: Description Table for use case 5

**3.6.7 Reaching the destination**

The following figure 3.9 and table 3.6 shows use case for the user after following the AR based navigation arrows and approaches the destination indicated by AR based object.

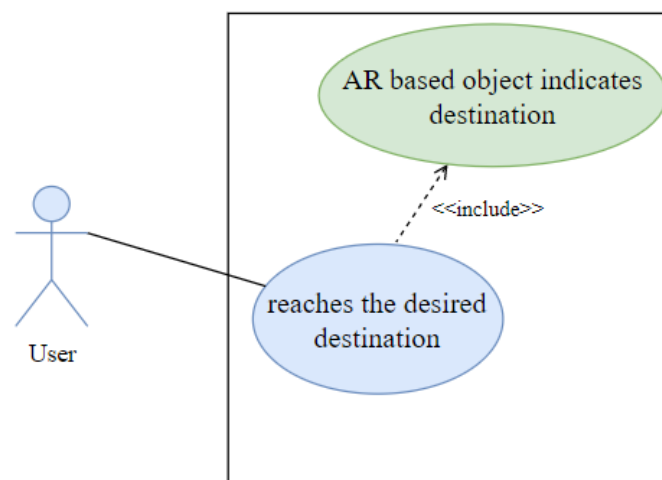


Figure 3.9: Use case for user reaching the desired location

**Tabular Description**

Use case ID	UC 6
Name	Reaching the desired destination
Description	After following the AR based arrows, the user will be approaching towards his/her destination. The destination will be marked by AR based balloon and the 3D bot will show animation upon reaching the destination.
Actor	User
Pre condition	User was following the AR based navigation arrows
Post condition	User is approaching towards his/her desired destination marked by AR based balloon
Success Scenario	User has successfully reached on his desired destination

Table 3.6: Description Table for use case 6

## Chapter 4

# System Design

In this chapter, we have discussed the development phase of our AR based navigation system. The objective of this chapter is to get information about the basic flow of our application to convert the proposed solution into the reality. This chapter includes the proposed system architecture, used to describe the abstract view and conceptual model of our application. The design methodology and interface of our system are discussed in detail to get the idea of interaction between user and system interface. We have also provided information about our system components, modules, interfaces, and data to satisfy specified requirements in earlier sections. The chapter's detail discussed below:

### 4.1 System Architecture

The architectural view of the system is shown below in the figure 4.1. It describes about the architectural design of the system.

- The user will need a smartphone in order to proceed with navigation.
- The navigation will be based on the algorithm to calculate the shortest path.
- The path from the user and the destination will be determined using the A\* search algorithm which is best in situation where we are dealing with maps.
- This will ensure the selection of the shortest path to reach the destination.
- The spatial anchors provided by AR Core are used as targets.
- The algorithm for assigning the path will be assigned in such a way that if the user gets off the track, he/she is notified regarding that.

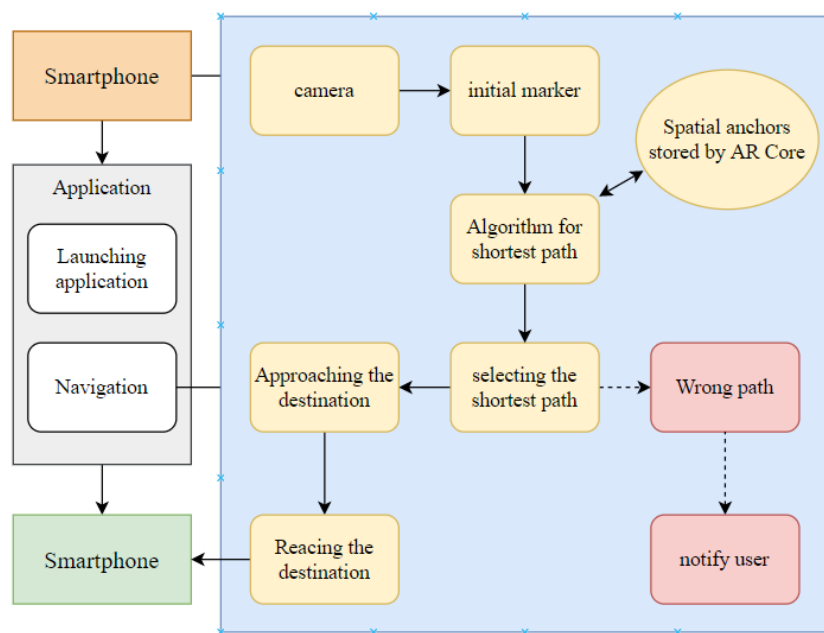


Figure 4.1: System Architecture design of the system

## 4.2 Design methodology

The development of the project is directly linked to the software development model we are using. In our case, we are using **incremental model**. Development of the project requires analysis of the closed environment we are implementing the idea and the planning of how this project will be achieved. After that we are designing the basic layout of the application and its features. After having a good idea of the requirements and specifications of the proposed project, the project is implemented.

## 4.3 Design Constraints

Design constraints in this project are:

- A user having least knowledge of using a smartphone.
- Our application will be initially basing on Android operating system. However, a user having an iPhone won't be able to use this application.
- Our application will require AR services in the smartphone. A smartphone having no AR services won't be able to run this application.
- Spatial anchors in our application will be pinned in a closed environment. Hence, there will be no support for open environment for the time being.

## 4.4 Context Diagram

The figure 4.2 illustrated below shows the contextual view of the system. The user interacts with the application. Selects his/her destination from the menu provided

and the navigator will guide the user towards the destination. User can scan the QR code which will result in resetting the session for the user and that place will be user's starting point.

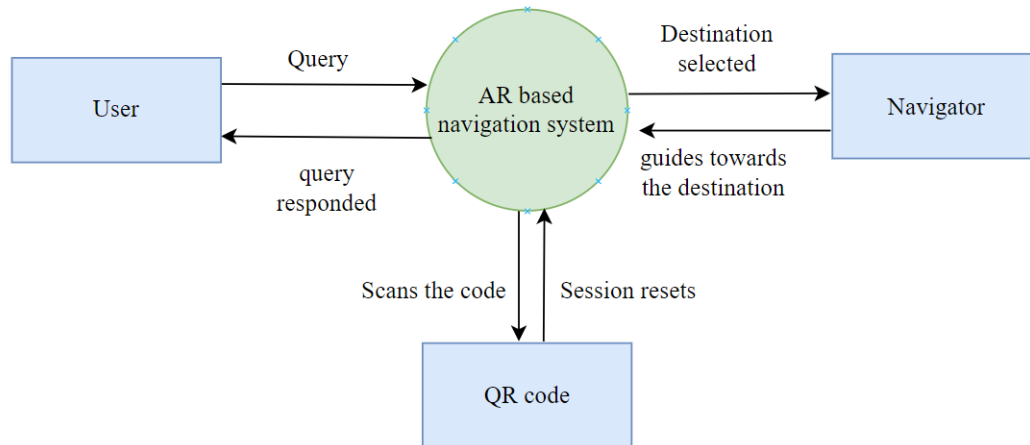


Figure 4.2: Contextual design of the system

## 4.5 Sequence Diagram

Sequence Diagram represents the interaction between various objects of the system in a single scenario. Based on our proposed project there are multiple scenarios on the basis of user and its interaction with the application and the interaction of system components itself.

### 4.5.1 Interaction with the application

The illustrated figure 4.3 below shows the interaction of the user and the application. User has to launch the application and the application should be in running state in order to proceed further.

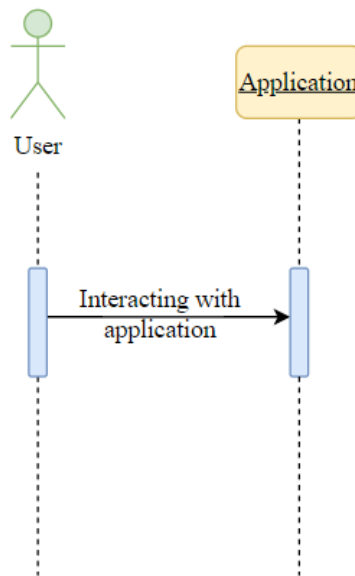


Figure 4.3: Sequence diagram for interaction with the application

#### 4.5.2 Selection of the desired location

The illustrated figure 4.4 below shows the sequence diagram for the user selecting the desired location from the menu.

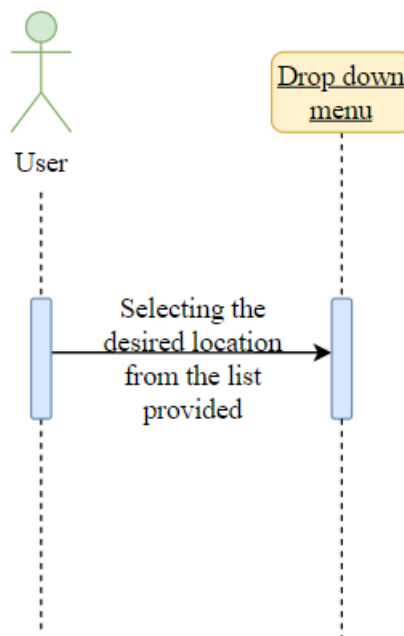


Figure 4.4: Sequence diagram for selection of the destination

### 4.5.3 Toggling the visibility of navigation line

The illustrated figure 4.5 below shows the sequence diagram for the user toggling the visibility of AR based line on or off .

