มหาวิทยาลัยอัสสัมชัญ
ASSUMPTION UNIVERSITY
of THAILAND

Senior Project Report

AR NAV

Lut Lat Aung,
Yel Lin,
Myat Min Phyo

Asst. Prof. Dr. Benjawan Srisura (Advisor)

CS 3200 Senior Project 1 (2/2024)

Senior Project Approval

Project title:	AR NAV
Academic Year:	2/2024
Authors:	Lut Lat Aung (6511163)
	Yel Lin (6520242)
	Myat Min Phyo (6511125)
Project Advisor:	Asst. Prof. Dr. Benjawan Srisura

The Senior Project committee's cooperation between the Department of Computer Science and Information Technology, Vincent Mary School of Engineering, Science and Technology, Assumption University had approved this Senior Project. The Senior Project in partial fulfilment of the requirement for the degree of Bachelor of Science in Computer Science and Information technology.

Approval Committee:

…………………………..
(Asst. Prof. Dr. Benjawan Srisura)
Project Advisor

…………………………..
(Asst. Prof. Dr. Rachsuda Setthawong)
Committee Member

…………………………..
(Asst. Prof. Dr. Darun Kesrarat)
Committee Member

Abstract

Finding rooms and facilities inside large buildings can be challenging for new students, guests, and visitors. **AR NAV** is an indoor navigation app designed specifically for the **VMES building** to provide real-time, interactive guidance. Using **Augmented Reality (AR) markers and arrows**, the app helps users navigate hallways, classrooms, and labs with ease. Unlike traditional navigation methods that rely on Wi-Fi or Bluetooth, AR NAV leverages **smartphone cameras and sensors**, eliminating the need for extra hardware.

The app features a **quick search** function for locating rooms, labs, and facilities, ensuring **faster and more convenient navigation**. By enhancing the user experience, AR NAV saves time and reduces confusion, especially for high school students, new students and visitors. However, the project **does not include outdoor navigation, multi-building support, or advanced AI features**. The goal is to create a **simple, efficient, and user-friendly AR-based indoor navigation solution** tailored for the VMES building.

Table Of Contents

Table Of Figures

Table Of Tables

Chapter 1: Introduction

1.1 Problem Statement

Navigating large indoor spaces like the VMES building can be challenging for visitors, new students, and high school students. Traditional methods, such as paper maps, posters or static signs, are often confusing and do not provide real-time directions. This leads to wasted time, frustration, and difficulty finding rooms, labs, or facilities. AR NAV solves this problem by offering an easy-to-use indoor navigation app that uses augmented reality (AR) to guide users with arrows and markers.

Figure 1: Difference between VR, MR, AR

VR (Virtual Reality) - User himself is portrayed as a virtual character in the game world and can interact with various virtual elements and NPCs. All interaction happens in system virtual environment and controlled by the user.

MR (Mixed Reality) - User can see the real world and virtual elements / characters simultaneously. Sometimes, virtual object can interact with user and other physical objects.

AR (Augmented Reality) - User can see the real world and virtual elements / characters simultaneously through smartphones and tablets. Limited Interactions with the virtual.

We use AR technology since there are financial and other limitations with VR / MR

1.2 Scope of the project

Table 1: AR NAV Project Scope Details

What is Included	What is not Included
Keyword search with floor instructions	Automatic elevator control
AR guidance with arrows and markers	Navigation for other buildings
Quick search for rooms, labs	Outdoor navigation support
Real-time updates for directions	Multi-language support
Basic user interface for easy use	User accounts or history tracking
VMES building only especially focuses on 1st, 2nd ,3rd, 4th ,6th ,10th floor	Every floor of VMES building
Offline functionality	Detailed room descriptions

AR NAV is designed to improve indoor navigation using augmented reality. The system helps users find their way around the VMES building by providing clear, real-time directions with AR guidance.

1.2.1 Features Included in AR NAV

- **Keyword Search with Floor Instructions**
  - ➢ Users can search for specific rooms or labs and get clear information about which floor that room or lab is located.
- **AR Guidance with Arrows and Markers**
  - ➢ Augmented reality overlays provide step-by-step navigation, making it easier to follow directions with arrows and markers.
- **Quick Search for Rooms and Labs**
  - ➢ Users can quickly find important locations without manually searching through maps, posters and direction signs.
- **Real-Time Updates for Directions**
  - ➢ The system continuously updates the user’s position to keep navigation accurate and responsive.

- **Simple and Easy-to-Use Interface**
  - The design is straightforward, making it accessible to all users without requiring prior experience.
- **Focus on Specific Floors**
  - Navigation is available for all floors of the VMES building primarily focusing on **1st, 2nd, 3rd, 4th, 6th, and 10th**.
  - Total Locations – users can navigate total 37 locations.
  - In 1st floor, can navigate 9 locations.
  - In 2nd floor, can navigate 5 locations.
  - In 3rd floor, can navigate 6 locations.
  - In 4th floor, can navigate 7 locations.
  - In 6th floor, can navigate 5 locations.
  - In 10th floor, can navigate 5 locations.
- **Offline Functionality**
  - Users can still access basic navigation even without an internet connection.

1.2.2 What is Not Included in AR NAV

Some features were not included in this version of AR NAV due to time constraints and project limitations:

- **Automatic Elevator Control** – The system does not integrate with or control elevators. (i.e. The system doesn't know which floor the user is at unless the user scans the QR code)
- **Navigation for Other Buildings** – Limited to the VMES building only.
- **Outdoor Navigation** – Designed strictly for indoor use.
- **Multi-Language Support** – Currently available in only English language.
- **User Accounts or History Tracking** – No login or saved navigation history.
- **Navigation for Every Floor** – Only specific floors are supported.
- **Detailed Room Descriptions** – Provides directions but not in-depth information about rooms.

1.3 Limitations

- **Dependence on AR-Compatible Devices** – Users need AR-supported smartphones or tablets, limiting accessibility.
- **Accuracy Challenges in Large Indoor Spaces** – AR tracking may struggle with signal loss, poor lighting, or featureless environments.
- **User Learning Curve** – Some users may find AR-based navigation unfamiliar or difficult to use.
- **Navigation Errors** – Occlusions (e.g., furniture, crowds) or moving objects may disrupt AR tracking and route accuracy.
- **AR Visuals Accuracy** – The AR Arrows will sometimes go through the walls and user can see behind the walls.
- **No Admin Page** – User can't update the destination's information without informing the developer.
- **Start Point Limit** – User has to start at the elevator point which is the main start point.

