Augmented Reality Based Indoor Navigation System

Group Members

Shahzaib Shakoor	18I-0875
Abdul Rafay Zafar	18I-0848
Muneeb Ahmad Nasir	18I-0766

Project Supervisor

Engr. Shahzad Ahmed

Department of Electrical Engineering

National University of Computer and Emerging Sciences, Islamabad

2023

Developer's Submission

"This report is being submitted to the Department of Electrical Engineering of the National University of Computer and Emerging Sciences in partial fulfillment of the requirements for the degree of BS in Electrical Engineering"

Augmented Reality Based Indoor Navigation System

Sustainable Development Goals

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	✓ **SDG 9**	**Industry, Innovation, and Infrastructure**
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well Being	✓ **SDG 11**	**Sustainable Cities and Communities**
SDG 4	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals

Range of Complex Problem Solving			
	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	✓
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	✓
4	Familiarity of issues	Involve infrequently encountered issues	✓
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	✓
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	
7	Consequences	Have significant consequences in a range of contexts.	✓
8	Interdependence	Are high level problems including many component parts or sub-problems	

Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	✓
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	✓
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	✓
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	✓
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Developer's Declaration

"We take full responsibility of the project work conducted during the Final Year Project (FYP) titled **"Augmented Reality Based Indoor Navigation System"**. We solemnly declare that the project work presented in the FYP report is done solely by us with no significant help from any other person; however, small help wherever taken is duly acknowledged. We have also written the complete FYP report by ourselves. Moreover, we have not presented this FYP (or substantially similar project work) or any part of the thesis previously to any other degree awarding institution within Pakistan or abroad.

We understand that the management of Department of Electrical Engineering of National University of Computer and Emerging Sciences has a zero-tolerance policy towards plagiarism. Therefore, we as an author of the above-mentioned FYP report solemnly declare that no portion of our report has been plagiarized and any material used in the report from other sources is properly referenced. Moreover, the report does not contain any literal citing of more than 70 words (total) even by giving a reference unless we have obtained the written permission of the publisher to do so. Furthermore, the work presented in the report is our own work and we have positively cited the related work of the other projects by clearly differentiating our work from their relevant work.

We further understand that if we are found guilty of any form of plagiarism in our FYP report even after our graduation, the University reserves the right to withdraw our BS degree. Moreover, the University will also have the right to publish our names on its website that keeps a record of the students who committed plagiarism in their FYP reports."

___________________	____________________	____________________
Shahzaib Shakoor	Abdul Rafay Zafar	Muneeb Ahmad Nasir
BS(EE) 2018-0875	BS(EE) 2018-0848	BS(EE) 2018-0766

____

Certified by Supervisor

____

Verified by Plagiarism Cell Officer

Dated: __________

Abstract

Indoor navigation systems play a crucial role in helping individuals navigate complex indoor environments such as shopping malls, airports, and large buildings. Traditional navigation methods, such as maps or signs, often prove inadequate in such settings, leading to confusion and time wasted. Augmented Reality (AR) technology presents a promising solution by superimposing digital information onto the real-world environment, enhancing the user's perception and understanding of their surroundings.

This paper presents an augmented reality-based indoor navigation system designed to provide seamless and intuitive navigation experiences within indoor spaces. The system leverages the capabilities of smartphones to deliver real-time navigation guidance to users. By utilizing the built-in cameras and sensors, the system detects and tracks the user's position and orientation accurately.

The proposed system incorporates QR codes strategically placed in the indoor environment. These QR codes are then mapped onto a virtual representation of the physical space, allowing for precise localization and path planning. Through the AR interface, users receive visual overlays, such as arrows, markers, and textual instructions, superimposed onto the camera feed, guiding them towards their desired destination.

Evaluation of the system's performance was conducted in university campus. The results demonstrate the system's effectiveness in accurately guiding users and reducing navigation time, compared to traditional methods.

The augmented reality-based indoor navigation system described in this paper presents a promising solution to enhance indoor navigation experiences. By leveraging the power of AR technology, users can effortlessly navigate complex indoor spaces, resulting in improved efficiency, reduced stress, and enhanced user satisfaction. Future work could focus on integrating additional features, such as machine learning optimization, voice-guided instructions or social interaction capabilities, to further enrich the user experience in indoor navigation scenarios.

Acknowledgements

We would like to extend our heartfelt gratitude to our supervisor, Dr. Shahzad Ahmed, for their exceptional guidance, unwavering support, and valuable insights throughout the development of the project. His expertise and dedication were instrumental in shaping the direction of our project and ensuring its success.

Furthermore, we extend our sincere thanks to the faculty members who generously volunteered their time and provided invaluable feedback during the evaluation phase of the project. Their active involvement and constructive criticism contributed significantly to the refinement and enhancement of the system's functionality and usability.

We are truly grateful for the support and guidance we received throughout this project, which played an essential role in its successful completion.

Table of Contents

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Indoor navigation systems are crucial for assisting individuals in complex indoor environments such as shopping malls, airports, hospitals, and museums. However, traditional navigation methods relying on GPS are often ineffective indoors due to limited satellite signals. To overcome this limitation, the integration of augmented reality (AR), Unity 3D, and QR code localization has emerged as a promising solution. By leveraging the capabilities of AR and Unity 3D, combined with the accuracy of QR code localization, an innovative indoor navigation system can be developed.

Motivation

The motivation for developing an AR-based indoor navigation system using Unity 3D and QR code localization is driven by the following factors:

- **Improved Navigation Accuracy:** GPS-based systems are ineffective indoors, leading to inaccurate positioning and guidance. By utilizing QR code localization, which offers higher accuracy and reliability, the proposed system can overcome the limitations of traditional navigation methods and provide accurate real-time navigation instructions.
- **Enhanced User Experience:** Traditional navigation systems lack an immersive and interactive experience. By incorporating AR technology and Unity 3D, the proposed system can overlay virtual information onto the real world, providing users with a visually enriched and intuitive navigation experience. This enhances user engagement and reduces navigational challenges in complex indoor environments.
- **Cost-Effectiveness:** QR codes provide a cost-effective solution for localization in indoor spaces. The use of QR codes as location markers eliminates the need for expensive infrastructure installations, making the proposed system accessible and affordable for various indoor environments.

1.1 Problem statement

"Design an accurate and user-friendly indoor navigation system. "

The problem addressed by this project is the need for an accurate and user-friendly indoor navigation system that overcomes the limitations of GPS-based systems. The proposed AR-based indoor navigation system aims to provide users with seamless navigation experiences in complex indoor environments. The system should offer accurate positioning, real-time guidance, and a user-friendly interface for intuitive interaction. By integrating AR technology, the system aims to provide an innovative solution for indoor navigation challenges.

1.2 Literature review

- S. K. Tadepalli, P. A. Ega, and P. K. Inugurthi, “Indoor navigation using augmented reality,” International Journal of Scientific Research in Science and Technology, vol. 7, no. 4, pp. 588–592, 2021.
    - The application utilizes Google ARCore, a platform for building augmented reality experiences, to sense and interact with the environment using a smartphone's camera.
    - Simultaneous Localization and Mapping (SLAM) is employed to track the user's real-time movement and update a 3D model of the building.
    - The NavMesh in Unity is used to create an area where the user object can move, based on the building's blueprint, including walls and doors.
    - The NavMesh allows the application to estimate the user's walkable areas and determine the accessible path.
    - The A* algorithm, an informed search algorithm, is employed to find the shortest path from the user's starting point to the destination.
    - A* combines the advantages of Dijkstra's Algorithm (favoring nearby vertices) and Greedy Best-First Search (favoring vertices close to the goal) by utilizing a heuristic to guide itself.
    - The application's methodology combines ARCore, SLAM, NavMesh, and A* to create an immersive augmented reality experience with accurate navigation capabilities.

- S. C. J, S. Dileep, S. C. S, and L. E. Sunny, “INDOOR NAVIGATION SYSTEM USING AUGMENTED REALITY,” International Research Journal of Engineering and Technology (IRJET), vol. 08, no. 06, pp. 2408–2412, Jun. 2021.
    - QR codes are used throughout the building to obtain location data for accurate indoor navigation in the proposed system.
    - The mobile application scans the QR codes to provide the user with precise navigation within the building.
    - Unity's NavMesh components are utilized to generate a navigation mesh for pathfinding inside the building, making navigation easier.
    - The navigation system uses augmented reality characters and navigation meshes to create an immersive experience and identify obstacles or hindrances in the user's path.
    - Augmented reality objects, such as directing arrows, are dynamically positioned to guide the user along the generated path.
    - The indoor navigation app utilizes augmented reality objects, specifically arrows, to visually indicate the direction the user needs to take after selecting a destination.
    - The system ensures that the arrows are visible and updated by constantly tracking the user's position and adjusting the placement of the arrows within a defined mesh.

1.3 Report Outline

This report is further divided into multiple chapters as listed below.