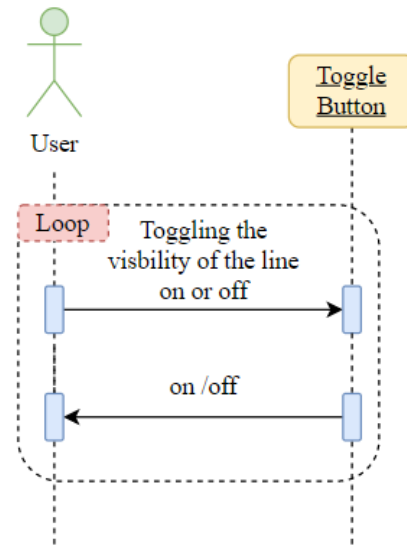


Figure 4.5: Sequence diagram for toggling the visibility of navigation line

### 4.5.4 Following Navigation Line

The illustrated figure 4.6 below shows the sequence diagram for the user who will follow the AR based navigation arrows in order to reach destination.

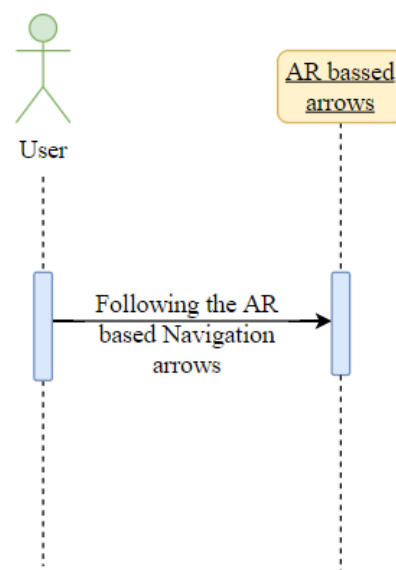


Figure 4.6: Sequence diagram for following navigation line

#### 4.5.5 Scanning the QR Code

The illustrated figure 4.7 below shows the sequence diagram for the user scanning the QR code. The scan of the QR code will result in resetting the session of the user.

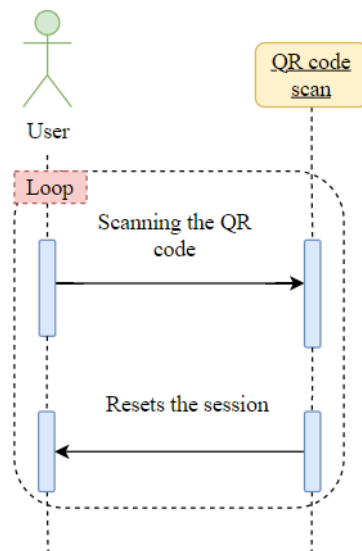


Figure 4.7: Sequence diagram for scanning the QR code

#### 4.5.6 Reaching the Destination

The illustrated figure 4.7 below shows the sequence diagram for the user after following the navigation arrows reaches the destination. There will be an AR based balloon that will indicate that it is the destination.

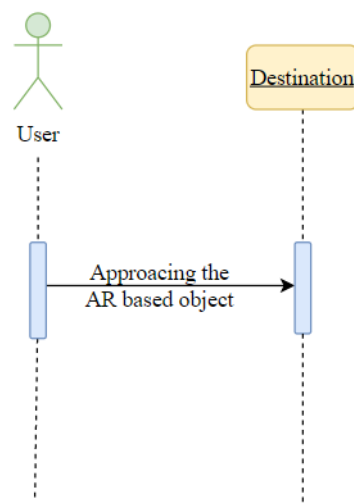


Figure 4.8: Sequence diagram for arriving on the destination

### 4.5.7 AR based Navigating Application

The illustrated figure 4.4 below shows the interaction of the user with the AR based navigating system. User will interact with the application. The components of the system interact as well in order to make it possible for the user to reach from the initial marker to the desired location.

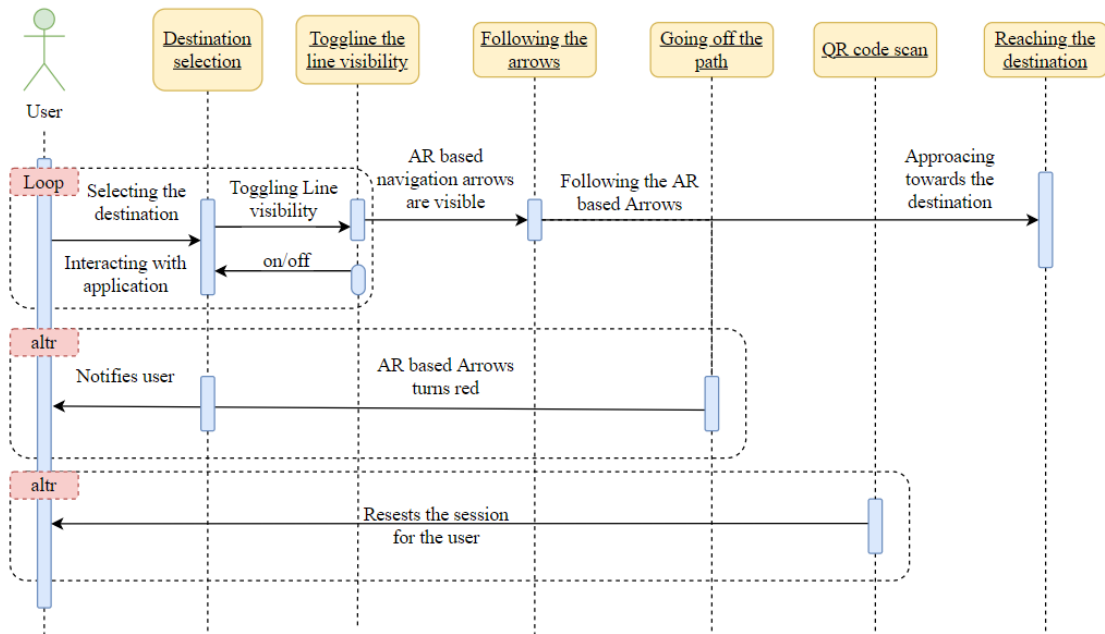


Figure 4.9: Sequence diagram for Overall system design

## 4.6 Activity Diagram

Activity diagram illustrates flow of activities from start to the end. This includes the flow of activity from user to the system and from system to components of the system. It shows how the action flows and the main procedure for the task is carried out. The diagram 4.5 shows that,

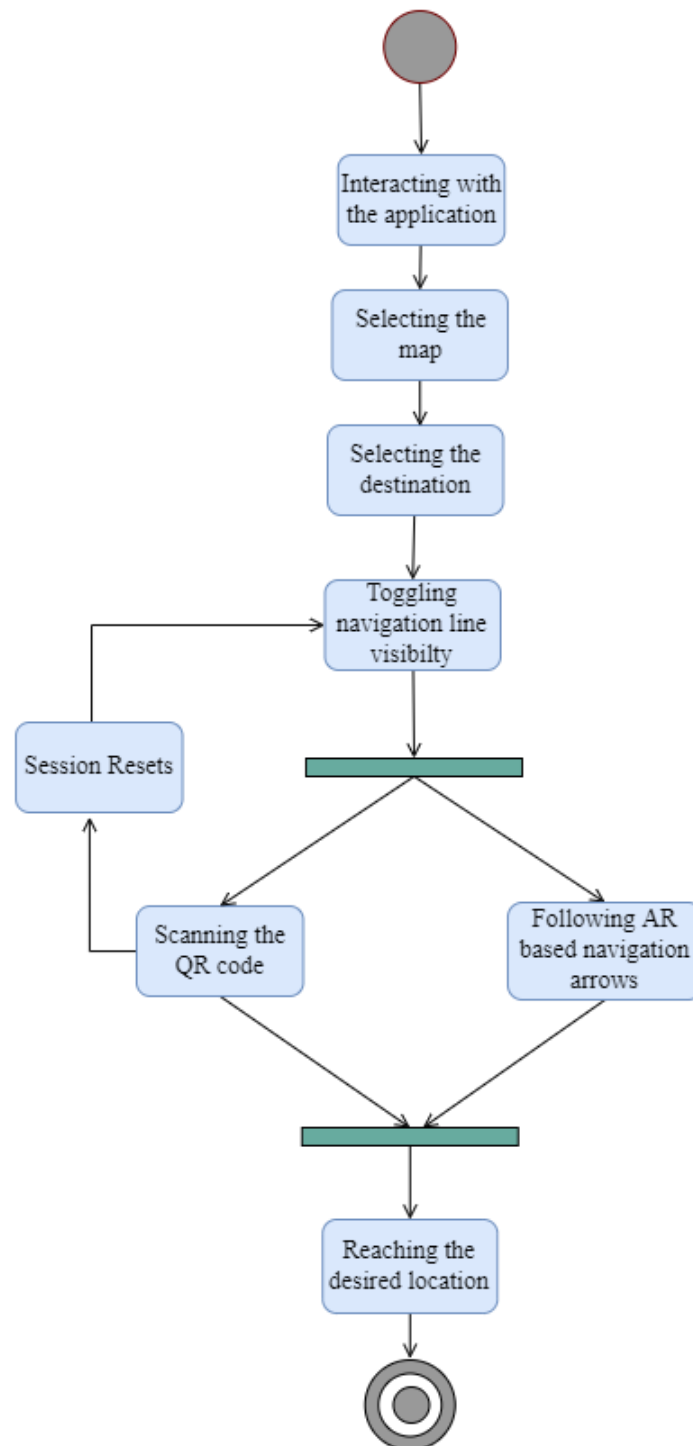


Figure 4.10: Activity Diagram for the system of the system

## 4.7 GUI Design

The application will provide easy to use and interactive interface. The user will be able to understand the application interface as it is not having any complexities. The application is a navigation application based on augmented reality that guides its users with in an indoor environment.

