ARFICT: Indoor Navigation Mobile Application using Augmented Reality (AR) for the Faculty of Information and Communication Technology (FICT)

By

Lee Chorng Huah

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ts. Dr. Saw Seow Hui, for providing invaluable technical support, mentorship, and guidance throughout my Final Year Project on ARFICT: Indoor Navigation Mobile Application using Augmented Reality (AR) for the Faculty of Information and Communication Technology (FICT). Your expertise and advice, particularly in navigating the technical aspects and refining my report, have been instrumental in shaping this project. A million thanks to you.

I am also deeply grateful to my family for their unwavering support, especially in helping me overcome challenges related to tool requirements and measurements during the project. Their assistance ensured that I had everything I needed to complete my work effectively. Furthermore, I sincerely thank my parents for their unconditional love and financial support, which made my entire university education possible. Their encouragement has been a constant source of motivation.

ABSTRACT

As commercial and educational institutions grow more complex, indoor navigation has become essential. Traditional GPS navigation, effective outdoors, struggles indoors due to signal interference from walls and ceilings. This is even more significant in large, multi-floor buildings such as university campuses, where GPS signals are blocked. Additionally, 2D maps are not capable of providing real-time, adaptive directions, leading to inefficiency and user frustration. Hence, the ARFICT project aims to develop an Augmented Reality (AR)-based indoor navigation system for the FICT building at UTAR, providing a real-time, interactive navigation experience that enhances wayfinding in complex indoor environments. The project's contribution goes beyond UTAR, forming a basis for AR-based navigation in various large indoor environments, such as shopping malls, airports, and hospitals. The use of external hardware, like Bluetooth beacons, is eliminated, thus offering an affordable solution for navigation in multi-floor buildings. The system employs a marker-based localisation method using QR codes decoded with the ZXing library to establish the user's initial position. Navigation paths are optimised using the A* algorithm, while ARCore integrates sensor data from the device's gyroscope and accelerometer to enhance tracking accuracy. Users are guided with AR-based path visualisation, distance indicators, and voice instructions, ensuring intuitive and efficient navigation across multiple floors. In summary, this project successfully developed an AR indoor navigation application to provide advanced indoor navigation, improve efficiency, and provide a reliable wayfinding for students, staff, and visitors in the multi-floor FICT building, enhancing the user experience in complex indoor environments.

Area of Study: Augmented Reality (AR), Mobile Application Development

Keywords: Indoor Navigation, Augmented Reality, QR Code Localisation, A* Pathfinding Algorithm, ARCore, Mobile Application

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AR	Augmented Reality
GPS	Global Positioning System
FICT	Faculty of Information and Communication Technology
CAGR	Compound Annual Growth Rate
QR	Quick Response
Wi-Fi	Wireless Fidelity
SUS	System Usability Scale
URE	User Range Error
BLE	Bluetooth Low Energy
RSSI	Received Signal Strength Indicator
AI	Artificial Intelligence
NeRF	Neural Radiance Fields
LiDAR	Light Detection and Ranging
UI	User Interface
NavMesh	Navigation Mesh
3D	Three-Dimensional
TTS	Text-to-Speech
IDE	Integrated Development Environment

CHAPTER 1

Introduction

Figure 1.1 Indoor Positioning and Indoor Navigation Market Revenue [1]

With the expansion and complexity of commercial and educational institutions, indoor navigation on smartphones has grown more and more popular. The global Indoor Positioning and Indoor Navigation Market size was valued at USD 1.28 billion in 2019 and is predicted to reach USD 30.20 billion by 2030, with a CAGR of 33.1% from 2020 to 2030 [1]. Besides, traditional GPS-based outdoor navigation systems are used on smartphones because of their high precision, which allows them to operate within a 4.9m radius under a clear sky [2]. However, GPS-based navigation cannot be used indoors because the signals from the satellites are attenuated, absorbed, and scattered by walls, roofs, and other objects, leading to errors in the system [3]. Large interior areas like office buildings, malls, and university campuses often feature intricate and multi-floor, which can be challenging to navigate [4]. Hence, there are relatively fewer commercial indoor navigation system developments than outdoor navigation systems.

Figure 1.2 Sample of AR Indoor Navigation Mobile Application [1]

Recent technological advancements, particularly Augmented Reality (AR), offer innovative solutions to these indoor navigation challenges. AR technology creates interactive and user-friendly navigational aids by overlaying digital information onto the physical world [5]. AR integrates digital content with real-world views, allowing users to receive real-time visual guidance such as arrows and markers on their mobile devices, as shown in Figure 1.2. This improves the ease and manageability of navigating complex indoor spaces. Also, AR technology uses smartphones' cameras and sensors [5] to offer seamless, context-aware navigation without additional hardware. The simplicity of using mobile devices, which are already widely owned and carried by users, provides an extra degree of accessibility and ease, enabling the seamless integration of navigation aids into daily activities.

The main goal of the ARFICT project is to develop an augmented reality (AR) mobile application for the Faculty of Information and Communication Technology (FICT) at the University Tunku Abdul Rahman (UTAR) to provide real-time, interactive indoor navigation that overcomes the limitations of traditional maps and enhances the user experience within the university’s indoor spaces.

1.1 Problem Statement and Motivation

1.1.1 Complex Building Layouts

Large buildings, especially the faculty in the university with multiple floors and numerous rooms, can present significant navigation challenges [6]. These complex layouts might be difficult to understand and navigate for those unfamiliar with the building, such as new students, faculty members, or visitors. Due to its complexity, people are more likely to get lost or arrive at their destinations late.

1.1.2 GPS Signal Limitations

GPS technology, designed for outdoor navigation, has problems with indoor navigation since it relies on satellite signals [7]. Accurate location can be challenging in structures of solid materials like steel and concrete, as these signals can be blocked entirely or considerably weakened [8]. Moreover, position data may be distorted by the multipath effect, which occurs when GPS signals bounce off floors and walls. These issues can lead to inaccuracies and unreliable guidance when navigating complex indoor environments.

1.1.3 Traditional 2D Maps

Conventional 2D indoor maps are static representations that provide a fixed view of a building's layout. These maps lack interactivity and do not provide real-time updates or adaptive guidance. These maps require users to interpret them independently, which might cause misconceptions, particularly in dynamic or complicated contexts where directions vary. Conventional maps don't alert users when they make a mistake or offer rerouting choices, which makes it difficult and confusing to discover the right route.

1.1.4 Time Wasted in Incorrect Locations

When users navigate using traditional direction boards or static 2D maps and end up in the wrong location or take a longer route, they waste valuable time trying to correct their path. This issue becomes more pronounced in busy or unfamiliar environments, where efficiency is crucial. The inefficiency of navigating with unclear or incorrect directions leads to frustration. It decreases productivity as users may have to backtrack or seek alternative routes to reach their destination in the shortest time possible.

1.2 Project Objectives

The primary aim of the project is to develop an advanced indoor navigation application for the Faculty of Information and Communication Technology (FICT) at Universiti Tunku Abdul Rahman (UTAR). The application integrates Augmented Reality (AR) technology, optimised routing, and marker-based localisation using QR codes to provide convenient and accurate navigation across complex multi-floor building environments. This project seeks to enhance the user experience by offering a more interactive, intuitive, and efficient alternative to traditional indoor navigation methods.

The sub-objectives of the project are as follows:

No.	Sub-Objectives
Sub-Objective 1	To develop a convenient indoor navigation mobile application.
Sub-Objective 2	To integrate AR technology into the navigation application
Sub-Objective 3	To optimise route calculation.
Sub-Objective 4	To localise the user's position by utilising QR codes

Table 1.2.1 List of sub-objectives

1.2.1 Sub-Objective 1

The project aims to develop convenient indoor navigation by creating a mobile application to assist users in navigating complex indoor environments in FICT, such as the multi-floor environment. The application will provide user-friendly features to simplify the navigation process, enabling individuals to find their way efficiently and accurately through interconnected and intricate building structures.

1.2.2 Sub-Objective 2

This objective focuses on integrating AR into the mobile navigation application. AR will enhance the user experience by displaying the guiding path onto the physical environment as viewed through a smartphone. This integration will provide a guiding path that helps users navigate indoor spaces more intuitively and effectively.

1.2.3 Sub-Objective 3

The project seeks to optimise route calculation by implementing the A* algorithm to determine the shortest and most efficient path to the user's destination. Additionally, the system will be able to reroute users to the nearest available location of the same type of destination, such as directing them to the closest toilet. This feature aims to minimise travel time, reduce the likelihood of users taking longer or incorrect routes, and improve overall navigation efficiency.

1.2.4 Sub-Objective 4

The project aims to localise the user's position within the complex indoor environment by utilising QR codes as recentering points. This feature will help maintain accurate positioning throughout navigation, ensuring the guiding path remains reliable and aligned with the user's real-world location.

1.3 Project Scope

This project developed an AR-based indoor navigation mobile application to help students at UTAR Kampar navigate the FICT building. The app enables users to scan QR codes for indoor localisation via barcode image processing, calculate the shortest navigation path, and provide real-time AR guidance. A mini map displays the user's current location and supports multi-floor navigation, improving orientation and overall navigation accuracy. Distance calculation labels are provided along the path to show the remaining distance to the destination in meters, while sound navigation guidance further enhances usability.

The application delivers a cost-effective solution by integrating mobile devices and augmented reality technologies to provide seamless, real-time guidance without requiring additional hardware such as Bluetooth beacons. With support for navigation across both floors of the FICT building and additional features to improve user experience, the system offers a reliable and efficient wayfinding solution for complex indoor environments.

1.4 Contributions

The ARFICT project provides an innovation to the growing demand for efficient indoor navigation systems in large, complex buildings. This project, which created an AR-based mobile application for the FICT building at UTAR, provides the foundation for more effective indoor navigation and has the potential to significantly improve the user experience in other similar environments such as malls, airports, and hospitals. Users can interact with their environment using AR, which provides them with intuitive real-time visual guidance. This innovation makes indoor navigation more accessible, efficient, and less frustrating, giving users an experience like outdoor navigation apps like Waze and Google Maps.

The project not only addresses common issues with traditional 2D maps and static guidance but also demonstrates AR's potential to change the way people navigate complex indoor spaces. The project contributes to the future of AR navigation systems by offering an interactive, immersive solution that pushes beyond the boundaries of what indoor navigation can achieve. This work is critical for providing more accessible and efficient navigation experiences for students, staff, and visitors in large-scale environments. The project's impact extends beyond UTAR, creating a way for broader adoption of AR technology in indoor navigation across multiple industries, resulting in smarter, more connected environments.

1.5 Report Organisation

This report is organised into seven chapters. Chapter 1 introduces the project background, outlining the problem statement and motivation, project objectives, scope, and contributions. Chapter 2 reviews indoor navigation methods, focusing on marker-based and marker-less approaches, and examines existing navigation systems. The advantages and limitations of these methods are discussed and summarised in a comparison table. Chapter 3 explains the project methodology and approach, including the proposed system architecture, user requirements, use case diagrams, activity diagrams, and the project timeline.

Chapter 4 presents the system design through flowcharts and block diagrams that illustrate the overall structure of the proposed application. Chapter 5 discusses the system implementation, covering the hardware and software setup, integration of components, and the challenges encountered during development and testing. Chapter 6 evaluates and discusses ARFICT, assessing its functionality, performance, and ability to achieve the intended objectives. It includes testing results, solutions to challenges, and a comparison with the previous app to highlight improvements. Finally, Chapter 7 concludes the report by summarising the project achievements and providing recommendations for future enhancements to improve navigation accuracy, coverage, and user experience.

CHAPTER 2

Literature Review

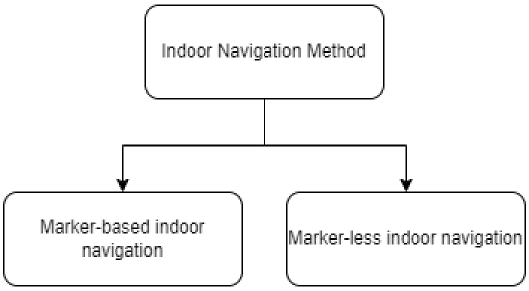


Figure 2.1 Indoor Navigation Method

This chapter will introduce indoor navigation methods, including marker-based and marker-less methods for indoor navigation. Also, this chapter will provide a review of the existing navigation system. Lastly, the advantages and limitations of the indoor navigation method and the reviewed existing navigation system are summarised in a table.

2.1. Marker-Based Indoor Navigation

A marker-based indoor navigation enhances navigation using predefined markers, such as QR codes, distinct images with visual features that are easy to extract, or even natural objects in the real environment [9] to overlay digital information onto the real world. This method ensures precise and reliable guidance, particularly in indoor environments like museums, malls, or airports where GPS is ineffective. It is a cost-effective solution that leverages simple technology with a camera-equipped device and includes AR software to create interactive user experiences.

2.1.1 QR Code

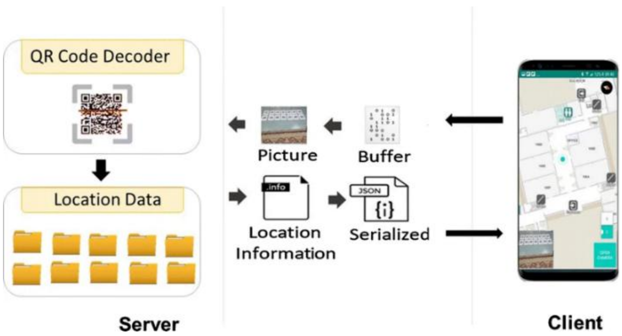


Figure 2.1.1.1 QRNav [10]

Visible markers, such as QR codes, provide a cost-effective solution for indoor navigation. Literature [10] proposed an indoor navigation system (QRNav) for the visually impaired, using QR codes to determine location information. These codes are placed at specific locations within a building and scanned by users to determine their position. Also, the 'Zxing' library was used for encoding and decoding the QR codes and worked well under low light conditions [10].

System	Average Error (Standard Deviation)	
	Route 1	Route 2
QRNav	3.3 (0.48)	5.5 (0.84)
BLE APP	4.3 (0.94)	8.7 (1.33)

Table 2.1.1.1 Average error in terms of the number of steps [10]

Also, in Literature [10], the average error in QRNav was 3.3, with a standard deviation of 0.48 in route 1 and 5.5, with a standard deviation of 0.84 in route 2, lower than the BLE beacon-based application, as shown in Table 2.1. This proved that QR codes provide more accurate and reliable navigation services.

Literature [11] proposed that this method stands out due to its affordability compared to other navigation techniques like GPS, Bluetooth, and Wi-Fi, which often involve higher infrastructure and maintenance costs. This shows that it has the advantages of accuracy and cost-effectiveness. While other methods, such as GPS, Bluetooth, and Wi-Fi, can suffer from lower accuracy and higher costs, QR codes offer a reliable and accurate navigation solution. This is particularly true for smaller spaces where QR codes can be placed strategically to ensure seamless navigation [12].

One of the main limitations of QR code-based navigation is that users must continuously find and scan new codes to update their position [13]. This process can be cumbersome and inconvenient, especially in larger or more complex environments. Additionally, the effectiveness of QR code-based systems can be impacted by external factors such as lighting conditions and weather. Proper lighting is necessary to scan codes [11] effectively.

2.1.2 AR Marker

Figure 2.1.2.1 Examples of ARToolKit marker [14]

AR markers are widely utilised in indoor navigation due to their simplicity, low cost, and ease of deployment. These markers are typically black-and-white patterns printed on paper, detectable by a camera-equipped device for virtual overlay to provide clear guidance to users. Various AR marker toolkits, such as ARToolKit [14], are available online, as shown in Figure 2.1.2.1.

Literature [15] proposed an indoor navigation system to aid visually impaired individuals using AR markers for spatial awareness. Another Literature [16] proposed a generic indoor navigation framework using ARToolKit markers that enable us to navigate buildings using computer vision techniques. The system showed a high positioning accuracy with a System Usability Scale (SUS) score of 92.0, indicating high user satisfaction with the accuracy. Both studies highlight that AR markers are inexpensive and easily deployed, making them ideal for large-scale projects [15], [16]. They offer precise localisation and support scalability, allowing new markers to be added as needed to expand the navigation network.

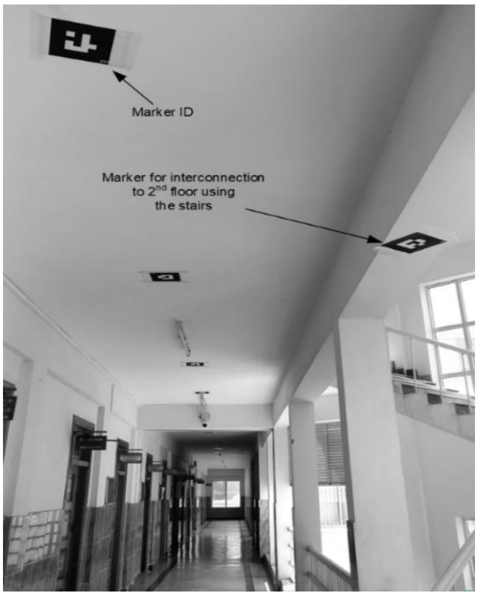
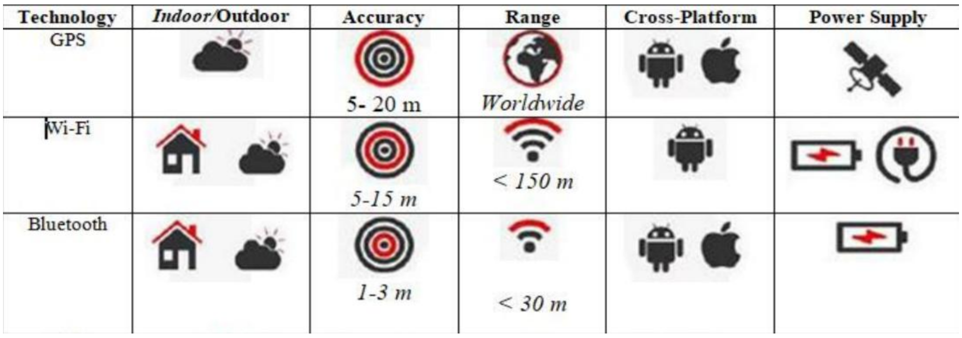


Figure 2.1.2.2 Marker deployment in the corridor of the building [16]

However, AR markers require sufficient lighting and unobstructed views to function effectively, and poor environmental conditions can lead to detection failures [15],[16]. Additionally, placing markers throughout an environment can cause visual clutter and impact aesthetics [16], as shown in Figure 2.1.2.2. The camera's range and field of view can also limit the detection of markers, potentially causing navigation errors [15]. Furthermore, AR marker systems are static and require manual updates when the environment changes, which can increase maintenance efforts.

In conclusion, while AR markers offer a cost-effective and accurate solution for indoor navigation, their use is constrained by environmental conditions, visual clutter, and the need for manual updates. Future work should focus on addressing these challenges to enhance the robustness and applicability of AR marker-based navigation systems.

2.2 Marker-Less Indoor Navigation



Technology	Indoor/Outdoor	Accuracy	Range	Cross-Platform	Power Supply
GPS		5- 20 m	Worldwide		
Wi-Fi		5-15 m	< 150 m		
Bluetooth		1-3 m	< 30 m		

Figure 2.2.1 Various Positioning Techniques in Marker-Less Indoor [17]

Marker-less navigation does not rely on physical or digital markers. Instead, it uses environmental features, sensors, and advanced algorithms to determine the user's location and guide them to maintain the relative positioning of virtual objects in the natural world [9]. Various position techniques are shown in Figure 2.2.1 [17].

2.2.1 Global Positioning System (GPS)

Marker-less navigation employs built-in sensors in devices such as smartphones and laptops to provide location information without relying on physical markers [17]. This approach utilises key technologies like GPS data, digital compass, accelerometers, and velocity meters to define the user's device location and position [18]. The basic GPS service gives users an average User Range Error (URE) of less than 7.8 meters 95 percent of the Time [19]. Thus, it can accurately determine geographical coordinates when unobstructed [8].

However, indoor navigation faces unique challenges due to the complexity of indoor environments. GPS, for instance, only works in open areas where satellite signal is consistently received to offer location positioning in outdoor environments [7]. GPS signals are significantly weakened indoors by obstructions [8] such as walls, furniture, and multipath effects, where signals bounce off surfaces, impacting the accuracy of positioning and navigation. Additionally, GPS is insufficient for determining floor levels in multi-floor buildings, making it ineffective for indoor positioning.

This overview captures the strengths and limitations of markerless navigation technologies and their application in AR while also addressing the difficulties faced by GPS in indoor environments.

2.2.2 Wireless Fidelity (Wi-Fi)

Wi-Fi offers a practical alternative to GPS for indoor navigation, benefiting from its widespread availability and easy integration with existing infrastructure. Many buildings already have Wi-Fi access points installed, such as those used for cash registers, public hotspots, and various commercial applications. Thus, Wi-Fi-based navigation has low hardware installation costs and is highly available. For navigation purposes, users do not need to connect directly to the Wi-Fi network by enabling Wi-Fi on their devices.

Wi-Fi-based indoor positioning generally provides accuracy within 5 to 15 meters [7]. Enhanced accuracy can be achieved through additional smartphone sensors, which help determine precise location and floor level [17].

While Wi-Fi-based navigation is advantageous in complex indoor environments like museums, shopping malls, railway stations, airports, hospitals, and office buildings, it also comes with challenges. The fingerprinting technique for Wi-Fi positioning involves collecting and mapping Wi-Fi signals to known locations, which can be time-consuming and require significant setup. The accuracy is influenced by the number of available Wi-Fi networks, signal reflections from walls and corridors, and obstructions like ceilings and the user's body. Also, the obstacles within buildings can lead to multipath effects, where signals reflecting off surfaces can degrade location accuracy [20].

Another consideration is the power consumption associated with Wi-Fi-based positioning systems. Despite the low cost of implementing Wi-Fi infrastructure in buildings, the power consumption for operating Wi-Fi-based navigation systems can be high [7]. This factor may impact the overall efficiency and sustainability of Wi-Fi-based navigation solutions, particularly in environments where energy efficiency is a concern.

Overall, while Wi-Fi provides a cost-effective and feasible solution for indoor navigation, especially where GPS is impractical, its implementation is accompanied by accuracy and power consumption challenges.

