BLUEWAY - BLUETOOTH BASED INDOOR NAVIGATION SYSTEM

A PROJECT REPORT

Submitted by

GURUPRASAD K S [CB.EN.U4CSE20421]
LOGESWARAN S R [CB.EN.U4CSE20435]
MANISH SARAVANAN [CB.EN.U4CSE20437]
PRAVEEN MUMAR M [CB.EN.U4CSE20449]

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

MAY 2024

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, COIMBATORE – 641 112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **BLUEWAY - BLUETOOTH BASED INDOOR NAVIGATION SYSTEM** submitted by GURUPRASAD (CB.EN.U4CSE20421), LOGESWARAN S R (CB.EN.U4CSE20435), MANISH SARAVANAN(CB.EN.U4CSE20437) , PRAVEEN KUMAR M(CB.EN.U4CSE20449) in partial fulfillment of the requirements for the award of Degree **Bachelor of Technology** in Computer Science and Engineering is a bonafide record of the work carried out under our guidance and supervision at the Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore.

Dr.Vijaya Kumar Sundar
(Assistant professor)
Department of CSE

Dr.Vidhya Balasubramanian
Chairperson
Department of CSE

Evaluated on:

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We, the undersigned solemnly declare that the project report **BLUEWAY - BLUETOOTH BASED INDOOR NAVIGATION SYSTEM** is based on our own work carried out during the course of our study under the supervision of Dr.Vijaya Kumar Sundar, (Assistant professor), Computer Science and Engineering, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgement has been made wherever the findings of others has been cited.

Guruprasad K S[CB.EN.U4CSE20421]
Logeswaran S R [CB.EN.U4CSE20435]
Manish Saravanan[CB.EN.U4CSE20437]
Praveen Mumar M[CB.EN.U4CSE20449]

ABSTRACT

This project presents a method for developing an efficient indoor navigation system that considers finding the shortest path between a source and destination while ensuring fault tolerance for real-time navigation. An android application has been developed for indoor navigation that uses Bluetooth Low Energy (BLE) of ESP32, cloud technology, and synchronised gyroscope for effective navigation. In this system, multiple BLE modules are strategically placed within buildings, and their locations are integrated with the building's map. When users select their destinations, the system generates the shortest path for navigation. As users start walking, the app utilises the beacons to analyse the users location and updates their location on the map accordingly. The Firebase real-time database verifies that all modules within the navigation system are operational, ensuring that the path is generated using active components. This ensures seamless operation and reliability of the system throughout the navigation process.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our beloved Satguru **Sri Mata Amritanandamayi Devi** for providing the bright academic climate at this university, which has made this entire task appreciable. This acknowledgment is intended to thank all those people involved directly or indirectly with our project. We would like to thank our, Vice Chancellor **Dr.Venkat Rangan.P** and **Dr.Bharat Jayaraman**, Dean, School of Computing, Amrita Vishwa Vidyapeetham for providing us with the necessary infrastructure required for the completion of the project. We express our thanks to **Dr.Vidhya Balasubramanian**, Chairperson, Department of Computer Science Engineering and Principal, School of Computing, Amrita Vishwa Vidyapeetham, **Dr.C.Shunmuga Velayutham and Dr.N.Lalithamani,** Vice Chairpersons of the Department of Computer Science and Engineering for their valuable help and support during our study. We express our gratitude to our guide, **Dr.Vijaya Kumar Sundar** for their guidance, support and supervision. We would also like to thank the entire members of the Department of Computer Science and Engineering. We would like to extend our sincere thanks to our family and friends for helping and motivating us during the course of the project. Finally, we would like to thank all those who have helped, guided, and encouraged us directly or indirectly during the project work. Last but not the least, we thank God for his blessings which made our project a success.

Guruprasad K S
Logeswaran S R
Manish Saravanan
Praveen kumar M

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

IOT	Internet of Things
BLE	Bluetooth low energy
LBS	Location Based Services
GNSS	Global navigation satellite system
IPS	Indoor positioning system
UWB	Ultra Wideband
RSSI	Received Signal Strength Indicator
TDOA	time difference of arrival
MDPI	more accurately than traditional methods

Chapter 1

INTRODUCTION

Navigation is indispensable in modern life, facilitating efficient movement from one location to another. While outdoor navigation using GPS has become very common, the need for indoor navigation is equally crucial. Indoors, complex environments like shopping malls, airports, and office buildings can confound individuals, leading to wasted time, frustration, and even safety concerns. Indoor navigation systems offer solutions to these challenges by providing accurate guidance within enclosed spaces. They enhance user experience by reducing uncertainty and improving efficiency, especially in critical situations such as emergencies where quick access to exits or facilities is vital. Moreover, indoor navigation opens up commercial opportunities, allowing businesses to guide customers to specific areas or promotions.