### 4.7.1 Front Screen Layout

This screen allows user to choose the desired location with in the environment. User can press the choose location button and choose the map.

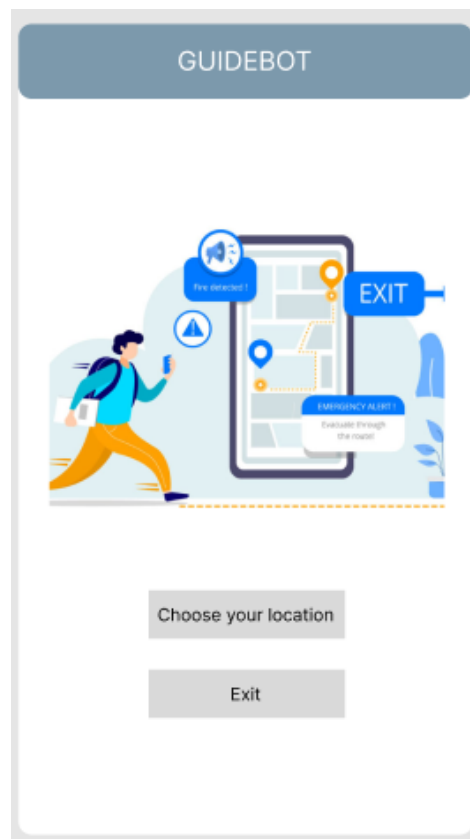


Figure 4.11: Front Screen Layout of the application

### 4.7.2 Map Screen

After user presses the button he can use the map provided by the application. He can choose the map and after that he can get to the desired locations inside that map.

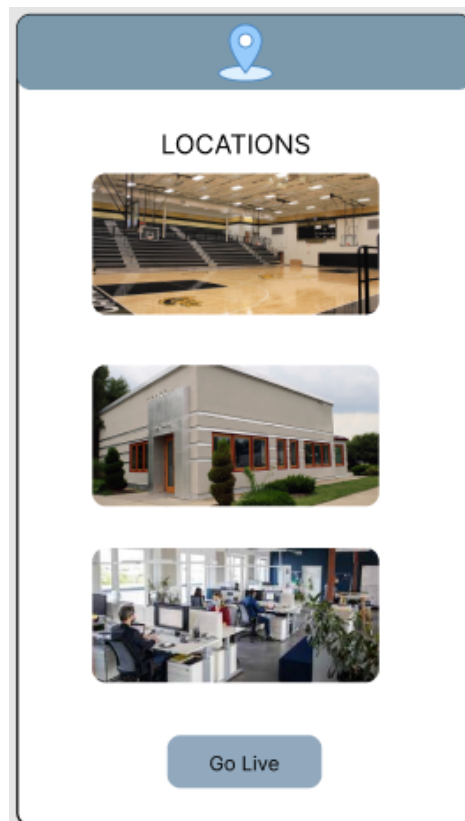


Figure 4.12: Map choosing screen of the application

### 4.7.3 Scanning the QR code

This screen allows user to scan the QR code that will as a result will reset their session from the place where they have scanned the QR code.

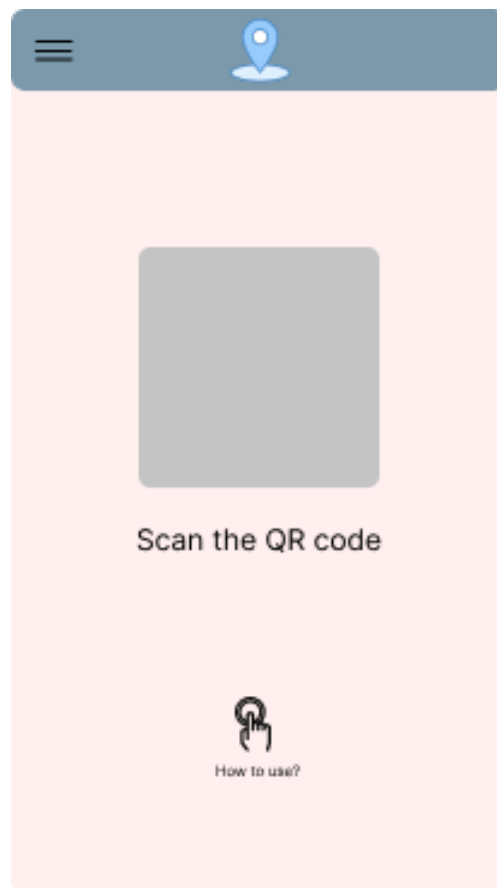


Figure 4.13: QR code scanning screen

#### 4.7.4 Navigation Screen Layout

This screen allows users view the AR based path and the mini map navigation within the indoor environment. The user can toggle the visibility of the AR based path. The target location will be marked by an AR based object to indicate that it is the destination.

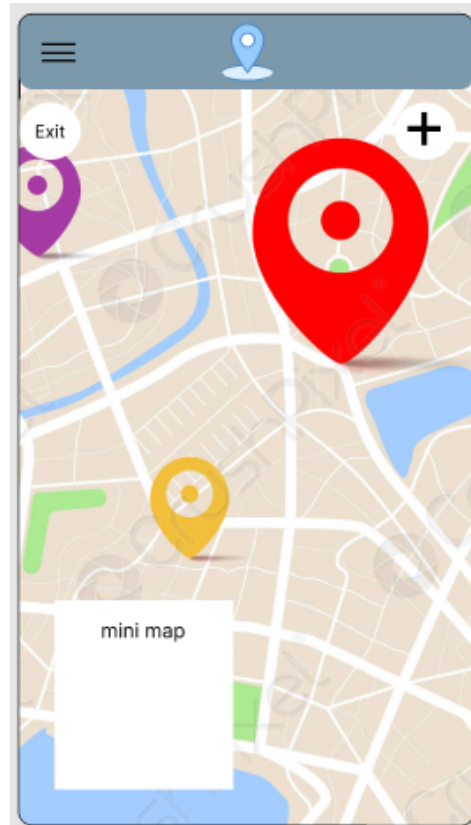


Figure 4.14: Navigation Guiding Screen

## Chapter 5

# System Implementation

This chapter includes all the steps and stages involved in the development of this application. In this phase, we will develop the application based on the requirement specification. We will be discussing tools, techniques, and procedures we have used in the implementation of the prototype and developed application in accordance with the system architecture and system requirements chapter.

### 5.1 System Architecture

The system architecture of the application can be divided into 3 different phases.

1. In first phase, we clicked different photos and analyze the environment in which we wanted to implement this indoor navigation.
2. In second phase, we created the map and placed the targets within the environment. A\* algorithm is used to calculate the shortest path from the source and various destinations and as a result path is generated from source to that particular target.
3. In third phase, an AR based path is shown from source to the destination. The visibility of that path can be toggled on/off. QR code can be scanned to initiate the session of the user again from the point where he scanned the QR code.

### 5.2 System Components

The application we developed has multiple components. Each component plays an important role in order to make this application working accurate. The components of the application are as follows,

- Analyzing the environment.
- Creating a 2D map of the environment
- Recreating 2D map into 3D map
- Placing the multiple targets with in the environment
- Determining the shortest path from source to the destination

- Generating the AR based path from source to the various targets
- Resetting the session

Each of the component of the application will be discussed below.

### **5.2.1 Analyzing the environment**

For creating the map for the indoor navigation, we will have to analyze the environment. It can be done by taking multiple pictures or by sketching the environment map. All the components especially walls, openings, and entrances have to be noted down clearly.

### **5.2.2 Creating a 2D map**

After the analysis of the environment, the 2D map of the environment has to be drawn. This map can be drawn using Paint or any other tool that allows us to draw different diagrams like Lucid chart, draw.io etc. The map should be drawn carefully. All the the rooms and areas in the environment must be replicated logically according to the measurements. Walls must be replicated accurately. It is one the complex stages of the project as to estimate the real environment and draw it using a tool was a challenge because the measurements don't sync. So, what is designed is to be tested at the final build so it actually maps the real world environment. If it doesn't work the map has to be fixed again.

### **5.2.3 Recreating 2D map into the 3D map**

Once the map was created in 2D, this same map has to be replicated in 3D. For creating a 3D map of the environment, we used Unity 3D. It allows to create 3D based models and applications. The main components of the environment that have to be redrawn in 3D, are walls and other structure that can deviate the path. As the walls serve the important function in determining the shortest path from the source and destination. So, these structures have to be mapped accurately in 3D as well. For mapping them in 3D we used 3D cube from the unity and adjusted them according to the 2D map layout we had before.

#### **Wall Heights**

The height of the walls is to be adjusted according to the environment and the target placed. The walls in the application are occluded that is the walls are present in the 3D map but are not visible. This as a result will hide walls while we are using the application, but they will be existing physically.

#### **Openings Size**

The size of the openings such as gates or are between two walls, should be kept enough that our model can pass through it. The model is provided by unity itself which helps us determine that source and targets have a path even before application has been built.

### Applying NavMesh

NavMesh is applied on 3D map 5.1, which creates a mesh on the map and find paths in the 3D environment. This helps in finding every possible path in the map. If the path size is very narrow in certain area of the environment, the agent size can be decreased in order to pass it through every opening. The detection of every possible path on the map is of great importance as it allows to make find the shortest path from the source to the target.