2.2.3 Bluetooth Low Energy (BLE) Beacon

Figure 2.2.3.1 Sample Product of a BLE Beacon

Bluetooth Low Energy (BLE) beacons are small, as shown in Figure 2.2.3.1, battery-powered devices used for indoor navigation and are easy to set up [21]. Their compact size and affordability make them ideal for widespread deployment in malls, airports, and offices. Operating on battery power, BLE beacons are more energy-efficient than other wireless signals [22] but require periodic battery replacement or recharging.

BLE beacons offer location tracking accuracy similar to Wi-Fi systems using radio frequency signals and Received Signal Strength Indicator (RSSI) fingerprinting. This allows for precise indoor positioning within the beacon's range. However, achieving high accuracy involves complex signal processing [7], and BLE systems can suffer from multipath effects where signals reflect off surfaces, impacting accuracy [21].

Despite these challenges, BLE beacons provide a practical solution for indoor navigation due to their low cost and effective performance. However, careful power consumption and signal processing management are necessary for optimal results.

2.3 Review of Existing Navigation Applications

2.3.1 Google Maps

Google Maps is a mapping and navigation application for desktop and mobile devices from Google. Maps provide turn-by-turn directions to a destination, along with 2D and 3D satellite views and public transit information. Maps also offers "Street Views," photos of the streets and surroundings. Google Maps for mobile offers location services for motorists that use the Global Positioning System (GPS) location of an iOS or Android mobile device, if available, along with data from wireless and cellular networks.

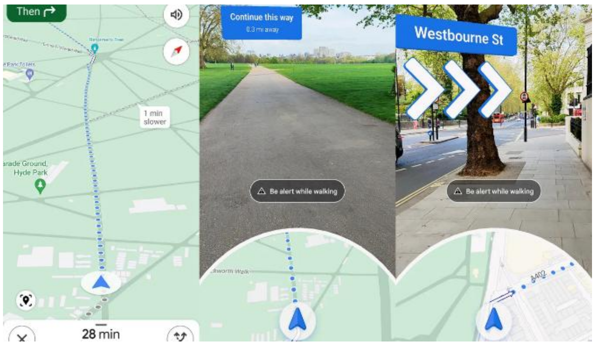


Figure 2.3.1.1 Google Map Live View AR Navigation [23]

Google Maps Live View integrates AR [23] to enhance pedestrian navigation by overlaying digital directions onto the real-world environment viewed through a smartphone camera, as shown in Figure 2.3.1.1. This feature utilises AI and computer vision to combine billions of Street View and aerial images into detailed, 3D digital models.

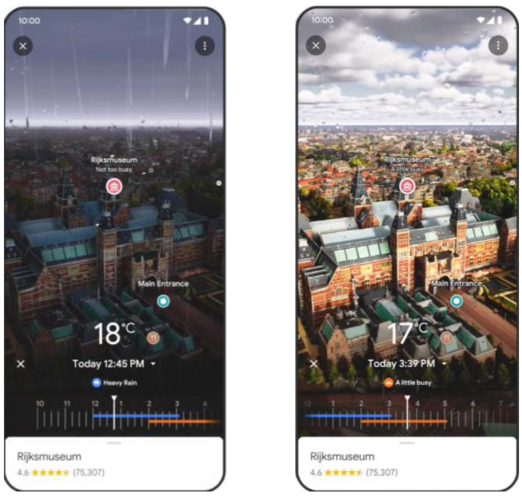


Figure 2.3.1.2 Google Map Real-time Weather [24]

Google Maps uses advanced AI techniques, such as neural radiance fields (NeRF) [24], to create realistic 3D representations of locations, capturing details like lighting and textures. NeRF is a recent innovation that uses AI algorithms to create 3D objects from 2D images [25]. Users can explore these environments, check real-time weather, traffic, and crowd density, and view inside establishments. This technology offers a more immersive navigation experience, but accuracy depends on Google's Street View data.

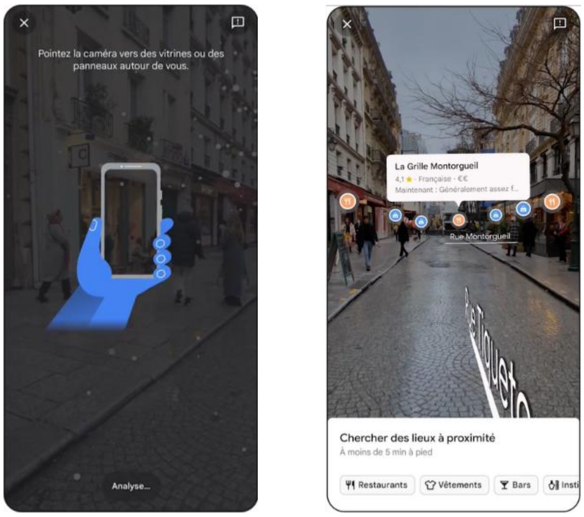


Figure 2.3.1.3 Google Map Search with AR Live View Feature [24]

Google Maps also provides the Search with Live View feature, as shown in Figure 2.3.1.3, which uses AI and augmented reality to help users find nearby amenities and view real-time information. However, it is currently available in select cities only, such as London, Los Angeles, New York, Paris, San Francisco, and Tokyo, with plans to expand to other places [24].

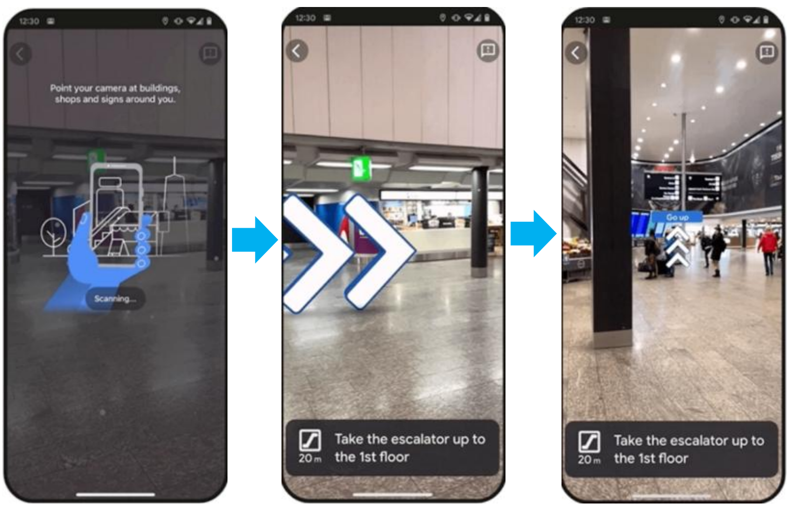


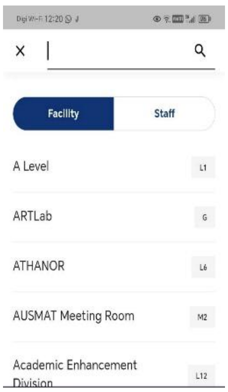
Figure 2.3.1.4 Google Map AR Indoor Navigation [24]

Besides, AR in Google Maps enhances navigation in complex indoor places like an unfamiliar airport, as shown in Figure 2.3.1.4. The AR arrows guide users to the closest amenities, such as restrooms and lounges.

2.3.2 Sunway MyCampus

Figure 2.3.2.1 Sunway MyCampus [26]

The Sunway MyCampus application [26] is a mobile platform developed to facilitate communication, enhance academic experiences, and streamline administrative processes for students, faculty, and staff at Sunway Campus in Bandar Sunway. The application provides a digital campus map with an indoor navigation feature, as shown in Figure 2.3.2.1. This is particularly useful for new students and visitors unfamiliar with the campus.



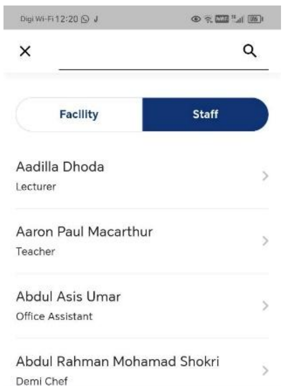


Figure 2.3.2.2 Facility Search Category [26] *Figure 2.3.2.3 Staff Search Category [26]*

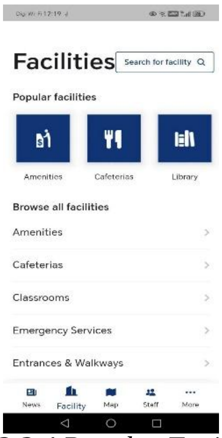


Figure 2.3.2.4 Popular Facilities [26]

The application allows the user to search for their desired location based on facility or staff, as shown in Figure 2.3.2.2 and Figure 2.3.2.3. Also, the user can view the popular facilities frequently searched by users, as shown in Figure 2.3.2.4, which allows users to find the most common facilities quickly.

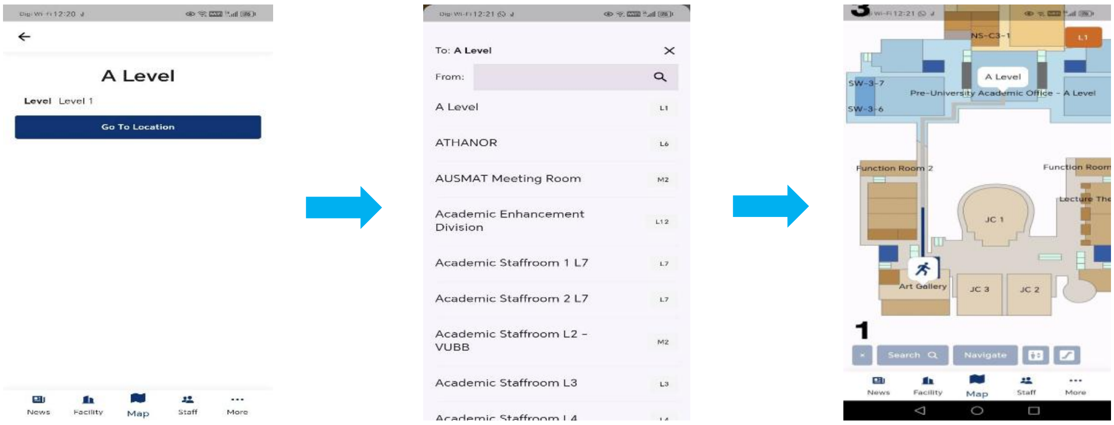


Figure 2.3.2.5 Static Map Navigation [26]

However, the navigation feature does not provide real-time navigation for the users. The user must choose their current location from a list of facilities or staff rooms after selecting the destination, rather than detecting the live location using a sensor or GPS, as shown in Figure 2.3.2.5. Thus, the user might get lost as the application does not provide a real-time location or show the current direction that the user is facing. This limits its effectiveness for users who need precise directions to specific locations, particularly indoors.

The app does not utilise AR technology to overlay navigational cues directly onto the user's real-world environment. The map does not provide detailed indoor floor plans or indoor mapping for multi-story buildings, making it challenging for users to navigate complex structures such as lecture halls or libraries. An AR-based solution would overcome the static nature of the map by providing real-time, interactive navigation that can adapt to the user's precise location. The static map may not provide sufficient guidance for new users or visitors to navigate efficiently. Thus, it is dependent on user familiarity with the campus.

2.3.3 Microsoft Path Guide

Microsoft Path Guide is a cutting-edge Android software that addresses the difficulties of indoor navigation. Path Guide does not rely on indoor maps or external hardware, in contrast to conventional systems. Instead, it records and navigates indoor locations using the smartphone's sensors, including barometers, gyroscopes, accelerometers, and electronic compasses [27]. This method makes a flexible, user-driven navigation solution possible and eliminates extra infrastructure requirements.

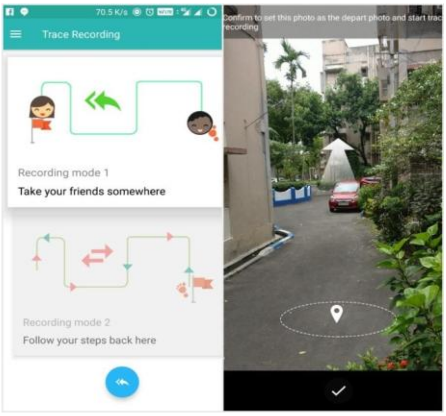


Figure 2.3.3.1 Microsoft Path Guide Recording Mode [28]

Path Guide operates on a peer-to-peer leader/follower model, allowing users to act as the “leader” to register movements in a public space by turning on the Recording Mode, as shown in Figure 2.3.3.1. In this process, Path Guide utilises the sensors already present in smartphones to record and navigate immediately without requiring specialised equipment or indoor maps, providing ease of use and accessibility to the users.

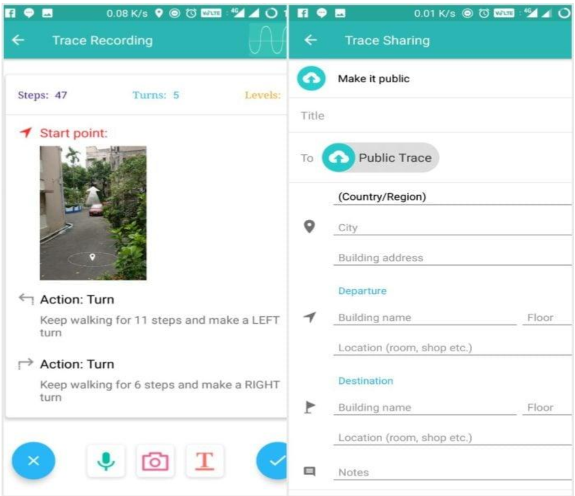


Figure 2.3.3.2 Microsoft Path Guide Trace Sharing [28]

Once the recording process is completed, a trace-sharing option comes up, which confirms whether the user wants to share the trace publicly [28], as shown in Figure 2.3.3.2. This approach makes the system inherently collaborative and adaptable, allowing users to contribute to and benefit from the navigation network. Besides, Path Guide enables the user to reverse the path back to its starting point from the destination point recorded previously.

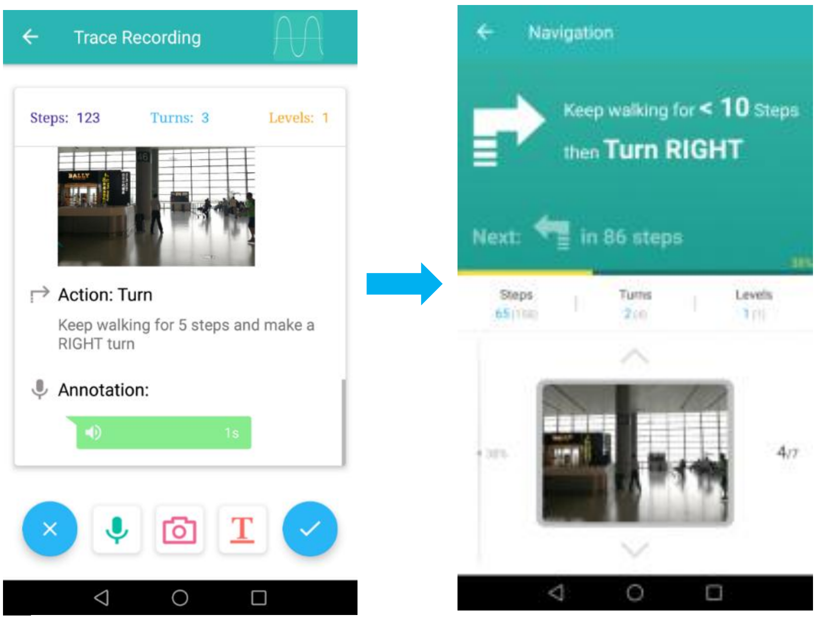


Figure 2.3.3.3 Microsoft Path Guide Annotation Support [27]

Also, Path Guide provides annotation support during trace recording, where text, audio, or photos can be added to a path, providing valuable context and interactivity [27].

Paths are uploaded to the cloud, which can be accessed and shared via a web browser, facilitating wide dissemination and use of navigation data [27].

However, Path Guide relies on users actively recording and sharing accurate navigation paths. If users do not contribute high-quality data, the system's effectiveness can be compromised, leading to gaps or inconsistencies in navigation information. Also, if there is no "leader" to record the path, the route might not exist for navigation. Besides, Path Guide lacks real-time updates and dynamic rerouting because it relies on static, user-recorded paths rather than live data. The app uses pre-uploaded navigation traces and does not process continuous data, which limits its ability to adapt to immediate changes such as construction or obstacles. Path Guide faces challenges with accuracy as sensor drift, magnetic interference, and variations in sensor quality can affect the precision of navigation instructions, causing potential deviations from the intended route.

2.3.4 Waze

Waze is a famous GPS navigation application that stands out due to its reliance on user-generated data [29]. Although primarily designed for outdoor navigation, Waze's features and community-driven approach provide valuable insights into its potential applications and limitations, including its relevance for indoor navigation contexts.

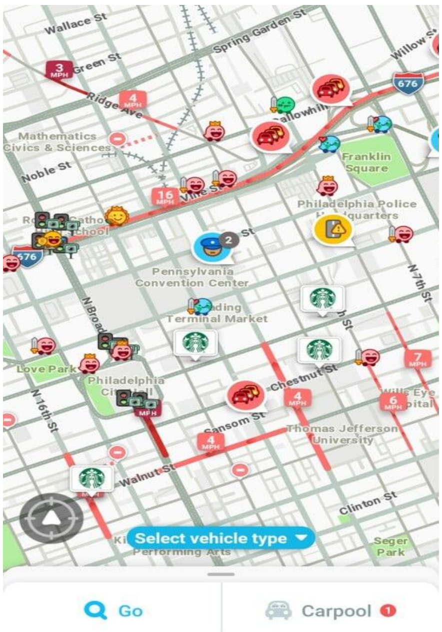


Figure 2.3.4.1 Waze Real-Time Information [31]

Waze's core strength lies in its crowd-sourced data model, with timely reporting on an average of 9.8 minutes earlier than a probe-based alternative [30]. Users contribute real-time information [31] regarding accidents, speed traps, road closures, and traffic conditions, as shown in Figure 2.3.4.1. The accuracy and timeliness of the traffic information provided by this user-generated data can significantly improve route planning and efficiency.

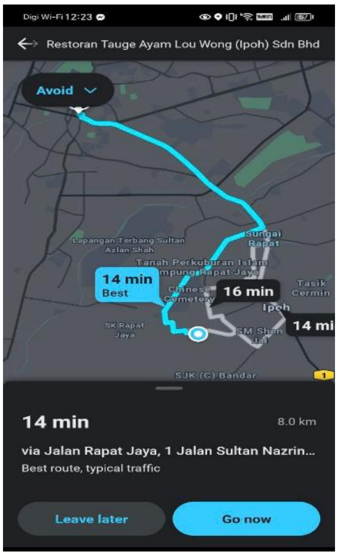


Figure 2.3.4.2 Waze Fastest/Shortest Route

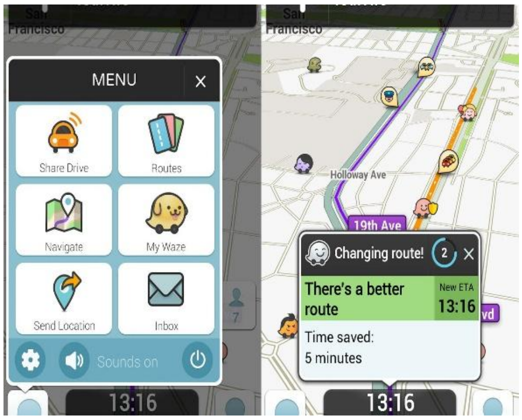


Figure 2.3.4.3 Waze Reroute [32]

Waze provides dynamic route options that adapt to the flow of traffic. The software calculates the fastest routes, as shown in Figure 2.3.4.2, and delivers turn-by-turn directions while considering traffic congestion and other factors. Waze will automatically reroute if there is congestion or an accident up ahead, which also depends on the community-sourced data [32], as shown in Figure 2.3.4.3. This versatility allows users to avoid delays and reach their destinations faster.

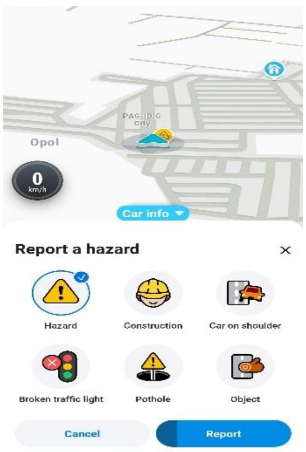


Figure 2.3.4.4 Waze User Report Feature

Also, it proved that the app fosters a strong community of drivers who actively share and receive information with interactive features, such as user reports, as shown in Figure 2.3.4.4.

However, Waze does not support indoor navigation as it relies on GPS [33], which is ineffective inside buildings where signal reception is poor. Also, the effectiveness of Waze’s routing and traffic information is contingent on the accuracy of user-generated data. Inaccurate or outdated reports can lead to suboptimal routing and potentially misleading information. Waze's reliance on crowd-sourced data can lead to variability in accuracy and reliability, especially in less populated areas. Lastly, Waze lacks augmented reality (AR) features. Without AR, it cannot provide immersive, visual navigation cues, limiting its use for detailed indoor navigation.

Waze is practical for outdoor navigation, providing real-time traffic updates, dynamic routing, and strong community engagement through crowd-sourced data. However, it lacks indoor navigation, AR features, data accuracy, and reliance on user reports. These limitations make it less suitable for indoor navigation applications.

2.3.5 AR Indoor Navigation Mobile Application for FICT [34]

One of the existing works closely related to this project is the AR Indoor Navigation Mobile Application for FICT, developed by Goh Brian Joon Jian, a student from the Faculty of Information and Communication Technology (FICT) at UTAR. The application was designed primarily for new intake students to navigate the FICT building efficiently without the need for registration or personal data collection.

Figure 2.3.5.1 Localisation via QR Code

Figure 2.3.5.2 Navigation Line

The system integrates AR-based navigation with QR code localisation, allowing users to scan a QR code to determine their position, as shown in Figure 2.3.5.1. It then calculates the shortest navigation path to the selected destination using a pathfinding algorithm, displaying an AR navigation line on the user's mobile device, as shown in Figure 2.3.5.2.