In chapter 2, proposed solution is discussed in detail. It includes details of the technologies used, block diagram, Flow chart of the process and interfacing of the sensors with the processing module.

Chapter 3 discusses the results acquired by testing our solution on the university campus. It further discusses the conclusions drawn from the obtained results and the recommendations/future work that is proposed for further enhancements.

Chapter 2 Solution Design & Implementation

This chapter discusses the complete design and implementation of the proposed solution. Section 2.1 discusses the technologies and methods used to implement the required functionality. Section 2.2 discusses the block diagram of the project. The details of flow chart are presented in section 2.3 whereas, Section 2.4 discusses the 3D environment setup in unity3D. Section 2.5 discusses in detail the QR code localization. Path Planning and Navigation Guidance is presented in Section 2.6. Section 2.7 discusses the integration with AR and location tracking using SLAM. Finally, section 2.8 discusses the user interface of the application.

2.1 Technologies and Methods

The development of an augmented reality-based indoor navigation system involves utilizing various methods and technologies to achieve accurate positioning, seamless AR integration, and intuitive user interaction. Here are brief details about some of the key methods and technologies used in the project:

Unity 3D

Unity 3D is a powerful game development engine that provides a comprehensive platform for creating AR applications. By utilizing Unity 3D, developers can import 3D models, design interactive user interfaces, implement SLAM algorithms or QR code recognition, and seamlessly integrate AR content with the help of AR Foundation and ARCore. Unity 3D's cross-platform support allows the project to be deployed on ARCore-compatible Android devices, ensuring broad accessibility and usability for users.

Augmented Reality (AR) Technology:

AR technology overlays virtual objects, information, and visual cues onto the real-world environment, enhancing the navigation experience. By leveraging AR Foundation and ARCore, the system can seamlessly integrate virtual content with the physical environment, allowing for the overlay of navigation guidance, such as arrows, markers, or labels, onto the user's view in real-time. AR technology improves spatial understanding, provides contextual information, and enhances the overall navigation experience within the navigation system.

AR Foundation and ARCore

AR Foundation is a framework provided by Unity that simplifies the development of AR applications across different platforms, including ARCore for Android devices. ARCore is a software development kit (SDK) developed by Google that enables augmented reality experiences on Android devices. By utilizing AR Foundation and ARCore, developers can leverage the capabilities of ARCore-compatible devices for accurate motion tracking, environmental understanding, and plane detection. These features allow the AR-based indoor navigation system to accurately map the environment, detect surfaces, and place virtual objects in the real world.

QR Code Localization

QR code localization is a technique commonly used in indoor navigation systems. QR codes are strategically placed at specific locations within the indoor environment. By scanning and recognizing these QR codes using the device's camera, the system can determine the user's position and align it with the corresponding digital map or floor plan. QR code localization provides a simple and effective way to achieve accurate indoor positioning without relying on GPS signals.

ZXing (Zebra Crossing)

ZXing is an open-source library used for decoding QR codes in real-time. By integrating this library into the project, the application can process the camera feed and extract information from QR codes. It simplifies the QR code recognition process, allowing the system to accurately determine the user's position based on the scanned QR codes.

Simultaneous Localization and Mapping (SLAM)

SLAM is a technique used in this project to estimate the user's position and map the environment in real-time. It is a part of ARCore. By combining the sensor data from the device, such as the camera feed and motion sensors, with SLAM algorithms, the system can create a dynamic map of the environment and determine the user's position within it. SLAM plays a crucial role in providing accurate positioning and tracking within the AR-based indoor navigation system, ensuring that virtual content aligns properly with the real-world environment.

NavMesh

NavMesh is a Unity feature used for pathfinding and navigation within a virtual environment. It provides a precomputed mesh (weighted graph) representation of the walkable areas in the environment. By generating a navigation mesh, the system can optimize pathfinding calculations, allowing for efficient and realistic movement within the virtual space. NavMesh ensures that the user can navigate the environment smoothly and find the shortest and safest routes to their destinations.

A* algorithm

The A* (A-star) algorithm is a widely used pathfinding algorithm that efficiently calculates the shortest path between two points in a graph or navigation mesh. It considers factors such as distance, obstacles, and costs to determine the most optimal route. The A* algorithm is utilized in the navigation system to generate optimized paths from the user's current location to their desired destination. It ensures efficient and effective navigation within the environment.

2.2 Block Diagram

Figure 2.1 shows the complete block diagram of the project. The details of each block are discussed below.

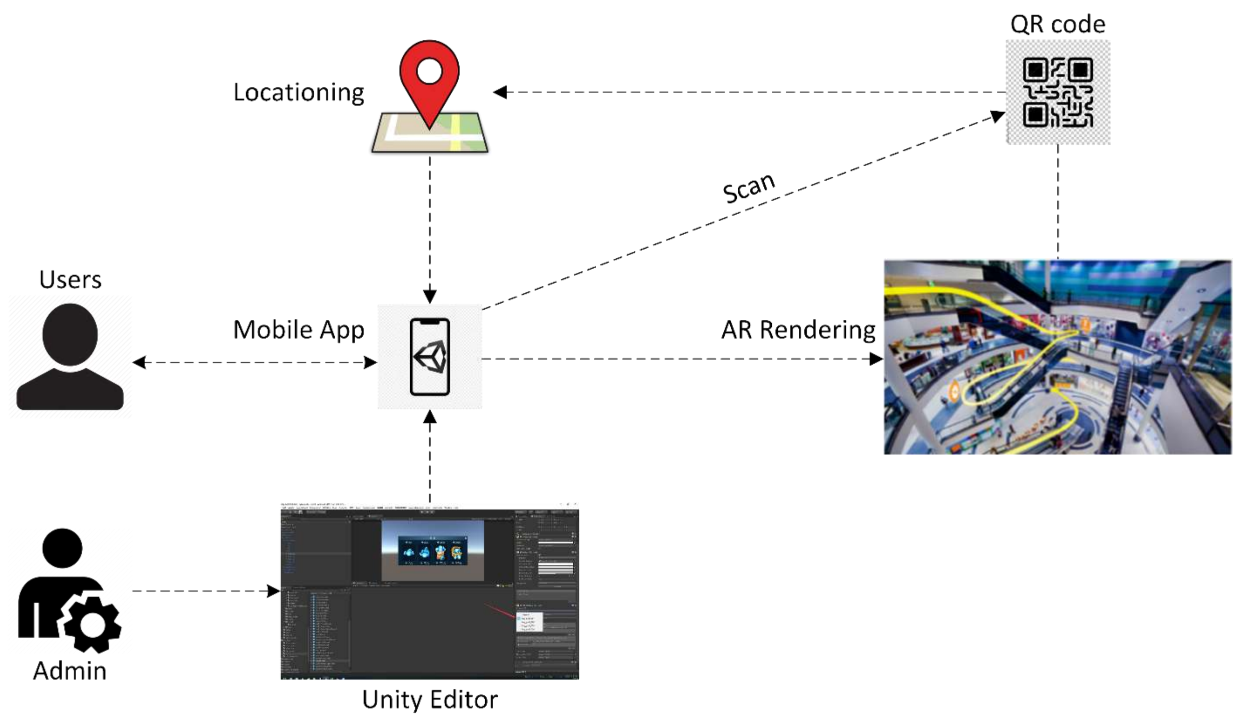

Figure 2.1 Block diagram of the project.

Unity Editor

The Unity Editor serves as the backend of the project, allowing developers to create, edit, and modify the 3D environment according to the physical layout of the indoor space. It provides a comprehensive set of tools and features for designing and managing the virtual aspects of the navigation system. Some key functionalities of the Unity Editor include:

- **3D Environment Creation:** The Unity Editor enables developers to create a virtual representation of the indoor environment by importing or building 3D models of the physical space. This includes the walls, floors, ceilings, and other structural elements.
- **Adding Destinations:** Within the Unity Editor, developers can add new destinations or points of interest within the virtual environment. These destinations serve as the targets for navigation and are associated with physical space.
- **QR Code Placement:** The Unity Editor facilitates the placement and positioning of QR codes within the virtual environment. Developers can accurately map the location of each QR code to correspond with its physical counterpart in the real world.

- **Updating Building Model:** If any changes occur in the physical building, such as renovations or expansions, the Unity Editor allows developers to update the building model accordingly. This ensures that the virtual representation remains consistent with the real-world environment.