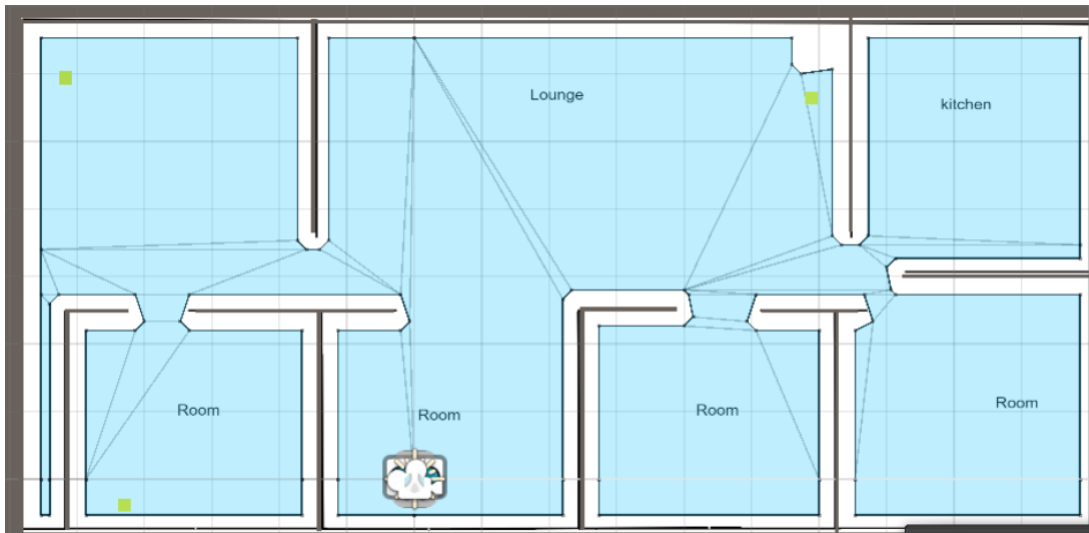


Figure 5.1: Applying the NavMesh on testing map

#### 5.2.4 Placing the targets

Once the 3D map is ready, we will now place multiple targets within the environment, that will serve as the basis for destination in that environment. These targets are 3D shapes based on AR and can be seen only using an AR camera. These targets have to be placed precisely in the locations that will be our assumed destinations according to the environment.

#### Target Heights

The height of the targets should be adjusted according to the environment. By default, targets are in the ground, and they are not visible. For this purpose, we have to adjust the heights of the targets so, that the path from source to the target is visible.

#### 5.2.5 Target Indicator

To indicate target, we are augmenting a AR based model of the hot air balloon 5.2. This can be seen from some distance indicating the user that he/she is approaching the destination.

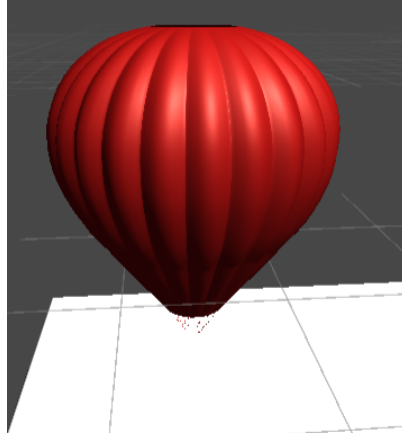


Figure 5.2: 3D model of the balloon

### 5.2.6 Determining the shortest path

After the placement of the targets, we have measured the shortest path from our fixed source to the multiple targets. The source can be changed by scanning the QR code placed physically. This scanning will reset the session for user as well. In all the cases whatever the source or destination is, the shortest path will be determined. For this purpose, A\* search algorithm is used. A\* algorithm not only allow us to determine the path between source and target, but it also allows the find the shortest path. It is the best searching algorithms where we have to use maps and graphs.

#### A\* Algorithm

A\* search algorithm is one of the best technique that is used in finding the paths and traversing the graphs especially in the case when we are dealing with informed searching problems. It is unique from other path finding techniques in such a way that it has **brains**. By the term brains it means that this algorithm is able to find the shortest path **smartly** from starting point to the end point. The following formula works as its basic,

$$f(n) = g(n) + h(n)$$

- Here **n** represents node
- **f** represents **sum** of g(n) and h(n)
- **g** represents the **cost** of moving from first cell to the goal cell, it is basically the sum of all the nodes traversed after leaving the first cell.
- **h** represents heuristic value, which is the **estimated cost** from moving from first cell to the goal cell, As the final cell hasn't been reached yet so actual cost can't be calculated.

**Play Example for A\* Algorithm**

Let us consider the following example in the figure 5.3. There are 5 nodes in total and every node is connected to its adjacent node. The path has the cost from one node to the other node. A is considered as the initial node and E is considered as the goal node.

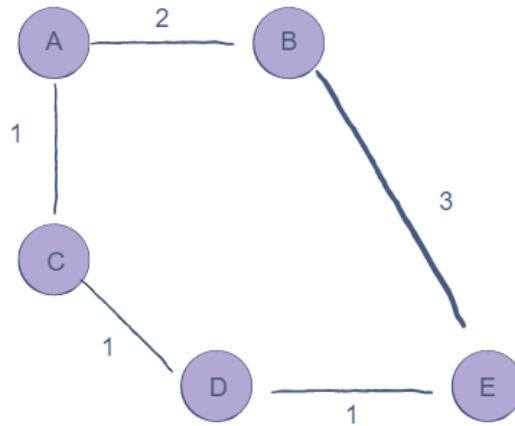


Figure 5.3: Set of nodes connected [5]



Figure 5.4: Root node A [5]

The above figure 5.4 shows the root node or initial node. The algorithm traverses every node from node A till the goal node.

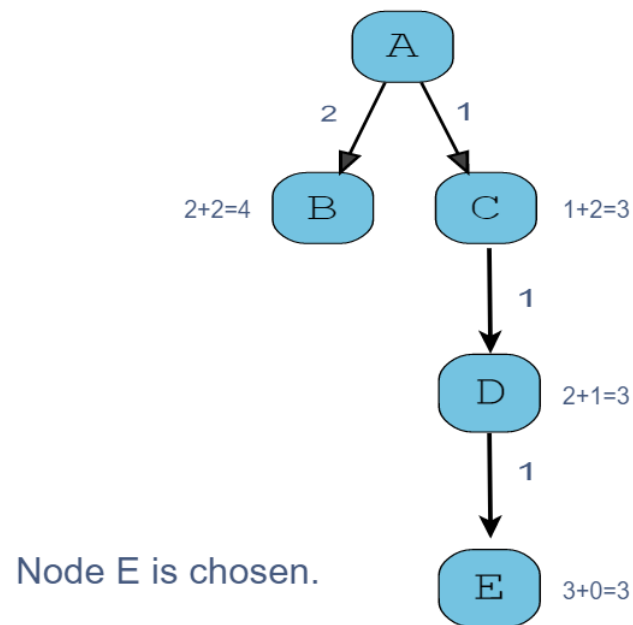


Figure 5.5: Traversing from root node to goal node [5]

The above figure 5.5 shows the path from initial node to the goal using A\* algorithm. At every step, the f-value is being recalculated by adding together the g and h values. The minimum f-value node is selected to reach the goal state.

#### Application A\* algorithm on the map we provided

The following figures 5.6, 5.7, shows the application of A\* on the map we are using. It determines the shortest path from the starting point to the destination we selected in the map.