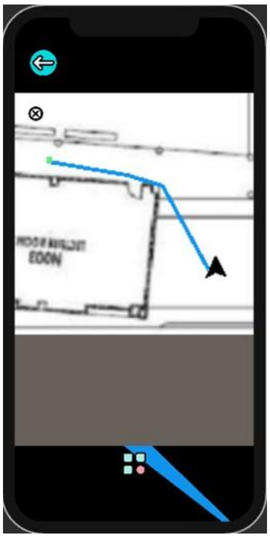


Figure 2.3.5.3 Mini Map Enlargement

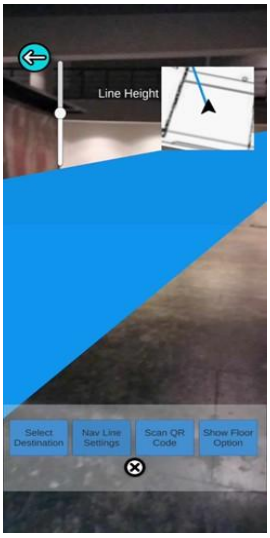


Figure 2.3.5.4 Adjustable Line Height

Additional features include floor selection, mini map enlargement, adjustable navigation line height, and destination drop-down options, making the app more interactive and user-friendly, as shown in Figure 2.3.5.3 and Figure 2.3.5.4.

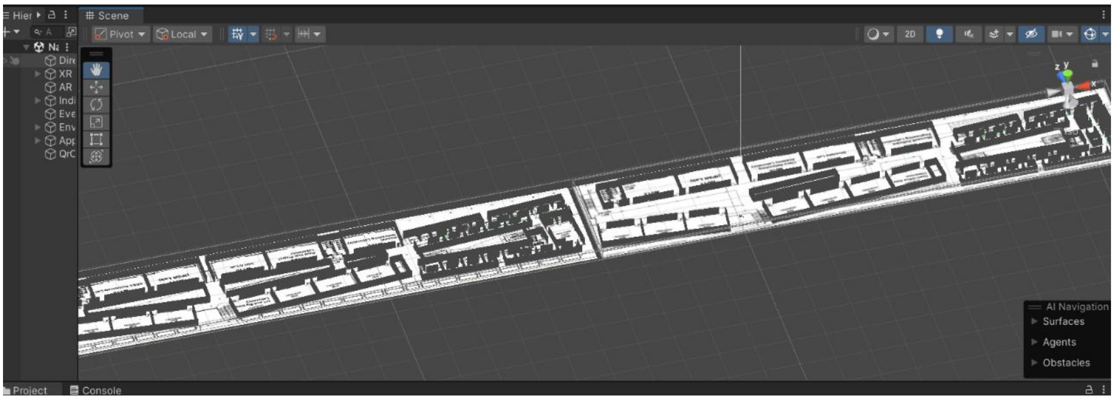


Figure 2.3.5.5 Separated Floor models in Unity

Despite its strengths, the application faces several limitations. The building dimensions were obtained using a virtual measuring app that relied on an AR ruler, which is not a reliable method for capturing accurate measurements. As a result, the scale in Unity did not accurately reflect the real-world environment, leading to localisation issues and reduced navigation accuracy. Furthermore, the floor models in Unity were developed separately, with no interconnected stairways between levels, as shown in Figure 2.3.5.5. As a result, the system could only navigate within a single floor along the same y-axis, requiring users to manually toggle the floor option and rescan a QR code when moving between floors. This disrupted seamless navigation across multiple levels and made the process less convenient. While the app successfully demonstrates the feasibility of AR-based indoor navigation with shortest path guidance, its limitations in accurate scaling and multi-floor integration highlight areas for further improvement.

2.4 Summarisation on Marker-based and Marker-Less Indoor Navigation

Method	Advantages	Limitations
Marker-based	• Provide precise guidance in an indoor environment • Cost-effective • Flexibility in installing new markers, deleting markers, or updating markers • Easy implementation without additional equipment	• Effectiveness can be reduced by poor lightning and obstructed view • Requires frequent placement and scanning of markers • Marker placement affects building aesthetics
Marker-Less	• No physical markers are needed • Adapt to dynamic environment changes	• The signal can be affected, which can impact the accuracy • It may require additional hardware • May increase power consumption

Table 2.4.1 Advantages and limitations of marker-based and marker-less indoor navigation

2.5 Summarisation of Existing Navigation System Reviewed

Name	Advantages	Limitations
Google	• Enhance pedestrian navigation with AR by overlaying directions • Integrate Neural Radiance Fields (NeRF) for realistic 3D representations • Provides real-time information	• Limited availability of AR features • AR may not always be effective in indoor environments with limited coverage • Dependence on the quality and availability of data
Sunway MyCampus	• Provides digital map and facility search features • User-friendly interface	• It does not offer real-time navigation or live location tracking • Does not integrate AR technology • It does not give detailed indoor floor plans
Microsoft Path Guide	• User-driven navigation without needing indoor maps or external hardware • Provide a collaborative system to record, share, and annotate navigation path • Cloud integration for navigation data sharing and accessibility • Integrate with AR technology for immersive navigation cues	• Relies on user contribution • Some paths may not exist if there is no “leader” to record the path first • It does not provide dynamic rerouting, as the navigation is based on static paths • Sensor drift and interference can affect the accuracy
Waze	• Provide real-time traffic information due to crowd-sourced data	• It does not provide indoor navigation as it relies on GPS

	• Provide dynamic routing based on current traffic and user reports • Strong community engagement	• Dependence on the accuracy of user-generated data • Does not integrate AR technology
FICT AR Indoor Navigaiton App	• Integrates AR-based navigation with QR code localisation for accurate positioning • Uses pathfinding algorithm to calculate the shortest route to destinations • Provides real-time navigation	• Resulting scale in Unity did not accurately match the real-world environment • Localisation issues and reduced navigation accuracy due to scaling errors • System only supported navigation within a single floor along the same y-axis • It requires the users to manually toggle floor options and rescan QR codes when switching level

Table 2.5.1 Advantages and limitations of existing navigation system reviewed

2.6 Comparison between ARFICT and the Existing Navigation System

This project develops an AR indoor navigation application to help students find destinations within the FICT building at UTAR Kampar. User localisation is achieved by scanning a QR code, after which the system calculates the shortest route using the A* algorithm. The user is then guided through the environment with a real-time AR navigation path. Unlike traditional systems, the app supports multi-floor navigation and integrates additional features to improve the overall user experience.

The marker-based method was chosen for its reliability, cost-effectiveness, and simplicity in indoor environments. It enables accurate localisation without requiring extra hardware, making it easier to manage compared to marker-less methods or beacon-based systems. Marker-less navigation, while flexible, is more complex and prone to issues such as signal interference and higher power consumption. GPS-based navigation is unsuitable indoors because walls and ceilings block satellite signals. Instead, ARFICT uses the device's sensors to continuously track user movement after QR code scanning, ensuring smooth and reliable navigation throughout the building.

Feature / System	AR	Indoor Navigation Implementation	Real-time Navigation	Navigation Technique
Google	Yes	Limited indoor AR features	Yes	GPS-based navigation, mainly for outdoor navigation
Sunway MyCampus	No	Static map without live tracking	No	Static map-based navigation path
Microsoft Path Guide	Yes	User-recorded paths without indoor map	No	Static path guidance from the recorded path
Waze	No	Not available (GPS-based only)	Yes	GPS-based routing for outdoor navigation, which utilises crowd-sourced traffic information
ARFICT	Yes	Marker-based using QR codes	Yes	QR code-based localisation, which utilises mobile device sensors to track user movement during navigation

Table 2.6.1 Comparison between ARFICT and the existing navigation system reviewed

In addition to core AR functionality, ARFICT integrates enhanced features such as audio-guided instructions, accurate multi-floor navigation, a multi-floor mini map, multiple navigation modes, and real-time distance calculation. Accuracy is significantly improved compared to the earlier FICT navigation app, as building measurements were obtained using a laser distance meter with near-zero error. In contrast, the previous app relied on AR ruler measurements, which were less precise and often inconsistent with the real environment.

System / Feature	FICT AR Indoor Navigation App by Goh Brian Joon Jian	ARFICT
AR	Yes	Yes
Indoor Navigation Technique	Marker-based using QR codes by using mobile device sensors to track user movement during navigation.	Marker-based using QR codes by using mobile device sensors to track user movement during navigation.
Real-time Navigation	Yes	Yes
Audio Navigation Support	No	Yes
Multi-Floor Navigation Support	No, navigation is limited to a single floor and only functions along the same y-axis.	Yes, supports multi-floor navigation with interconnected floors in the Unity building model.
Mini Map	Yes, but only supports single-floor maps without support for gestures to move across the map or zoom in/out.	Yes, supports multi-floor maps for better visualisation and navigation, with user gestures such as swiping to move across the map and pinch-to-zoom for zooming in/out.
Accuracy	Low	High
Adjustable Line Height	Yes	Yes
Navigation Mode	Line only	Line, arrow
Distance Calculation Label	No	Yes
Reroute to a Closer Destination	No	Yes

Table 2.6.2 Comparison between previous FICT AR Indoor Navigation App and ARFICT

CHAPTER 3

System Methodology/Approach

This chapter presents an overview of the proposed solution, such as system architecture diagrams, user requirements, use case diagrams, use case descriptions and activity diagrams to give a broad understanding of the project's approach. The section also covers development methodologies and the timeline for project completion.

3.1 System Architecture

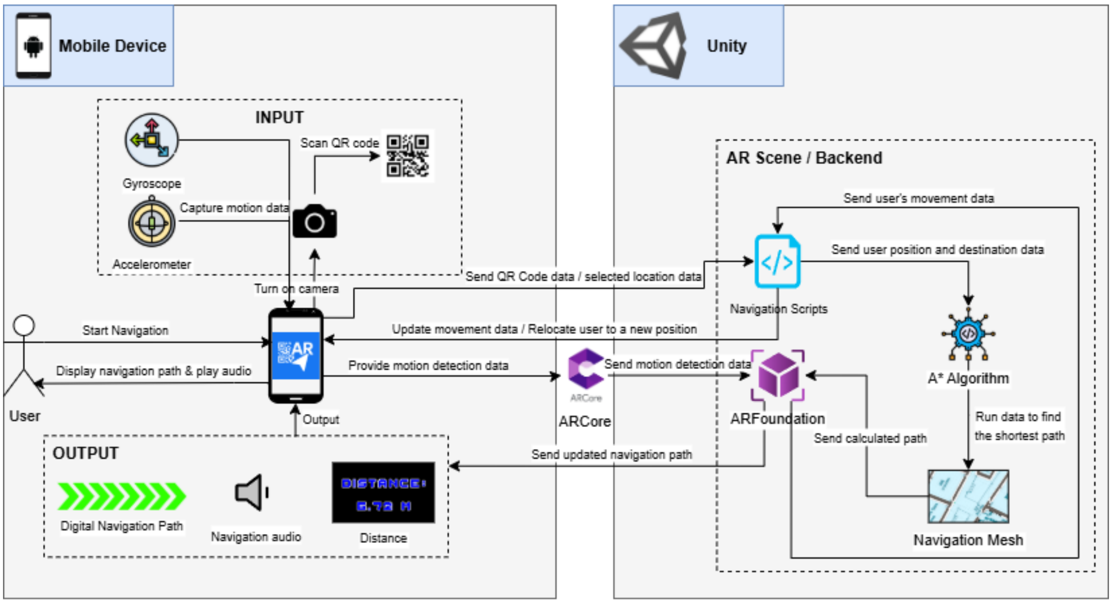


Figure 3.1.1 System Architecture Diagram

The system architecture integrates an Android mobile device with Unity and artificial intelligence, specifically the A* pathfinding algorithm operating on a Navigation Mesh, which defines the walkable areas for efficient AR-based indoor navigation. This architecture is structured into three primary layers, which are the presentation layer, the service layer, and the application logic layer.

Presentation Layer (Mobile Device)

The mobile device functions as the presentation layer, serving as both the input and output interface. The device camera captures real-world images for QR code scanning, which is processed using the ZXing library to determine the user's starting location. The mobile device also displays the augmented navigation path and provides user interaction through the screen and speakers.

Service Layer (ARCore)

The service layer is represented by ARCore, which bridges the physical environment with the application logic. ARCore provides motion tracking, environment understanding, and anchoring, enabling accurate spatial awareness. This allows the system to map and track the user’s position within the building in real time.

Application Logic Layer (Unity with ARFoundation)

Unity operates as the application logic layer, where the navigation algorithms and AR content are implemented. The navigation path is calculated using the A* algorithm applied to a predefined NavMesh. Unity, supported by ARFoundation, integrates ARCore’s spatial data to generate and manage augmented elements. The Unity rendering engine displays the navigation path, either as directional arrows or guiding lines, directly within the user’s environment.

3.2 Use Case Diagram and Description

This section outlines the interactions between the user and AR indoor navigation app.

3.2.1 Use Case Diagram

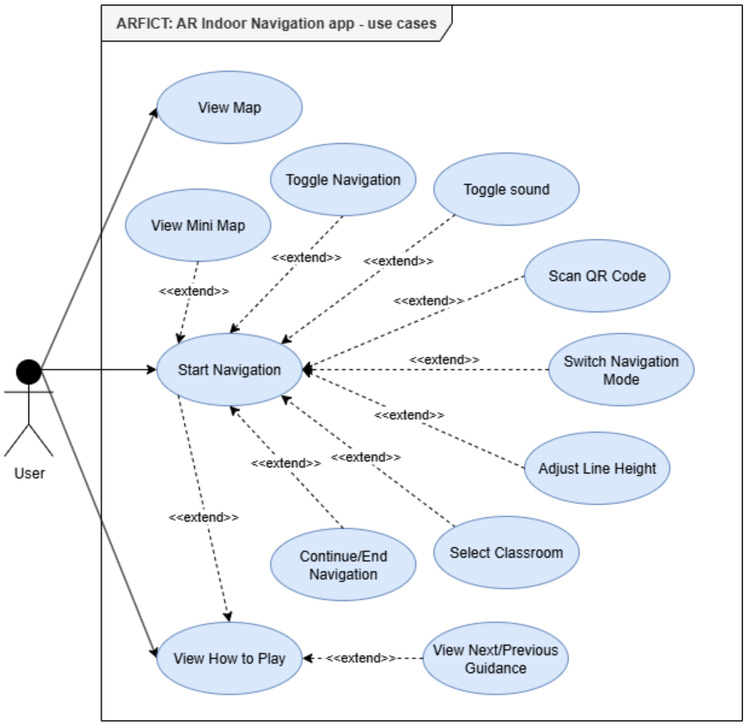


Figure 3.2.1.1 Use Case Diagram

The application's use case diagram in Figure 3.2.1.1 presents three main use cases, which are UC01 Start Navigation, UC02 View How to Play, and UC03 View Map. UC01 Start Navigation is the primary flow where users scan a QR code to set their location, optionally select a destination, and configure navigation options such as showing or hiding the path, switching modes, adjusting line height, viewing the mini map, and toggling sound. UC02 View How to Play provides step-by-step instructions on how to use the application, with an option to proceed to UC01 Start Navigation. UC03 View Map allows users to view the building's floor plans for orientation and destination selection. Together, these use cases capture the core functionality of the application.

3.2.2 Use Case Description

This section provides detailed descriptions of the three primary use cases, which serve as the foundation for use case testing. Among these, UC01 Start Navigation is the main use case, covering scanning a QR code, selecting a destination, configuring navigation, and receiving real-time AR guidance. UC02 View How to Play provides instructions and a direct option to start navigation, while UC03 View Map allows users to view floor plans for orientation and destination selection, ensuring the app meets user needs.

UC01 – Start Navigation

Field	Value
ID	UC01
Purpose	To allow the user to scan a QR code, select a destination, configure navigation options, and receive real-time AR guidance until reaching the destination.
Primary Actor	User
Trigger	User selects Start Navigation from the main menu.
Sub-Functions	• Scan QR Code • Select Destination • Toggle Navigation • Switch Navigation Mode • Adjust Line Height • Reroute to a Closer Destination

		• View Mini Map • Toggle Sound • Continue/End navigation
Pre-conditions		• Application is installed and running. • Camera permission is granted. • Device sensors (camera, gyroscope, accelerometer) are active.
Scenario Name	**Step**	**Action**
Main Flow	1	User launches Start Navigation screen.
	2	System activates device camera for navigation.
	3	User performs available actions on the screen.
Alternate Flow – ***Scan QR Code***	3.1.1	User clicks (+) button.
	3.1.2	System displays feature panel.
	3.1.3	User clicks scan QR code button.
	3.1.4	System displays QR code scanning panel.
	3.1.5	User scans a QR code.
	3.1.6a	If QR code is valid, system updates user's location, displays success message, and provides audio confirmation.
	3.1.6b	If the QR code is invalid, the system displays an error message with audio confirmation and then returns to Step 3.1.2.
Alternate Flow – ***Select Destination***	3.2.1	User clicks Select button.
	3.2.2	System displays a list of destination in the dropdown menu.
	3.2.3	User selects a destination.
	3.2.4	System calculates shortest path using A* algorithm, displays navigation path, success message, and provides audio confirmation.
Alternate Flow – ***Toggle Navigation***	3.3.1	User clicks (+) button.
	3.3.2	System displays feature panel.
	3.3.3	User click toggle navigation button.

	3.3.4	System enables/disables the navigation features such as distance label and navigation path, displays confirmation message, and provides audio confirmation.
Alternate Flow – ***Switch Navigation Mode***	3.4.1	User clicks (+) button.
	3.4.2	System displays a feature panel.
	3.4.3	User clicks switch navigation mode button.
	3.4.4	System switches navigation path between line/arrow, displays confirmation message, and provides audio confirmation.
Alternate Flow – ***Adjust Line Height***	3.5.1	User clicks (+) button.
	3.5.2	System displays feature panel.
	3.5.3	User clicks adjust line height button.
	3.5.4	System displays adjustable slider with confirmation message and audio notification.
	3.5.5	User moves slider handle.
	3.5.6	System updates line height according to slider value.
Alternate Flow – ***Reroute to a Closer Destination***	3.6.1	User selects a destination with multiple locations of the same name, such as toilet.
	3.6.2	System calculates and navigates the user to the nearest location.
	3.6.3	User walk away from the destination.
	3.6.4	If a closer location is detected, the system automatically reroutes the user to the new destination.
	3.6.5	System displays a confirmation message and provides audio notification of the reroute.
Alternate Flow – ***View Mini Map***	3.7.1	User clicks mini map.
	3.7.2	System enlarges and centers mini map.
	3.7.3	User drags to move, zoom in/out.
	3.7.4	User clicks (X) button.
	3.7.5	System restores mini map to original position.
Alternate Flow –	3.8.1	User clicks speaker button.

Toggle Sound	3.8.2	System enables/disables sound.
Alternate Flow – *Continue/End Navigation*	3.9.1	User reaches the destination.
	3.9.2	System displays dialog and provides audio notification.
	3.9.3a	If user clicks continue, system closes dialog and maintains session.
	3.9.3b	If user click end, system closes dialog and clears all navigation data.
Post-conditions		• User's position is updated based on scanned QR code. • Shortest path to selected destination is calculated and displayed. • Navigation path shown with adjustable line height and distance labels. • Mini map updated with current location and multi-floor support. • Sound guidance enabled/disabled based on user preference. • System reroutes user to the closest available destination if a nearer option is detected. • Completion dialog shown upon reaching destination. • If navigation ends, all navigation data and AR elements are cleared.

Table 3.2.2.1 UC01 Start Navigation Use Case Description

UC02 – View How to Play

Field	Value
ID	UC02
Purpose	To view the guidance on how to use the app.
Primary Actor	User
Trigger	User selects How to Play from the main menu.
Sub-Functions	• View Next/Previous Guidance

	• Start Navigation	
Pre-conditions	• Application is installed and running.	
Scenario Name	**Step**	**Action**
Main Flow	1	User launches How to Play screen.
	2	System displays first guidance title and image.
	3	User clicks next, previous, or start navigation button.
Alternate Flow – *View Next/Previous Guidance*	3.1.1	System displays next/previous guidance title and image.
Alternate Flow – *Start Navigation*	3.2.1	User launches Start Navigation screen and proceeds with UC01.
Post-conditions	• The user successfully views the navigation guidance and may proceed to start navigation.	

Table 3.2.2.2 UC02 How to Play Use Case Description

UC03 – View Map

Field	**Value**	
ID	UC03	
Purpose	To view the floor plan of the FICT building.	
Primary Actor	User	
Trigger	User selects View Map from the main menu.	
Sub-Functions	-	
Pre-conditions	• Application is installed and running.	
Scenario Name	**Step**	**Action**
Main Flow	1	User launches View Map screen.
	2	System displays floor plan images.
Post-conditions	• The user successfully views the floor plan of the FICT building.	

Table 3.2.2.3 UC02 View Map Use Case Description

3.3 Activity Diagram

This section presents three activity diagrams that illustrate the key flows of the project's use cases. Among these, the most significant is UC01 Start Navigation Activity Diagram, as it represents the core function and delivers the greatest value to the project.