As the advancement in digital technology and its devices, Location Based Services (LBS) are considerably growing. LBS has long used Global navigation satellite system (GNSS) to navigate, get accurate and reliable information about the outdoor environment. The signals from GNSS are weak, when it penetrates through walls, making it impossible to get accurate indoor LBS. Because GNSS are ineffective indoors, it takes a lot of research and development to create an Indoor positioning system (IPS). For the development of IPS, numerous technologies and methods have been investigated. As a result of various access points (APs) and multiple signal sources, the IPS is realised.

There are many ways for establishing the source signal such as Bluetooth low energy (BLE), Ultra Wideband (UWB), radio frequency identification (RFID) tags, or use already installed APs, such as Wi-Fi, geomagnetic fields. A dead-reckoning technique called "signal-free solutions" in IPS uses commercially available mobile sensors to detect changes in location. Received Signal Strength Indicator (RSSI), time of arrival (TOA), time difference of arrival (TDOA), and angle of arrival (AOA) are few of the IPS concepts for sensing radio signals. Due to its simplicity of implementation and lack of additional hardware requirements, RSS has been commonly employed for IPS design. Compared to the outdoor environment, it is difficult to predict the indoor radio signal transmission. Even when users are in strange environments, IPS aims to help them get to their final destination.

Ideally, deploying an indoor navigation system must be easy and cost effective. Most of the time, signals received from Wi-Fi devices present in a building are used as reference, however, these are not meant to be used for this purpose. It is because of this that other technologies are better suited for indoor navigation, such as Bluetooth Low Energy BLE, which may be a good alternative. BLE is a subsystem of the traditional Bluetooth technology capable of broadcasting data using a minimal amount of power. This makes it ideal for devices operating on small batteries which need to function uninterrupted for long periods of time. For the purpose of indoor navigation, BLE devices known as beacons seem to be the best choice.

The aim is to develop an efficient indoor navigation system using BLE technology with the consideration of the shortest path between source and destination and fault tolerance to ensure the functioning of the Bluetooth modules. A smartphone is required for the system, along with the installation of an application. The proposed method obtains blueprint data of an organisation and updates them at the top of the map, the smartphone application is based on the building's floor plan. The mobile application will assist the third-party user in locating inside a building and determining the quickest route between the source and destination after updating the floor plan details in the map with the signal received from the beacons.

1.1 Background

This provides concise background information related to the various technological concepts and platforms that are fundamental to developing an indoor navigation system. Understanding these elements is crucial to fully appreciate the integration of mobile apps, cloud, ESP32 building an indoor navigation system.

1.1.1 Bluetooth Low Energy

BLE was developed for the purpose of broadcasting data without the need of a connection. BLE devices are normally used in one-way communications where receivers pick up data passively and are unable to establish connections with the broadcaster. Due to this, data on the device cannot be manipulated, except for specific cases in which dedicated USB cables can be used to upload firmware. BLE is a subsystem of the conventional Bluetooth 4.0 protocol, designed to provide device communication with minimal power consumption. The number of transmission channels has been reduced from 79 to 40 2 MHz wide channels, 3 of which (37, 38 and 39) are used for advertising. The key advantages of the BLUE are Reduced energy consumption, Wide coverage, Higher data rates.

1.1.2 Shortest Path Algorithm

The shortest path algorithm computes the shortest path, assisting users in reaching their desired locations efficiently. This not only reduces the time users spend searching for destinations but also decreases the manpower required for guiding individuals in business establishments and educational institutions. In essence, the algorithm streamlines navigation, optimising both time and resources for users and organisations alike.

1.1.3 Fault Tolerance

The Firebase real-time database receives acknowledgments from the ESP32 modules, listing the modules that are functioning. This allows the user to choose alternate paths if necessary, ensuring fault tolerance within the system. By providing real-time information on the status of modules, the system enhances reliability and resilience, enabling users to navigate with confidence even in the event of module failures.

1.1.4 ESP32

The ESP32 is a versatile microcontroller widely used in Internet of Things (IOT) applications. It features a dual-core processor, Wi-Fi and Bluetooth connectivity, and a rich set of peripheral interfaces, making it suitable for a wide range of projects. The Bluetooth connectivity in it can be used for positioning and the Wi-Fi connectivity in it can be used for fault tolerance in the firebase. The ESP32 supports low-power operation, making it suitable for battery-powered applications.

1.2 Motivation

Indoor navigation systems are crucial for navigating intricate, enclosed spaces like shopping malls and hospitals, where GPS signals are frequently unreliable or inaccessible. While outdoor navigation systems predominantly rely on GPS and are effective in guiding users through open environments, the challenges posed by indoor spaces necessitate specialised solutions. These systems significantly reduce search times, particularly in critical situations such as emergencies in hospitals or fire outbreaks, by providing clear guidance to exits and essential facilities. This creates a need for an alternate technology for indoor navigation.