Mobile App

The Mobile App serves as the user-facing front-end of the project, providing an intuitive interface for users to interact with navigation. It offers a range of functionalities and features designed to enhance the user experience. Some key features of the mobile app include:

- **User-Friendly Interface:** The Mobile App offers a user-friendly interface that guides users through the navigation process. It provides clear instructions and intuitive controls for seamless interaction with the system.
- **QR Code Localization:** Users can utilize the Mobile App to scan QR codes placed within the physical environment. The app's camera functionality and QR code recognition capabilities enable accurate localization of the user within the virtual environment.
- **Destination Selection:** After localizing themselves, users can select their desired destination within the Mobile App. The app provides a list of available destinations or points of interest to choose from.
- **Path Creation and Visualization:** Once the destination is selected, the Mobile App generates a navigational path from the user's current location to the chosen destination. This path is displayed on a mini map, providing visual guidance to the user.
- **AR Markers:** By pressing the "visualize" button, the Mobile App activates the augmented reality feature and overlays AR markers along the generated path. These markers serve as visual cues, guiding the user along the correct route in the physical environment.

2.3 Flow Chart

Figure 2.2 shows the complete flow chart of the project. The details are discussed below.

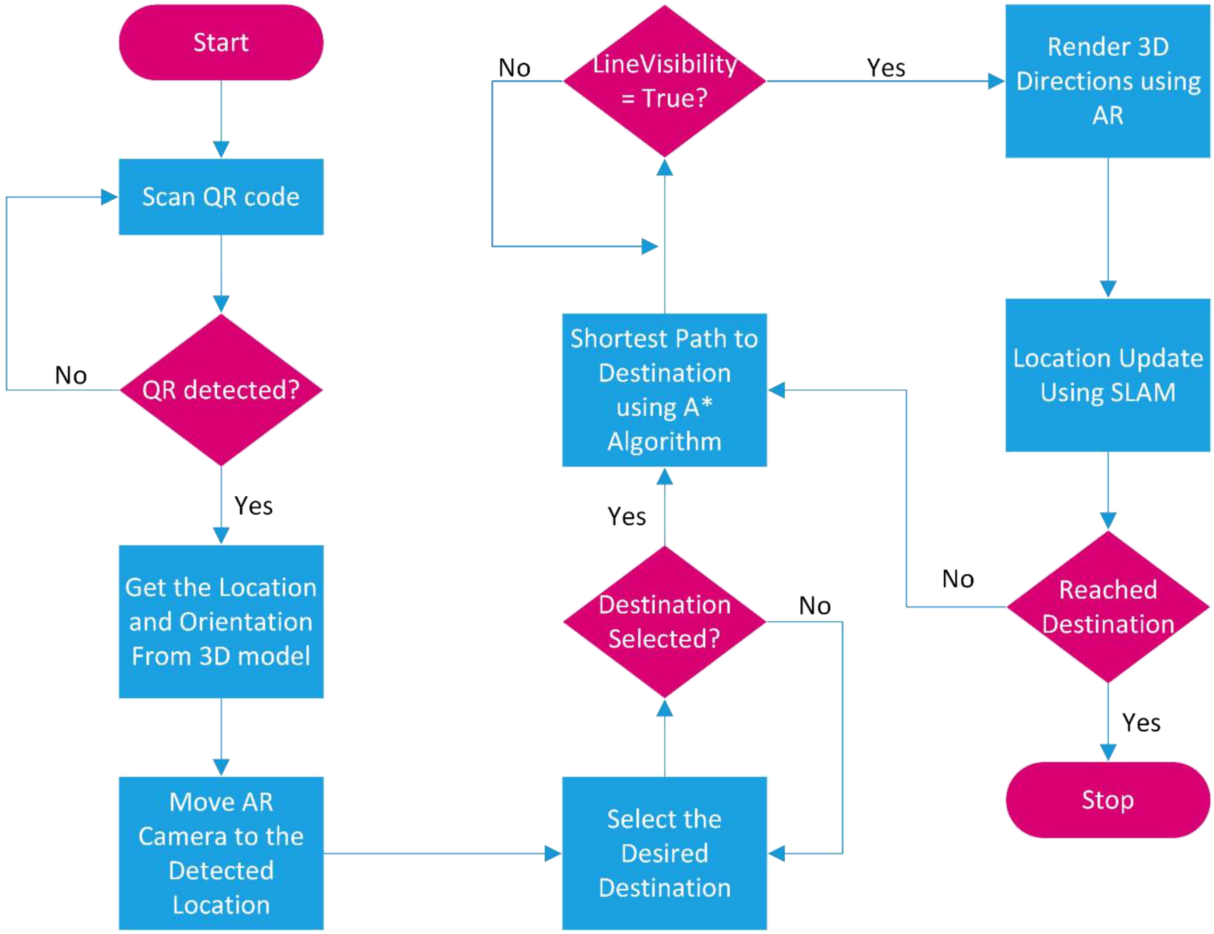

Figure 2.2 Flow chart of project.

QR Code Scanner

The application initiates by checking if a QR code is detected. The system will not proceed further until a QR code is successfully scanned.

Localization

Upon scanning the QR code, the application compares the information obtained from the code with the QR codes present in the virtual environment. If there is a match, the system receives the location and orientation data, allowing the agent (user) to be accurately localized at the corresponding position in the virtual environment.

Destination Selection

After successful localization, the user is prompted to select their desired destination within the environment. The system awaits the user's input for the destination selection.

Path Calculation

Once the user selects the desired destination, the A* algorithm is applied to the NavMesh, using the current location and the selected destination as inputs. The A* algorithm processes this information and generates the shortest path from the user's current location to the desired destination within the virtual environment.

A* Algorithm

Figure 2.3 shows the flow chart of the A* algorithm.

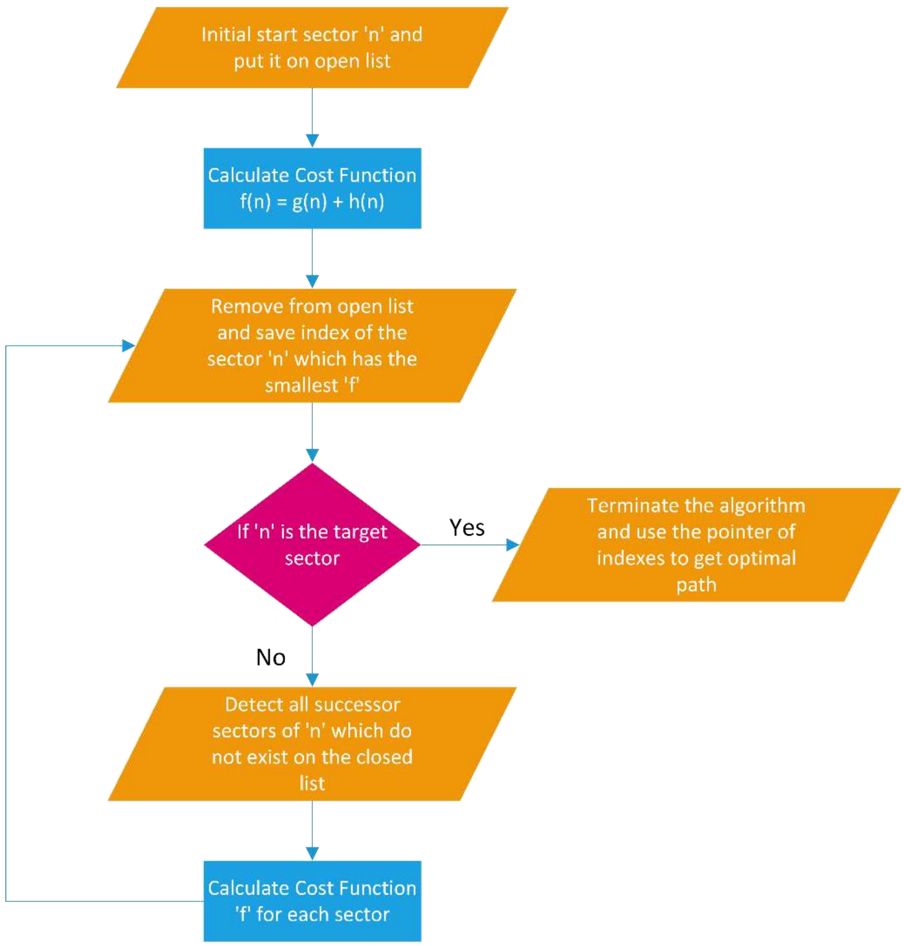

Figure 2.3 Flow chart of A* algorithm.

The A* algorithm is a widely used pathfinding algorithm that efficiently calculates the shortest path between two points in a graph or a grid-based environment. It is commonly employed in navigation systems, including the augmented reality-based indoor navigation system.