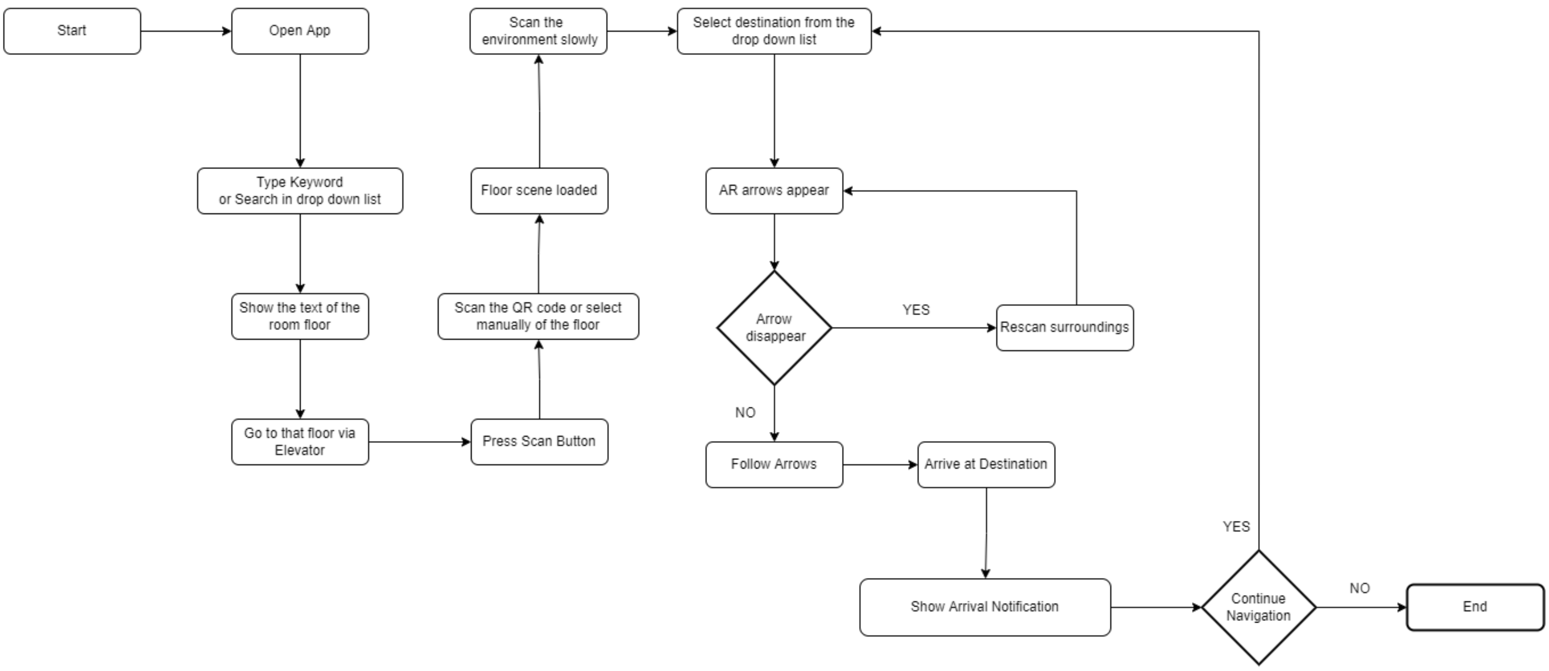

Figure 2: Work Flow Diagram of AR NAV

Chapter 2: Related Work

2.1 Existing Indoor Navigation Systems

1. Google Maps (Indoor Features):
   - Provides basic indoor maps for airports, malls and large venues.
   - **Limitations:** Requires pre-mapped locations, lacks AR guidance, and does not support real-time updates.

2. Apple Indoor Maps:
   - Offers detailed indoor layouts for partner locations (e.g. Shopping centers).
   - **Limitations:** Limited to specific partnered buildings and no AR features.

3. Indoor Atlas:
   - Uses geomagnetic fields and smartphone sensors for navigation.
   - Strengths: Works without Wi-Fi or beacons.
   - **Limitations:** Complex setup and lower accuracy in crowded areas.

2.2 AR-Based Solutions

1. Google Live View:
   - Uses AR arrows overlaid on streets for outdoor navigation.
   - Strengths: Intuitive AR interface.
   - Limitations: Not designed for indoor use.

2. ARCore/ARKit Apps:
   - Some apps use smartphone cameras and sensors (like SLAM (Simultaneous Localization and Mapping)) for indoor navigation.
   - Strengths: No need for Bluetooth beacons or Wi-Fi.
   - Limitations: Often require pre-scanned 3D maps, which are time-consuming to create.

2.3 How AR NAV Differs?

Most existing solutions either:

- Require expensive hardware (Bluetooth beacons, Wi-Fi routers).
- Lack of real-time updates or AR features.
- Are limited to outdoor use or specific partnered locations. Examples: Apps like Google Live View show AR arrows on streets but can’t guide us inside buildings (e.g. classrooms or labs).

AR NAV’s Approach:

- Utilizes AR markers and arrows for real-time guidance.
- Uses smartphone cameras and Lidar sensor (no extra hardware).
- Focus on real-time AR guidance for the VMES building.
- Provides a simple, cost-effective solution for first-time visitors.
- It makes room searching easier for new students and guests.

Chapter 3: Proposed Methodology

3.1 Methodology

We found a solution that will guide users with visuals elements using augmented reality technology. We use Vuforia Engine which is a software development kit (SDK) for creating AR apps. With the SDK, we can add advanced computer vision functionality to our application, allowing it to recognize images, objects and spaces with intuitive options to configure our app to interact with the real world. We use Vuforia with Unity platform to develop UI interaction and logical integration for the application.

3.1.1 Tools we use summary

Unity – A game engine used to develop and integrate AR navigation with 3D elements.

Vuforia – An AR SDK that helps recognize and track markers for indoor navigation.

3.2 Development Process

The project employs the waterfall methodology aligned with the Software Development Life Cycle (SDLC), processing through sequential phases:

- Research and Strategic Planning
- Development Environment Configuration
- Core Navigation System Development
- User Interface (UI) Design and Interaction Optimization

3.2.1 Research and Strategic Planning

Before starting development, we performed in-depth research to find the best way to build an augmented reality (AR) navigation system. This included:

- Investigating different AR tools and frameworks to assess their strengths and limitations.
- Reviewing existing AR navigation systems to pinpoint gaps and opportunities for improvement.
- Gathering user feedback to design a simple, easy-to-use interface and navigation process.

Using these insights, we developed a detailed project plan with clear goals, chosen technologies, and step-by-step strategies to ensure the system met user needs effectively.

3.2.2 Development Environment Configuration

3.2.2.1 Setting Up the Unity Project Environment

1. Install Unity & Required Packages

- Download Unity Hub from Unity's official website.
- Install Unity Editor (LTS version recommended for stability).
- Open Unity Hub and create a new 3D project (URP is optional for better graphics).

2. Install AR & Navigation Packages

- Open Unity and go to **Window** → **Package Manager**.
- Install the following packages:
  - Vuforia Engine AR (For AR tracking & device support)
  - AI Navigation (For AI pathfinding)

  If you want other packages to try something new, you can download the compatible packages as well.

3. Set Up AR Camera

- Create a camera, rename it "AR camera" and add Vuforia Behavior (Script) Component in Unity hierarchy

3.2.2.2 How to Get a Vuforia License Key

1. Create a Vuforia Developer Account

- Go to [Vuforia Developer Portal]
- Sign up or log in.

2. Generate a License Key

- Go to **Develop → License Manager**.
- Click **Get Development Key**.
- Enter the project name (e.g., **AR NAV or any name**).
- Agree to terms and click **Confirm**.
- Copy the **License Key** shown on the screen.

3. Integrate Vuforia into Unity

- Open **Unity** and go to **Window → Package Manager**.
- Install **Vuforia Engine** (From the Unity Asset Store or Vuforia's website). If you have already done that in installing packages, not necessary to do again.
- Go to **AR camera→ Inspector → Vuforia Behavior → Open Vuforia Engine Configuration** .
- Paste your **License Key** in the **App License Key** field.

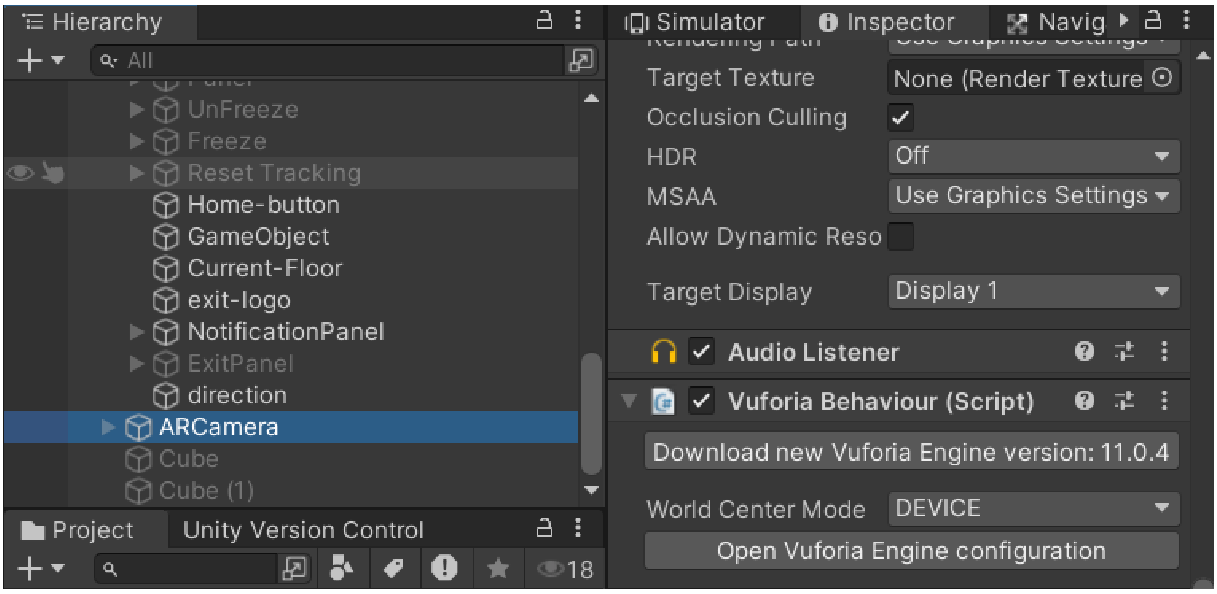

Figure 3: Unity Developer Environment Setup

After Vuforia is successfully added to Unity, we can start importing the assets to Unity.