1.3 Problem statement

Given the limitations of navigation systems in complex indoor structures there is a need to develop an indoor navigation system that must be able to achieve the following: The indoor navigation system must dynamically integrate the real-time active status of devices to determine the most efficient routing paths. The system must utilise advanced pathfinding algorithms to continuously update and optimise routes based on the operational status of sensors and beacons.

Chapter 2

LITERATURE SURVEY

In the rapidly evolving field of indoor navigation, the integration of advanced technologies like Bluetooth Low Energy (BLE) and Received Signal Strength Indicator (RSSI), alongside sophisticated pathfinding algorithms and robust fault tolerance mechanisms, marks a significant technological frontier. This literature survey explores these critical components, each fundamental to developing systems that provide reliable and precise navigation within indoor environments where traditional GPS systems falter.

Indoor navigation systems are essential in complex structures like shopping malls, airports, hospitals, and industrial sites, where precise localization is crucial for safety, efficiency, and convenience. The primary challenge in these settings is the lack of satellite signal penetration, which necessitates alternative methods to determine locations within buildings accurately. BLE technology, utilising RSSI, offers a promising solution by measuring the strength of the signal from multiple fixed points within the building to triangulate positions. This method's accuracy hinges significantly on the sophistication of the underlying algorithms that process RSSI values and the robustness of the system's design to accommodate environmental variabilities and potential system failures.

Foundational research by (1) demonstrated the initial feasibility of RSSI for indoor positioning, setting the stage for numerous enhancements in accuracy and reliability through advanced filtering and calibration techniques. As the complexity of indoor environments and the demand for more precision in navigation have increased, so too has the reliance on machine learning to refine data interpretation and decision-making processes within these systems. This has led to the development of models capable of adapting to dynamic conditions in real time, a critical feature for indoor navigation systems operating in highly variable environments.

Parallel to these developments, the survey delves into the evolution of pathfinding algorithms, such as Dijkstra's and A* algorithms, tailored specifically for indoor settings. These algorithms are crucial for mapping out efficient routes within complex indoor spaces and have been enhanced by integrating real-time data and machine learning to dynamically adjust paths in response to environmental changes.

Moreover, the robustness of indoor navigation systems is significantly bolstered by fault tolerance mechanisms. These systems are designed to ensure continuous operation despite failures, whether from individual sensor malfunctions or broader network issues. Techniques such as redundancy, distributed architectures, and cloud computing have been pivotal in enhancing the reliability of these systems. They ensure that the navigation service remains operational, thus maintaining safety and operational continuity in critical settings.

2.1 Indoor Navigation Using Bluetooth RSSI

Indoor navigation systems utilising Bluetooth Low Energy (BLE) and RSSI have become increasingly vital in overcoming the limitations of GPS in indoor environments. The integration of BLE with RSSImeasurement for positioning and navigation within structures such as buildings and underground complexes highlights significant advancements in this technology.

One of the foundational studies by (1) demonstrates the feasibility of using RSSI values for indoor positioning, establishing a benchmark for accuracy and reliability in such systems . Subsequently, studies like the one by (1) have explored enhancements in RSSI-based systems using filtering and calibration techniques to improve location accuracy.

Recent advancements have focused on integrating machine learning algorithms to refine the estimation processes. For instance, (3) employed Kalman filters to address the inherent variability and instability of RSSI signals due to environmental factors . Concurrently, (4)proposed a framework using fingerprinting techniques to enhance the precision of RSSI based indoor navigation systems.

Further research by (5) explored the spatial diversity of BLE beacons to optimise signal distribution and minimise localization error . This approach was complemented by . (6), who utilised multiple models for RSSI signal propagation to improve the robustness and accuracy of indoor positioning systems.

The integration of deep learning techniques has also been examined, with significant contributions from (7), who developed a deep neural network model to predict indoor positions from RSSI data more accurately than traditional methods (MDPI). This model addressed issues such as signal fluctuation and multi-path interference, common in complex indoor environments.

Moreover, the role of BLE in enhancing indoor navigation systems has been highlighted in the research by (8), where upgraded hardware capabilities allowed for better handling of RSSI values and increased system efficiency . The adoption of advanced algorithms for dynamic and real-time navigation was further explored by (9), demonstrating significant improvements in latency and computational overhead.

To address the challenges of scalability and integration, studies by (10) have introduced cloud-based solutions for managing data from multiple sensors and users in large-scale indoor environments . These solutions not only enhance the scalability of RSSI based indoor navigation systems but also improve their adaptability to various commercial and industrial applications.