Here's an explanation of how the A* algorithm works:

- **Initialization:** The algorithm begins by initializing two lists the Open List and the Closed List.
    - The Open List stores the nodes that are yet to be evaluated.
    - The Closed List keeps track of the nodes that have already been evaluated.
- **Start Node:** The algorithm sets the starting node as the initial position from where the navigation will begin.
- **Cost Evaluation:** Each node in the graph or grid is assigned three values: G cost, H cost, and F cost.
    - G cost represents the cost to reach the current node from the start node.
    - H cost estimates the heuristic cost from the current node to the destination node (typically calculated using methods like Euclidean distance or Manhattan distance).
    - F cost is the sum of G cost and H cost, representing the total estimated cost from the start node to the destination node passing through the current node.
- **Adding Start Node to Open List:** The starting node is added to the Open List for evaluation.
- **A* Algorithm Loop:** The algorithm enters a loop that continues until either the destination node is reached or there are no more nodes to evaluate.
- **Selecting Node with Lowest F Cost:** From the nodes in the Open List, the algorithm selects the node with the lowest F cost for evaluation.
- **Evaluating Neighboring Nodes:** The algorithm examines the neighboring nodes of the selected node. For each neighboring node, it calculates:
    - The G cost (the cost to reach the neighboring node from the current node)
    - The H cost (the estimated cost from the neighboring node to the destination node)
    - The F cost of the neighboring node is calculated by summing the G cost and H cost.
- **Updating Costs and Parent Nodes:** If a neighboring node has a lower G cost when reached from the currently selected node, the algorithm updates the G cost and sets the parent node of the neighboring node to the current node. This step ensures that the algorithm is selecting the most optimal path based on the lowest cost.
- **Adding Neighboring Nodes to Open List:** The neighboring nodes that have been evaluated are added to the Open List for further consideration.
- **Destination Check:** At each iteration, the algorithm checks if the destination node has been reached. If so, the algorithm exits the loop.
- **Backtracking the Path:** Once the destination node is reached, the algorithm backtracks from the destination node to the start node, following the parent nodes. This process determines the optimal path from the start to the destination, as it traces the nodes with the lowest G cost.
- **Path Completion:** The algorithm completes once the shortest path from the start node to the destination node is determined.

The A* algorithm combines the evaluation of node costs, heuristic estimates, and path backtracking to efficiently find the shortest path. By considering both the actual cost and the estimated cost to reach the destination, A* algorithm ensures effective navigation within the augmented reality environment, providing the user with the most optimized route to their desired location.

2.4 3D Environment Setup in Unity 3D

The 3D environment setup in Unity 3D is a crucial aspect of the project. This section focuses on the steps involved in creating a realistic and immersive virtual environment for navigation, starting with the creation of floor plans in Microsoft Visio.

To begin with, the floor plans of the indoor space are created using Microsoft Visio. The floor plans created in Microsoft Visio, as shown below serve as a blueprint for designing the virtual environment. Microsoft Visio provides a user-friendly interface for creating accurate and detailed floor plans, including walls, rooms, doors, and other architectural elements.

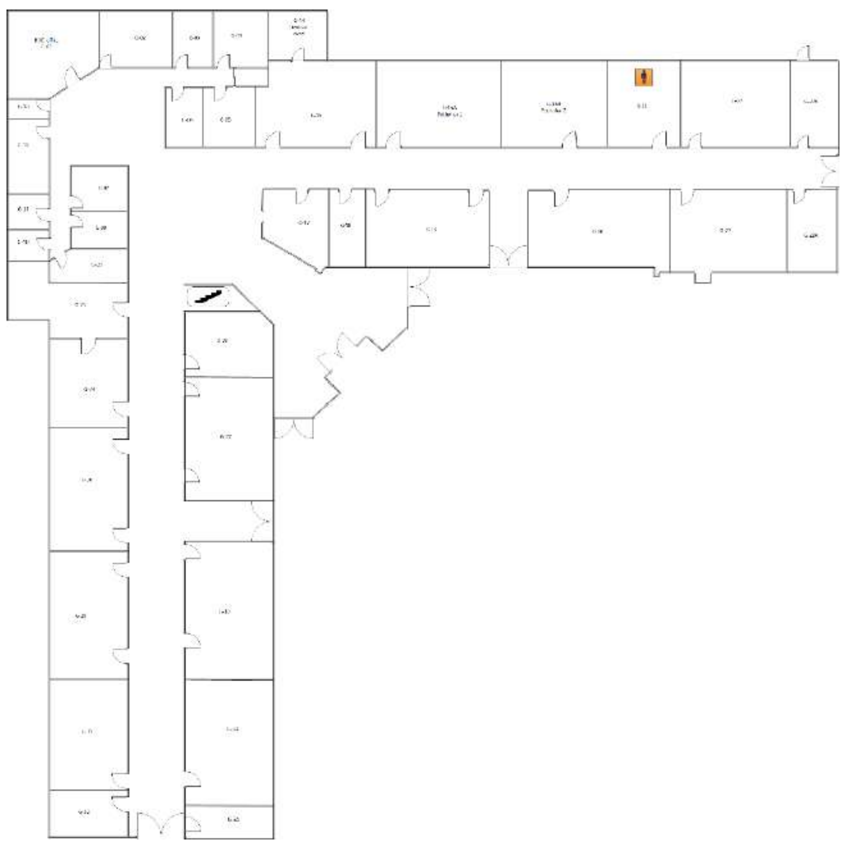

Figure 2.4 Floor plan created on ms visio.

Once the floor plans are finalized in Visio, they are exported in a compatible format, such as a 2D image or a vector-based file. These floor plans are then imported into Unity 3D, where they serve as the foundation for constructing the 3D model of the virtual environment.

Using the imported floor plans as a reference, 3D models of walls, floors, furniture, and other objects are created within Unity 3D, as showcased below. The modeling process involves placing and arranging the virtual objects to match the layout and structure of the physical environment depicted in the floor plans.

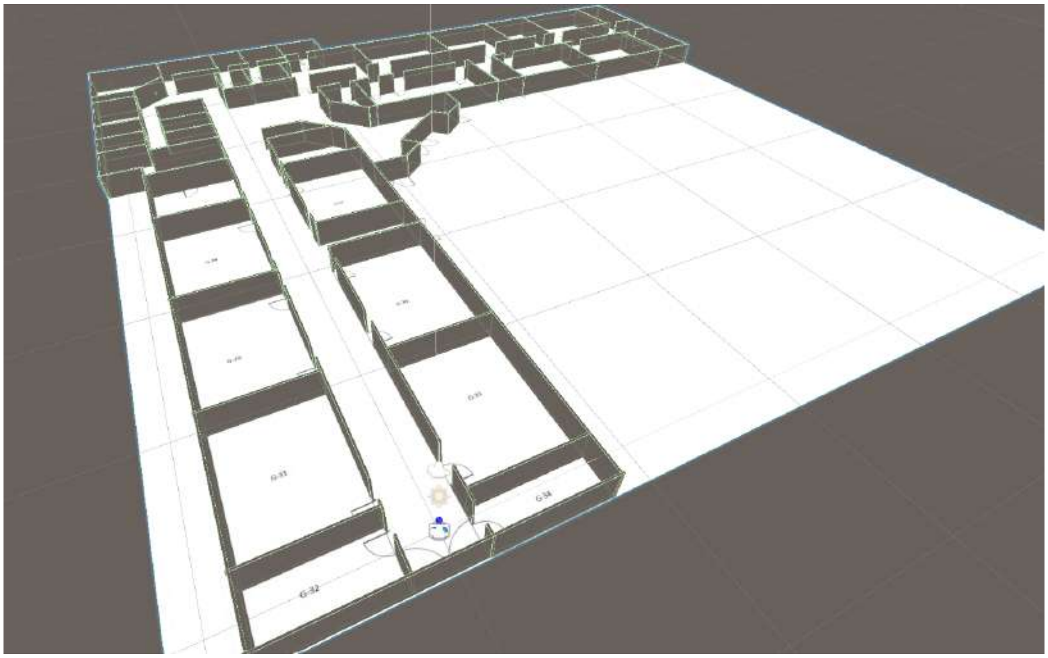

Figure 2.5 3D model created on unity 3D

The combination of Microsoft Visio for floor plan creation and Unity 3D for 3D modeling allows for a streamlined workflow in designing and constructing the virtual environment. The use of floor plans as a reference ensures accuracy and consistency with the physical space, while Unity 3D provides the necessary tools and capabilities to bring the 3D environment to life.

2.5 QR Code Localization

QR code localization is a fundamental aspect of the proposed solution for indoor navigation. This section focuses on the details of QR code localization and its implementation.

Firstly, QR codes are strategically placed within the physical environment at specific locations. These QR codes serve as reference points for the system to identify the user's position.

In the mobile application, a QR code scanner is implemented to detect and read the QR codes. When the user scans a QR code using the device's camera, the application processes the scanned information to determine the user's location within the virtual environment.

The application compares the scanned QR code information with the pre-defined QR code database, which contains the corresponding virtual positions and orientations. By matching the scanned information with the database, the application localizes the user accurately.

The localization information obtained from the QR codes is then used as input for the subsequent path planning and navigation guidance stages. It enables the system to calculate the shortest path from the user's current location to the desired destination.

QR code localization provides an effective and reliable method for indoor positioning, as QR codes can be easily placed and scanned within the environment. It allows for precise localization, ensuring accurate navigation within the virtual environment.