UC01 – Start Navigation Activity Diagram (Main Use Case)

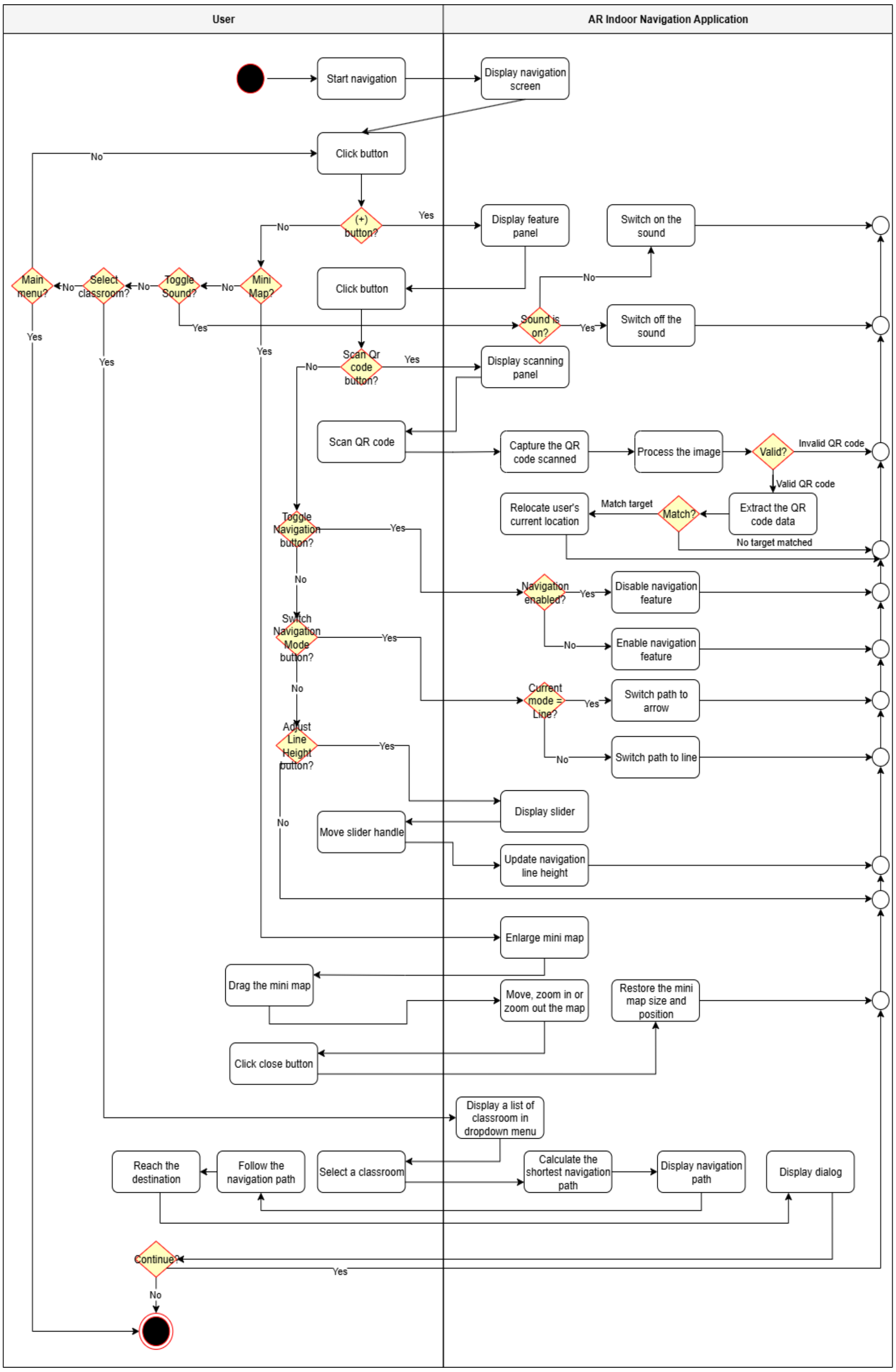


Figure 3.3.1 Start Navigation Activity Diagram

UC02 – View How to Play Activity Diagram

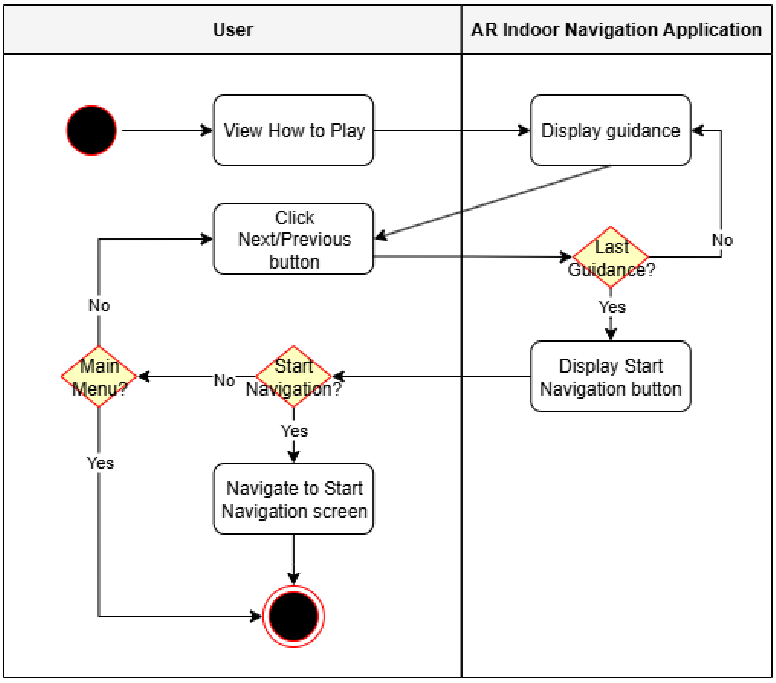


Figure 3.3.2 View How to Play Activity Diagram

UC03 – View Map Activity Diagram

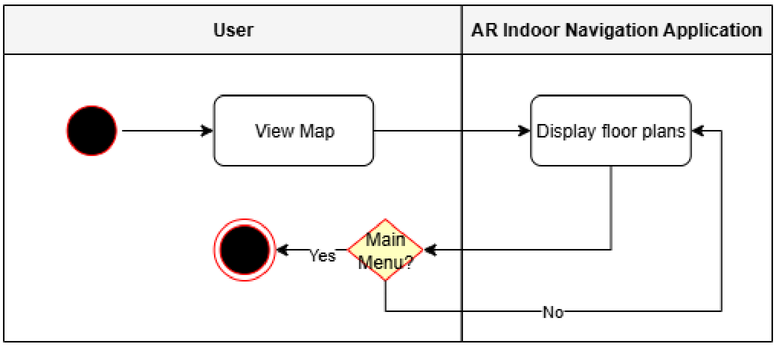


Figure 3.3.3 View Map Activity Diagram

3.4 Storyboard

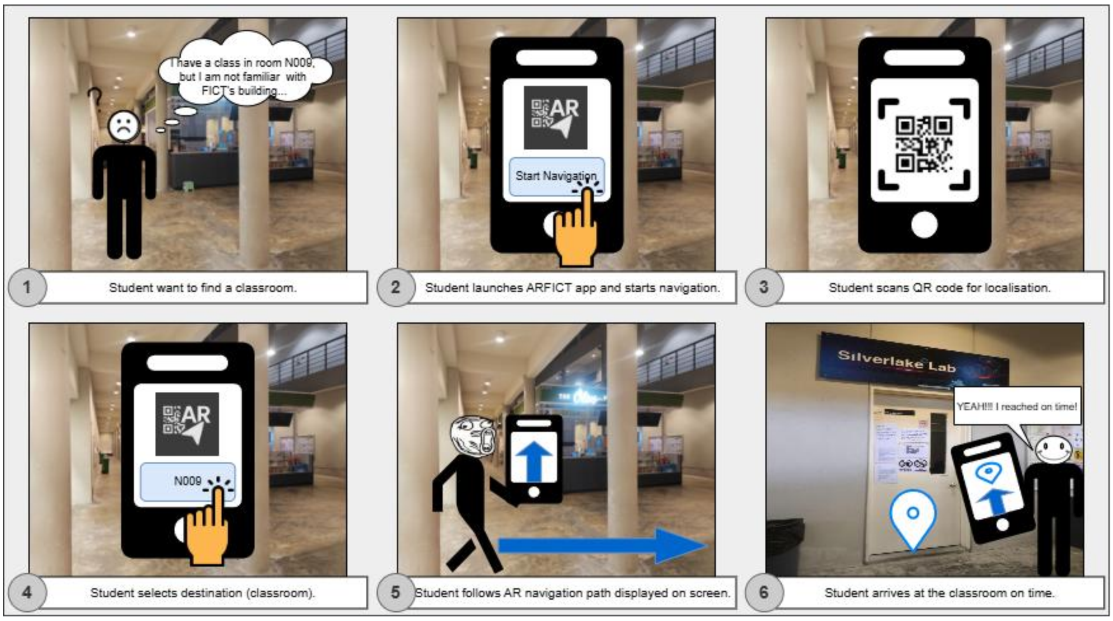


Figure 3.4.1 Indoor navigation app ARFICT's storyboard

The storyboard represents the typical user experience when using the ARFICT application for indoor navigation within the FICT building. It begins with a student who needs to find a specific classroom but is unfamiliar with the building's layout. The student starts the ARFICT application and starts the navigation feature to solve the problem. The student then scans a QR code inside the building to determine their current location using indoor localisation. After localising, the student chooses the desired destination classroom from a list of available locations in the app. The application then shows an AR navigation path on the mobile device's screen to guide the student through the building. Finally, the student arrives at the destination, which is marked by a virtual indicator in the AR view, indicating the navigation's endpoint.

3.5 Development Methodology

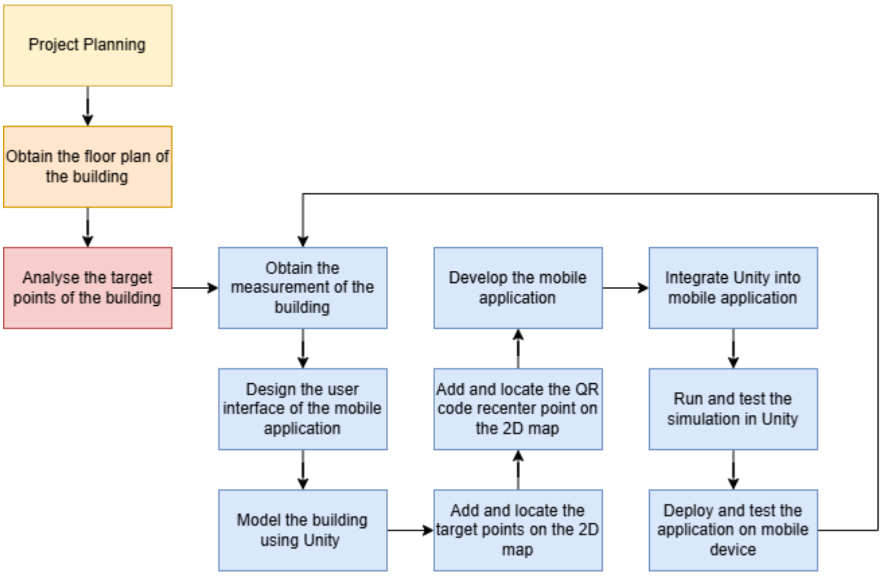


Figure 3.5.1 ARFICT Project Agile Methodology Workflow

The AR-based indoor navigation application is developed iteratively, rapidly prototyped, and continuously tested using an Agile methodology, as illustrated in Figure 3.2.1. This approach allows for continuous feedback loops, making necessary adjustments based on testing results [35] without disrupting the project's progress.

The process begins with project planning, followed by obtaining the floor plan and analysing reference points for the AR system. Building measurements are then collected, first using an iPad Pro and later refined with a laser distance meter for higher precision. This iterative process primarily focuses on achieving accurate measurements and precise target point placement to ensure alignment with the real-world environment.

Next, the user interface is designed to enhance usability, and the building's 3D model is created in Unity. Navigation target points and QR code recenter points are added for localisation, while additional features such as a mini map and navigation mode options further improve user experience.

Mobile app development integrates Unity's navigation features, including 3D model alignment and positioning through target and recenter points. Testing begins in Unity simulations, followed by real-world testing on mobile devices for refinement. This cycle continues to ensure seamless and adaptable indoor navigation, providing the user with the most reliable and accurate guidance.

3.6 Timeline

Task \ Week	1	2	3	4	5	6	7	8	9	10	11	12	13
Study the AR navigation methods	■												
Obtain floor plan		■											
Identify target locations		■											
Design the mobile app UI		■											
Measure the building dimensions			■										
Model the building in Unity				■	■								
Add target points in Unity					■								
Implement pathfinding with shortest route calculation					■								
Implement QR code detection and localisation						■							
Develop basic mobile app UI							■						
Conduct simulation testing on Unity Editor								■					
Deploy the app for further testing at FICT building									■				
Conduct minor adjustments on the prototype										■			
Produce the FYP1 Report											■	■	
FYP1 Presentation													■

Table 3.6.1 FYP1 Timeline

Task \ Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Study the navigation methods for multiple floors														
Design and plan the navigation to first floor														
Identify target locations														
Update and expand the building model to first floor														
Add target points for navigation and localisation														
Add additional features to support navigation														
Enhance the mobile app UI														
Conduct simulation testing on Unity Editor														
Deploy the app for further testing at FICT building														
Conduct adjustments on the prototype														
Perform testing after the adjustment														
Produce the FYP2 Report														
FYP2 Presentation														

Table 3.6.2 FYP2 Timeline

CHAPTER 4

System Design

This section presents the system design through a system flowchart and system block diagram, illustrating the proposed application.

4.1 System Flowchart

This section presents the system flowchart, which includes both the overall system flow and the subsystem flowcharts. These flowcharts provide a simple and clear illustration of the system design, showing how the processes are organised and connected within the proposed application.

4.1.1 Overall System Flowchart

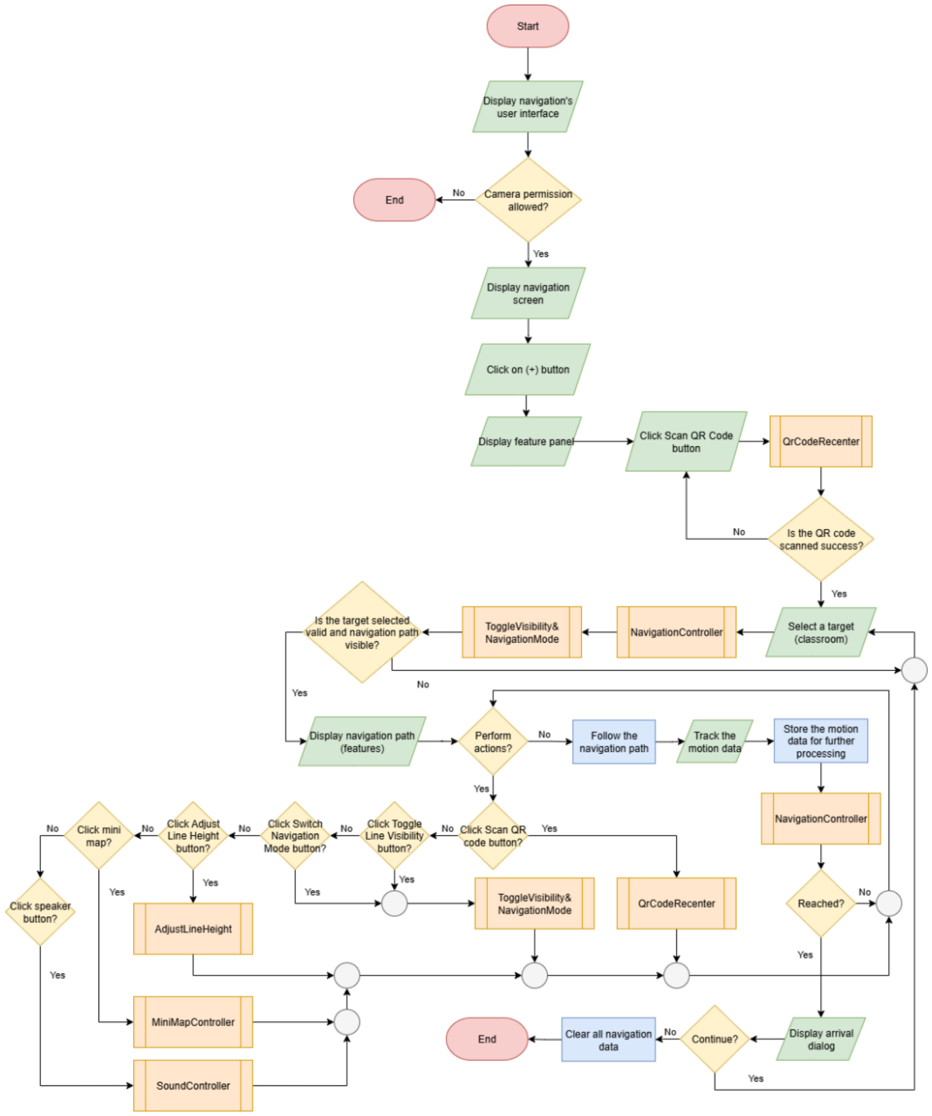


Figure 4.1.1.1 Overall System Flowchart

The flowchart in Figure 4.1.1.1 illustrates the overall system flow of the AR-based indoor navigation application, ARFICT. It incorporates navigation path visualisation, QR code recentering, and interactive feature controls. The process begins with the display of the navigation interface, followed by a camera permission check. If permission is granted, the system presents the navigation screen, where the user can access features through the menu panel.

The user determines their current location by scanning a QR code. If the scan is successful, the system re-centres the position and allows the user to select a target destination. Once a valid target is chosen and the navigation path is visible, the system displays the path and updates it dynamically to guide the user. The user follows the path while the system tracks motion data and continuously checks whether the destination is reached. Upon arrival, an arrival dialogue is displayed.

Throughout navigation, the user may also interact with additional features such as toggling line visibility, switching visualisation modes, adjusting line height, enabling/disabling the mini map, or muting/unmuting sound. If the user decides not to continue, all navigation data is cleared, and the session ends.

4.1.2 Subsystem Flowchart: QR Code Recenter

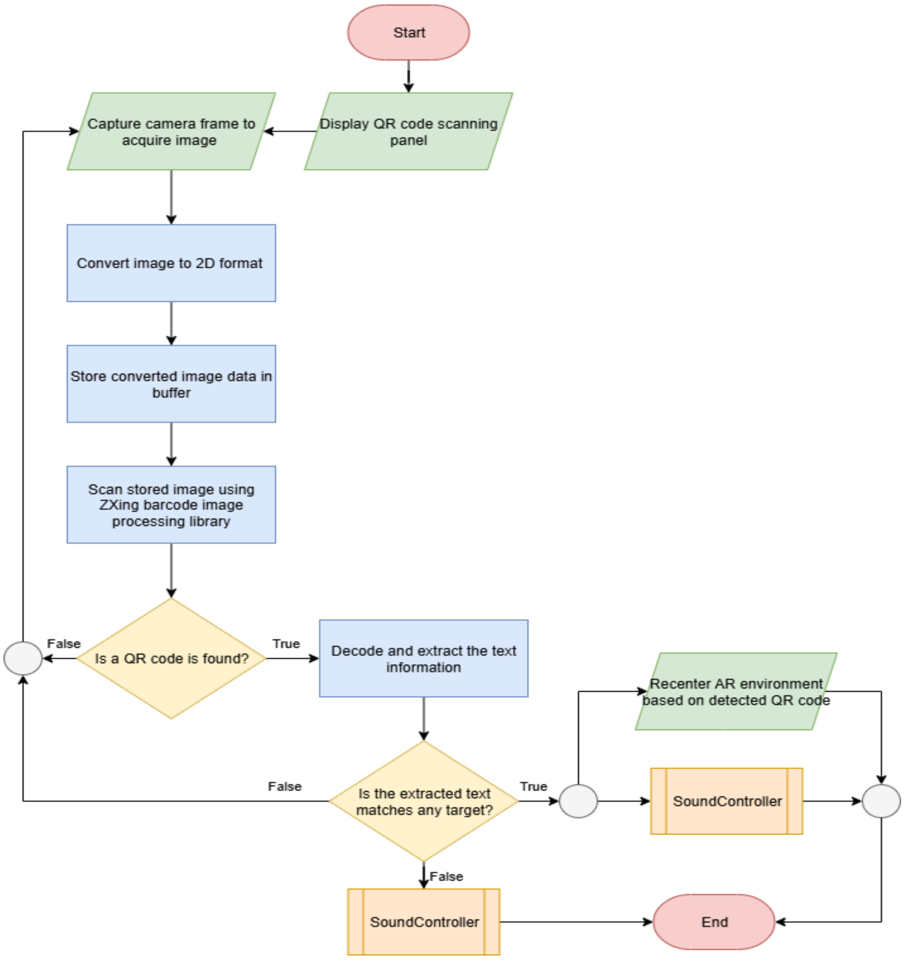


Figure 4.1.2.1 QR Code Recentre Flowchart

The subsystem flowchart illustrates the QR code recentring module of ARFICT, as shown in Figure 4.1.2.1. The process begins with the display of a QR code scanning panel to capture an image, which is then stored in a buffer and converted into a 2D format. The system uses the ZXing barcode processing library to detect a QR code within the stored image.

If no QR code is detected or if the extracted text does not match any predefined target, the image is captured again to retry the process. When a valid QR code is detected and the text matches a target, the system accurately updates the user's location by recentring the AR environment. A voice notification is also played to inform the user of the successful recentring.

This procedure ensures that the AR navigation system aligns correctly with the user's position by using QR code markers for precise indoor positioning.

4.1.3 Subsystem Flowchart: Navigation Controller

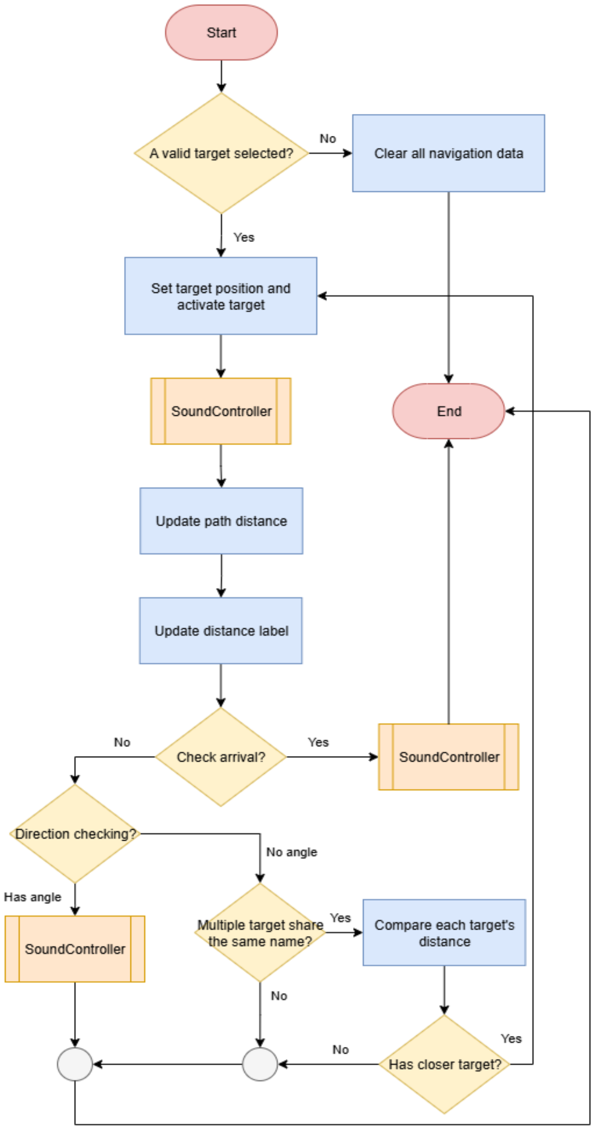


Figure 4.1.3.1 Navigation Controller Flowchart

The subsystem flowchart in Figure 4.1.3.1 shows the flow of managing the navigation process throughout the system. The process begins with validating the user's target selection. If no valid target is chosen, the system clears all navigation data and terminates the process. If a valid target is selected, the system sets the target's position and activates navigation. An audio cue signals the start, and the system continuously updates the remaining path distance to keep the user informed.