3.2.2.3 How do we Input Real World Environment into our system?

Figure 4: Scanning Environment with Vuforia Creator App

We use Vuforia Creator App. It is a mobile tool that allows you to scan, create, and generate Area Targets using your smartphone or tablet. This is useful for indoor navigation, AR experiences, and location-based augmented reality

3.2.2.4 What is an Area Target?

An Area Target in Vuforia is a 3D spatial map of an environment that allows AR applications to track and overlay digital content in real-world spaces. This is often used in indoor navigation, museums, industrial sites, and large rooms.

3.2.2.5 Steps to Get an Area Target Using Vuforia Creator App

- Install the Vuforia Creator App
  1. Download and install the Vuforia Creator app from the Google Play Store or Apple App Store. Now, it is only available at Apple App Store.
  2. Ensure your device meets the camera and AR hardware requirements for scanning.
     - Android phone / tablet with Lidar sensor (i.e. Samsung)
     - iPhone / iPad every version pro (above version 12 pro) with Lidar sensor
- Create a New Scan
  1. Open the Vuforia Creator App on your device.
  2. Tap "Capture Area" to start capturing an Area Target.

3. Move your device slowly around the environment, making sure to scan walls, objects, floors, and ceilings to generate a detailed map.

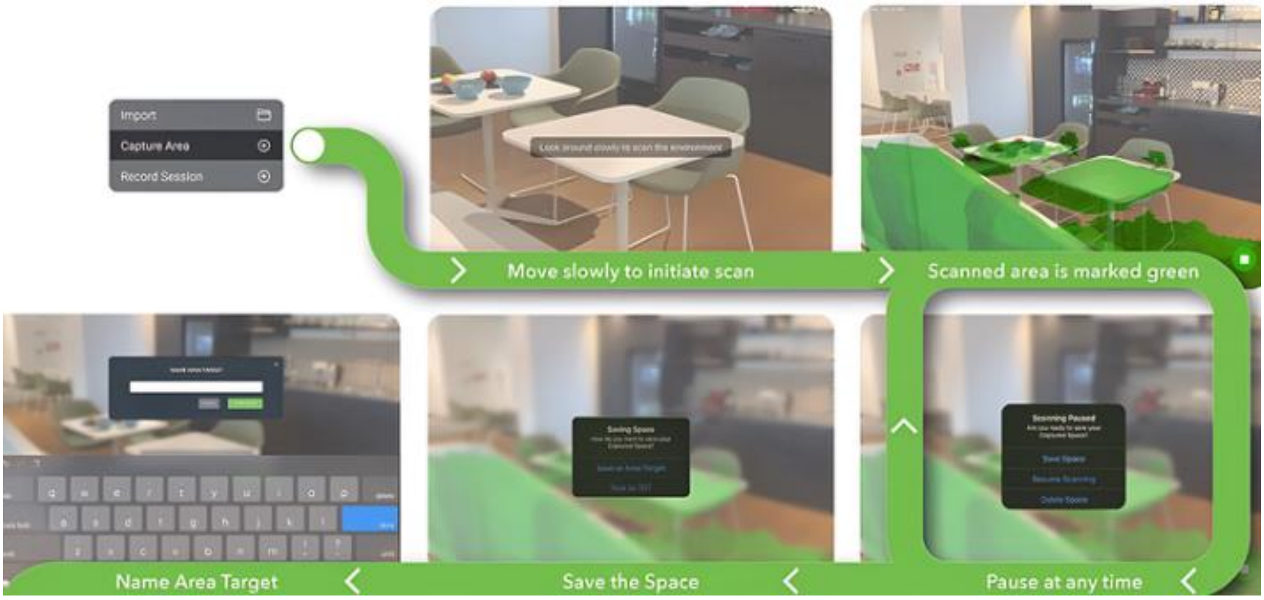

Figure 5: Vuforia Creator Scanning and Saving Area Target Process

- Tips for a Good Scan:
  1. Walk steadily and ensure good lighting conditions.
  2. Cover as much area as possible, including edges, corners, and key objects.
  3. Avoid fast movements and reflections, as this may cause tracking errors.
- Process and Upload the Scan
  1. After scanning, tap "Finish" to process the Area Target.
  2. The app will generate a 3D map (Area Target dataset).
  3. We import the Area Target data set directly into Unity's assets.

The Area target is ready to be used as a default base environment ground. Sometimes, the scanned area targets come separately. In that case, we must connect them manually.

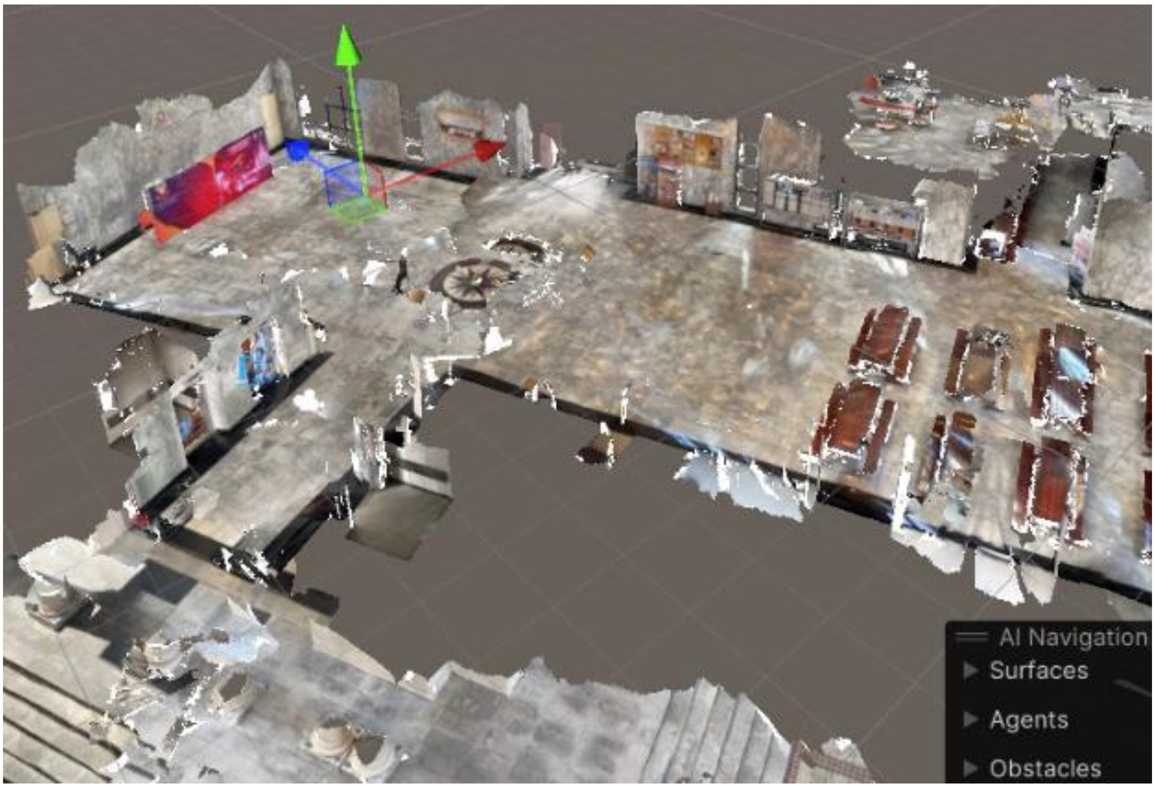

Figure 6: 1st Floor Area Target Connected Manually After Importing to Unity

We import 1st, 2nd, 3rd, 4th, 6th and 10th floors area targets and then connect them if they come separately to implement navigation system later as well. In figure 6, it is the complete area target map of the 1st floor after we carefully assemble all the area targets that we scan separately. That's the environment set up done.

3.2.3 Core Navigation system Development

The first step is to define a walkable space in the Unity's Virtual Environment for the user. We will use NavMesh for that.

3.2.3.1 What is NavMesh ?

NavMesh specifies navigable areas in the environment, including areas where characters can walk, as well as obstacles. As you can see in Figure 7 (right), the green cylinder is the user / camera which represents the user's phone in the real world. As the user walks in the real world, the system also needs to update the user's location in virtual environment. That's why we need to define a walkable area using NavMesh. NavMesh is a part of Unity's AI Navigation Plug in.