2.6 Path Planning and Navigation Guidance

Path planning and navigation guidance are integral components of the proposed solution. This section delves into the details of how the system calculates the optimal path and provides guidance to the user.

After the user's location has been localized using QR code information, path planning algorithms are employed to calculate the shortest path from the user's current location to the desired destination. The A* algorithm, a popular pathfinding algorithm, is commonly used for this purpose. The A* algorithm considers factors such as distance, obstacles, and path cost to determine the optimal route.

The NavMesh feature in Unity 3D is utilized to generate a navigation mesh (weighted graph), which represents the walkable areas within the virtual environment. The NavMesh acts as a map that guides the path planning algorithm and ensures that the calculated path adheres to the walkable surfaces.

Once the optimal path is determined, the application provides navigation guidance to the user. This guidance can be visualized using augmented reality (AR) markers or other intuitive methods. AR markers can be overlaid on the user's device camera view, displaying the path as virtual markers in the real-world environment.

The user receives real-time instructions and feedback, such as turn-by-turn directions. This guidance system assists the user in navigating through the virtual environment, making the indoor navigation experience intuitive and user-friendly.

2.7 Integration with AR and Location Tracking using SLAM

The integration of augmented reality (AR) and location tracking using Simultaneous Localization and Mapping (SLAM) enhances the immersive and interactive aspects of the proposed solution. This section focuses on how AR and SLAM technologies are incorporated into the project.

AR Foundation, which leverages AR Core, is utilized to enable the integration of augmented reality within the mobile application. AR features, such as virtual markers, overlays, and visual guidance, are overlaid on the user's real-world environment, providing a seamless blending of the virtual and physical worlds.

SLAM technology is employed to track the user's location and orientation within the virtual environment in real-time. It utilizes the device's camera, motion sensors, and sophisticated algorithms to simultaneously build a map of the environment and estimate the user's position and orientation within that map. SLAM achieves this by iteratively fusing sensor data and updating the map and pose estimates as the user moves through the environment.

The SLAM algorithm starts by capturing and analyzing the camera feed, extracting visual features and landmarks from the scene. These features are then matched and tracked across consecutive frames to determine the motion of the camera. By leveraging the motion data from the camera and other sensors, SLAM can estimate the user's position and orientation relative to the environment. This provide accurate location tracking as showcased in the table.

Waypoint \ **Technology**	**GPS**	**MARKER**	**AR SLAM**
Waypoint 1: 0m	0 m	0 m	0 m
Waypoint 2: 0.75 m	0.1 m	0.1 m	0.025 m
Waypoint 3: 1.5 m	0.2 m	0.1 m	0.01 m
Waypoint 4: 2.25 m	0.15 m	0.25 m	0.015 m
Waypoint 5: 3 m	0.3 m	0.3 m	0.02 m
Waypoint 6: 3.75 m	0.1 m	0.05 m	0.01 m
Total Variation	0.85 m	0.8 m	0.08 m

Table 2.1 **Technology comparison**

As the user moves, SLAM continuously refines the map and updates the user's position and orientation estimates. This allows for accurate tracking of the user's movements, even in dynamic environments with changing lighting conditions or occlusions.

The SLAM-generated map serves as a reference for aligning the virtual objects, markers, and overlays within the user's real-world view. By registering the virtual content to the real-world environment, the augmented reality experience becomes seamlessly integrated with the user's surroundings.

SLAM technology provides several advantages in the context of indoor navigation. It enables precise positioning and tracking of the user, ensuring that the virtual elements are accurately overlaid on the physical environment. This accuracy is essential for providing reliable guidance and navigation instructions to the user as they traverse the indoor space.

Furthermore, SLAM allows for dynamic updates to the map and user pose estimates, accommodating changes in the environment or user movements. This adaptability ensures that the navigation system remains robust and responsive, providing an optimal user experience.

By integrating SLAM technology into the proposed solution, the augmented reality-based indoor navigation system achieves high accuracy in location tracking, real-time mapping, and alignment of virtual content with the physical environment. This integration enhances the overall usability, immersion, and effectiveness of the application, providing users with a seamless and interactive navigation experience.

2.8 User Interface of the Application

The user interface (UI) of the application is a crucial aspect in ensuring a user-friendly and intuitive experience. This section focuses on the design and implementation of the application's UI.

The UI is designed to be visually appealing and easy to navigate. It includes features such as a QR code scanner interface, destination selection options, and navigation guidance displays.

The QR code scanner interface provides clear instructions and visual feedback to guide the user in scanning the QR codes accurately. It ensures a seamless and efficient localization process.

The destination selection feature allows users to input their desired destination within the virtual environment. It can include options such as search functionality, pre-defined points of interest, or manual input.

Navigation guidance displays provide real-time instructions and visual cues to guide the user along the calculated path. The display includes an AR line to assist the user in reaching their destination.

The UI is designed to be intuitive and user-friendly, allowing users to easily navigate through the application and interact with the augmented reality-based indoor navigation system.

By paying careful attention to the design and implementation of the user interface, the application ensures a seamless and enjoyable user experience, enhancing the overall effectiveness and usability of the augmented reality-based indoor navigation system.

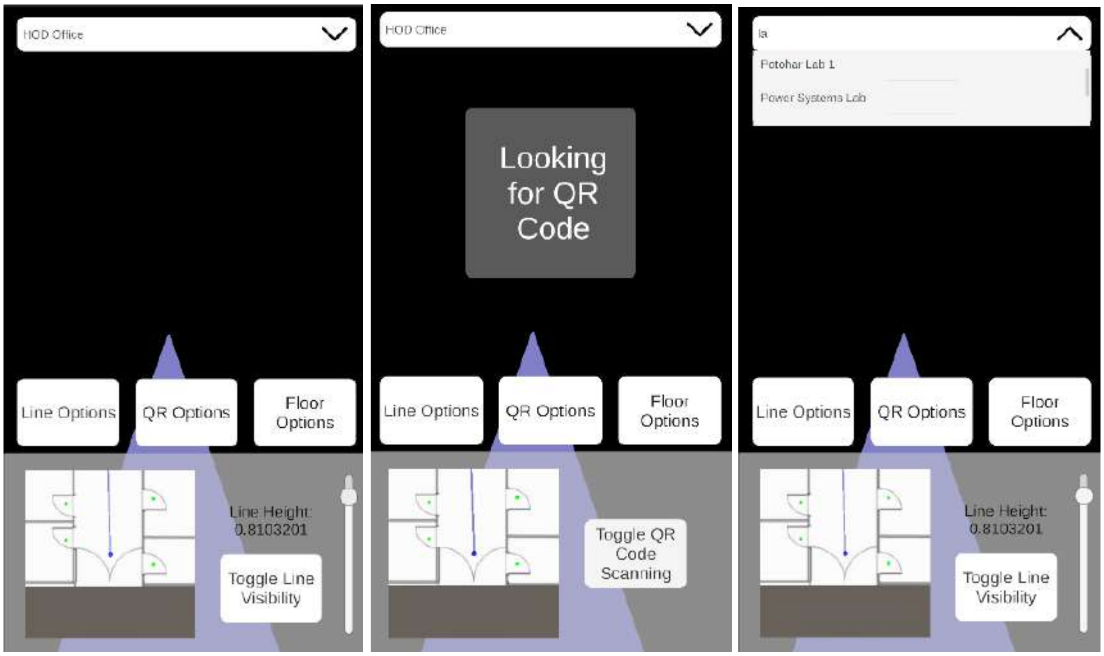

Figure 2.6 UI of mobile application.

Chapter 3 Result and Recommendations

The implementation of the augmented reality-based indoor navigation has yielded promising results. This chapter provides a description of the results obtained from the project, along with the corresponding simulations and testing.

To evaluate the functionality and performance of the system, extensive simulations and testing were conducted on both unity 3D and android phone. The simulations aimed to assess the accuracy of the QR code localization, the effectiveness of the path planning algorithm, and the user experience during navigation.

3.1 QR Code Localization Results

The QR code localization process demonstrated reliable results, with the system accurately identifying and localizing the user's position based on the scanned QR codes. The system successfully matched the scanned QR code information with the virtual QR codes in the 3D environment, ensuring accurate localization within the virtual space.

3.2 Results of Path Planning

The path planning algorithm was thoroughly evaluated by simulating different navigation scenarios. Users were assigned various starting points and destinations within the virtual environment, and the system calculated optimal paths using the A* algorithm. The effectiveness of the algorithm was assessed based on factors such as path length, avoidance of obstacles, and efficient route selection. The results demonstrated the algorithm's ability to generate optimal paths, ensuring efficient navigation from one point to another. As showcased by the images below:

Figure 3.1 Path calculation simulation (a)

Figure 3.2 Path calculation simulation (b)

3.3 Results of Android Device Testing

In addition to the Unity 3D simulations, the system was also tested on Android devices to evaluate its performance in a real-world setting. Extensive testing was conducted to assess the system's responsiveness, stability, and accuracy on different Android devices. The application consistently delivered reliable performance, demonstrating its suitability for practical use on mobile devices.