During navigation, two conditions are checked, which are arrival and direction. On arrival, the system plays an audio cue and ends the process. Otherwise, a direction check is performed, and when a valid angle is determined, a directional audio cue is provided.

If multiple targets share the same name, the system compares their distances and re-routes the user to the nearer one by restarting the navigation process.

Overall, the subsystem flowchart outlines a structured process that validates user selection, provides real-time guidance, and ensures accurate arrival detection for a smooth navigation experience.

4.1.4 Subsystem Flowchart: Sound Controller

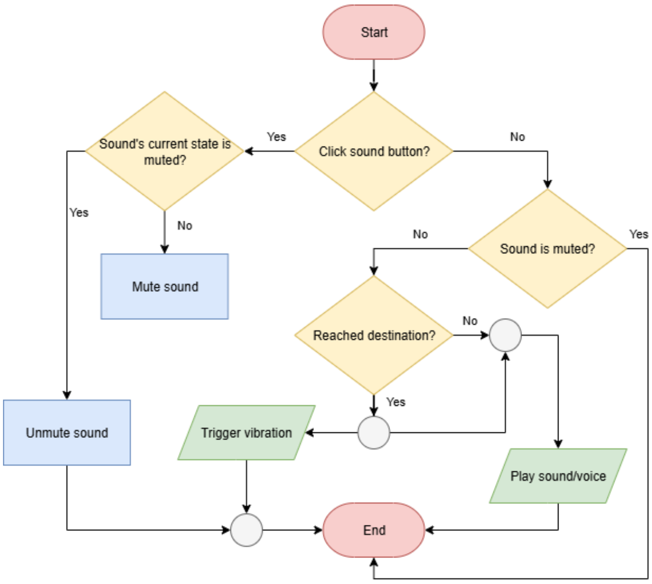


Figure 4.1.4.1 Sound Controller Flowchart

The subsystem flowchart in Figure 4.1.4.1 illustrates the flow of managing the system’s audio feedback and settings through the Sound Controller module. The process begins by checking whether the user clicks the sound button. If clicked, the system determines the current state and toggles between mute and unmute before ending the process.

If the sound button is not clicked, the system checks the mute state. When muted, no sound is played, but a vibration is triggered upon reaching the destination. When unmuted, the system provides voice cues during navigation and triggers a vibration at the destination. This ensures users receive consistent feedback through sound or haptics based on their preference.

4.1.5 Subsystem Flowchart: Toggle Visibility and Navigation Mode

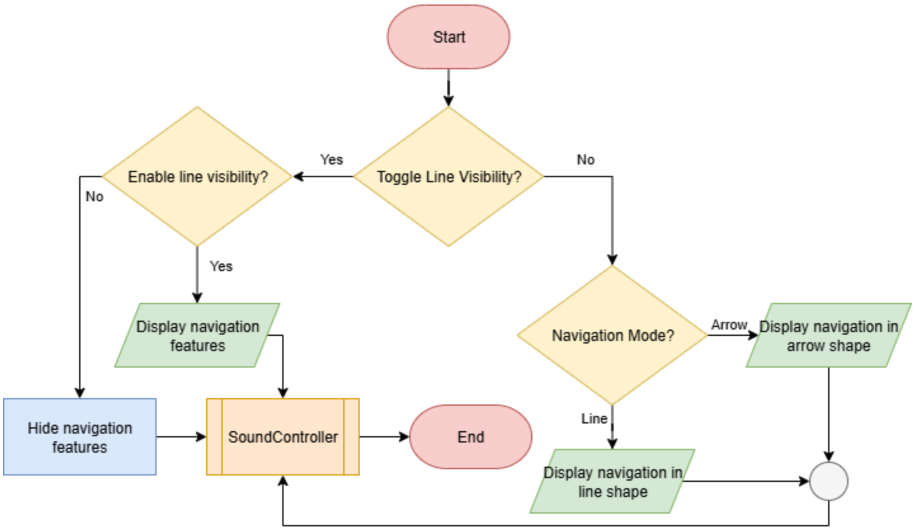


Figure 4.1.5.1 Toggle Visibility & Navigation Mode Flowchart

The subsystem flowchart in Figure 4.1.5.1 illustrates the flow of managing navigation path display through the Toggle Visibility and Navigation Mode module. The process begins when the user clicks the Toggle Line Visibility button. The system then checks the current state of the path. If line visibility is disabled, the navigation features are hidden; otherwise, they are enabled and displayed. In both cases, a sound cue is triggered via the SoundController module to acknowledge the action.

If the Toggle Line Visibility button is not clicked, the system proceeds to check the Navigation Mode. The user can switch between Arrow and Line modes, and the system displays the navigation path based on the selected option. A sound cue is also triggered to confirm the change.

Overall, the flowchart demonstrates how the subsystem enables users to control both the visibility and representation of the navigation path, ensuring flexibility without interrupting the navigation experience.

4.1.6 Subsystem Flowchart: Adjust Line Height

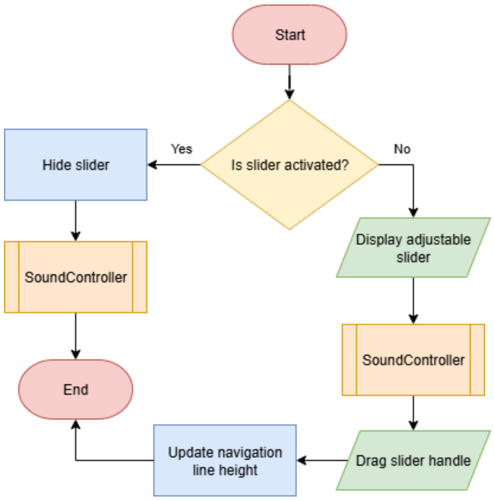


Figure 4.1.6.1 Adjust Line Height Flowchart

The subsystem flowchart in Figure 4.1.6.1 illustrates the flow of adjusting the navigation line height. The process begins by checking whether the user activates the slider function by clicking the button. If the slider is not activated, the system displays an adjustable slider for user interaction and triggers a sound cue via the SoundController module to confirm its display. The user can then drag the slider handle to set the desired height, and the system updates the navigation line accordingly before ending the process.

If the slider is already active, the system hides the slider from the user interface and plays a sound cue to acknowledge the action. This ensures the slider does not obstruct the navigation view when not in use. Once the slider is hidden, the process terminates.

Overall, the flowchart demonstrates how the subsystem provides a simple mechanism for showing, adjusting, and hiding the slider to enhance user control without disrupting navigation.

4.1.7 Subsystem Flowchart: Mini Map Controller

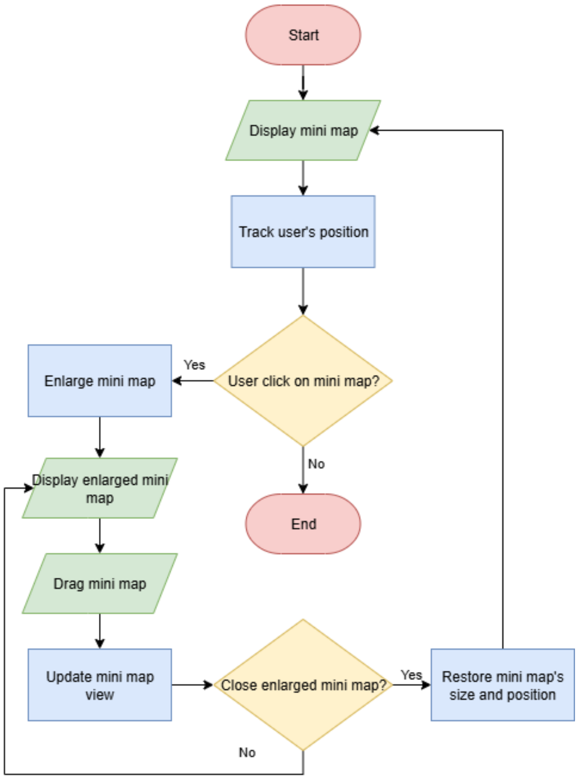


Figure 4.1.7.1 Mini Map Controller Flowchart

The Mini Map Controller flowchart in Figure 4.1.7.1 manages the display and interaction of the mini map. The process starts by displaying the mini map and tracking the user’s position. If the mini map is clicked, it enlarges to provide a detailed view, allowing the user to drag and pan while the system updates the map in real time. When the user closes the enlarged view, the mini map returns to its original state, resuming position tracking. Overall, the flow illustrates a cycle of display, interaction, and restoration that keeps the mini map both functional and adaptive during navigation.

4.2 System Block Diagram

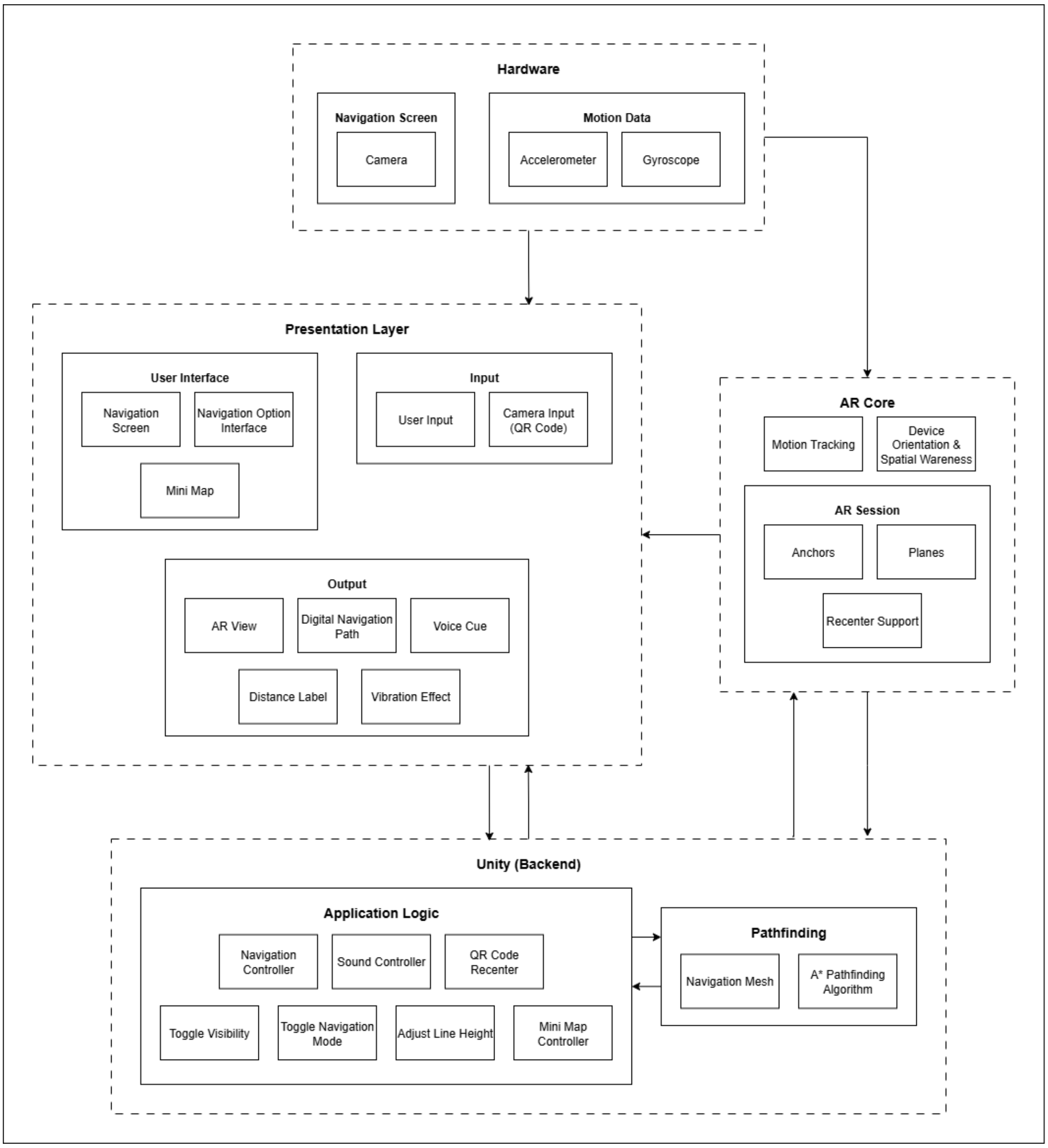


Figure 4.2.1 System Block Diagram

The system block diagram illustrates the architecture of the ARFICT application, which is structured into several interconnected layers which are Hardware, AR Core Service, Presentation Layer, and the Unity backend. This setup shows how the system processes input, handles core logic, and provides feedback to the user, creating a seamless and integrated experience.

The system's foundation is the Hardware layer, providing raw data from the device's physical components. This includes motion data from the accelerometer and gyroscope, and a camera to capture the real-world environment. The raw camera feed is sent to both the AR Core for processing and the Presentation Layer for direct display. The AR Core Service acts as the augmented reality engine, using this data to perform crucial tasks like motion tracking and managing the AR session, which identifies anchors and planes. AR Core then sends processed data and the rendered AR view to the backend and presentation layers.

The Unity backend contains the core logic and processing hub. Its application logic houses various controllers that manage all functionality, from navigation and sound to handling user-configurable settings. This layer communicates with the pathfinding module, which uses the A* algorithm to calculate the shortest route. The backend receives data from the AR Core and user input from the Presentation Layer, processes it, and sends commands back to the Presentation Layer to update the user's view and provide feedback.

The Presentation Layer is the user-facing component responsible for all visual and interactive elements. Its user interface includes the navigation screen, controls, and mini-map. It receives raw camera data directly from the hardware and the final rendered view from the AR Core and backend. It handles user input, such as button clicks and QR code scans, sending this data to the Unity backend for processing. Processed data is communicated to the user through various outputs, including the AR view, digital navigation path, and voice cues.

In summary, the system operates as a continuous loop. Hardware captures environmental data, which the AR Core processes to understand the physical space. The Unity backend uses this data, along with user input from the Presentation Layer, to execute core logic and calculate navigation paths. The results are then sent back to the Presentation Layer to provide real-time augmented reality feedback to the user. This layered architecture ensures that the application is modular, efficient, and responsive.

CHAPTER 5

System Implementation

This chapter presents the hardware and software setup of the ARFICT system and explains how they are integrated to support AR-based indoor navigation. It also describes the system operation, main features, and the challenges encountered during implementation and testing.

5.1 Hardware Setup

The hardware involved in this project is a laptop and an Android mobile device. A laptop is used to design and develop the mobile application and AR technology in Unity.

Description	Specifications
Model	Asus Tuf Gaming F15
Processor	Intel Core i5-10300H
Operating System	Windows 11
Graphic	NVIDIA GeForce GTX 1650
Memory	24GB DDR4 RAM
Storage	1 TB SSD

Table 5.1.1 Specifications of Laptop

A mobile device is used to deploy the ARFICT application and test it in the FICT building.

Description	Specifications
Model	Honor 70
Processor	Qualcomm Snapdragon 778G Plus
Operating System	Android v14
RAM	8GB + 2GB (HONOR RAM Turbo)
Storage	256GB
Display	6.67 inches OLED
Rear Camera	54 MP
Front Camera	32 MP

Table 5.1.2 Specifications of Mobile Device

Next, an Apple iPad Pro 2021 tablet is initially used to measure the dimensions of the FICT building using the built-in Measure app. The app utilises the tablet's LiDAR sensor to detect depth and accurately calculate real-world dimensions.

Description	Specifications
Model	Apple iPad Pro 2021
Processor	Apple M1
Operating System	iPadOS 18.4
RAM	16GB
Storage	1TB
Display	11.0 inches, Liquid Retina IPS LCD
Rear Camera	12 MP
Front Camera	12 MP

Table 5.1.3 Specifications of Tablet

Lastly, a laser distance meter is used as the final measuring device to accurately measure the dimensions of the FICT building.

Description	Specifications
Model	Mileseey S2 Laser Distance Meter
Measuring Range	100M
Measuring Accuracy	±2mm
Measuring Unit	m/in/ft /ft+in

Table 5.1.4 Specifications of Laser Distance Meter

5.2 Software Setup

This section covers the software setup, including Unity for development, ZXing for QR code scanning, Visual Studio Code for scripting, and ARCore for AR functionality on Android devices.

5.2.1 Unity and its packages

The application is developed using Unity version 2022.3.59f1. Several major Unity packages are integrated to provide AR functionality, navigation, UI enhancements, and an efficient development workflow. These major packages automatically install their required dependency packages, ensuring smooth operation and integration. Table 5.2.1.1 lists the major packages, their versions, and descriptions.

Unity Version	2022.3.59f1	
Description	A cross-platform game engine used to develop 2D, 3D, AR, and VR applications, providing tools for scene building, scripting, animation, and real-time rendering.	
Package	**Version**	**Description**
AI Navigation	1.1.5	Provides high-level NavMesh components for building and using NavMeshes at runtime and edit time.
AR Foundation	5.1.6	Enables the creation of multi-platform augmented reality (AR) applications within Unity.
Google ARCore XR Plugin	5.1.6	Provides native Google ARCore integration for use with Unity's multi-platform XR API.
TextMeshPro	3.0.7	Offers advanced text rendering and layout features, replacing default UI Text and legacy Text Mesh.
ProBuilder	5.2.4	Allows building, editing, and texturing custom geometry within Unity, useful for in-scene level design and prototyping.
UI Rounded Corners	3.5.0	Provides components and shaders to add rounded corners to UI elements.
Visual Studio Code Editor	1.2.5	Integrates Visual Studio Code as the code editor for Unity, supporting an efficient scripting workflow.

Table 5.2.1.1 Key Unity Packages for AR Application Development

These major packages formed the foundation of the development environment and facilitated the creation of a fully functional AR navigation system with customised UI, scene modelling, and real-time AR interaction.

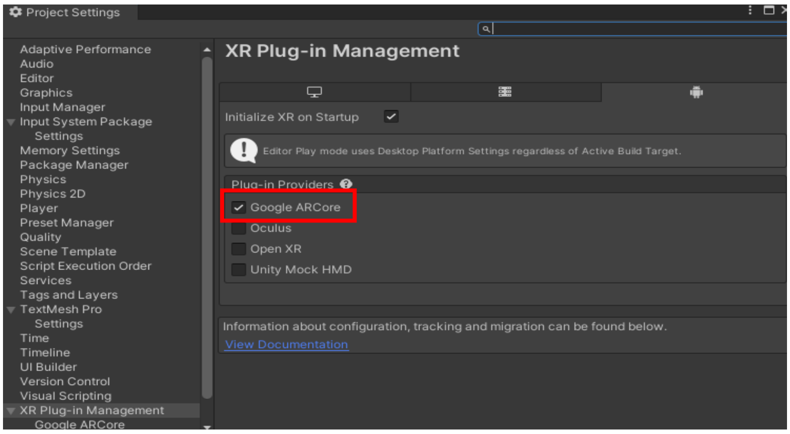


Figure 5.2.1.1 Google ARCore as Plug-in Providers

Google ARCore was selected as the plug-in provider to enable AR functionality for Android devices. This was done by opening Project Settings in Unity, going to XR Plug-in Management under Android, and enabling Google ARCore. This setup allows the AR-based indoor navigation application to utilise ARCore's augmented reality features on Android devices.

5.2.2 ZXing.Net

ZXing.Net is a library used for decoding and generating various types of barcodes, including QR Code, PDF 417, EAN, UPC, Aztec, Data Matrix, and Codabar. It is integrated into the project to enable scanning and processing of QR codes within the AR-based indoor navigation application.

Library	ZXing.Net
Version	0.16.8
Github Repo	https://github.com/micjahn/ZXing.Net
Description	Supports decoding and generating barcodes within images, enabling QR code scanning functionality.

Table 5.2.2.1 ZXing.Net Installation

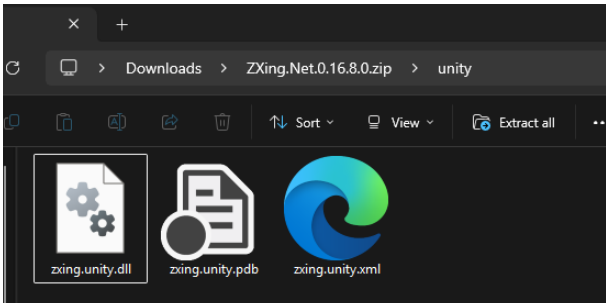


Figure 5.2.2.1 Unzip ZXing

After downloading the ZXing.Net library as a ZIP file from the GitHub repository, the file is unzipped, and the contents are navigated to the GitHub directory. All the relevant files inside this directory are then copied to be added to the Unity project. This step ensures that the library's core files are available for integration.

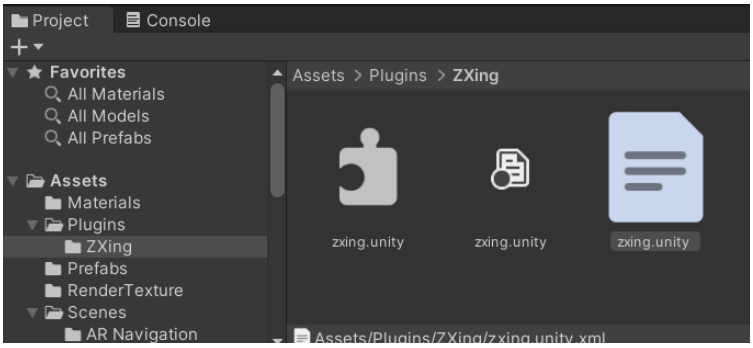


Figure 5.2.2.2 Paste the Files into Unity's Plugin Folder

The copied files are then pasted into Unity's Plugins folder, which is specifically created and named ZXing. This folder allows Unity to recognise and utilise the ZXing.Net library, enabling the application to decode and generate QR codes during runtime.