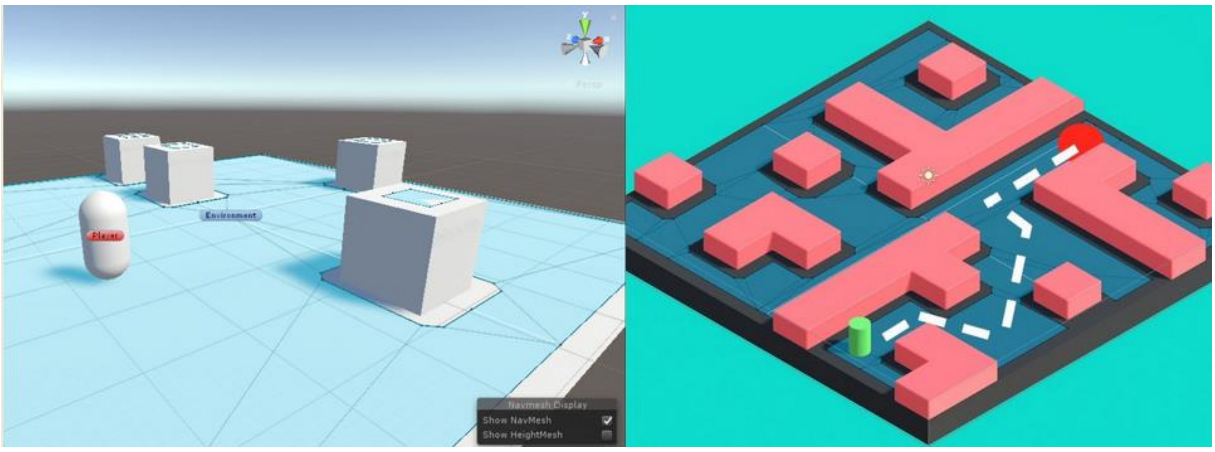

Figure 7: Example of NavMesh in Unity (left) / AI Navigation Path Calculation (right)

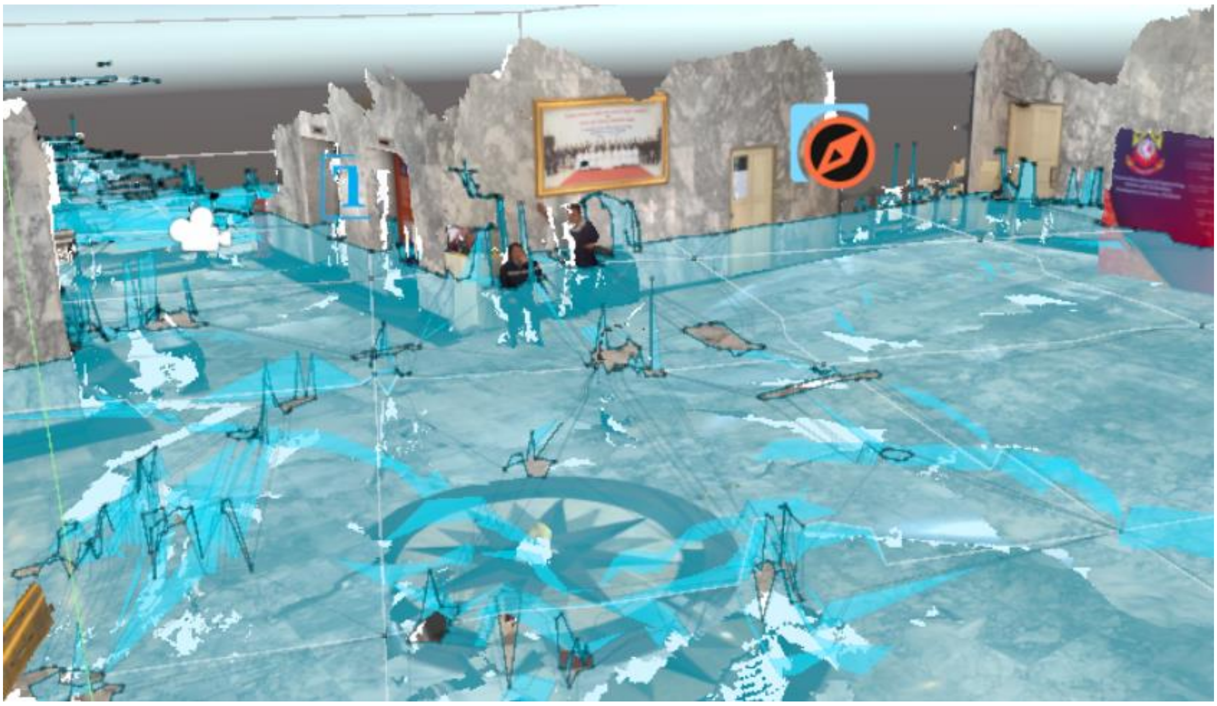

Figure 8: NavMesh Area in 1st Floor Area Target

In Figure 8, the blue area is **NavMesh** where the user can walk, and navigation takes place. NavMesh has some rigged areas and spikes due to the quality of Area Target. Poor Area Target of the 1st Floor is the result of a difficult scanning process as the area has similar walls and reflective floors.

3.2.3.2 Adding Virtual Elements To The Environment

Figure 9: Virtual Elements in 2nd Floor Area Target Environment

We add virtual elements to visualize the environment for the user. Markers are added for elevator areas. Room numbers are on the room walls and Floor number can be seen near the elevator. This makes the user interface decorated with details that the user can process.

3.2.3.3 Path Visualization

NavMesh and Unity's AI Navigation Plug in calculate the shortest path between user's location and the selected destination (e.g. MG Room in figure 10 (right)). The calculated path is visualized with 3D Arrow overlay on the screen for the user.

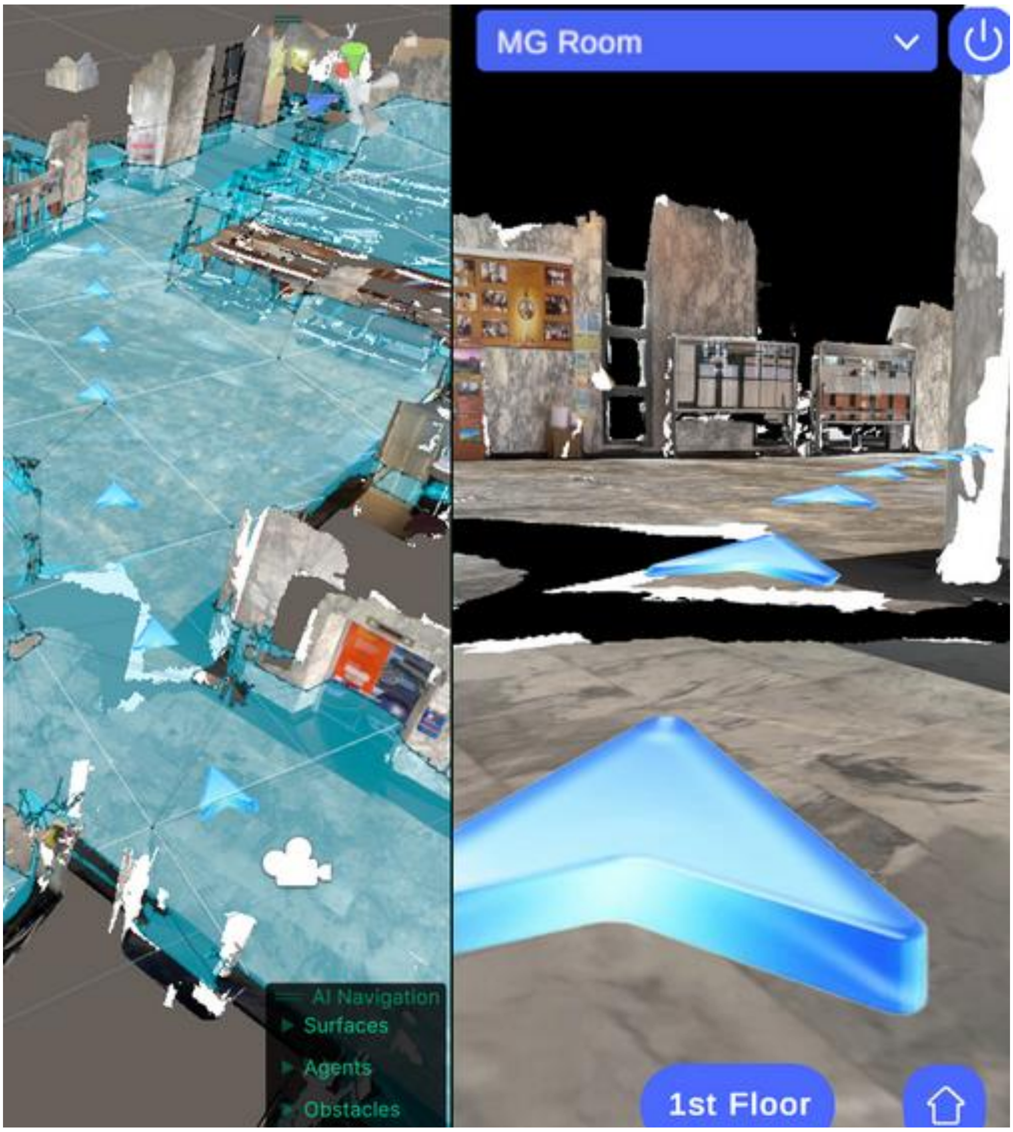

Figure 10: Navigating the Path with Arrows in Unity Simulation

Developer Simulation in Unity POV (left) / User POV (right)

3.2.3.4 ROI analysis

1. **Enhanced User Experience** – Provides seamless indoor navigation, reducing confusion and stress.
2. **Increased Efficiency** – Saves time for users and improves productivity by reducing reliance on physical signage.
3. **Competitive Advantage** – Positions businesses as innovative and enhances brand engagement.
4. **Environmental Benefits** – Reduces paper waste and maintenance costs for static signs.
5. **Data-Driven Insights** – Provides analytics on user movement, helping optimize space and navigation.