To gauge the user experience during navigation, a group of participants performed simulated navigation tasks using the Android application. Their feedback on the usability, intuitiveness, and overall satisfaction with the system was collected and analyzed. Participants reported a positive experience, highlighting the clear navigation instructions, accurate positioning, and seamless integration of augmented reality elements within their real-world surroundings.

3.4 Conclusion

The augmented reality-based indoor navigation system developed using Unity 3D, QR code localization, and integration with AR and SLAM technologies has successfully addressed the need for effective navigation in complex indoor environments. This project has demonstrated the feasibility and practicality of using augmented reality to enhance indoor navigation experiences.

Through extensive simulations and testing, the system showcased its accuracy in QR code localization, providing precise positioning of users within the virtual environment. The integration of the A* algorithm for path planning ensured optimal route calculations, taking into account obstacles and walkable surfaces. The navigation guidance, presented through augmented reality overlays, delivered clear and intuitive instructions to users, enabling them to navigate through intricate indoor spaces with ease.

The integration of AR and SLAM technologies further enhanced the system's performance, allowing real-time tracking of users' positions and orientations within the environment. This integration enabled accurate alignment of virtual markers and overlays with the physical surroundings, creating a seamless and immersive navigation experience.

User feedback and testing confirmed the system's usability and effectiveness. Participants reported a positive user experience, noting the system's intuitive interface, accurate positioning, and helpful navigation guidance. The combination of simulations in Unity 3D and real-world testing on Android devices ensured the system's reliability and suitability for practical use.

Overall, the developed augmented reality-based indoor navigation system has shown great potential in overcoming the limitations of traditional navigation methods in complex indoor settings. By leveraging the power of Unity 3D, QR code localization, and AR/SLAM technologies, the system provides users with a seamless and immersive navigation solution.

In conclusion, the augmented reality-based indoor navigation system offers a promising solution for efficient and intuitive navigation in complex indoor spaces. Its successful implementation opens up possibilities for enhancing user experiences, improving accessibility, and transforming how people navigate and interact with indoor environments.

3.5 Recommendations / Future Work

Incorporating Machine Learning:

To further enhance the system's capabilities, integrating machine learning algorithms can provide personalized navigation recommendations based on user preferences and historical data. Machine learning techniques can also be utilized to improve the accuracy of localization and path planning algorithms, adapting to the specific characteristics of different indoor environments.

Utilizing Manhattan Mobility Model:

Instead of relying solely on NavMesh for path planning, incorporating the Manhattan mobility model can enhance the system's flexibility. By creating custom waypoints based on the Manhattan distance metric, the system can generate more efficient and optimized paths, especially in complex indoor environments with irregular layouts and obstacles.

Voice-Guided Instructions:

Enhancing the user experience by implementing voice-guided instructions can provide a hands-free navigation experience. Integrating speech synthesis capabilities into the application will allow users to receive real-time navigation instructions without the need to visually interact with the device.

Real-Time Obstacle Detection:

Implementing real-time obstacle detection using computer vision or sensor-based technologies can improve the safety and reliability of the navigation system. By continuously monitoring the environment for obstacles and dynamically adjusting the navigation path, the system can ensure a seamless and obstacle-free navigation experience.

Integration with Smart Building Systems:

Further integration with smart building systems, such as IoT devices, can enable enhanced functionality. By leveraging data from sensors and smart devices within the building, the system can provide real-time information about occupancy, room availability, and environmental conditions, enriching the navigation experience and facilitating efficient utilization of indoor spaces.

Performance Optimization:

Continual research and development efforts should focus on optimizing the system's performance and accuracy. This includes optimizing computational algorithms, minimizing latency, and improving the responsiveness of the system to ensure smooth and seamless navigation experiences.

User Testing and Feedback:

Conducting extensive user testing and gathering feedback from a diverse range of users in various real-world indoor environments will provide valuable insights for further system improvements. This iterative process will help identify any usability issues, address user preferences, and refine the system to better meet the needs of different user groups.

By implementing these recommendations and conducting further research and development, the augmented reality-based indoor navigation system can continue to evolve, offering enhanced functionality, improved performance, and superior user experience. These advancements will contribute to the widespread adoption of the system in various indoor settings, including shopping malls, airports, museums, and large-scale buildings, revolutionizing the way people navigate and interact within these spaces.

Appendix-A: Project Codes

A-1: Target Selection

```
using System;
using UnityEngine;

[Serializable]
public class Target
{
    public string Name;
    public GameObject PositionObject;
}
```

A-2: Set Navigation Target, Calculate and Render Path

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.AI;
using UnityEngine.UI;
using TMPro;

public class SetNavigationTarget : MonoBehaviour
{
    [SerializeField]
    private List<Target> navigationTargetObjects = new List<Target>();
    [SerializeField]
    private Slider navigationYOffset;

    private NavMeshPath path;
    private LineRenderer line;
    private Vector3 targetPosition = Vector3.zero;

    public SearchableDropDown SearchableDropDown;
    private List<string> options = new List<string>();
    private bool lineToggle = false;
    private int currentFloor = 0;

    // Start is called before the first frame update
    private void Start()
    {
        path = new NavMeshPath();
        line = transform.GetComponent<LineRenderer>();
        line.enabled = lineToggle;
    }

    // Update is called once per frame
    private void Update()
    {
        if (lineToggle && targetPosition != Vector3.zero)
        {
            NavMesh.CalculatePath(transform.position, targetPosition, NavMesh.AllAreas,
path);
```

Appendix

```
            line.positionCount = path.corners.Length;
            Vector3[] calculatedPathAndOffset = AddLineOffset();
            line.SetPositions(calculatedPathAndOffset);
        }
    }
    public void SetCurrentNavigationTarget(string selectedValue)
    {
        targetPosition = Vector3.zero;
        string selectedText = selectedValue;
        Target currentTarget = navigationTargetObjects.Find(x =>
x.Name.ToLower().Equals(selectedText.ToLower()));
        if (currentTarget != null)
        {
            if (!line.enabled)
            {
                ToggleVisibility();
            }
            targetPosition = currentTarget.PositionObject.transform.position;
        }
    }
    public void ToggleVisibility()
    {
        lineToggle = !lineToggle;
        line.enabled = lineToggle;
    }

    public void ChangeActiveFloor(int floorNumber)
    {
        currentFloor = floorNumber;
        SetNavigationTargetDropDownOptions(currentFloor);
    }

    private void SetNavigationTargetDropDownOptions(int floorNumber)
    {
        options.Clear();
        SearchableDropDown.ClearDropdown();

        if (line.enabled)
        {
            ToggleVisibility();
        }

        if (floorNumber == 0)
        {
            options.Add("HOD Office");
            options.Add("Main Entrance");
            options.Add("Potohar Lab 2");
            options.Add("Potohar Lab 1");
            options.Add("Mens Room (Ground Floor)");
            options.Add("Power Systems Lab");
            options.Add("Room G-22A");
            options.Add("Room G-23A");
            options.Add("Machine Lab");
            options.Add("Room G-20");
            options.Add("Room G-19");
            options.Add("Academic Office");
            options.Add("Room G-15");
            options.Add("Meeting Room");
```

```
            options.Add("Girls Common Room (Ground Floor)");
            options.Add("Room G-26");
            options.Add("Room G-27");
            options.Add("Room G-30");
            options.Add("Room G-33");
            options.Add("Room G-34");
            options.Add("Room G-32");
            options.Add("Room G-31");
            options.Add("Room G-29");
            options.Add("Room G-28");
            options.Add("Room G-03");
            options.Add("Prof Ata ul Aziz Office");
            options.Add("Prof Sana Saleh Office");
            options.Add("Prof Maria Nasir Office");
            options.Add("Prof Momal Bukhari Office");
            options.Add("Prof Faryal Gulla Office");
            options.Add("Faculty Kitchen (Ground Floor)");
            options.Add("Faculty Mens Room (Ground Floor)");
            options.Add("Prof Mukhtarullah Office");
            options.Add("Prof Amir Hafeez Office");
            options.Add("Prof Amir Muneer Office");
            options.Add("Ground Floor Stairs");
        }

        if (floorNumber == 1)
        {
            options.Add("Faculty Lounge");
            options.Add("Physics Lab");
            options.Add("FeedBack Lab");
            options.Add("Electronics Lab");
            options.Add("Circuit Lab");
            options.Add("Kohsar Lab");
            options.Add("IC Design Lab");
            options.Add("Radio Frequency Circuit System and Sensors Lab");
            options.Add("Maker Lab");
            options.Add("Computer Server Room");
            options.Add("Prof Khalilullah Office");
            options.Add("Room B-117");
            options.Add("Room B-126");
            options.Add("Room B-127");
            options.Add("Room B-130");
            options.Add("Room B-129");
            options.Add("Room B-132A");
            options.Add("Communication and Antenna Lab");
            options.Add("Mens Room");
            options.Add("Room B-122A");
            options.Add("Room B-123A");
            options.Add("Prof Niaz Ahmed Office");
            options.Add("Prof Muhammad Saeed Office");
            options.Add("Prof Arshad Hassan Office");
            options.Add("Prof Awais Ayub Office");
            options.Add("Prof Waseem Ikram Office");
            options.Add("Prof Muhammad Jafar Office");
            options.Add("Prof Shahid Qureshi Office");
            options.Add("Prof Muhammad Ibrar Office");
            options.Add("Prof Shahzad Saleem Office");
            options.Add("Prof Shahzad Ahmad Office");
            options.Add("Prof Azhar Rauf Office");
```