5.2.3 Integrated Development Environment (IDE)

The application development is carried out using Visual Studio Code (VS Code) as the Integrated Development Environment (IDE). VS Code provides a lightweight, cross-platform environment suitable for C# development and Unity integration. Several extensions are installed to enhance the C# development workflow and support Unity project management.

Version	1.103.2 (user setup)	
Extension	**Version**	**Description**

C#	2.87.31	Provides rich language support for C# and is shipped with C# Dev Kit.
C# Dev Kit	1.41.11	Helps manage code with a solution explorer and integrated unit testing across platforms.
IntelliCode for C# Dev Kit	2.2.3	Offers AI-assisted development features with contextual code suggestions using machine learning.
.NET Install Tool	2.3.7	Provides a unified way to install local .NET Runtime and SDK versions required by other extensions.
Unity	1.1.3	Offers a streamlined Unity development experience for C# developers in VS Code.

Table 5.2.3.1 VS Code Extensions Used for Development

In addition, the Microsoft .NET SDK version 9.0 is installed to provide a C# runtime and development tools required by VS Code extensions and project compilation. This SDK ensures compatibility with C# 9.0 language features used in the project.

SDK Name	Microsoft .NET SDK
Version	9.0
Link	https://dotnet.microsoft.com/en-us/download
Description	Provides the .NET runtime and development tools needed to compile and run C# projects, including Unity scripts.

Table 5.2.3.2 Microsoft .NET SDK Installation

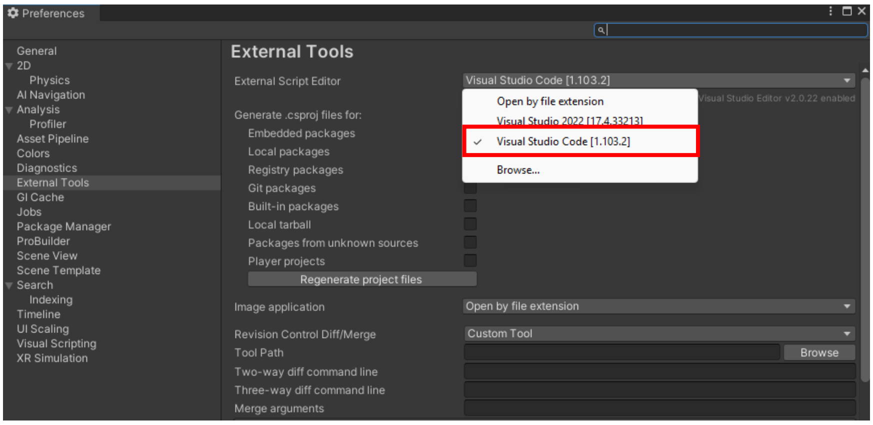


Figure 5.2.3.1 Unity's External Script Editor

In Unity, the external script editor is configured by opening the Preferences menu, selecting External Tools, and choosing Visual Studio Code, as shown in Figure 5.2.3.1. This setup links Unity with VS Code, allowing C# scripts to open directly in the IDE for editing, debugging, and efficient project management.

5.2.4 ARCore

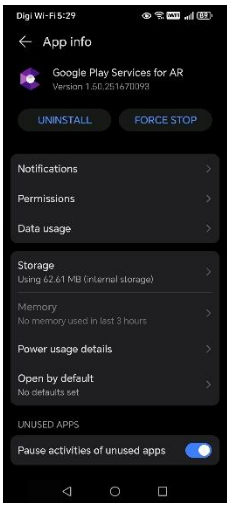


Figure 5.2.4.1 ARCore

Figure 5.2.4.1 shows the installation of the Google Play Services for AR app on an Android device. This service is required for ARCore functionality, allowing the AR-based indoor navigation application to perform augmented reality features such as motion tracking, environmental understanding, and surface detection, ensuring accurate placement of digital content within the real-world environment.

5.3 Project Implementation

This section describes the implementation of the ARFICT system, including building modelling in Unity, setting up target locations and recenter points, designing the user interface, generating QR codes for localisation, and integrating voice guidance for navigation.

5.3.1 FICT Building Modelling in Unity

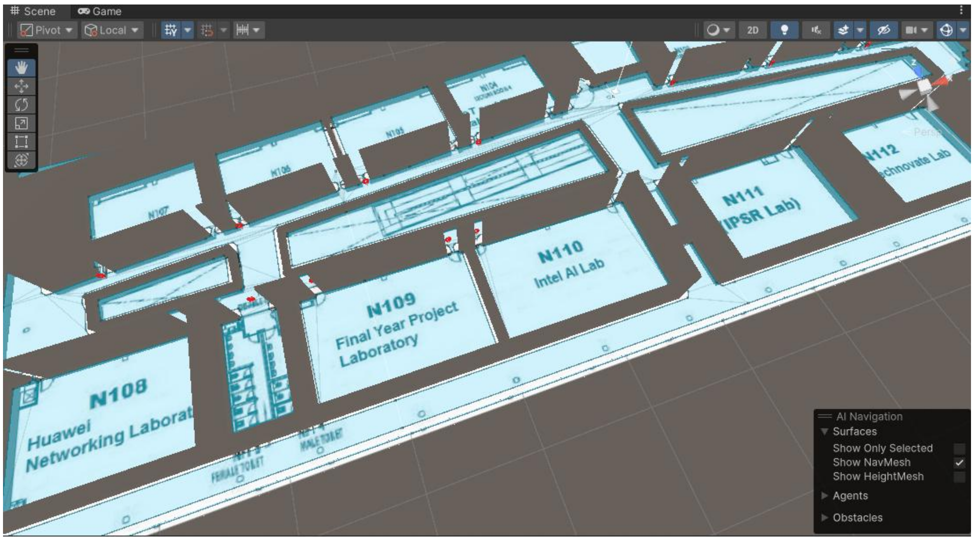


Figure 5.3.1.1 FICT Building Model

The FICT building model is developed in Unity with a scale of 101 units in length and 36 units in width for each floor. The ground floor height is set to 4.4 meters. Measurements are obtained using a laser distance meter to ensure high precision and accuracy, enabling the model to closely align with the real-world environment. Since Unity units correspond directly to meters, the measured dimensions are applied consistently during modelling.

The floor plan is sourced from the FICT official website, which simplifies the modelling process. During development, walls were positioned to serve as obstacles for baking the navigation mesh (NavMesh). This NavMesh provides the underlying graph for pathfinding, allowing the system to use the A* algorithm to find the shortest possible paths for users. Staircases connecting the ground and first floors were also constructed, ensuring the NavMesh correctly links levels and enables seamless multi-level navigation.

5.3.2 Target Location Setup

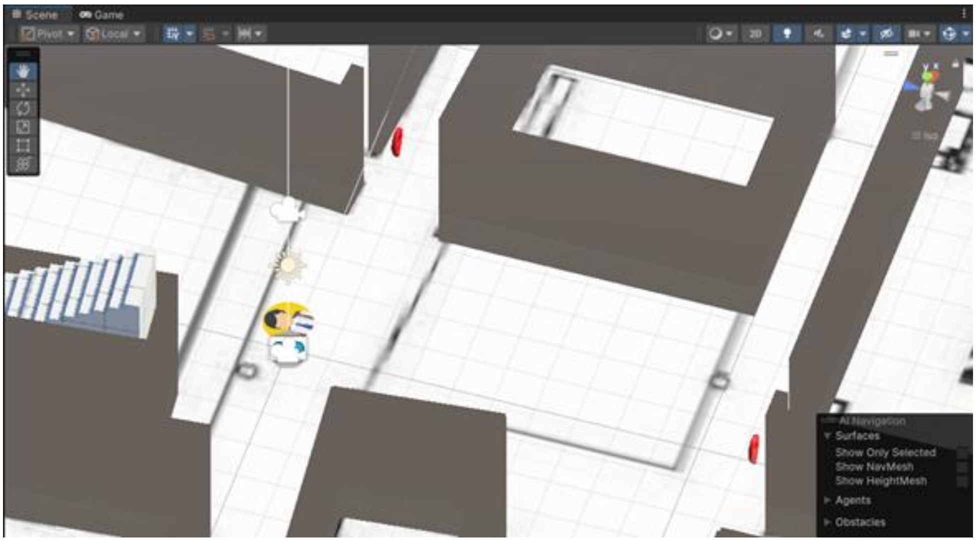


Figure 5.3.2.1 Target Setup

An indicator target is created using a canvas icon to represent the user’s current location in the building. This indicator is also synchronised with the mini map so that users can view their position both in the AR environment and on the simplified floor plan. This dual representation improves spatial awareness and helps users track their movement more effectively.

Each destination is marked with a red 3D map pointer that serves as the navigation destination. When a user selects a destination, the system highlights the corresponding marker and generates the navigation path towards it. These destination markers make it easy for users to identify rooms quickly within the AR interface.

In addition, several recenter targets are strategically placed at key locations throughout the building. These targets are linked with QR codes that can be scanned to correct the user’s position in case of AR tracking drift. The application ensures that the virtual navigation path remains accurately aligned with the real-world environment by recentring through QR code scanning

5.3.3 Application User Interface

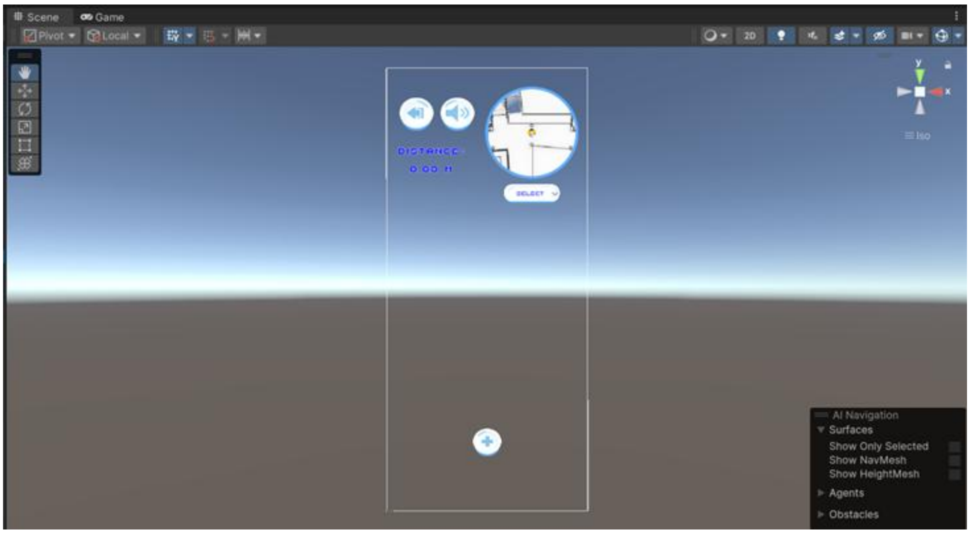


Figure 5.3.3.1 ARFICT's User Interface

ARFICT application user interface (UI) is developed in Unity to provide an interactive and user-friendly experience. UI elements such as buttons, panels, and the mini map are designed using Unity's canvas system, which allows them to be layered seamlessly on top of the augmented reality view. Each clickable object, such as navigation buttons and the room selection dropdown menu, is linked to Unity scripts to trigger corresponding functions, including starting navigation, switching between navigation modes, or scanning QR codes.

The mini map is integrated into the interface to display the user's current location and the selected destination, enhancing spatial awareness within the building. Buttons are included to allow users to toggle navigation modes, enable or disable background music, and adjust navigation line height as needed. The design prioritises simplicity and clarity, ensuring that users can interact with the application intuitively while navigating the building in real time.

5.3.4 QR Code Generator

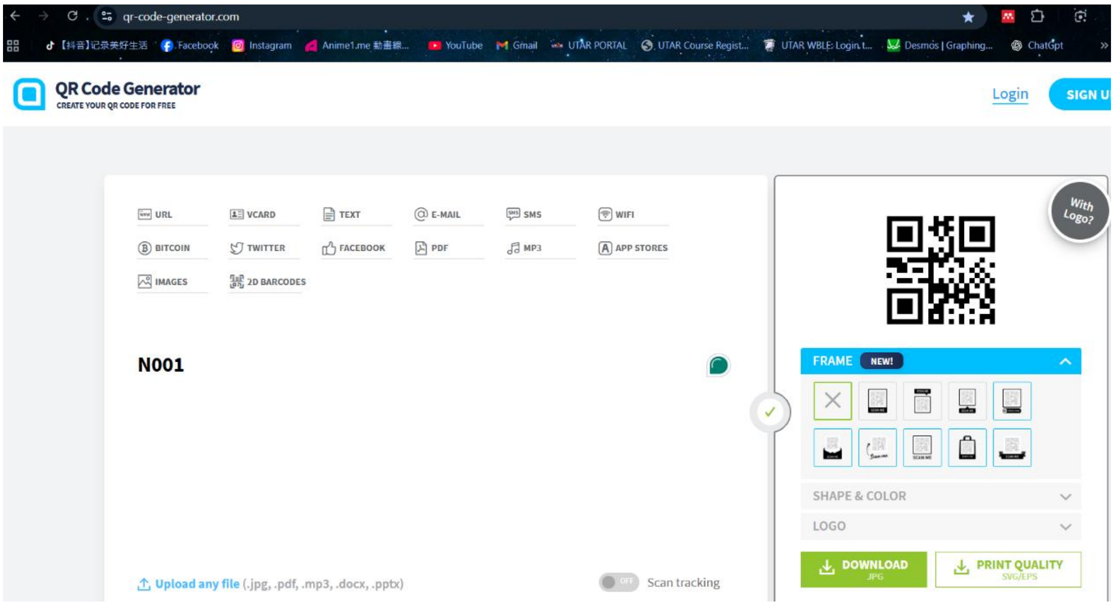


Figure 5.3.4.1 QR Code Generator Website

A free QR code generator website is used to create the QR codes required for localisation within the ARFICT application. Each QR code contains plaintext data that corresponds to a specific target object name in the Unity project. When the user scans a QR code, the application decodes the plaintext and matches it with the predefined target. This process allows the system to accurately reposition the user’s current location in the virtual environment, ensuring proper alignment with the physical building layout.

The use of QR codes helps address potential issues with AR tracking drift by providing fixed reference points for recentring. The user can recalibrate their position by scanning a QR code placed at key locations, which improves both navigation accuracy and the overall user experience.

5.3.5 Navigation Voice Generator

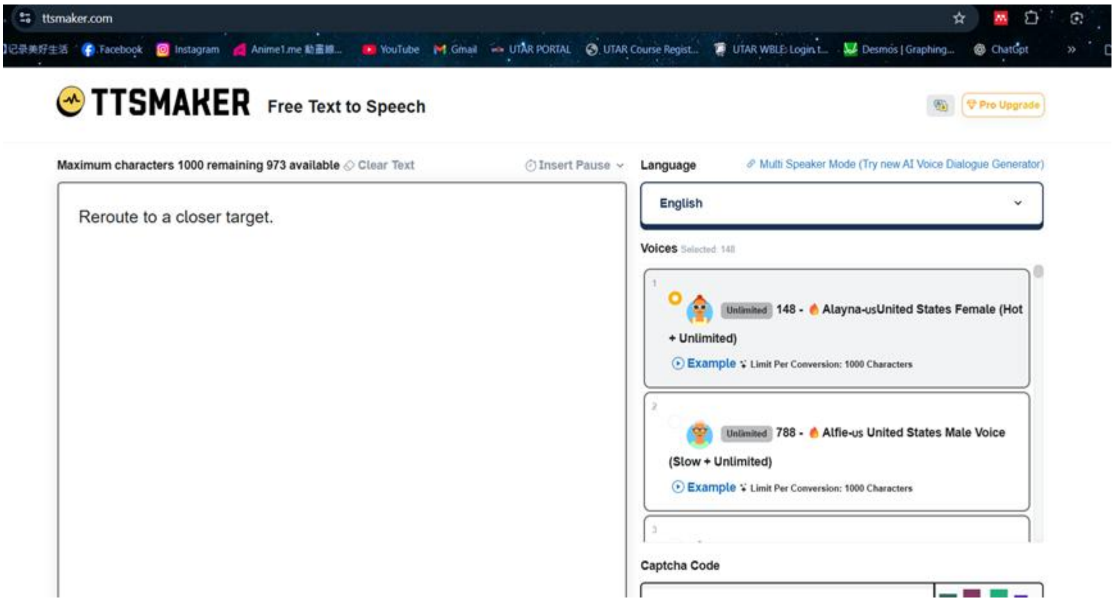


Figure 5.3.5.1 Text-to-Speech Generator Website

The website TTSMAKER is used to generate navigation voice instructions through its free text-to-speech feature. Text inputs, such as navigation directions, are converted into audio files, which are then integrated into the application. These audio clips are assigned to objects within Unity and linked to scripts that trigger playback when specific conditions are met, such as reaching a destination or making a turn.

This approach ensures that users receive real-time voice guidance during navigation, enhancing accessibility and providing a more intuitive experience compared to relying solely on visual cues.

5.4 System Operations as Graphical User Interface

This section outlines the main system operations of the ARFICT application through its graphical user interface. It explains how users interact with features such as the main menu, navigation options, QR code scanning, destination selection, map viewing, and other interactive functions that support the indoor navigation process.

5.4.1 Main Menu

Figure 5.4.1.1 Main Menu

The main menu is displayed when the ARFICT application is launched. It provides users with three primary feature buttons that serve as entry points to different functions of the system. The Start Navigation button directs users to the navigation screen, which is the core feature of the application. The View Map button opens a page displaying the floor plans of the FICT building, allowing users to explore the layout before navigating. The How to Navigate button leads to a tutorial page that guides users on how to use the navigation screen and its features effectively.

5.4.2 Start Navigation

Figure 5.4.2.1 Start Navigation

When the user enters the Start Navigation screen, the system requests permission to access the device camera for augmented reality functionality. The interface displays key elements such as buttons, a dropdown menu, and a mini map. The top-left corner contains a quit button to return to the main menu and a speaker icon button to toggle navigation voice guidance. On the top-right, a mini map displays the user's current location on the floor plan in real time, with a dropdown menu positioned below it that lists the available destinations. At the bottom centre, a feature button is placed which expands into a panel containing additional navigation functions.

5.4.3 QR Code Scanner

Figure 5.4.3.1 QR Code Scanning Panel

The feature panel includes four buttons that support the navigation process. One of these is the Scan QR Code button, which activates the scanning panel to capture a QR code for localisation, as shown in Figure 5.4.3.1. This feature is used when the application is first launched and the user needs to reposition their location before starting navigation, or when the user’s position becomes inaccurate or lost due to extended usage.

Figure 5.4.3.2 QR Code Scanned Success

Figure 5.4.3.3 Invalid QR Code

If the scanned QR code matches a recenter target object's name, the application recentres the user's position in the building, plays a confirmation sound, and displays the message "QR Code Scanned Success". If the QR code is invalid, an error sound is played, and a failure message is shown to the user.

5.4.4 Destination Selection

Figure 5.4.4.1 List of Destination Selection

Figure 5.4.4.2 Navigation Started

Once localisation is completed, the user can click the dropdown menu located below the mini map to display a list of available destinations, as shown in Figure 5.4.4.1. After selecting a destination, the system plays a confirmation sound and displays the message "Navigating to Target" to inform the user. At the same time, a navigation line and the destination's target object are activated both on the screen and on the mini map by default, guiding the user towards the chosen location, as illustrated in Figure 5.4.4.2. The shortest route to the target is calculated using the A* algorithm on the navigation mesh, ensuring an efficient and accurate path within the multi-floor building.

During the navigation process, the system provides voice guidance to support the user's movement through the building. The navigation voice instructs the user to turn left, turn right, continue straight, or make a U-turn depending on the user's current position, orientation, and the planned path. The combination of A*-based pathfinding, visual guidance on the screen and mini map, together with real-time auditory instructions, offers a more accessible and intuitive navigation experience by reducing disorientation and improving usability.

5.4.5 Navigation Visibility

Figure 5.4.5.1 Navigation Feature Off

Figure 5.4.5.2 Navigation Feature On

The Toggle Navigation button allows the user to switch the navigation features on or off. When switched off, it hides the distance to the destination and the navigation path that overlays on both the screen and the mini map. When switched on, these features are reactivated. In both cases, the system provides feedback by displaying a message and playing a sound to indicate the action performed.

5.4.6 Navigation Mode

Figure 5.4.6.1 Arrow Mode Navigation

Figure 5.4.6.2 Line Mode Navigation

The Switch Navigation Mode button allows the user to change how navigation guidance is presented according to their preference. The line mode displays a path overlaid with arrows leading towards the destination by default. Alternatively, the arrow mode places a single directional arrow in front of the screen that points the way forward. When the button is clicked, the system provides feedback by showing a message indicating the current navigation mode and playing a confirmation sound. Both modes offer users a different style of guidance, enhancing flexibility and user experience.

5.4.7 View Mini Map

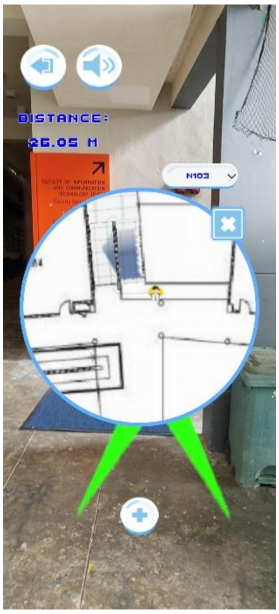


Figure 5.4.7.1 View Mini Map

The user can click the mini map to enlarge it at the centre of the screen for a clearer and more detailed view of the building layout. The user can interact with the mini map by dragging to adjust the visible area or by zooming in and out to better understand their surroundings and navigation path. This feature allows users to check their overall route or explore alternative directions more easily. Once finished, clicking the close button restores the mini map to its original size and position on the screen, ensuring it remains available as a quick reference during navigation.

5.4.8 Adjust Line Height

Figure 5.4.8.1 Turn On Adjustable Slider

The Adjust Line Height button allows the user to modify the vertical position of the navigation path during navigation. This feature is useful when the path appears too high or too low on the screen after extended use, making it less convenient to follow. When the button is clicked, a slider appears, as shown in Figure 5.4.8.1, accompanied by a sound and a confirmation message.