These benefits make AR NAV a **valuable investment** beyond financial returns.

3.2.4 User Interface (UI) Design and Interaction Optimization

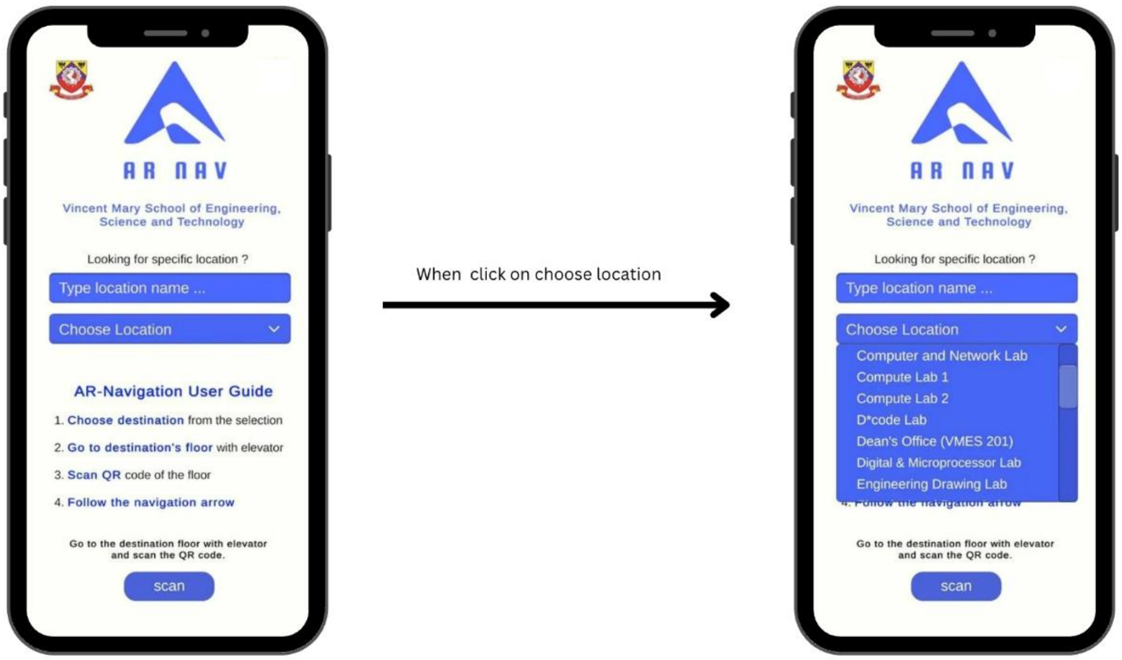

Figure 11: UI Image 1

When the user starts the application, the home page displays easy to read user guide on how to use the app. User can choose the destination from the selection drop down menu.

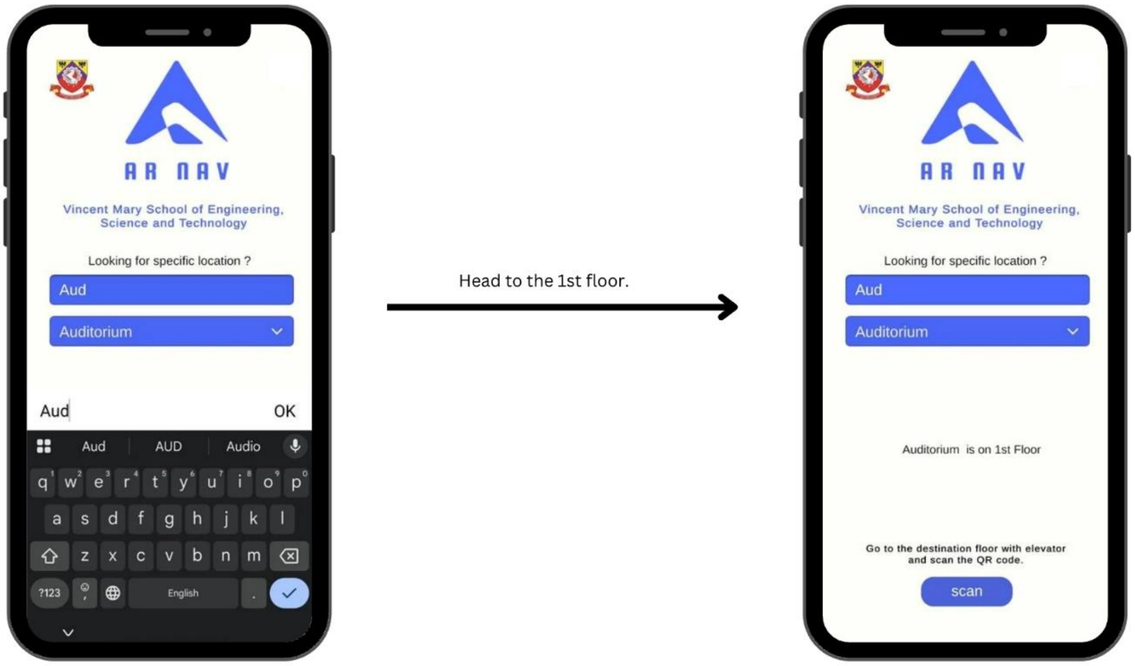

Figure 12: UI Image 2

The process is simple. The user chooses a destination that they want to go to via typing or selecting from drop down. The app display which floors the user's selected destination is located. In Figure 12, it shows "Auditorium is on 1^{st} Floor". If the destination is located on another floor (e.g. 2^{nd} floor or 6^{th} floor), the user must go with elevator by himself.

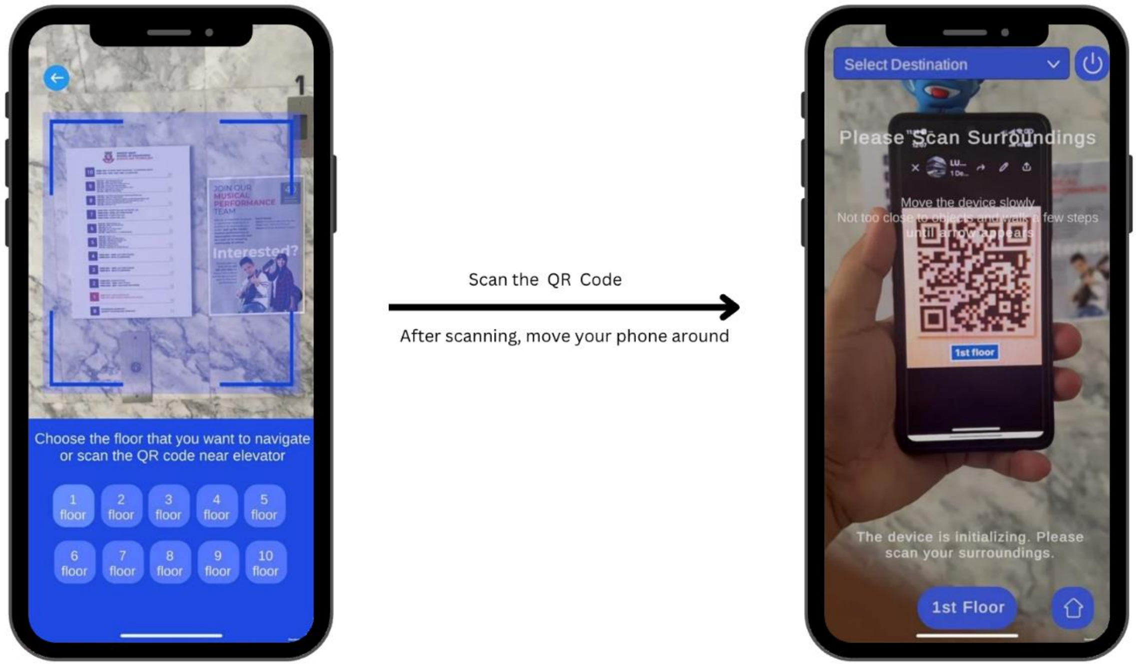

Figure 13: UI Image 3

When the user arrives at the destination's floor, the user can scan the QR code or choose from the given keypad to select which floor he is in. After scanning the QR, the floor scene is loaded, and user is given instructions to scan the environment around.