```
            options.Add("Prof Farhan Khalid Office");
            options.Add("Faculty Mens Room");
            options.Add("Ladies Room");
            options.Add("Stairs");
        }

        if (floorNumber == 2)
        {
            options.Add("Visiting Ficulty Lounge");
            options.Add("Room B-231");
            options.Add("Room B-232");
            options.Add("Room B-229");
            options.Add("Rohtas Lab");
            options.Add("Room B-225");
            options.Add("Ladies Room (2nd Floor)");
            options.Add("Room B-226");
            options.Add("Room B-227");
            options.Add("Room B-230");
            options.Add("Room B-232A");
            options.Add("Prof Hamda Khan Office");
            options.Add("Room B-212A");
            options.Add("Prof Khadija Farooq Office");
            options.Add("Prof Sara Aziz Office");
            options.Add("Faculty Mens Room (2nd Floor)");
            options.Add("Prof Aisha ijaz Office");
            options.Add("Prof Mehwish Hassan Office");
            options.Add("Faculty Ladies Room (2nd Floor)");
            options.Add("Prof Tayyab Nadeem Office");
            options.Add("Prof Usman Ashraf Office");
            options.Add("Prof Muhammad Ali Office");
            options.Add("Prof Irfan Shah Office");
            options.Add("Prof Muhammad Ibrahim Office");
            options.Add("Prof Shahzad Mahmood Office");
            options.Add("EPIC Lab");
            options.Add("Huawei Lab");
            options.Add("TDA Lab");
            options.Add("SEAL Lab");
            options.Add("IPC Lab");
            options.Add("DLD Lab");
            options.Add("Room B-222");
            options.Add("Room B-220");
            options.Add("Mens Room (2nd Floor)");
            options.Add("MIP Lab");
            options.Add("QUEST Lab");
            options.Add("Stairs (2nd Floor)");
        }

        SearchableDropDown.ChangeOptions(options);
    }



    private Vector3[] AddLineOffset()
    {
        if (navigationYOffset.value == 0)
        {
            return path.corners;
        }
```

```
        Vector3[] calculatedLine = new Vector3[path.corners.Length];
        for (int i = 0; i < path.corners.Length; i++)
        {
            calculatedLine[i] = path.corners[i] + new Vector3(0,
navigationYOffset.value, 0);
        }
        return calculatedLine;
    }
}
```

A-3: QR Code Detection

```
using UnityEngine;
using TMPro;
using UnityEngine.XR.ARFoundation;
using ZXing;

public class GetImageAlternative : MonoBehaviour
{
    [SerializeField]
    private ARCameraBackground arCameraBackground;
    [SerializeField]
    private RenderTexture targetRenderTexture;
    [SerializeField]
    private TextMeshProUGUI qrCodeText;

    private Texture2D cameraImageTexture;
    private IBarcodeReader reader = new BarcodeReader(); // create a barcode reader
instance

    private void Update()
    {
        Graphics.Blit(null, targetRenderTexture, arCameraBackground.material);
        cameraImageTexture = new Texture2D(targetRenderTexture.width,
targetRenderTexture.height, TextureFormat.RGBA32, false);
        Graphics.CopyTexture(targetRenderTexture, cameraImageTexture);

        // Detect and decode the barcode inside the bitmap
        var result = reader.Decode(cameraImageTexture.GetPixels32(),
cameraImageTexture.width, cameraImageTexture.height);

        // Do something with the result
        if (result != null)
        {
            qrCodeText.text = result.Text;
        }
    }
}
```

A-4: QR Code Localization

```
using System.Collections.Generic;
using Unity.Collections;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;
using ZXing;

public class QrCodeRecenter : MonoBehaviour
{
    [SerializeField]
    private ARSession session;
    [SerializeField]
    private ARSessionOrigin sessionOrigin;
    [SerializeField]
    private ARCameraManager cameraManager;
    [SerializeField]
    private List<Target> navigationTargetObjects = new List<Target>();
    [SerializeField]
    private GameObject qrCodeScanningPanel;

    private Texture2D cameraImageTexture;
    private IBarcodeReader reader = new BarcodeReader();
    private bool scanningEnabled = false;


    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            SetQrCodeRecenterTarget("Start");
        }
    }

    private void OnEnable()
    {
        cameraManager.frameReceived += OnCameraFrameReceived;
    }

    private void OnDisable()
    {
        cameraManager.frameReceived -= OnCameraFrameReceived;
    }

    private void OnCameraFrameReceived(ARCameraFrameEventArgs eventArgs)
    {
        if (!scanningEnabled)
        {
            return;
        }

        if (!cameraManager.TryAcquireLatestCpuImage(out XRCpuImage image))
        {
            return;
        }

        var conversionParams = new XRCpuImage.ConversionParams
```

```
        {
            // Get the entire image.
            inputRect = new RectInt(0, 0, image.width, image.height),

            // Downsample by 2.
            outputDimensions = new Vector2Int(image.width / 2, image.height / 2),

            // Choose RGBA format.
            outputFormat = TextureFormat.RGBA32,

            // Flip across the vertical axis (mirror image).
            transformation = XRCpuImage.Transformation.MirrorY
        };

        // See how many bytes you need to store the final image.
        int size = image.GetConvertedDataSize(conversionParams);

        // Allocate a buffer to store the image.
        var buffer = new NativeArray<byte>(size, Allocator.Temp);

        // Extract the image data
        image.Convert(conversionParams, buffer);

        // The image was converted to RGBA32 format and written into the provided
buffer
        // so you can dispose of the XRCpuImage. You must do this or it will leak
resources.
        image.Dispose();

        // At this point, you can process the image, pass it to a computer vision
algorithm, etc.
        // In this example, you apply it to a texture to visualize it.

        // You've got the data; let's put it into a texture so you can visualize it.
        cameraImageTexture = new Texture2D(
            conversionParams.outputDimensions.x,
            conversionParams.outputDimensions.y,
            conversionParams.outputFormat,
            false);

        cameraImageTexture.LoadRawTextureData(buffer);
        cameraImageTexture.Apply();

        // Done with your temporary data, so you can dispose it.
        buffer.Dispose();

        // Detect and decode the barcode inside the bitmap
        var result = reader.Decode(cameraImageTexture.GetPixels32(),
cameraImageTexture.width, cameraImageTexture.height);

        // Do something with the result
        if (result != null)
        {
            SetQrCodeRecenterTarget(result.Text);
            ToggleScanning();
        }

    }
```

```
    private void SetQrCodeRecenterTarget(string targetText)
    {
        Target currentTarget = navigationTargetObjects.Find(x =>
x.Name.ToLower().Equals(targetText.ToLower()));
        if (currentTarget != null)
        {
            session.Reset();

            sessionOrigin.transform.position =
currentTarget.PositionObject.transform.position;
            sessionOrigin.transform.rotation =
currentTarget.PositionObject.transform.rotation;
        }
    }

    public void ChangeActiveFloor(string floorEntrance)
    {
        SetQrCodeRecenterTarget(floorEntrance);
    }