Figure 5.4.8.2 Adjust Line Height

Figure 5.4.8.3 Turn Off Slider

The user can then drag the slider handle to adjust the navigation path's height in real time, as shown in Figure 5.4.8.2, ensuring a more comfortable viewing experience. Once the adjustment is complete, clicking the button again hides the slider, with another sound and message provided as feedback.

5.4.9 Reroute to a Closer Target

Figure 5.4.9.1 Navigating to Toilet

Figure 5.4.9.2 Reroute Target

If the selected destination has multiple targets with the same name, such as toilets, the system continuously calculates and compares the distance between the user's current location and each target. The system ensures that the shortest and most efficient path is always selected using the A* algorithm on the navigation mesh. When a closer target is detected during navigation, the system automatically updates the navigation path and reroutes the user to the nearest location. A notification message and sound are triggered to inform the user that a reroute has been performed, as shown in Figure 5.4.9.2, to avoid confusion

5.4.10 Arrive at Destination

Figure 5.4.10.1 Arrive at Destination

The user follows the navigation path until reaching the destination. The arrival process is triggered when the system detects that the user is within 1.5 metres of the target location. At this point, a message dialogue appears on the screen, accompanied by a voice prompt, to clearly indicate that the user has arrived. During this stage, no further actions can be performed until the user chooses either to continue or to end the navigation process. Selecting the continue button allows the user to carry on navigating or access other features, while selecting the end button clears and resets the navigation data, requiring the user to select a new destination if they wish to start another navigation. This process ensures that the navigation session is properly concluded, preventing errors and maintaining a smooth user experience.

5.4.11 View Map

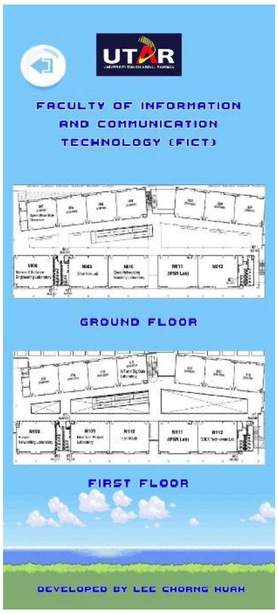


Figure 5.4.11.1 View Map (Floor Plan)

The View Map button in the main menu opens a page displaying the floor plans of the FICT building, including both the ground and first floors. This feature serves as a static reference to help users understand the building layout before or during navigation.

5.4.12 View How to Play

Figure 5.4.12.1 First Tutorial

Figure 5.4.12.2 Second Tutorial

The View How to Play button takes the user to the tutorial page, which contains four step-by-step guides on how to navigate within the app. The user can move through the tutorials by clicking the buttons below the images to go forward or return to the previous step.

Figure 5.4.12.3 Third Tutorial

Figure 5.4.12.4 Last Tutorial

Once the final tutorial is reached, a Start button appears instead of the Next button, allowing the user to proceed directly to the Start Navigation screen.

5.5 Implementation Issues and Challenges

One of the main challenges faced in the implementation of the system is obtaining accurate dimensions of the multi-floor FICT building. This requires multiple rounds of measurement to capture precise values for length, width, and height using a laser distance meter. The collected data is then used to construct the building model in Unity, which must be carefully aligned with the real-world environment. The model undergoes repeated simulation in Unity as well as real-world testing to ensure accuracy. This process is challenging and time-consuming, as even small discrepancies in measurement can affect the navigation accuracy and overall reliability of the application.

Another challenge encountered is the placement of QR code recenter points in the virtual environment and their correspondence to real-world locations. The localisation process depends heavily on the QR code's physical placement and the angle at which it is scanned by the user. Since scanning positions and angles vary each time, inconsistencies in alignment can sometimes result in inaccurate localisation. Careful adjustment and repeated real-world testing are required to ensure that the QR codes function correctly and reliably within the navigation system.

The third challenge relates to the scripting of navigation voice directions. The system is designed to play specific voice prompts, such as "turn left" or "turn right," based on the user's orientation and the angle of movement relative to the navigation path. However, managing the angle thresholds with precision is difficult, as inaccurate scripting can cause the wrong direction to be triggered. This issue requires extensive testing in real-world scenarios using mobile devices to fine-tune the conditions and ensure that the navigation voice consistently provides correct and helpful guidance.

5.6 Concluding Remark

In summary, the implementation of the ARFICT indoor navigation system highlights the practical integration of augmented reality, indoor positioning, and user interface design into a single mobile application. The system is developed through Unity and ARCore, supported by additional components such as QR code localisation, audio guidance, and interactive controls that collectively enhance the navigation experience. The development process involved not only creating a virtual model of the multi-floor FICT building but also ensuring that it aligned accurately with the real-world environment through repeated measurement, testing, and refinement. Each functional module, including destination selection, mini map interaction, navigation path display, and voice guidance, is carefully integrated to provide a smooth and intuitive user experience.

At the same time, several challenges are encountered during implementation, such as managing building dimensions, optimising QR code placement, and ensuring precise triggering of navigation instructions. These issues are systematically addressed through iterative testing, adjustment of parameters, and real-world validation using mobile devices. As a result, the system achieves a balance between technical feasibility and user usability, ensuring that the navigation process remains reliable and accurate in practical use. Overall, this chapter demonstrates how the conceptual design of ARFICT is transformed into a working prototype, laying a strong foundation for future improvements and potential large-scale deployment.

CHAPTER 6

System Evaluation and Discussion

This chapter evaluates ARFICT by examining its functionality, performance, and ability to achieve the intended objectives. It presents testing results, discusses challenges and solutions, and compares advancements over the previous app to show how ARFICT delivers accurate, reliable, and seamless multi-floor indoor navigation within the FICT building.

6.1 Testing Setup and Result

This section presents the testing process carried out to validate the functionality of the AR-based indoor navigation application. Each feature are tested systematically to ensure it operated as expected under defined conditions. The results confirm whether the application met its functional requirements and provided a smooth user experience.

6.1.1 Navigation Features Test Case

This subsection tests the core navigation functions, including QR code scanning, destination selection, toggling and switching navigation modes, adjusting line height, rerouting, mini map interaction, and sound guidance. It also checks arrival handling and navigation termination. All features are validated to work correctly, ensuring accurate and reliable real-time guidance.

<table>
<tr><td colspan="2">Module</td><td colspan="4">TC01 – Navigation</td></tr>
<tr><td colspan="2">Related Use Case ID</td><td colspan="4">UC01 – Start Navigation</td></tr>
<tr><td colspan="2">Objective</td><td colspan="4">To ensure all navigation features operate correctly for a smooth user experience.</td></tr>
<tr><td colspan="2">Preconditions</td><td colspan="4">• User selects Start Navigation from the main menu.
• Camera permission is granted.
• Device sensors (camera, gyroscope, accelerometer) are active.</td></tr>
<tr><td colspan="6">Test Case and Result</td></tr>
<tr><td>ID</td><td>Description</td><td>Test Steps</td><td>Expected Result</td><td>Actual Result</td><td>Status</td></tr>
<tr><td colspan="2">Test Feature</td><td colspan="4">Scan QR Code</td></tr>
</table>

TC0101	Scan a valid QR code	1. Tap Scan QR Code. 2. Scan a valid QR code.	System updates user's location with success message and voice.	As expected	Pass
TC0102	Scan an invalid QR code	1. Tap Scan QR code. 2. Scan an invalid QR code.	System shows error message with voice.	As expected	Pass
Test Feature		Select Destination			
TC0103	Select a destination	1. Tap Select dropdown menu. 2. Choose a destination.	System displays navigation path with confirmation message and voice	As expected	Pass
Test Feature		Toggle Navigation			
TC0104	Turn on navigation feature	1. Navigation is off. 2. Tap Toggle Navigation.	Navigation path and distance label are displayed.	As expected	Pass
TC0105	Turn off navigation feature	1. Navigation is on. 2. Tap Toggle Navigation.	Navigation path and distance label are hidden.	As expected	Pass
Test Feature		Switch Navigation Mode			
TC0106	Switch to arrow path	1. Path is in line mode. 2. Tap Switch Navigation Mode.	Path changes from line to arrow.	As expected	Pass
TC0107	Switch to line path	1. Path is in arrow mode.	Path changes from arrow to line.	As expected	Pass

		2. Tap Switch Navigation Mode.			
Test Feature		Adjust Line Height			
TC0108	Turn on slider	1. Slider is off. 2. Tap Adjust Line Height.	Slider is displayed.	As expected	Pass
TC0109	Adjust line height	1. Drag slider handle.	Path height changes according to slider value.	As expected	Pass
TC0110	Turn off slider	1. Slider is on. 2. Tap Adjust Line Height.	Slider is hidden.	As expected	Pass
Test Feature		Reroute Navigation Path			
TC0111	Reroute to a closer destination	1. Select a destination with multiple targets. 2. Walk away from the original path.	System automatically reroutes the user to the closer destination with a voice prompt and message.	As expected	Pass
Test Feature		View Mini Map			
TC0112	Enlarge mini map	1. Tap mini map.	Mini map is enlarged and centered on screen.	As expected	Pass
TC0113	Move mini map	1. Drag mini map	Mini map moves according to drag gesture.	As expected	Pass
TC0114	Zoom in/out mini map	1. Pinch mini map.	Mini map zooms in or out based on gesture.	As expected	Pass

TC0115	Close mni map	1. Tap close button.	Mini map is restored to original size and position.	As expected	Pass
Test Feature		Toggle Sound			
TC0116	Turn on sound	1. Tap speaker off button.	Sound is enabled and icon changes to speaker on.	As expected	Pass
TC0117	Turn off sound	1. Tap speaker on button.	Sound is disabled and icon changes to speaker off.	As expected	Pass
Test Feature		Navigation Sound Guidance			
TC0118	Continue straight guidance	1. Walk along straight path.	System plays "Continue straight" voice.	As expected	Pass
TC0119	Turn left guidance	1. Approach a left turn.	System plays “Turn left” voice.	As expected	Pass
TC0120	Turn left guidance	1. Approach a right turn.	System plays “Turn right” voice.	As expected	Pass
TC0121	Make U-turn guidance	1. Face 180° opposite from destination.	System plays “Make a U-turn” voice.	As expected	Pass
Test Feature		Reach Destination			
TC0122	Reach destination	1. Follow the navigation path. 2. Reach destination within 1.5m.	System displays arrival dialog and plays arrival voice.	As expected	Pass
Test Feature		Continue/End Navigation			
TC0123	Continue navigation	1. Tap Continue on dialog.	Dialog closes and navigation data remain active.	As expected	Pass

TC0124	End navigation	1. Tap End on dialog.	Dialog closes and all navigation data are cleared.	As expected	Pass

Table 6.1.1.1 Navigation Features Test Case

6.1.2 View How to Play Feature Test Case

This subsection verifies the tutorial function, ensuring users can move between tutorial pages and start navigation from the final page. Results confirmed the feature works smoothly and effectively guides users before navigation.

Module	TC02 – View Tutorials				
Related Use Case ID	UC02 – View How to Play				
Objective	To verify that users can navigate tutorial pages correctly and start navigation from the final tutorial.				
Preconditions	User selects How to Play from the main menu.				
Test Case and Result					
ID	**Description**	**Test Steps**	**Expected Result**	**Actual Result**	**Status**
Test Feature	Navigate between tutorials				
TC0201	View next tutorial	1. Tap next button.	System proceeds to the next tutorial page.	As expected	Pass
TC0202	View previous tutorial	1. Tap previous button.	System back to the previous tutorial page.	As expected	Pass
TC0203	Start navigation	1. Tap start button (last tutorial).	System navigates to the Start Navigation screen.	As expected	Pass

Table 6.1.2.1 View Tutorials Feature Test Case

6.2 Project Challenges

One of the major challenges faced during the project was AR tracking drift. The sensors used for localisation in the AR environment often experienced drift, which led to misalignment between the virtual navigation path and the actual surroundings. A QR code recentering technique was implemented to address this issue, which allows the system to recalibrate and maintain accurate positioning during navigation.

Another challenge was ensuring building modelling accuracy in Unity. Initially, the 3D model of the FICT building was developed based on the available floor plans, but this approach resulted in scaling errors and misaligned dimensions. A laser distance meter was later used to obtain precise measurements, which significantly improved the accuracy and reliability of the 3D model.

A further challenge involved navigation across multiple floors. Designing a navigation mesh (NavMesh) capable of handling staircases and inter-floor transitions was complex. A NavMesh is a data structure in Unity that defines the walkable surfaces of a 3D environment, simplifying the geometry into polygons for efficient pathfinding. In this project, the NavMesh determined where users could move, including hallways, open spaces, and stairs. Configuring the NavMesh for multiple floors required careful setup to ensure smooth connections between levels. Multiple rounds of testing and adjustments were carried out to achieve seamless and reliable pathfinding across both floors of the building.

In summary, the project faced challenges related to AR tracking drift, building modelling accuracy, and multi-floor navigation. QR code recentering with continuous testing of placement was implemented to maintain accurate localisation, precise measurements using a laser distance meter improved the 3D building model, and careful NavMesh configuration enabled smooth inter-floor navigation. These solutions effectively addressed the issues, resulting in a more accurate and reliable AR navigation system.

6.3 Advancements of ARFICT over Previous FICT Indoor Navigation App

This subsection presents a comparison between the previous AR Indoor Navigation App for FICT, developed by Goh Brian Joon Jian, and the ARFICT application. The purpose of this comparison is to highlight the improvements and advancements introduced in ARFICT. Both applications adopt marker-based navigation with QR codes and mobile device sensors, but ARFICT extends the functionality by providing additional features that enhance usability, accuracy, and user experience. These include multi-floor navigation, audio navigation support, a more interactive mini map, multiple navigation modes, and advanced features such as rerouting and distance calculation. Table 6.3.1 summarises the differences between the two applications.

System / Feature	FICT AR Indoor Navigation App by Goh Brian Joon Jian	ARFICT
AR	Yes	Yes
Indoor Navigation Technique	Marker-based using QR codes by using mobile device sensors to track user movement during navigation.	Marker-based using QR codes by using mobile device sensors to track user movement during navigation.
Real-time Navigation	Yes	Yes
Audio Navigation Support	No	Yes
Multi-Floor Navigation Support	No, navigation is limited to a single floor and only functions along the same y-axis.	Yes, supports multi-floor navigation with interconnected floors in the Unity building model.
Mini Map	Yes, but only supports single-floor maps without support for gestures to move across the map or zoom in/out.	Yes, supports multi-floor maps for better visualisation and navigation, with user gestures such as swiping to move across the map and pinch-to-zoom for zooming in/out.
Accuracy	Low	High
Adjustable Line Height	Yes	Yes
Navigation Mode	Line only	Line, arrow
Distance Calculation Label	No	Yes
Reroute to a Closer Destination	No	Yes

Table 6.3.1 Comparison between previous FICT AR Indoor Navigation App and ARFICT

6.3.1 Similarities and Shared Features

This subsection highlights the common features found in both applications, such as the use of AR technology, marker-based indoor navigation with QR codes, real-time navigation, adjustable line height and mini map. While these features are shared, ARFICT enhances them with a more user-friendly interface, including action confirmation messages and voice cues.

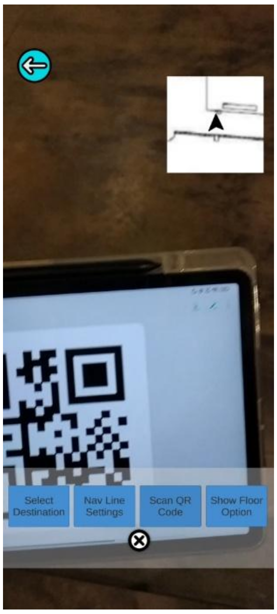


Figure 6.3.1.1 Previous App Scanned QR

Figure 6.3.1.2 ARFICT Scanned QR

In the previous application, the QR code scanning feature lacked clarity, as it did not inform the user whether the scan was successful or unsuccessful, as shown in Figure 6.3.1.1. In contrast, ARFICT validates the scanned QR code and provides feedback through confirmation messages and audio cues, as shown in Figure 6.3.1.2. This improvement is important as it enhances usability by reducing user uncertainty and improving navigation reliability.

Figure 6.3.1.3 Previous App Height Slider

Figure 6.3.1.4 ARFICT Height Slider

Both applications include an adjustable navigation line height slider. However, in the previous app, the slider was positioned at the top left corner, as shown in Figure 6.3.1.3, which was less user-friendly as users had to move their finger upward to make adjustments. ARFICT addressed this issue by relocating the slider to the bottom right of the interface for easier accessibility, as shown in Figure 6.3.1.4. Additionally, the activation of the slider is enhanced with confirmation messages and audio cues, further improving user interaction.

Figure 6.3.1.5 Arrive at Destination

In terms of user interface, ARFICT offers a more user-friendly and interactive experience through its improved design. One of the key enhancements is that ARFICT notifies users once they have reached their destination, as shown in Figure 5.4.10.1, a feature absent in the previous app. This notification is provided through both visual confirmation and audio cues, ensuring that users are clearly informed without needing to constantly monitor their screen. Such feedback not only enhances usability but also improves accessibility, particularly for users who may be navigating while multitasking or focusing on their surroundings.

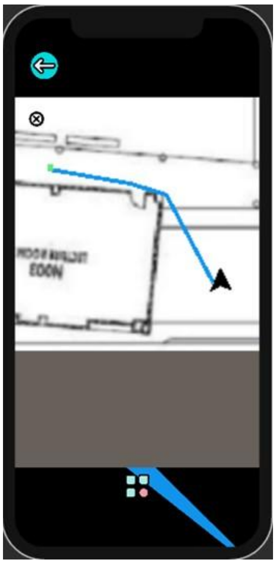


Figure 6.3.1.6 Previous App's Mini Map

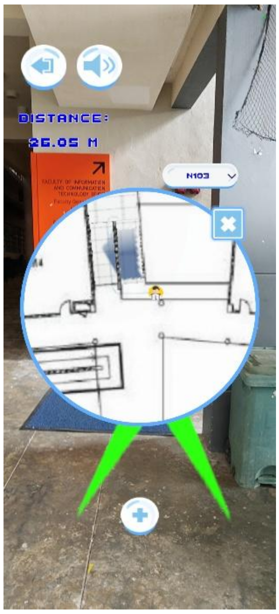


Figure 6.3.1.7 ARFICT's Mini Map

In ARFICT, the mini map design was enhanced in terms of user interface compared to the previous application. The mini map now supports multi-floor navigation, allowing users to move seamlessly between two building levels without needing to rescan a QR code to switch the virtual environment and floor plan, as required in the earlier app. Additionally, the enlarged mini map in ARFICT provides improved interaction by supporting user gestures such as dragging to move across the map and zooming in or out, features that were not available in the previous application.

6.3.2 Accuracy Improvements

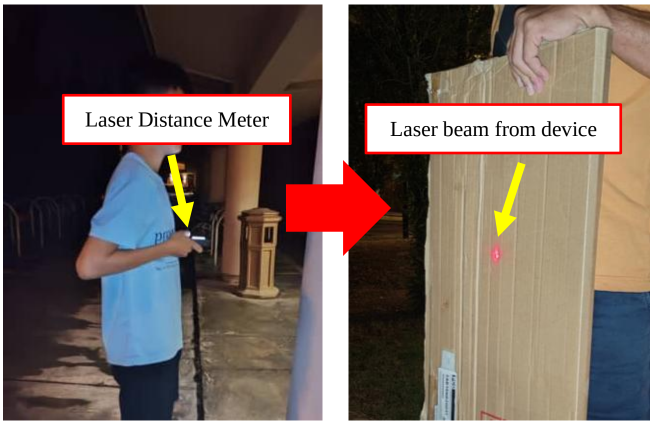


Figure 6.3.2.1 ARFICT's Measuring Technique

The accuracy of the navigation in the ARFICT app has significantly improved, allowing users to be guided precisely to their intended destinations. This improvement is mainly attributed to the reliable measuring tools and techniques employed during the development phase. Specifically, a laser distance meter with a measurement accuracy of ±2 mm was used to capture the building's dimensions, as shown in Figure 6.3.2.1.

This high level of accuracy means that any recorded measurement could only deviate by a maximum of 2 mm from the actual value, which is negligible in the context of architectural measurements. As a result, the building model constructed in Unity is based on true-to-scale (1:1) dimensions, ensuring that the navigation path generated in the AR environment closely aligns with the real-world layout of the FICT building. This accurate modelling directly contributes to a smoother navigation experience, where virtual guidance matches the physical space without noticeable discrepancies.

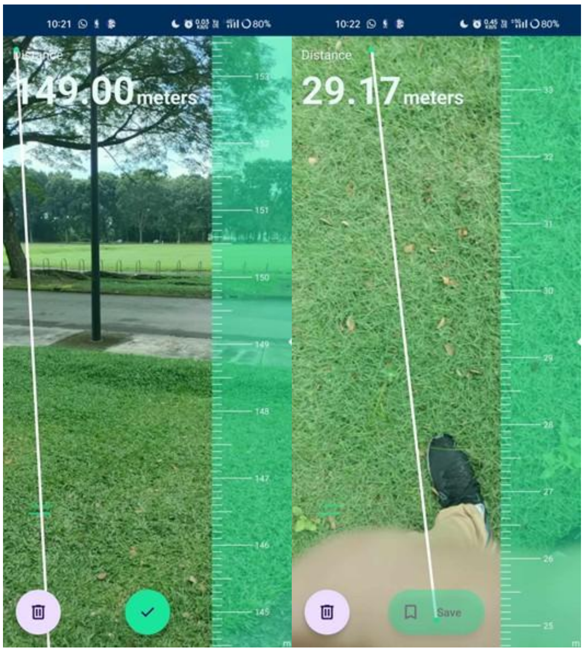


Figure 6.3.2.2 Previous Project's Measuring Technique

In contrast, the previous indoor navigation app suffered from lower accuracy due to its reliance on a mobile device camera with an AR ruler tool, as shown in Figure 6.3.2.2. This method introduced significant errors in measurement, especially for long-distance readings. The accuracy was easily compromised by factors such as hand instability, camera angle shifts, and software lag when attempting extended measurements. Consequently, the building model created from these measurements did not fully match the real-world structure, leading to navigation errors where users could be directed slightly away from their intended destinations.