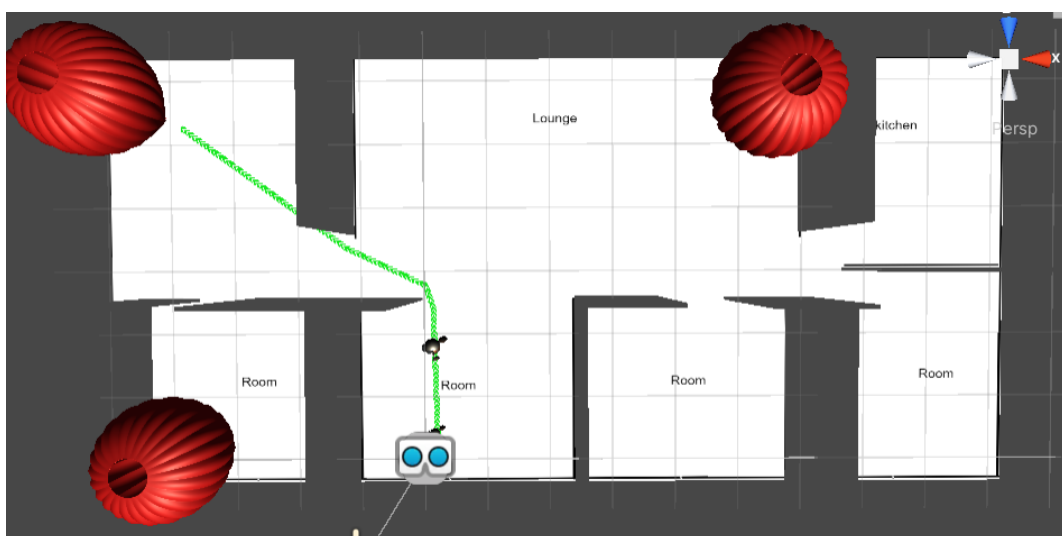


Figure 5.6: Traversing from my room to the main entrance

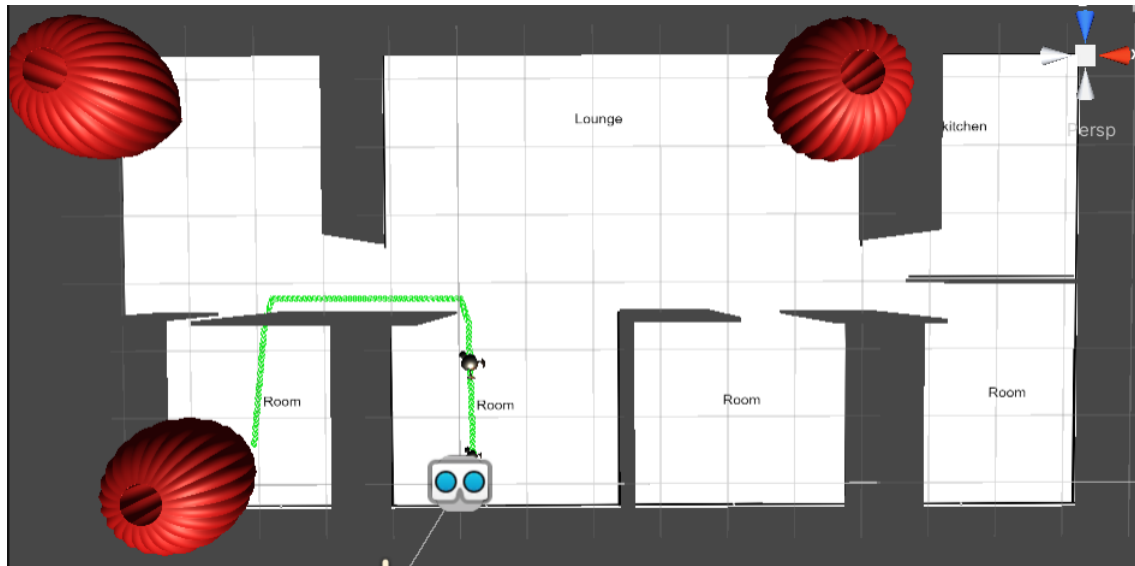


Figure 5.7: Traversing from my room to the guest room

### 5.2.7 Generating the AR based indicator

Finally, after the determination of the shortest path, the last thing is to generate an AR based line from the source to the target. This line will indicate the user the direction towards the destination. By following the line generated, the user can reach its destination.

#### Navigation Indicator Size

The size of the AR based navigation line is adjusted in the way it can pass through openings and is visible to the user so, the can follow that AR based line in order to reach from source to the destination.

#### AR based bot

There will be an AR based bot 5.8 that will move with the user while the user navigates towards the destination. The bot is a prebuilt 3D model that is imported from **Unity Asset Store**. The functionality for the bot that it will move along us is added through C# code.

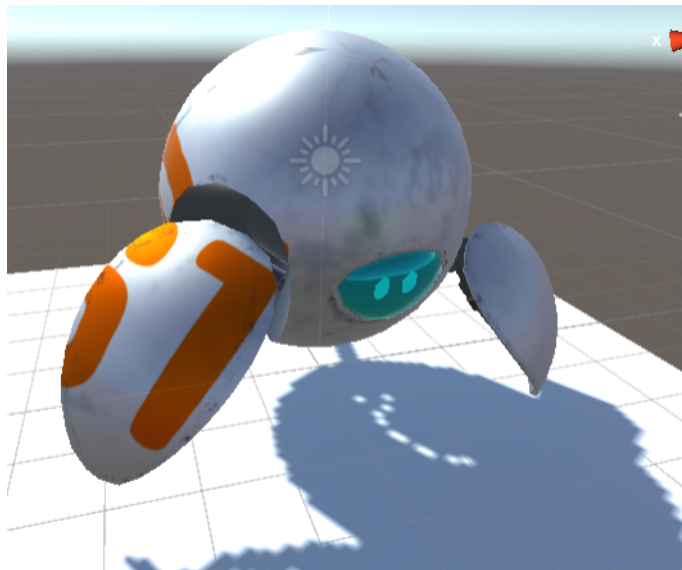


Figure 5.8: 3D model of the bot

### 5.2.8 Resetting the session

User can reset the session and the position of the AR camera by scanning the QR codes placed physically in the environment. Once the user scans the QR code, that position will be considered as the starting point or source in the environment.

#### Position of the QR code

QR codes are placed at the assumed positions in the indoor environment that will be more likely to be the place where the user may scan the code. This place can be the destination or the initial starting position,

## 5.3 Tools and Technology Used

Tools and technologies used in the implementation of this project are as follows,

### 5.3.1 AR Core

AR core is a SDK that allows to develop AR based applications for android OS. We used AR core mainly as we were developing the application for android. In our application we used it in order to place spatial anchors that are our targets. These anchors can be placed anywhere in the spatial context.

### 5.3.2 AR Foundation

AR foundation is a cross platform framework that allows the development of AR based application for both iOS and android. We considered AR foundation here because it will help us in order to enhance the scope of our application where we can develop the application for iOS as well.

### 5.3.3 Unity 3D

Unity 3D is a cross platform game engine that has a complete 3D environment. It allows to develop a full application for multiple platforms out there. For our application, unity has the major contribution. It enabled us in making the application providing all the 3D essentials that were required to build the application. It has all the necessary functionalities and services to build a 3D application.

### 5.3.4 ZXing .NET

We are using Zebra Crossing (ZXing) for the QR code reading system. It's implementation is already provided in .NET for Unity 3D. We just imported it from GitHub [38]. It is a library which supports decoding and generating of barcodes (like QR Code, PDF 417, EAN, UPC, Aztec, Data Matrix, Codebar) within images.

### 5.3.5 Visual Studio

Visual Studio is an IDE provided by Microsoft. It has multiple language support and help design . for various platforms. We used visual studio as it can easily be integrated by unity 3D. We wrote code and scripts in C# for the functionality of our application.

### 5.3.6 Inkscape

Inkscape is a vector graphics editor used to create vector images. We We have designed 3D buttons for our application using this application. Benefit of using this tool is that we can make vector images that can be scaled easily without pixelating the image. Moreover, this application has allowed us to make some diagrams in the report as well.

## 5.4 Language Used

The Main Language used for this application is C#. To perform a functionality on a button or an Event handler in unity we need a programming language and c# served us here. Unity has its own IDE known as mono behavior but to access the best control of the code we used Microsoft Visual Studio. C# is also web language and we used .NET framework for the application, so, C# was the best language for this application.

## Chapter 6

# System Testing and Evaluation

In this chapter, we will be discussing the testing and evaluation of our application that is GuideBot. We have incorporated different techniques in order to test our application. This application serves its user navigation guidance in an indoor environment. The testing phase of the application includes testing the components, modules, and functionality of the application. All this evaluation and results will be compared with the requirements specified in the software requirement chapter in order to measure the progress of the project.

### 6.1 Graphical User Interface Testing

While designing the GUI we kept in mind that it should be interactive and easy to use. We kept it simple but it provides complete functionality according to the application. Buttons and labels are placed at handy positions, so they can be accessed easily. Scanning the QR code and tracking the Navigation path is highly user friendly. Several testers were allowed to interact with the application and they didn't report bugs in the final builds of the application. Rather they found application's interface simple and fully functional.

### 6.2 Usability Testing

Usability testing was conducted in order to ensure the user friendliness, simplicity and ease of use of the application. We aimed at developing an application that was easy to use and requires minimum user interaction but still serve the purpose. Our application after being tested by number of users was able to find the destination they were looking for and the navigating path was clear. They were guided accurately towards the destination. Hence, the usability of the application was quite satisfactory.

### 6.3 Software Performance Testing

Software performance testing was conducted in order to determine how the system performs under in terms of responsiveness and stability under certain workload and conditions. There were slight jitters while using the application after the initial launch of the application but as soon as the application stays in the RAM, they are

gone. Overall, our application was successful in fulfilling the efficient performance criteria and users had a smooth and interactive experience with the application.

## 6.4 Exception Handling

In this application, there were not much of the exceptions that we had to handle. Although there is one that is when the user is following the AR based line and there is a turn, line may be seemed passing to the wall. It is obvious because AR based line does not exist physically so, it may seems to be passing but it surely navigates to the exact destination.

## 6.5 Installation Testing

Installation testing was done in order to ensure that the application is install-able and runs perfectly on the devices it was intended to work on. As our application is for android devices for now, hence, we used only smartphones running android operating system. For the purpose of testing, multiple devices having different android versions were used. The least compatible android version for our application is **Android 7 Nougat**. Our application was installed and was working as desired on the systems having same or greater android versions.