    public void ToggleScanning()
    {
        scanningEnabled = !scanningEnabled;
        qrCodeScanningPanel.SetActive(scanningEnabled);
    }
}
```

A-5: Searchable Drop Down

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using TMPro;
//using UnityEditor.EditorTools;
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;
using UnityEngine.UIElements;
using Button = UnityEngine.UI.Button;
using Image = UnityEngine.UI.Image;

public class SearchableDropDown : MonoBehaviour
{
    [SerializeField] private Button blockerButton;
    [SerializeField] private GameObject buttonsPrefab = null;
    [SerializeField] private int maxScrollRectSize = 300;
    [SerializeField] private List<string> avlOptions = new List<string>();


    private Button ddButton = null;
    private TMP_InputField inputField = null;
    private ScrollRect scrollRect = null;
    private Transform content = null;
    private RectTransform scrollRectTrans;
    private bool isContentHidden = true;
```

```
    private List<Button> initializedButtons = new List<Button>();

    public delegate void OnValueChangedDel(string val);
    public OnValueChangedDel OnValueChangedEvt;

    void Start()
    {
        Init();
    }

    /// <summary>
    /// Initilize all the Fields
    /// </summary>
    private void Init()
    {
        ddButton = this.GetComponentInChildren<Button>();
        scrollRect = this.GetComponentInChildren<ScrollRect>();
        inputField = this.GetComponentInChildren<TMP_InputField>();
        scrollRectTrans = scrollRect.GetComponent<RectTransform>();
        content = scrollRect.content;

        //blocker is a button added and scaled it to screen size so that we can close
the dd on clicking outside
        blockerButton.GetComponent<RectTransform>().sizeDelta = new
Vector2(Screen.width, Screen.height);
        blockerButton.gameObject.SetActive(false);

blockerButton.transform.SetParent(this.GetComponentInParent<Canvas>().transform);

        blockerButton.onClick.AddListener(OnBlockerButtClick);
        ddButton.onClick.AddListener(OnDDButtonClick);
        scrollRect.onValueChanged.AddListener(OnScrollRectvalueChange);
        inputField.onValueChanged.AddListener(OnInputvalueChange);
        inputField.onEndEdit.AddListener(OnEndEditing);

        AddItemToScrollRect(avlOptions);

    }

    /// <summary>
    /// public method to get the selected value
    /// </summary>
    /// <returns></returns>
    public string GetValue()
    {
        return inputField.text;
    }

    public void ResetDropDown()
    {
        inputField.text = string.Empty;

    }

    //call this to Add items to Drop down
    public void AddItemToScrollRect(List<string> options)
    {
        foreach (var option in options)
```

```
        {
            var buttObj = Instantiate(buttonsPrefab, content);
            buttObj.GetComponentInChildren<TMP_Text>().text = option;
            buttObj.name = option;
            buttObj.SetActive(true);
            var butt = buttObj.GetComponent<Button>();
            butt.onClick.AddListener(delegate { OnItemSelected(buttObj); });
            initializedButtons.Add(butt);
        }
        ResizeScrollRect();
        scrollRect.gameObject.SetActive(false);
    }


    /// <summary>
    /// listner To Input Field End Editing
    /// </summary>
    /// <param name="arg"></param>
    private void OnEndEditing(string arg)
    {
        if (string.IsNullOrEmpty(arg))
        {
            Debug.Log("no value entered ");
            return;
        }
        StartCoroutine(CheckIfValidInput(arg));
    }

    /// <summary>
    /// Need to wait as end inputField and On option button  Contradicted and message
was poped after selection of button
    /// </summary>
    /// <param name="arg"></param>
    /// <returns></returns>
    IEnumerator CheckIfValidInput(string arg)
    {
        yield return new WaitForSeconds(1);
        if (!avlOptions.Contains(arg))
        {
           // Message msg = new Message("Invalid Input!", "Please choose from
dropdown",
           //                 this.gameObject, Message.ButtonType.OK);
           //
           //             if (MessageBox.instance)
           //                 MessageBox.instance.ShowMessage(msg);

            inputField.text = String.Empty;
        }
        //else
        //    Debug.Log("good job " );
        OnValueChangedEvt?.Invoke(inputField.text);
    }
    /// <summary>
    /// Called ever time on Drop down value is changed to resize it
    /// </summary>
    private void ResizeScrollRect()
    {
        //TODO Dont Remove this until checked on Mobile Deveice
```

```
        //var count = content.transform.Cast<Transform>().Count(child =>
child.gameObject.activeSelf);
        //var length = buttonsPrefab.GetComponent<RectTransform>().sizeDelta.y * count;

        LayoutRebuilder.ForceRebuildLayoutImmediate((RectTransform)content.transform);
        var length = content.GetComponent<RectTransform>().sizeDelta.y;

        scrollRectTrans.sizeDelta = length > maxScrollRectSize ? new
Vector2(scrollRectTrans.sizeDelta.x,
            maxScrollRectSize) : new Vector2(scrollRectTrans.sizeDelta.x, length + 5);
    }

    /// <summary>
    /// listner to the InputField
    /// </summary>
    /// <param name="arg0"></param>
    private void OnInputvalueChange(string arg0)
    {
        if (!avlOptions.Contains(arg0))
        {
            FilterDropdown(arg0);
        }
    }

    /// <summary>
    /// remove the elements from the dropdown based on Filters
    /// </summary>
    /// <param name="input"></param>
    public void FilterDropdown(string input)
    {
        if (string.IsNullOrEmpty(input))
        {
            foreach (var button in initializedButtons)
                button.gameObject.SetActive(true);
            ResizeScrollRect();
            scrollRect.gameObject.SetActive(false);
            return;
        }

        var count = 0;
        foreach (var button in initializedButtons)
        {
            if (!button.name.ToLower().Contains(input.ToLower()))
            {
                button.gameObject.SetActive(false);
            }
            else
            {
                button.gameObject.SetActive(true);
                count++;
            }
        }

        SetScrollActive(count > 0);
        ResizeScrollRect();
    }

    /// <summary>
```

```
    /// Listner to Scroll rect
    /// </summary>
    /// <param name="arg0"></param>
    private void OnScrollRectvalueChange(Vector2 arg0)
    {
        //Debug.Log("scroll ");
    }

    /// <summary>
    /// Listner to option Buttons
    /// </summary>
    /// <param name="obj"></param>
    private void OnItemSelected(GameObject obj)
    {
        inputField.text = obj.name;
        foreach (var button in initializedButtons)
            button.gameObject.SetActive(true);
        isContentHidden = false;
        OnDDButtonClick();
        //OnEndEditing(obj.name);
        StopAllCoroutines();
        StartCoroutine(CheckIfValidInput(obj.name));
    }

    /// <summary>
    /// listner to arrow button on input field
    /// </summary>
    private void OnDDButtonClick()
    {
        if(GetActiveButtons()<=0)
            return;
        ResizeScrollRect();
        SetScrollActive(isContentHidden);
    }
    private void OnBlockerButtClick()
    {
        SetScrollActive(false);
    }

    /// <summary>
    /// respondisble to enable and disable scroll rect component
    /// </summary>
    /// <param name="status"></param>
    private void SetScrollActive(bool status)
    {
        scrollRect.gameObject.SetActive(status);
        blockerButton.gameObject.SetActive(status);
        isContentHidden = !status;
        ddButton.transform.localScale = status ? new Vector3(1, -1, 1) : new Vector3(1,
1, 1);
    }

    /// <summary>
    /// Return numbers of active buttons in the dropdown
    /// </summary>
    /// <returns></returns>
    private float GetActiveButtons()
    {
```

```
        var count = content.transform.Cast<Transform>().Count(child =>
child.gameObject.activeSelf);
        var length = buttonsPrefab.GetComponent<RectTransform>().sizeDelta.y * count;
        return length;
    }

    public void ClearDropdown()
    {
        avlOptions.Clear();
        foreach (var btn in initializedButtons)
        {
            Destroy(btn.gameObject);
        }
        initializedButtons.Clear();
        ResizeScrollRect();
        scrollRect.gameObject.SetActive(false);
    }

    public void ChangeOptions(List<string> options)
    {
        avlOptions = options;
        AddItemToScrollRect(avlOptions);
    }
}
```

A-5: UI for AR Line Adjustment

```
using TMPro;
using UnityEngine;

public class SetUiText : MonoBehaviour
{
    [SerializeField]
    private TMP_Text textField;
    [SerializeField]
    private string fixedText;

    public void OnSliderValueChange(float numericValue)
    {
        textField.text = $"{fixedText}: {numericValue}";
    }
}
```

Bibliography

[1] S. K. Tadepalli, P. A. Ega, and P. K. Inugurthi, “Indoor navigation using augmented reality,” International Journal of Scientific Research in Science and Technology, vol. 7, no. 4, pp. 588–592, 2021.

[2] S. C. J, S. Dileep, S. C. S, and L. E. Sunny, “INDOOR NAVIGATION SYSTEM USING AUGMENTED REALITY,” International Research Journal of Engineering and Technology (IRJET), vol. 08, no. 06, pp. 2408–2412, Jun. 2021.

[3] U. Technologies, “Navigation System in Unity,” Unity, 2023. [Online]. Available: https://docs.unity3d.com/Manual/nav-NavigationSystem.html.

[4] M. W. Technologies, “What is SLAM (simultaneous localization and mapping) – matlab & simulink,” What Is SLAM (Simultaneous Localization and Mapping) – MATLAB & Simulink - MATLAB & Simulink, 2018. [Online]. Available: https://www.mathworks.com/discovery/slam.html.

[5] Public Health Information for Scotland, "Chronic liver disease: introduction," 20 March 2018. [Online]. Available: https://www.scotpho.org.uk/health-wellbeing-and-disease/chronic-liver-disease/introduction/.