ARFICT ensures that the virtual navigation path is precisely overlaid onto the physical environment by leveraging a professional-grade laser measurement tool. This advancement directly addresses the limitations of the previous app, offering users a much more reliable and accurate AR indoor navigation experience.

6.3.3 Multi-Floor Navigation Support

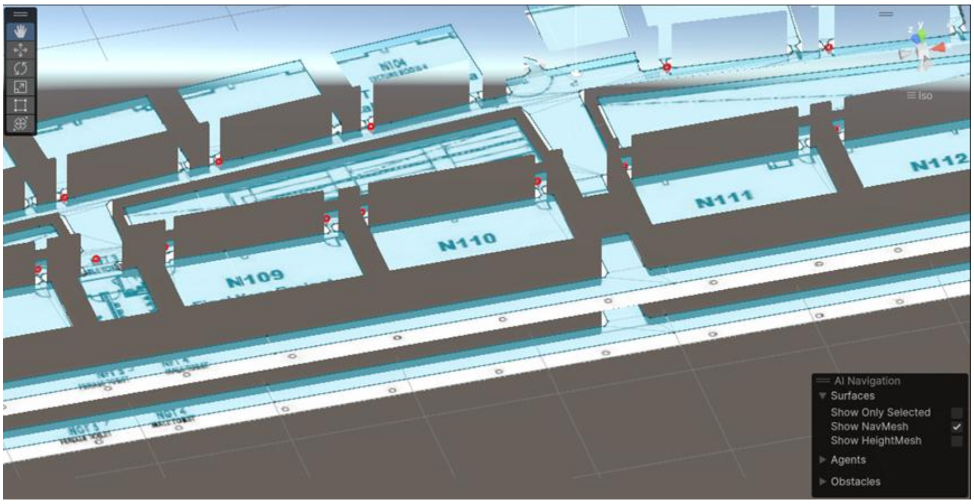


Figure 6.3.3.1 ARFICT's multi-floor FICT modelling

In this project, the building model in Unity was developed to support multi-floor navigation, with the navigation mesh (NavMesh) successfully baked across both floors, as shown in Figure 6.3.3.1. This approach allows users to navigate seamlessly across floors, including movement through staircases, without the need for additional steps. The interconnected NavMesh ensures that the system can calculate paths not only within the same floor but also between different floors, guiding users accurately to their desired destinations anywhere in the building.

This achievement was made possible by using a laser distance metre with a measurement accuracy of ±2 mm, which ensured that the multi-floor building dimensions in Unity were precisely aligned with the real-world environment. The high accuracy of the measurements allowed the NavMesh to be baked consistently across multiple floors, enabling the system to provide reliable navigation without misalignment or drift.

Figure 6.3.3.2 Previous Project's FICT modelling

In contrast, the previous project implemented the two floors independently, creating separate virtual environments for each level, as shown in Figure 6.3.3.2. Since the floors were not connected through a unified NavMesh, the staircase was not integrated into the navigation model. As a result, users were required to scan a QR code every time they moved to another floor to reload the virtual environment and floor plan. This process caused interruptions in the navigation flow and limited usability, as the app could not provide directions to destinations located on other floors.

The successful integration of a multi-floor NavMesh in ARFICT eliminates these limitations by enabling a continuous and uninterrupted navigation experience. Users no longer need to manually reset the environment or rescan QR codes when transitioning between floors. Instead, they can simply follow the AR guidance across levels, just as naturally as navigating within a single floor. This advancement greatly enhances convenience, efficiency, and reliability for users navigating the FICT building.

6.3.4 Audio Navigation Support

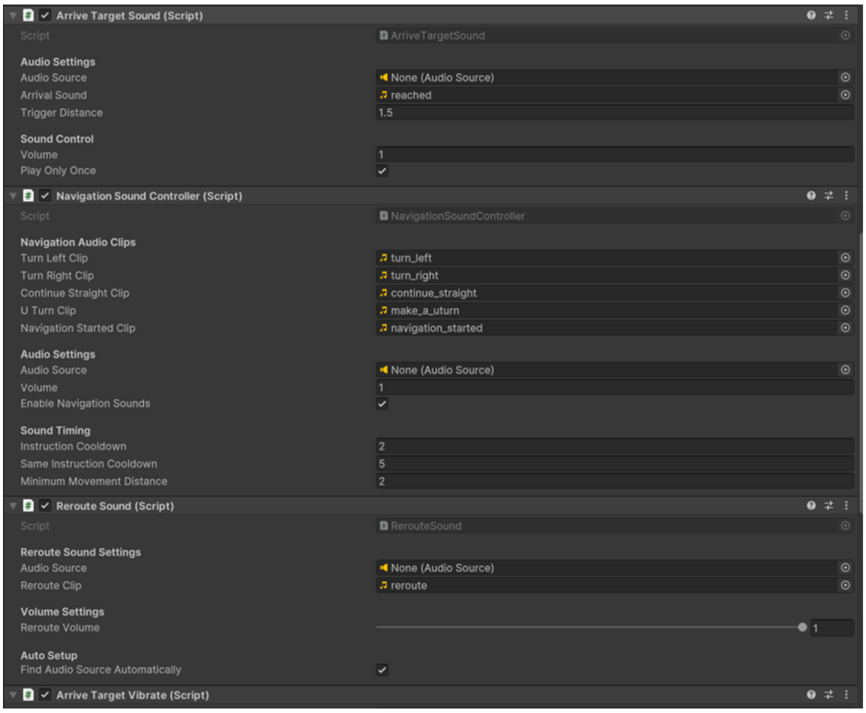


Figure 6.3.4.1 ARFICT's Sound Feature

This subsection explains how ARFICT integrates audio feedback for navigation guidance, providing users with real-time voice cues and sound notifications that respond to different actions. For example, when scanning a QR code, the system plays an audio cue to indicate whether the scan was successful or failed. During navigation, the system also provides directional prompts such as "turn left," "turn right," and an arrival notification when the destination is reached, as shown in Figure 6.3.4.1. These action-performed cues are controlled by Unity's scripting feature, which dynamically triggers the appropriate sound based on user interaction and system events.

Unlike the previous app, which only relied on visual navigation cues, ARFICT delivers multimodal feedback by combining both visual and auditory guidance. This advancement improves accessibility, particularly for users who may be visually impaired or unable to constantly monitor the screen.

ARFICT ensures that users receive continuous and inclusive feedback throughout the navigation process by providing audio responses for various actions such as QR code scanning, navigation direction changes, and arrival confirmation.

6.3.5 Additional Features (Navigation Modes & Distance Calculation)

Figure 6.3.5.1 ARFICT's Arrow Mode

Figure 6.3.5.2 ARFICT's Line Mode

ARFICT offers two navigation modes, which are Line Mode by default and Arrow Mode, as shown in Figure 6.3.5.1 and Figure 6.3.5.2. In Line Mode, a navigation line is generated along the shortest calculated path to the destination, giving users a clear visual guide to follow. In Arrow Mode, an arrow is displayed in front of the device screen, continuously pointing in the direction of the destination. The app displayed a real-time distance label that updates dynamically as the user moves, allowing them to monitor how far they are from their destination. This dual-mode functionality provides flexibility, letting users choose their preferred style of guidance.

Figure 6.3.5.3 Previous App's Navigation

In contrast, the previous app only supported a single navigation mode, which was the line path, as shown in Figure 6.3.5.3. It did not include a distance measurement label, limiting the amount of information available to the user during navigation. As a result, users could only rely on the line visual without knowing their remaining distance. The addition of Arrow Mode and real-time distance tracking in ARFICT represents a major improvement in usability, accuracy, and user choice.

6.4 Objectives Evaluation

The evaluation of the project objectives demonstrated that both the main objective and sub-objectives were successfully achieved. The primary aim of developing ARFICT, an indoor navigation mobile application using Augmented Reality (AR) for the Faculty of Information and Communication Technology (FICT), was accomplished. The final prototype integrated AR technology, optimised route calculation, and marker-based localisation using QR codes, providing accurate and convenient navigation across a complex multi-floor building environment. The system enhanced the overall user experience by offering a more interactive and intuitive alternative to conventional indoor navigation methods.

The first sub-objective, which was to develop a convenient indoor navigation mobile application, was successfully achieved. ARFICT allowed users to navigate seamlessly within the FICT building, including movement between interconnected floors. Features such as destination selection, mini map interaction, continue or end navigation options, and voice-guided prompts contributed to a user-friendly experience, ensuring that the navigation process remained efficient and accessible.

The second sub-objective, focused on integrating AR technology, was also accomplished. The guiding path was displayed in real time within the physical environment using AR, with options to switch between line mode and arrow mode as well as to adjust path height for better visibility. This integration provided users with an intuitive navigation experience by overlaying virtual guidance directly onto their surroundings. The implementation of QR code recentering further enhanced the AR feature, ensuring that the guiding path remained stable and accurate throughout the navigation process.

The third sub-objective, which aimed to optimise route calculation, was successfully realised through the implementation of the A* algorithm. The algorithm generated the shortest and most efficient path to the selected destination, reducing travel time and preventing unnecessary detours. Furthermore, ARFICT supported rerouting to the nearest available location of the same type, such as the closest toilet, improving overall navigation efficiency. This functionality was tested successfully and contributed to a more adaptive and reliable navigation system.

The fourth sub-objective, which focused on localising the user's position using QR codes, was effectively achieved. QR codes were employed as recentering points to recalibrate the user's position and reduce AR tracking drift. Continuous testing of QR code placement ensured consistent positioning accuracy, enabling users to realign themselves whenever tracking errors occurred. This feature proved essential for maintaining reliable navigation across the entire building.

In conclusion, all project objectives were met as intended. ARFICT combined AR-based path guidance, accurate localisation, and optimised routing to create a robust indoor navigation solution. The results demonstrated that the objectives were not only achieved but also contributed to the development of a practical and effective system capable of enhancing user experience within multi-floor indoor environments.

6.5 Concluding Remark

In conclusion, the evaluation of ARFICT demonstrates that the application achieved its intended objectives by delivering a functional and user-friendly AR-based indoor navigation system. Core features such as real-time navigation, multi-floor support, mini map visualisation, and tutorial guidance were successfully implemented and validated. Challenges, including AR tracking drift, measurement accuracy, and navigation across multiple floors, were effectively managed through QR code recentering, refined building modelling, and navigation mesh optimisation. ARFICT showcases the potential of AR technology to enhance indoor navigation within the FICT building, offering users a reliable and interactive tool. The project not only fulfils its current goals but also establishes a strong foundation for future enhancements and wider applications.

CHAPTER 7

Conclusion and Recommendations

This chapter concludes the ARFICT project by summarising its achievements and providing recommendations for future enhancements to improve navigation accuracy, coverage, and user experience.

7.1 Conclusion

The ARFICT project successfully developed an advanced AR-based indoor navigation mobile application for the Faculty of Information and Communication Technology (FICT). The system effectively integrated augmented reality, optimised route calculation, and QR code-based localisation to provide accurate, reliable, and user-friendly navigation across a complex multi-floor environment. All project objectives were achieved, including convenient navigation, AR integration, efficient routing, and precise user localisation.

During testing, ARFICT demonstrated strengths, such as a simple and intuitive interface that combined visual AR path guidance with sound prompts, allowing users to navigate easily even on their first attempt. The system relied on QR code scanning and proper orientation alignment to maintain accurate localisation, which proved effective for consistent navigation across the building. Testing confirmed that the application operated as intended, providing seamless movement between interconnected floors and ensuring users could follow the most efficient paths to their destinations.

The most important and challenging aspect of the project was obtaining accurate dimensions of the multi-floor FICT building. Initial reliance on floor plans led to scaling errors and misaligned structures, which could have compromised navigation accuracy. This issue was overcome by using a laser distance meter to gather precise measurements, significantly improving the reliability of the 3D model and the navigation system. Other challenges, such as AR tracking drift and multi-floor pathfinding, were successfully addressed through QR code recentering and careful NavMesh configuration.

Overall, ARFICT presents a practical and effective solution for indoor navigation, offering an improved user experience compared to conventional methods and providing a solid foundation for future feature enhancements, such as AI-assisted navigation and more interactive user interfaces.

7.2 Recommendations

Future iterations of ARFICT could extend navigation coverage to additional areas, such as lecturers' offices, providing users with a more comprehensive indoor navigation experience. Beacon-based localisation or other positioning hardware could be integrated alongside QR codes to reduce reliance on frequent QR code rescanning. This can ensure more stable and accurate positioning throughout the building. Moreover, incorporating AI technologies, such as deep learning or lightweight language models, could enhance user localisation and navigation by predicting user movement, providing intelligent rerouting suggestions, and offering context-aware guidance when QR code signals are weak or temporarily unavailable.

Improving the user interface would further enhance the overall experience. A colourful, dynamic mini map, animated navigation paths, and customisable AR path styles could make navigation clearer and more engaging. Visual cues, such as highlighted corridors or animated arrows, would help users intuitively follow routes in complex multi-floor environments, reducing confusion and improving efficiency.

Finally, expanding ARFICT to support iOS devices would increase accessibility for a wider range of users. Additionally, integrating alternative localisation hardware, such as Bluetooth beacons or Wi-Fi-based positioning systems, could further stabilise navigation, particularly in areas where QR codes are sparse or difficult to scan.

REFERENCES

[1] 'Indoor Positioning and Indoor Navigation Market Statistics – 2030 | NMSC'. Accessed: Sep. 01, 2024. [Online]. Available: https://www.nextmsc.com/report/indoor-positioning-and-indoor-navigation-market?srsltid=AfmBOorJ4rmMYFJJ3eFRa7uDQAIuFqEudoXh4YxgN5bLUZEWF5vuFDxW

[2] 'What is the accuracy of the GPS in the Trakbond device?' Accessed: Sep. 01, 2024. [Online]. Available: https://trakbond.zohodesk.com/portal/en/kb/articles/what-is-the-accuracy-of-the-gps-in-the-trakbond-device-8-8-2018

[3] 'Experimental Evaluation of Indoor Localization Methods for Industrial IoT Environment', *J Sci Ind Res (India)*, vol. 81, no. 03, Mar. 2022, doi: 10.56042/jsir.v81i03.50926.

[4] 'Challenges and Solutions: Implementing Indoor Navigation in Large Public Spaces | by VAISHAK | Medium'. Accessed: Sep. 01, 2024. [Online]. Available: https://medium.com/@VAISHAK_CP/challenges-and-solutions-implementing-indoor-navigation-in-large-public-spaces-3bc9e1a7b0d

[5] 'What is augmented reality (AR)? | Definition from TechTarget'. Accessed: Sep. 01, 2024. [Online]. Available: https://www.techtarget.com/whatis/definition/augmented-reality-AR

[6] 'Multi-building Navigation for Colleges and Universities | Mappedin'. Accessed: Sep. 01, 2024. [Online]. Available: https://www.mappedin.com/resources/blog/multi-building-navigation/colleges-universities/

[7] N. C. Syazwani, N. Haliza Abdul Wahab, N. Sunar, S. H. S Ariffin, K. Yinn Wong, and Y. Aun, 'Indoor Positioning System: A Review', *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, p. 2022, Accessed: Aug. 28, 2024. [Online]. Available: www.ijacsa.thesai.org

[8] ‘Global Positioning System - an overview | ScienceDirect Topics’. Accessed: Aug. 28, 2024. [Online]. Available: https://www.sciencedirect.com/topics/engineering/global-positioning-system

[9] J. C. P. Cheng, K. Chen, and W. Chen, ‘Comparison of Marker-Based and Markerless AR: A Case Study of An Indoor Decoration System’, in *Lean and Computing in Construction Congress - Volume 1: Proceedings of the Joint Conference on Computing in Construction*, Edinburgh: Heriot-Watt University, Jul. 2017, pp. 483–490. doi: 10.24928/JC3-2017/0231.

[10] J. Kunhoth, A. Karkar, S. Al-Maadeed, and A. Al-Attiyah, ‘Comparative analysis of computer-vision and BLE technology based indoor navigation systems for people with visual impairments’, *Int J Health Geogr*, vol. 18, no. 1, p. 29, Dec. 2019, doi: 10.1186/s12942-019-0193-9.

[11] ‘(14) (PDF) A Novel Method for Indoor Navigation Using QR Codes’. Accessed: Aug. 29, 2024. [Online]. Available: https://www.researchgate.net/publication/295210900_A_Novel_Method_for_Indoor_Navigation_Using_QR_Codes

[12] B. Hartmann, N. Link, and G. F. Trommer, ‘Indoor 3D position estimation using low-cost inertial sensors and marker-based video-tracking’, in *IEEE/ION Position, Location and Navigation Symposium*, IEEE, May 2010, pp. 319–326. doi: 10.1109/PLANS.2010.5507248.

[13] X. H. Ng and N. Lim, ‘Design of a Mobile Augmented Reality-based Indoor Navigation System’.

[14] H. Kato, M. Billinghurst, and I. Poupyrev, ‘ARToolKit version 2.33’, 2000.

[15] G. Yang and J. Saniie, ‘Indoor navigation for visually impaired using AR markers’, in *2017 IEEE International Conference on Electro Information Technology (EIT)*, IEEE, May 2017, pp. 1–5. doi: 10.1109/EIT.2017.8053383.

[16] D. Khan, S. Ullah, and S. Nabi, ‘A Generic Approach toward Indoor Navigation and Pathfinding with Robust Marker Tracking’, *Remote Sens (Basel)*, vol. 11, no. 24, p. 3052, Dec. 2019, doi: 10.3390/rs11243052.

[17] A. Gupta, K. Bhatia, K. Gupta, and M. Vardhan, 'A Comparative Study of Marker-Based and Marker-Less Indoor Navigation in Augmented Reality', *International Research Journal of Engineering and Technology*, 2018, Accessed: Aug. 28, 2024. [Online]. Available: www.irjet.net

[18] 'Marker-based, markerless, or location? Different types of Augmented Reality explained | by Aryel | Medium'. Accessed: Aug. 28, 2024. [Online]. Available: https://aryelhq.medium.com/marker-based-markerless-or-location-different-types-of-augmented-reality-explained-b0e6cbddac7b

[19] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed, 'A Review of Indoor Localization Techniques and Wireless Technologies', *Wirel Pers Commun*, vol. 119, no. 1, pp. 289–327, Jul. 2021, doi: 10.1007/s11277-021-08209-5.

[20] C. Yang and H. R. Shao, 'WiFi-based indoor positioning', *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, Mar. 2015, doi: 10.1109/MCOM.2015.7060497.

[21] Y.-C. Pu and P.-C. You, 'Indoor positioning system based on BLE location fingerprinting with classification approach', *Appl Math Model*, vol. 62, pp. 654–663, Oct. 2018, doi: 10.1016/j.apm.2018.06.031.

[22] C. Zhou, J. Yuan, H. Liu, and J. Qiu, 'Bluetooth Indoor Positioning Based on RSSI and Kalman Filter', *Wireless Personal Communications: An International Journal*, vol. 96, no. 3, pp. 4115–4130, Oct. 2017, doi: 10.1007/S11277-017-4371-4.

[23] 'How to use augmented reality (AR) directions in Google Maps | T3'. Accessed: Aug. 27, 2024. [Online]. Available: https://www.t3.com/news/how-to-use-ar-google-maps-android-directions

[24] 'New Google Maps features including immersive view, Live view updates, electric vehicle charging tools and glanceable directions.' Accessed: Aug. 27, 2024. [Online]. Available: https://blog.google/products/maps/sustainable-immersive-maps-announcements/

[25] A. S. A. Rabby and C. Zhang, 'BeyondPixels: A Comprehensive Review of the Evolution of Neural Radiance Fields', Jun. 2023.

[26] 'Sunway MyCampus - Apps on Google Play'. Accessed: Aug. 31, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=my.edu.sunway.mycampus&hl=en_US

[27] 'Path Guide: A New Approach to Indoor Navigation - Microsoft Research'. Accessed: Aug. 31, 2024. [Online]. Available: https://www.microsoft.com/en-us/research/blog/path-guide-new-approach-indoor-navigation/

[28] 'How Microsoft Path Guide has the Potential to Define the Future of Indoor Navigation - TechPP'. Accessed: Aug. 31, 2024. [Online]. Available: https://techpp.com/2017/05/30/microsoft-path-guide-indoor-navigation/

[29] A. K. M. M. Hossain, 'Crowdsourced Indoor Mapping', *Geographical and Fingerprinting Data for Positioning and Navigation Systems: Challenges, Experiences and Technology Roadmap*, pp. 97–114, Jan. 2019, doi: 10.1016/B978-0-12-813189-3.00005-8.

[30] M. Amin-Naseri and M. C. Shelley, 'Adopting and incorporating crowdsourced traffic data in advanced transportation management systems'.

[31] 'Waze Review | PCMag'. Accessed: Sep. 01, 2024. [Online]. Available: https://www.pcmag.com/reviews/waze

[32] 'Waze could come pre-installed on your next Android phone - The Verge'. Accessed: Sep. 01, 2024. [Online]. Available: https://www.theverge.com/2015/3/3/8140389/waze-google-mobile-apps-android

[33] T. Laor and Y. Galily, 'In WAZE we trust? GPS-based navigation application users' behavior and patterns of dependency', *PLoS One*, vol. 17, no. 11, p. e0276449, Nov. 2022, doi: 10.1371/journal.pone.0276449.

[34] B. J. J. Goh, 'An Augmented Reality (AR) indoor navigation mobile application for the Faculty of Information and Communication Technology

(FICT)’, Oct. 2024, Accessed: Aug. 31, 2025. [Online]. Available: http://eprints.utar.edu.my/6636/

[35] ‘What is an Agile Workflow? | Miro’. Accessed: Sep. 02, 2024. [Online]. Available: https://miro.com/agile/what-is-agile-workflow/

[36] ‘zxing/zxing: ZXing (“Zebra Crossing”) barcode scanning library for Java, Android’. Accessed: Apr. 10, 2025. [Online]. Available: https://github.com/zxing/zxing

[37] ‘Faculty of Information and Communication Technology’. Accessed: Apr. 08, 2025. [Online]. Available: https://fict.utar.edu.my/contact_us.php

APPENDIX

POSTER