## 6.6 Test Cases

### 6.6.1 Launching Application Test Case

<b>Test case ID</b>	TC01
<b>Description</b>	Testing the application launch
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC1
<b>Initial Condition</b>	Application is installed
<b>Input</b>	Launching the application
<b>Expected Output</b>	Application should start
<b>Actual Output</b>	Application is launched and running
<b>Status</b>	PASS

Table 6.1: Table for Test Case 1

### 6.6.2 AR Camera Working Test

<b>Test case ID</b>	TC02
<b>Description</b>	Testing the AR camera interface in the application
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC1
<b>Initial Condition</b>	Application is running
<b>Input</b>	Viewing through the Application's camera
<b>Expected Output</b>	Application interface should be displaying real time view
<b>Actual Output</b>	Application interface is showing the view
<b>Status</b>	PASS

Table 6.2: Table for Test Case 2

### 6.6.3 Position on Mini Map

<b>Test case ID</b>	TC03
<b>Description</b>	Testing the position on mini map
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC1
<b>Initial Condition</b>	Application is running
<b>Input</b>	Standing at the marked position
<b>Expected Output</b>	User's position should display on the mini map
<b>Actual Output</b>	User's position is displaying and indicated by a sphere
<b>Status</b>	PASS

Table 6.3: Table for Test Case 3

#### 6.6.4 Navigation Line visibility on Mini Map

<b>Test case ID</b>	TC04
<b>Description</b>	Testing the line visibility on mini map
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC3
<b>Initial Condition</b>	Destination has been selected
<b>Input</b>	Standing at the marked position
<b>Expected Output</b>	Navigation line should display on the mini map from the source i.e. user to the target i.e. destination
<b>Actual Output</b>	Navigation line is displaying on mini map
<b>Status</b>	PASS

Table 6.4: Table for Test Case 4

#### 6.6.5 AR based navigation line visibility in application's camera interface

<b>Test case ID</b>	TC05
<b>Description</b>	Testing the visibility of AR based line
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC3
<b>Initial Condition</b>	Destination has been selected
<b>Input</b>	Standing at the marked position
<b>Expected Output</b>	AR based Navigation line should display on the AR camera interface from the source i.e. user to the target i.e. destination
<b>Actual Output</b>	Navigation line is displaying on AR camera interface
<b>Status</b>	PASS

Table 6.5: Table for Test Case 5

### 6.6.6 Reaching the destination selected

<b>Test case ID</b>	TC06
<b>Description</b>	Testing the reaching to the destination selected
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC6
<b>Initial Condition</b>	User is following the navigation line
<b>Input</b>	Destination selected
<b>Expected Output</b>	User should reach at the destination
<b>Actual Output</b>	User has reached the destination he selected
<b>Status</b>	PASS

Table 6.6: Table for Test Case 6

#### Reaching to the destination 1

<b>Starting Point</b>	My Room
<b>Ending Point</b>	Main Entrance
<b>Status</b>	PASS
<b>Shortest path</b>	Yes

Table 6.7: Testing the application for destination 1

#### Reaching to the destination 2

<b>Starting Point</b>	My Room
<b>Ending Point</b>	TV Lounge
<b>Status</b>	PASS
<b>Shortest path</b>	Yes

Table 6.8: Testing the application for destination 2

**Reaching to the destination 3**

<b>Starting Point</b>	My Room
<b>Ending Point</b>	Guest Room
<b>Status</b>	PASS
<b>Shortest path</b>	Yes

Table 6.9: Testing the application for destination 3

**What we have observed?**

The accuracy of the user reaching the target and the selection of the shortest path every time is possible as we are replicating the actual environment in 3D and the usage of A\* Algorithm ensures that the path that is selected every time to reach the destination from the source is the shortest path possible.

**6.6.7 Visibility of the AR based targets on the destination**

<b>Test case ID</b>	TC07
<b>Description</b>	Testing the visibility of AR based targets
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC6
<b>Initial Condition</b>	User is following the navigation line
<b>Input</b>	Destination selected
<b>Expected Output</b>	AR based targets should display on the selected destination
<b>Actual Output</b>	AR based targets are present at the target locations and are visible through application camera interface
<b>Status</b>	PASS

Table 6.10: Table for Test Case 7

### 6.6.8 Toggling navigation line visibility

<b>Test case ID</b>	TC08
<b>Description</b>	Testing the toggling of navigation line visibility
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC3
<b>Initial Condition</b>	User has selected the destination
<b>Input</b>	Pressing the line visibility toggle button
<b>Expected Output</b>	Line should be visible or invisible depending on the button pressing
<b>Actual Output</b>	Navigation line is hides and appears when the button is pressed
<b>Status</b>	PASS

Table 6.11: Table for Test Case 8

### 6.6.9 Selecting the destination from the list provided

<b>Test case ID</b>	TC09
<b>Description</b>	Testing the selection of destination
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC2
<b>Initial Condition</b>	User wants to select the destination
<b>Input</b>	Selecting the destination from the list
<b>Expected Output</b>	The destination should be selected and the path to that destination should be generated
<b>Actual Output</b>	Destination has been selected and the path is visible from the source towards destination
<b>Status</b>	PASS

Table 6.12: Table for Test Case 9

### 6.6.10 Scanning the QR Code

<b>Test case ID</b>	TC10
<b>Description</b>	Testing the scanning of QR code
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC4
<b>Initial Condition</b>	The AR camera interface is open
<b>Input</b>	Placing the smartphone's camera in front of QR code
<b>Expected Output</b>	The QR code should be scanned
<b>Actual Output</b>	QR code has been scanned by the AR camera interface
<b>Status</b>	PASS

Table 6.13: Table for Test Case 10

### 6.6.11 Scanning QR Code resets the session

<b>Test case ID</b>	TC11
<b>Description</b>	Testing the QR code scan resets the session
<b>Applicable</b>	Run on all android devices
<b>Use Case</b>	UC4
<b>Initial Condition</b>	QR code has been scanned successfully
<b>Input</b>	Placing the smartphone's camera in front of QR code
<b>Expected Output</b>	The scanning of QR code should result in resetting of position
<b>Actual Output</b>	Session has reset and position of the AR camera i.e position of user has changed.
<b>Status</b>	PASS

Table 6.14: Table for Test Case 11

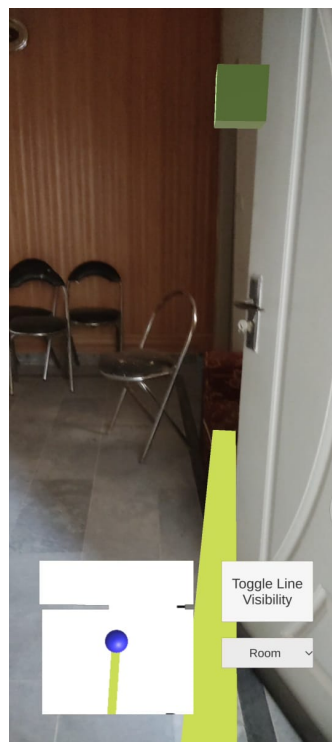
## 6.7 Initial Build Test of the Application



(a) Loading Screen



(b) Choosing location



(c) AR based line and target

Figure 6.1: Screenshots from the initial builds of the application

Above screenshots illustrate the testing of the one of the initial builds of the application. In the initial tests the targets were simple. An AR based cube was used to indicate destination. Simple line was used as the navigation path to be followed. After the working of the simple builds we made the interaction more interactive.

## 6.8 Final Build Test of the Application

The following figures 6.2, 6.3, 6.4 and 6.6 shows the working of the final build of the application. According to them,

- Application was successfully launched.
- Map was selected perfectly.
- Destination with in the map was chosen.
- The visibility of the navigation line can be toggled.
- The destination can be indicated precisely with AR based balloon.