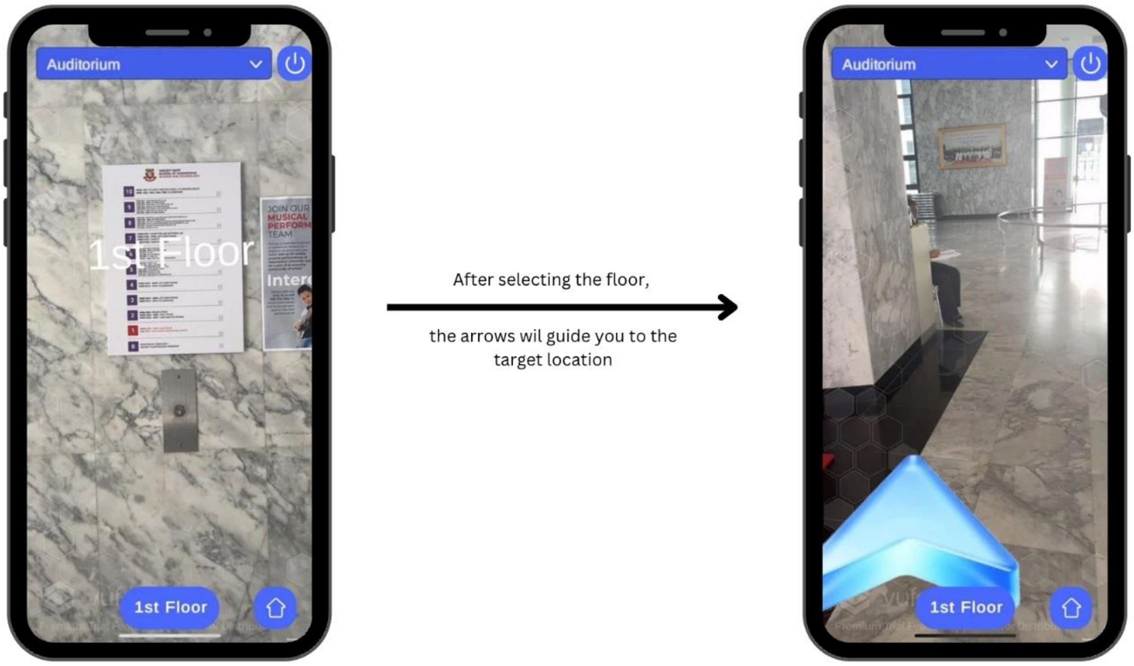

Figure 14: UI Image 4

After scanning the environment around, users can choose the destination from the drop-down menu and the arrow will guide the user to the destination.

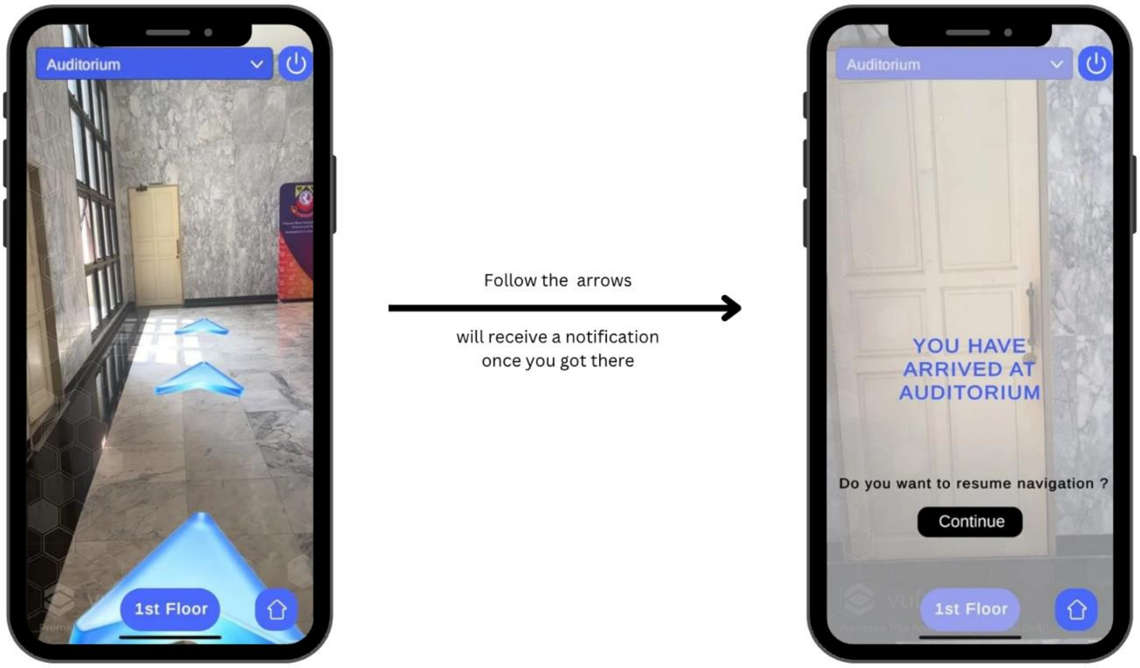

Figure 15: UI Image 5

The user follows the arrow path and finally when the user arrives at the doorstep of the destination, the arrival notification will show, and it will give the user an option to continue the navigation or exit the app using the exit button on the top right.

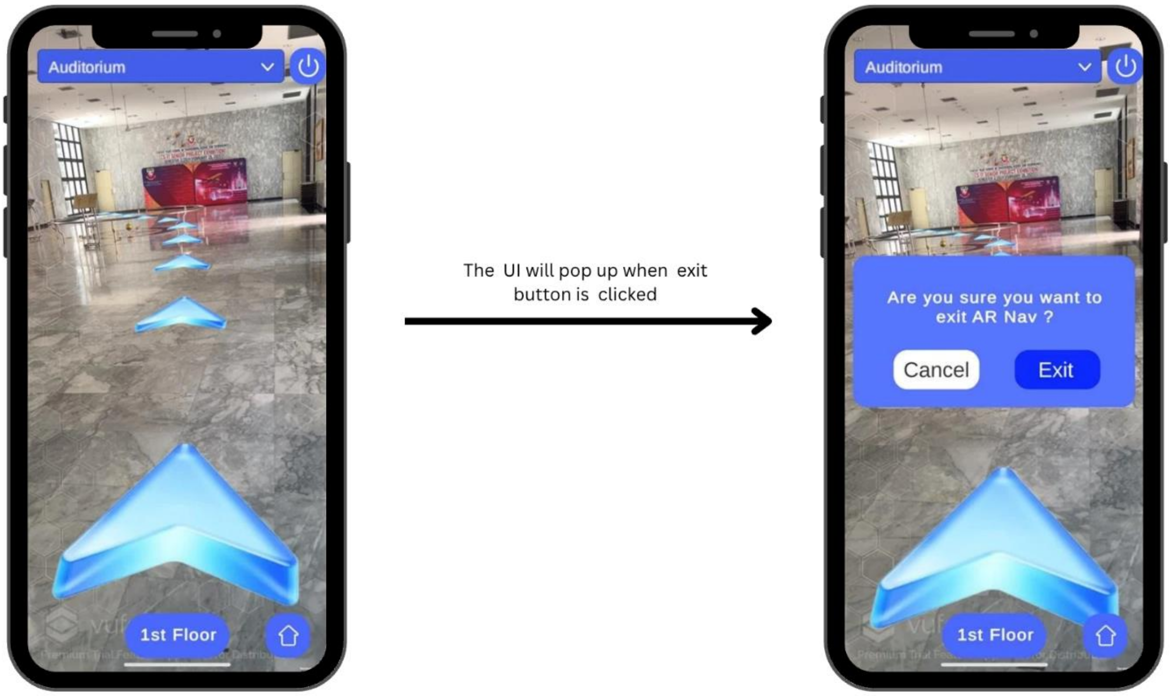

Figure 16: UI Image 6

If you click the exit button on the top right, Exit UI will pop up.