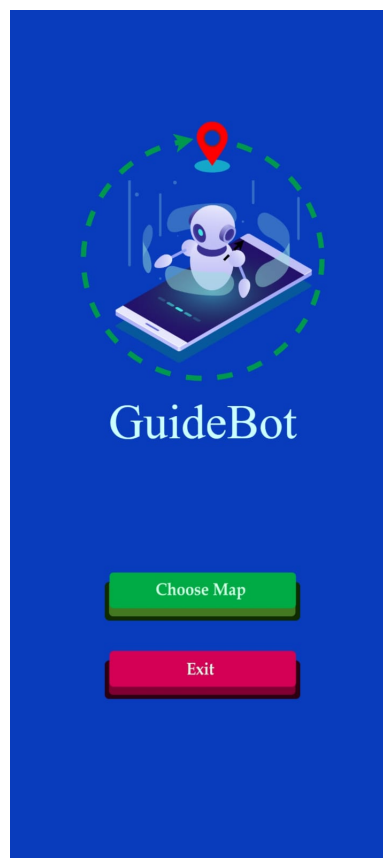


Figure 6.2: Application is launched and running

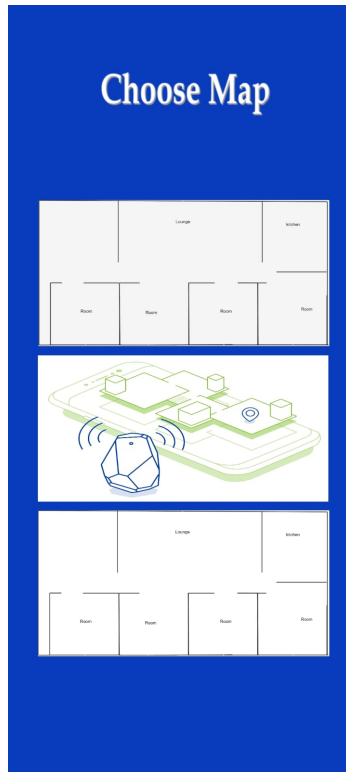


Figure 6.3: Maps can be selected successfully

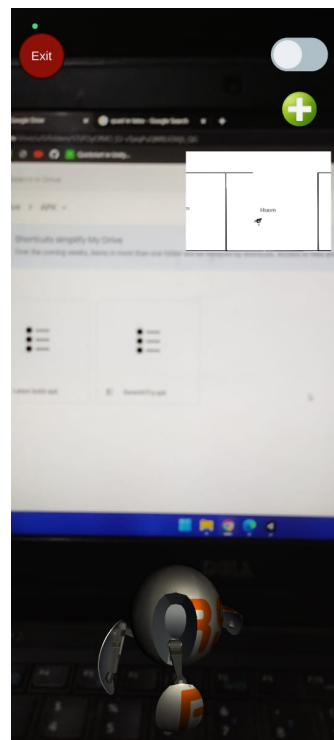


Figure 6.4: AR camera view and AR based bot visible

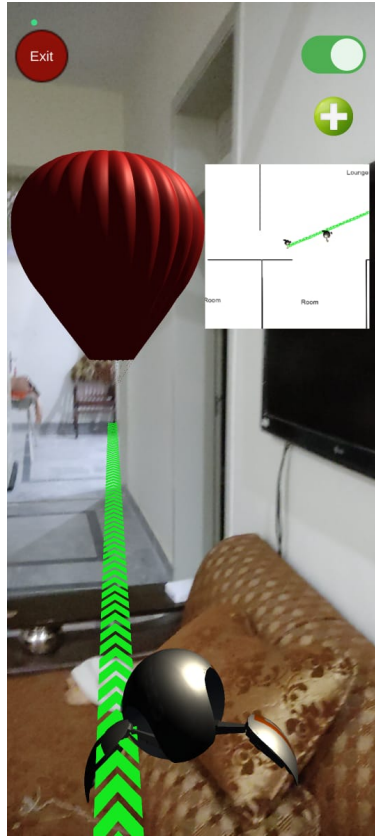


Figure 6.5: Successfully visible AR based line and target

# Chapter 7

## Conclusions

In this chapter, we will summarize what we have been implementing and how much we have been successful in it. We developed an application which helps us in navigation in an indoor environment. It uses spatial anchors and Augmented Reality in order to achieve its objective. It can easily and interactively navigate a user in an indoor environment precisely. We tried our application using few maps. We were successful in the navigation process after some trials in each map. The spatial anchors provided by AR core serves the basis of the navigation. The fastest algorithm that can help us in finding the shortest path was A\* algorithm and we in-cooperated it for the purpose of finding shortest path from source to the destination. We made the design of our application minimal, so, everyone can understand it easily. We have added a 3D bot and other 3D models in order to make the experience joyful. The bot has animation on the bases of action.

### 7.1 Future Work

We have further plans regarding our application for future. Some of them are listed below,

1. Our application is primarily for android users, so we will develop an iOS version of this application.
2. We will be adding more destinations in the environment.
3. We will be incorporating more maps.
4. we will increase the number of environments in which this application can be in cooperated.
5. We will be adding 3D models in order to make the user experience more interactive.
6. We will be adding chat assistance in the application in future.
7. We will be adding more interaction features related to the application and its purpose.
8. We will be adding audio features and notifiers to aid in navigation.

## 7.2 Conclusion

To conclude, we developed our Final Year Project, that is GuideBot, keeping in mind the ease it can provide to its user and the new technology we are bringing in our country. We developed this application and report solely with a lot of efforts and hard work. The technology we were using was totally new to us but for the purpose of learning, we decided to implement our FYP in it. We focused on putting our best to learn the new technology and bringing the best we were able to put in front.

# Appendix A

## User Manual

### A.1 Introduction

User manual helps the user of the application to get a concept about the usability and working of the application. We have attached various screen snaps along with brief description for the ease of the user. The user manual for the application is as follows,

## A.2 Loading Screen

In the following figure A.1, the loading screen of the application is illustrated that appears when we launch the application. As this application is developed using Unity 3D, so, the loading screen has unity logo.



Figure A.1: Loading screen of the application

### A.3 Front Screen

In the following figure A.2, the front screen of the application is displayed that appears after successful launch of the application and when it is running perfectly. There are 2 buttons. One exits the application and other let you choose the map.

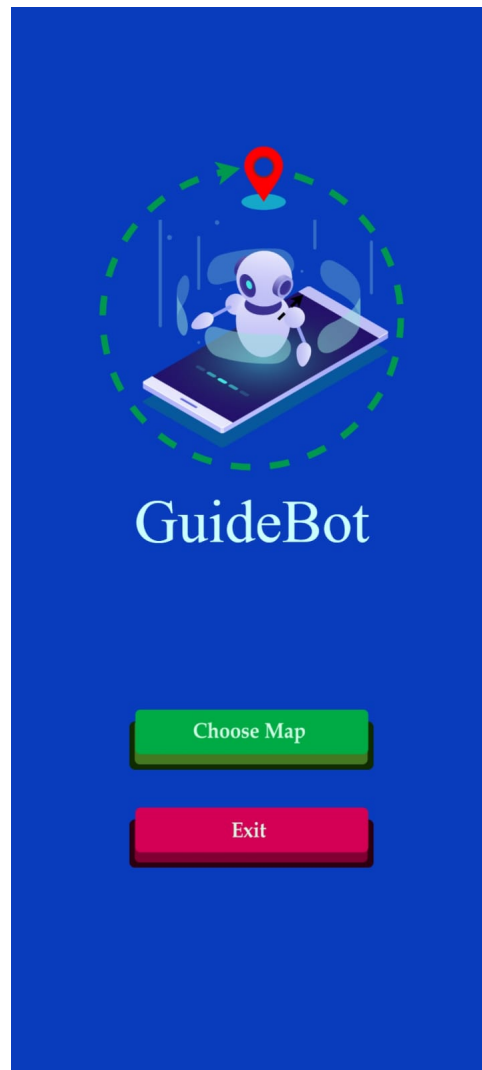


Figure A.2: Front screen of the application

## A.4 Map Selection screen

The figure A.3 below displays the screen of the application where we can choose map. We can choose any of the map.

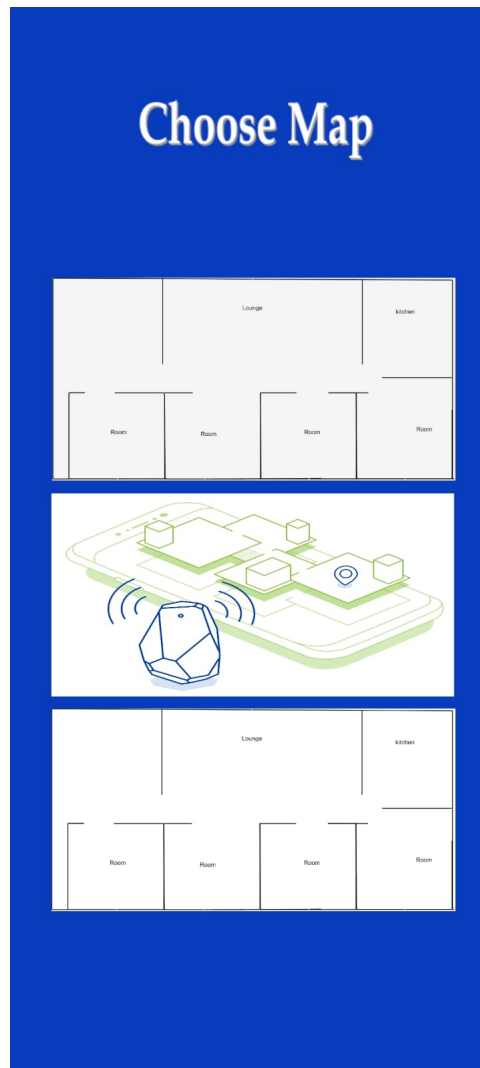


Figure A.3: Map selection screen of the application

## A.5 Live Camera Screen

In the following figure A.4, the screen displays the view of the camera when the application goes live. It is by using the AR camera provided by the phone itself. The screen has an AR based bot, a mini map that provides the overall view, navigation line visibility toggle and a button by which we can choose destination with in the map.



Figure A.4: Live AR camera screen of the application

## A.6 Destination Selection Screen

The figure A.5 below displays the menu that allows us to choose location with in the map. To open this menu we have to press that ”+” button and then this menu appears. Location can be selected just by clicking the destination.

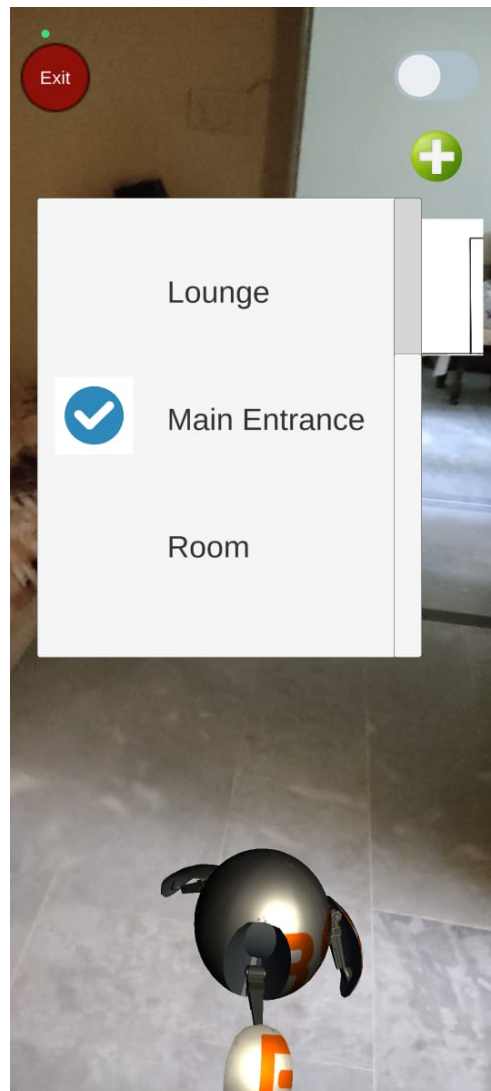


Figure A.5: Destination selection screen of the application

## A.7 Navigation Line Visibility

### A.7.1 AR based navigation line toggled on

The figure A.6 below shows AR based navigation line when toggle is set to on. This can be turned on or off any time once we have selected the destination.



Figure A.6: Navigation line visibility turned on

### A.7.2 AR based navigation line toggled off

In the figure A.7, we can see the screen when the visibility of the navigation line is turned off. It can be turned on any time.



Figure A.7: Navigation line visibility turned off

## A.8 Destination indication

In the following figure A.8, it is shown that once we are approaching our destination, we can see an AR based balloon. This balloon will indicate that we have reached our destination that we selected before.

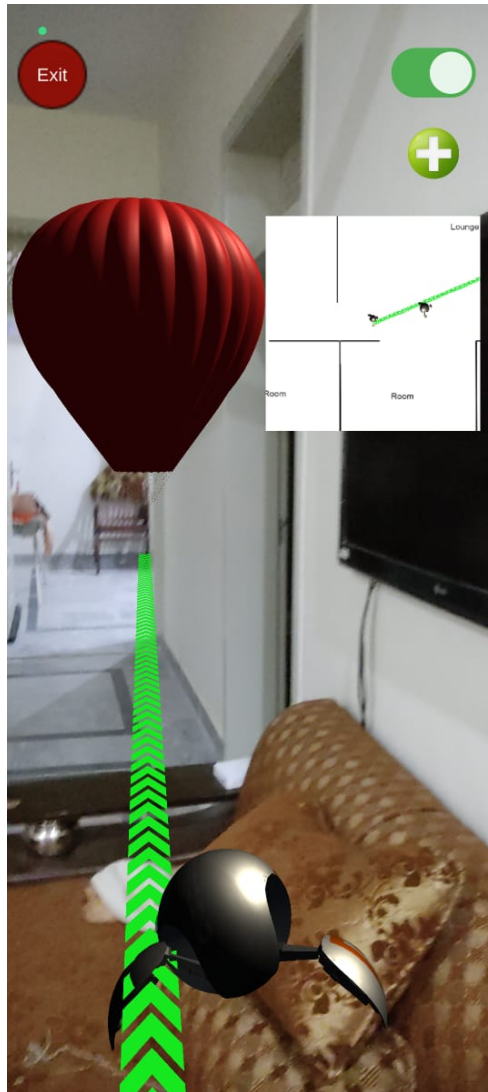


Figure A.8: 3D model of hot air balloon while approaching destination

# References

- [1] Indoor navigation using augmented reality apps. In *Indoor Navigation*. indoornavigation.com. <https://www.indoornavigation.com/knowledge/indoor-navigation-using-augmented-reality-apps>.
- [2] Ar indoor navigation technology connects retailers with shoppers. In *IOT for all*. iotforall.com. <https://www.iotforall.com/ar-indoor-navigation>.
- [3] Wikipedia. Augmented reality. [https://en.wikipedia.org/wiki/Augmented\\_reality](https://en.wikipedia.org/wiki/Augmented_reality).
- [4] Navigation map road and direction navigative symbols and navigator app. In *Pixa Stock*. pixastock.com. <https://www.pixastock.com/illustration/71558610>.
- [5] What is the a\* algorithm? In *Educative*. educative.io. [www.educative.io/edpresso/what-is-the-a-star-algorithm](http://www.educative.io/edpresso/what-is-the-a-star-algorithm).
- [6] Augmented reality. In *Wikipedia*. wikipedia.org. [https://en.wikipedia.org/wiki/Augmented\\_reality](https://en.wikipedia.org/wiki/Augmented_reality).
- [7] Autonomous supermarket robot assistance using machine learning. In *IEEE Explore*. IEEE. <https://ieeexplore.ieee.org/abstract/document/9441833>.
- [8] Indoor navigation with augmented reality. In *VIEWAR*. viewar.com. <https://guidebot.viewar.com/>.
- [9] Guidebot. a tour guide system based on mobile robots. In *Sage Journals*. SagePub. <https://journals.sagepub.com/doi/full/10.5772/56901>.
- [10] Augmented reality (ar) navigation in the shopping mall use case. In *5GPPP 5G-CORAL*. YouTube. [https://www.youtube.com/watch?v=REn6\\_qiUMws&ab\\_channel=5GPPP5G-CORAL](https://www.youtube.com/watch?v=REn6_qiUMws&ab_channel=5GPPP5G-CORAL).
- [11] Borko Furht. *Handbook of augmented reality*. Springer Science & Business Media, 2011.
- [12] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.
- [13] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.

- [14] Wikitude. Wikitude augmented reality: the world's leading cross-platform ar sdk. 2021. <https://www.wikitude.com/>.
- [15] Wikipedia. Sensorama. <https://en.wikipedia.org/wiki/Sensorama>.
- [16] Ronald Azuma. Overview of augmented reality. In *ACM SIGGRAPH 2004 Course Notes*, pages 26–es. 2004.
- [17] Laurence F. Johnson and Rachel S. Smith. 2005 horizon report johnson. *2005 Horizon Report*, 2005.
- [18] Ar home. *Google AR & VR*. <https://arvr.google.com/ar/>.
- [19] Daniel R Montello. *Navigation*. Cambridge University Press, 2005.
- [20] Outdoor navigation. *Outdoor Navigation - Navipedia*. [https://gssc.esa.int/navipedia/index.php/Outdoor\\_Navigation#:~:text=Outdoor%20Navigation%20consists%20in%20the,environments%20such%20as%20nature%20trails.&text=Outdoor%20Navigation%20is%20somewhat%20different,a%20turn%20by%20turn%20navigation](https://gssc.esa.int/navipedia/index.php/Outdoor_Navigation#:~:text=Outdoor%20Navigation%20consists%20in%20the,environments%20such%20as%20nature%20trails.&text=Outdoor%20Navigation%20is%20somewhat%20different,a%20turn%20by%20turn%20navigation).
- [21] Indoor positioning system. *Wikipedia*. [https://en.wikipedia.org/wiki/Indoor\\_positioning\\_system](https://en.wikipedia.org/wiki/Indoor_positioning_system).
- [22] Low Weng Fai and Lukman Audah. In campus location finder using mobile application services. In *AIP Conference Proceedings*, volume 1883, page 020022. AIP Publishing LLC, 2017.
- [23] Dennis Roddy. *Satellite communications*. McGraw-Hill Education, 2006.
- [24] Mark Altaweel. Augmented reality and computer vision in navigation. *GIS Lounge*. <https://www.gislounge.com/augmented-reality-and-computer-vision-in-navigation/>.
- [25] Wolfgang Narzt, Gustav Pomberger, Alois Ferscha, Dieter Kolb, Reiner Müller, Jan Wieghardt, Horst Hörtnner, and Christopher Lindinger. Augmented reality navigation systems. *Universal Access in the Information Society*, 4(3):177–187, 2006.
- [26] Magda Chelly and Nel Samama. New techniques for indoor positioning, combining deterministic and estimation methods. In *ENC-GNSS 2009: European Navigation Conference-Global Navigation Satellite Systems*, pages 1–12, 2009.
- [27] Nobuo Kawaguchi. Wifi location information system for both indoors and outdoors. In *International Work-Conference on Artificial Neural Networks*, pages 638–645. Springer, 2009.
- [28] J Joseph. Qr code based indoor navigation with voice response. *International Journal of Science and Research*, 3(11):923–926, 2014.
- [29] Alessandro Mulloni, Daniel Wagner, Istvan Barakonyi, and Dieter Schmalstieg. Indoor positioning and navigation with camera phones. *IEEE Pervasive Computing*, 8(2):22–31, 2009.

- [30] Gerhard Reitmayr and Dieter Schmalstieg. Location based applications for mobile augmented reality. In *Proceedings of the Fourth Australasian user interface conference on User interfaces 2003-Volume 18*, pages 65–73, 2003.
- [31] Chen Feng and Vineet R Kamat. Augmented reality markers as spatial indices for indoor mobile aecfm applications. In *Proceedings of 12th international conference on construction applications of virtual reality (CONVR 2012)*, pages 235–24, 2012.
- [32] Gi Ho Nam, Han Sol Seo, Min Su Kim, Yun Kyoung Gwon, Cheol Min Lee, and Dong Myung Lee. Ar-based evacuation route guidance system in indoor fire environment. In *2019 25th Asia-Pacific Conference on Communications (APCC)*, pages 316–319. IEEE, 2019.
- [33] Jia-Hua Wu, Cheng-Ta Huang, Zhi-Ren Huang, Yen-Bai Chen, and Sheng-Chang Chen. A rapid deployment indoor positioning architecture based on image recognition. In *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 784–789. IEEE, 2020.
- [34] Georg Gerstweiler, Emanuel Vonach, and Hannes Kaufmann. Hymotrack: A mobile ar navigation system for complex indoor environments. *Sensors*, 16(1):17, 2016.
- [35] Umair Rehman and Shi Cao. Augmented-reality-based indoor navigation: A comparative analysis of handheld devices versus google glass. *IEEE Transactions on Human-Machine Systems*, 47(1):140–151, 2016.
- [36] Tara Rustagi and Kyungjin Yoo. Indoor ar navigation using tilesets. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pages 1–2, 2018.
- [37] Ar core. In *Google Developers*. AR Core. <https://developers.google.com/ar>.
- [38] Zxing .net github repository. In *Zebra Crossing v0.16.8.0*. Michael Jahn. <https://github.com/micjahn/ZXing.Net>.