Chapter 4: Design of The System (or Work)

4.1 Functional Requirements Specification

4.1.1 Stakeholders

- **Primary Users**: Students, Faculty, Visitors (who use the AR navigation system)
- **Developers**: The project team responsible for building and maintaining the system

4.1.2 Actors and Goals

- **User (Student/Visitor)** – Uses the system for navigation.
- **System (AR Navigation)** – Processes data and provides navigation guidance.

4.1.3 Use Cases

UC-01: User search Room's Location

- Actor: Highschool students / Guests / Freshmen / Anyone who uses the app
- Precondition: Users open the app Home page
- Postcondition: User knows which floor the room is located on.
- Action: User looks up the room via typing or searching in dropdown and the app tells the floor of the room.

UC-02: Users want to know which way to go to selected room

- Actor: Highschool students / Guests / Freshmen / Anyone who use the app
- Precondition: User is already located at the floor of the room's location
- Postcondition: User arrives at the door of the room
- Action: User selects the destination's location in the dropdown menu and scans the environment around to get started. The navigation arrow (3D arrow) will appear after the area is recognized. Unless the area is recognized due to the poor scanning quality of the phone, this may take long.

4.1.4 Functional Requirements

1. AR Guidance

- The app shall display **AR arrows and markers** over the camera view to guide users.
- The app shall update arrows **in real time** as the user moves.
- If arrows disappear (due to poor tracking), the user is encouraged to **rescan the surroundings** when navigation process start.

2. Search Functionality

- Users shall be able to **type keywords** (e.g., “True Lab”) to find locations.
- The app shall display a **text response** (e.g., “True Lab is on the 6th floor”).
- Users shall see a **dropdown list** of rooms/labs on the selected floor.

3. Floor Navigation

- The app shall instruct users to **go to the correct floor** (e.g., via elevator).
- Users shall **scan a QR code** near the elevator **or** manually select the floor in the app.

4. Environment Scanning

- After reaching the floor, the app shall prompt the user to **scan the environment slowly** by moving the device.
- Scanning shall initialize AR tracking for accurate navigation.

5. Real-Time Navigation

- The app shall **recalculate the path instantly** if the user takes a wrong turn.
- Arrows should adjust based on the user’s **live position** (using phone sensors).

6. Room Labels

- The app shall display **static labels** for rooms (e.g., “True Lab”).
- Labels shall **not include detailed descriptions** (e.g., room purpose or class info).

7. Notifications

- The app shall show a **“You have arrived” notification** at the destination.
- Users shall click a **“Continue” button** to either:
  - Restart navigation (go to home page).
  - Select a new destination on the same floor.

8. Error Handling

- If AR tracking fails, the user must **re-scan the environment** until success.
- The app shall **not crash** if the camera is blocked or sensors malfunction.

4.2 System Design (consider items that applied)

4.2.1 Process / Workflow / Sequence Diagram

- User searches the desired Room's Floor via Home Page
- User goes to the room's floor with elevator
- User scans the QR code
- User selects the destination after the floor scene is loaded
- Navigation Path is generated and visualized with 3D Arrow
- When the user arrives at destination, arrival notification is displayed
- User chooses two options: continue or exit the app.

4.2.2 System Architecture

Two main functions for the system architecture –

- **QR Code Scanner Function / Floor Changing**
- **AR Navigation System**

4.2.2.1 QR Code Scanner / Floor Changing Function

The QR Code Scanner detects the floor and switches to the corresponding AR environment.

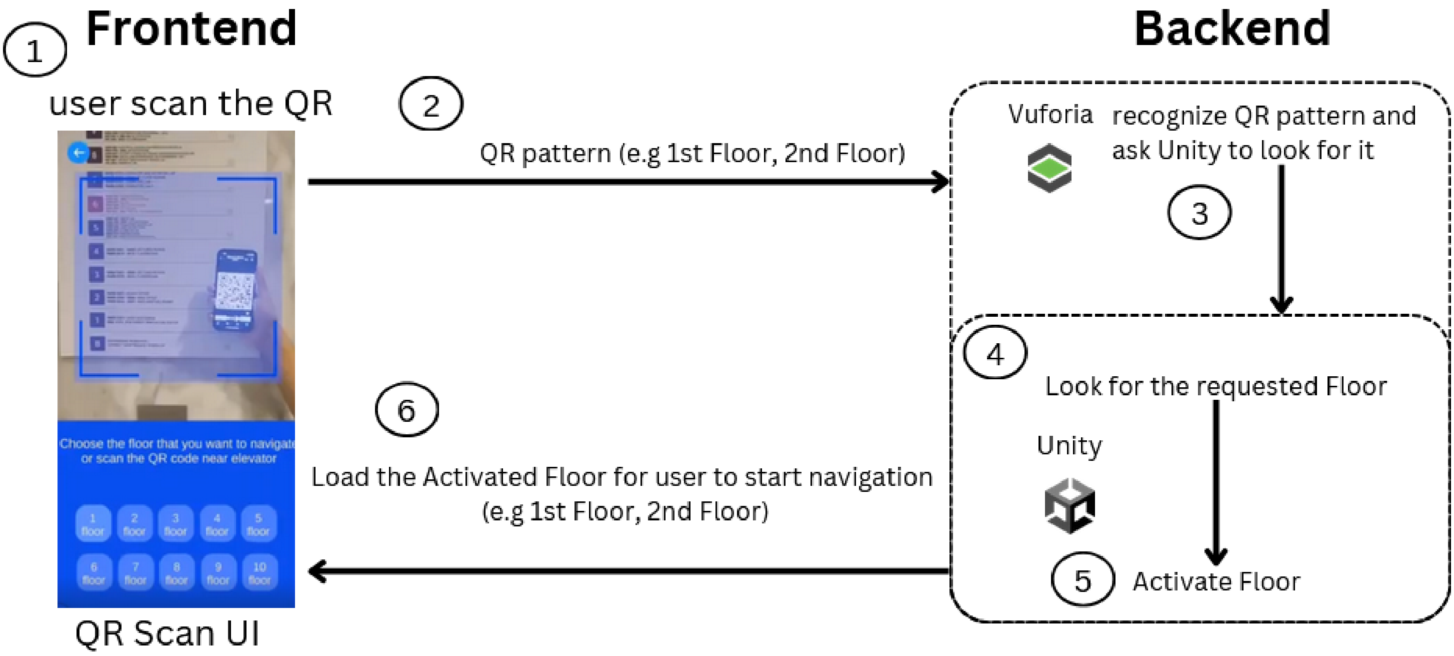

Figure 17: Data Flow and Backend Implementation of QR Scanning

Data Flow and Backend Implementation Process of QR Scanning

- User scans the QR code near the elevator
- The captured QR pattern is sent to Unity to process
- Vuforia Engine matches the captured QR with the floor QR stored in the system.
- Unity looks for the matched Floor according to QR.
- Unity activates the located floor with script
- Activated Scene is ready for users to start navigation.

4.2.2.2 AR Navigation System

After the floor scene is loaded, Vuforia tracks the real-world environment, aligns virtual elements, and Unity AI Navigation calculates the path.

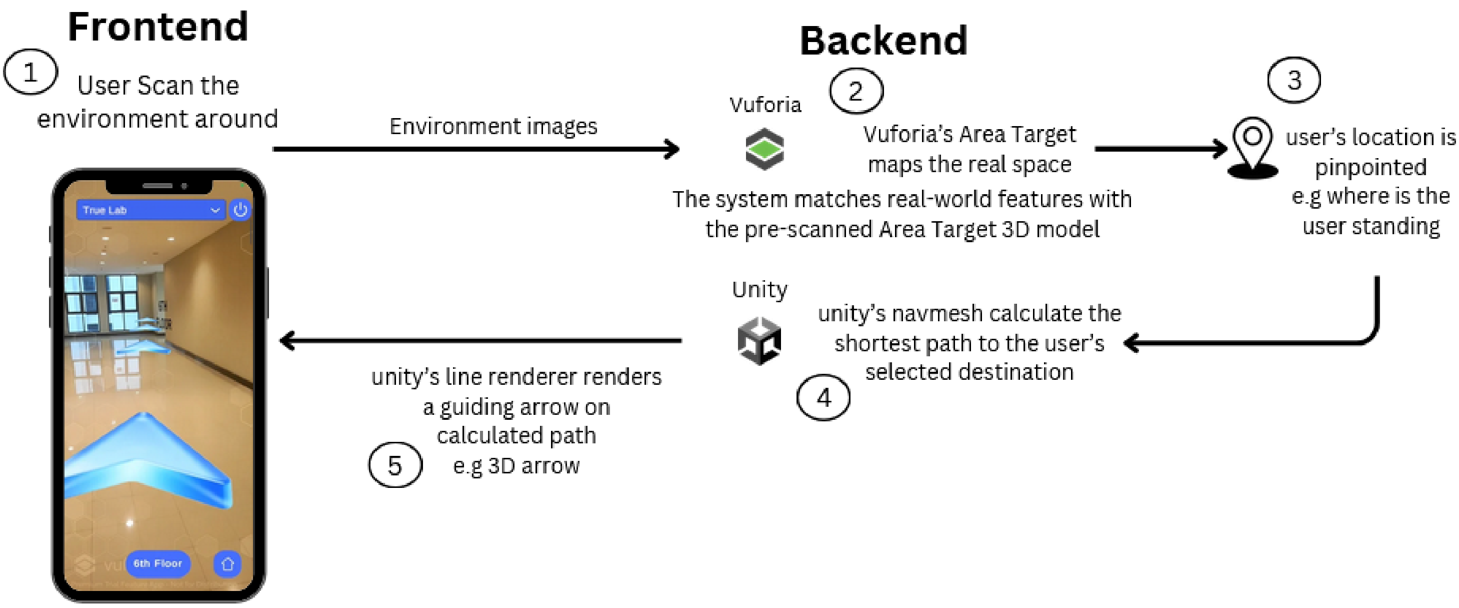

Figure 18: Data Flow and Backend Implementation of AR Navigation System

Data Flow and Backend Implementation Process of AR Navigation

- Users scan the environment around to capture images of the environment (e.g. walls, floor, doors and reflections)
- Vuforia Engine processes the captured images and mapped the real space with the pre-scanned virtual Area Target 3D model.
- User's location (e.g. where is the user standing) can be pinpointed after step 2
- Unity's AI Navigation Library calculates the path from user to the destination in the virtual world.
- Unity Line Renderer renders the guiding arrow on the calculated path.

4.2.3 Database Design

4.2.3.1 Does the System Use a Database?

No, the AR Navigation System does not require a traditional database because it operates using real-time AR tracking and AI navigation.

4.2.3.2 Why the System Does NOT Need a Database?

Unlike systems that store and retrieve data dynamically, our project:

- Uses Vuforia Area Targets to recognize environments instead of querying location data from a database.
- Relies on Unity NavMesh AI Navigation to calculate paths dynamically, eliminating the need for a stored map database.
- Predefines all floor layouts and destinations in Unity, meaning there is no need for an external data storage system.

Since the system **does not need to store or update destinations dynamically**, a database is not necessary.

4.2.3.3 When Would a Database Be Needed?

A database would be required only if:

- Locations and floors change dynamically and need to be stored externally.
- An admin panel is needed to update navigation paths remotely.
- The system tracks user history or analytics for navigation usage.

Since our system works with fixed floor layouts and predefined navigation paths, it does not require a database.

Chapter 5: Result

Figure 19: Exhibition Day Presentation Image 1

Figure 20: Exhibition Day Presentation Image 2

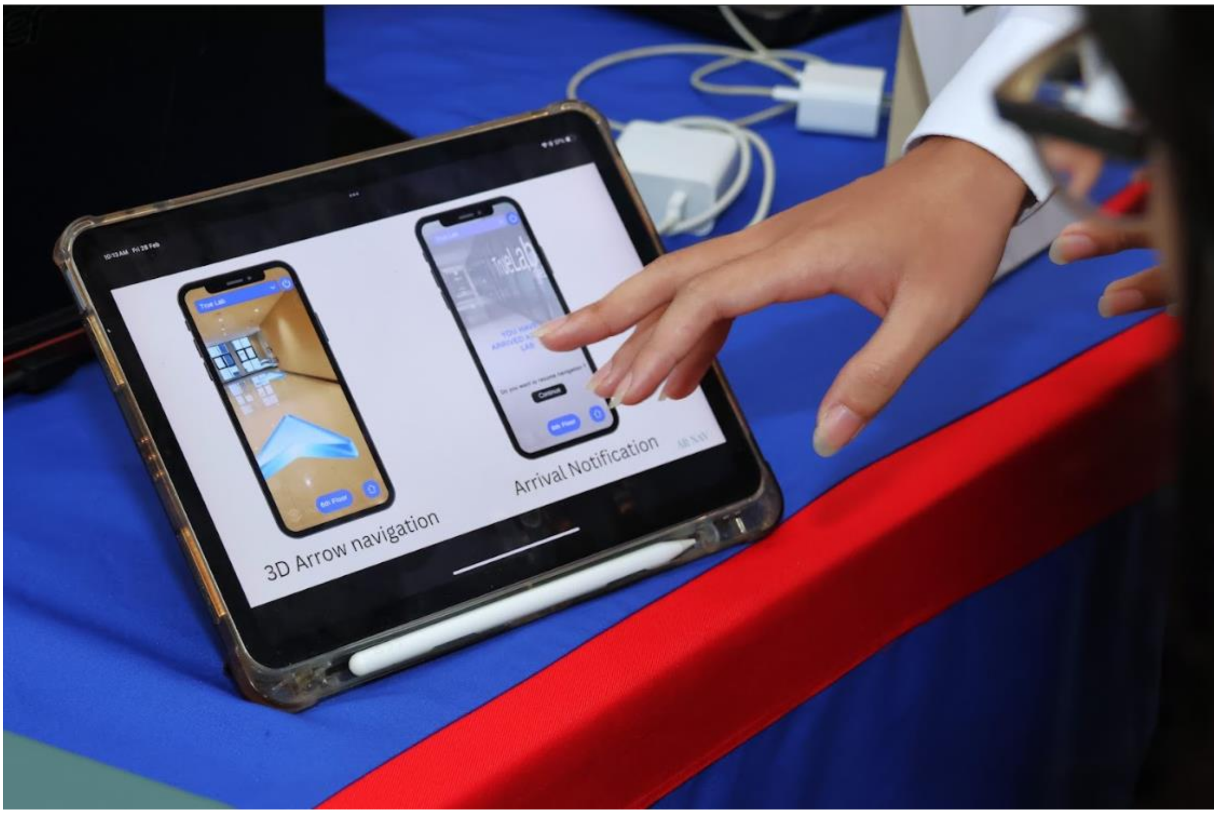

Figure 21: Exhibition Day Presentation Image 3

Chapter 6: Conclusion

In this project, we developed **AR NAV**, an advanced augmented reality navigation system designed to enhance real-time spatial awareness and guidance. We effectively addressed the difficulties of providing precise and engaging AR-based navigation, creating a smooth and intuitive experience for users.

6.1 Effectiveness of the Proposed Method

The analysis and evaluation in previous chapters confirm that **AR NAV** effectively meets its objectives. The system's **real-time mapping, intuitive interface, and adaptive navigation** contributed to a smooth and efficient experience. Key takeaways include:

- **Accurate Real-Time Navigation** – Users experienced minimal delays in location updates, ensuring precise guidance in dynamic environments.

- **Enhanced User Experience** – Feedback from users highlighted high satisfaction levels, particularly in terms of ease of use, visual clarity, and responsiveness of AR overlays.
- **Scalability and Reliability** – The system demonstrated the ability to function effectively across different locations and user densities without significant performance degradation, making it viable for large-scale deployment.

Overall, this project successfully delivered a **real-time AR navigation solution**, improving spatial awareness and enhancing user engagement.

6.2 Future Enhancements and Recommendations

While **AR NAV** has successfully achieved its core objectives, several areas for further development have been identified to enhance its capabilities:

- **Direction Indicator:**
  - ➢ When the user is going in the wrong direction, there will be a small arrow on the screen directing to the arrow path's direction.
- **Room Descriptions:**
  - ➢ When the user is navigating the path, he can see the room descriptions of the rooms that he walks past. The descriptions will include available time, room name and information.
- **The Whole Building support:**
  - ➢ In the future, we plan to make this AR navigation app support available for the whole building and possibly the whole school.

With these improvements, **AR NAV** has the potential to become a leading **AR-powered navigation tool**, offering enhanced precision, usability, and adaptability across various applications.

References

[1] AR Indoor Navigation with Vuforia Engine : Joshua Drewlow, 20 Oct 2021

[2] AR indoor navigation from scratch tutorial video : FireDragonGameStudio, 3 Aug 2024

[3] Getting Started with Vuforia Engine in Unity : Vuforia Developer portal.

[4] How to use Vuforia Creator App : Vuforia Developer portal.

[5] VR, MR, AR difference explanation: Alyssa Small, 29 May 2020

[6] What is NavMesh Tutorial : Brackeys, 15 March 2018

[7] Learn Unity Navigation using NavMesh in 10 minutes : Viju Dudiya, 9 August 2018

[8] Unity NavMesh Manual : Unity Website

[9] Setting up Unity project for Vuforia : Unity Website documents