The evolution of RSSI based indoor navigation systems from basic signal strength measurement to sophisticated, machine-learning-enhanced solutions exemplifies significant technological progress. The integration of advanced computational methods, enhanced signal processing techniques, and new BLE standards continues to push the boundaries of what these systems can achieve, offering more reliable, accurate, and efficient solutions for indoor positioning and navigation.

2.2 Pathfinding Algorithms in Indoor Navigation

Pathfinding algorithms are pivotal in the realm of indoor navigation, enabling precise and efficient routing within complex environments such as buildings, malls, and campuses. This literature survey delves into various pathfinding strategies that have been adapted for indoor navigation systems, highlighting significant developments and the integration of advanced computational techniques.

2.2.1 Fundamental Pathfinding Techniques

Dijkstra's Algorithm

Dijkstra's algorithm remains a cornerstone for indoor navigation systems, providing a robust method for finding the shortest path in a weighted graph. Its application in indoor settings facilitates real-time user guidance by calculating the most efficient route from point A to point B. (11) explored enhancements to Dijkstra's algorithm, optimising its performance in dynamic indoor environments where obstacles and user density can change rapidly.

A Algorithm*

The A* algorithm, known for its efficiency and accuracy, utilises heuristics to speed up the pathfinding process. It has been particularly effective in indoor navigation by considering various metrics such as distance, time, and even crowd density to determine the optimal path. A study by (22) demonstrated the adaptability of the A* algorithm in a multi-floor shopping mall scenario, showing significant improvements in navigation precision and computational time.

2.2.2 Advanced Techniques and Integrations

Graph-based Pathfinding

Graph-based techniques have seen extensive use due to their ability to model complex indoor spaces accurately. These methods involve constructing a graph representation of the indoor space, where nodes represent specific locations, and edges represent accessible paths (19) incorporated real-time data into graph-based models to dynamically update paths based on current conditions, such as congestion or temporary obstructions.

Machine Learning in Pathfinding

The integration of machine learning with traditional pathfinding algorithms has opened new avenues for predictive and adaptive navigation systems. By learning from historical data, these systems can predict areas of congestion and suggest alternate routes proactively. An innovative approach by (20) used reinforcement learning to continuously improve the path selection process as the algorithm encountered various environmental factors during operation.

2.2.3 Multi-Modal and Dynamic Pathfinding

Multimodal Routing

Multi-modal pathfinding algorithms consider different types of movements and transportation modes within indoor environments, such as elevators, escalators, and footpaths. This is crucial in large multi-use spaces. Research by (24) developed a model that seamlessly integrates these modes, providing users with the quickest or least effort-intensive paths depending on their preferences and physical needs.

Dynamic Pathfinding for Real-Time Adaptation

Dynamic pathfinding adapts to real-time changes within the environment, an essential feature for large public spaces that experience varying levels of human traffic. (23) introduced a dynamic pathfinding system that adjusts routes based on real-time IoT sensor data, significantly enhancing navigational efficiency in crowded or rapidly changing conditions . The field of pathfinding algorithms for indoor navigation is evolving rapidly, with advancements that not only enhance the accuracy of navigation but also address real-time changes and multi-modal transportation needs. Future research is likely to focus on further integrating AI to refine the adaptability and predictive capabilities of these systems.

Dijkstra's algorithm is highly regarded in the field of indoor navigation due to its robustness and effectiveness in finding the shortest path between points. Its systematic approach ensures that it can calculate the most optimal route efficiently, which is crucial for real-time applications where rapid response is necessary. Unlike other algorithms that may require more complex calculations or additional information about the environment, Dijkstra's algorithm operates effectively with straightforward, weight-based criteria. This makes it not only reliable but also simpler to implement and adapt to various indoor environments, contributing to its popularity and widespread use in navigation systems.

2.3 Fault Tolerance Mechanisms for indoor navigation

Fault tolerance in indoor navigation systems is crucial to ensure reliability and continuous service, especially in environments where navigation aids are critical, such as hospitals, industrial sites, and large commercial complexes. This literature survey examines various strategies and technological innovations that have been developed to enhance fault tolerance in these systems. Fault tolerance refers to the ability of a system to continue functioning in the event of a failure of some of its components. Early studies by (17) highlighted the necessity of designing systems that can operate under partial failures without degrading the overall system performance significantly.

2.3.1 Redundancy Techniques

Redundancy is a fundamental approach in fault tolerance, where multiple components perform the same function. This ensures that if one component fails, others can take over without loss of functionality. Li and Parker (2016) explored various models of redundancy in sensor networks, particularly emphasising the role of redundant data paths in maintaining system operations during individual sensor failures.

2.3.2 Fault Tolerance Techniques

Distributed Architectures

Moving from centralised to distributed system architectures enhances fault tolerance by decentralising the navigation tasks across multiple nodes. (19)demonstrated that distributed systems could effectively reduce the risk of total system failure and improve resilience by localising the effects of a fault to a small part of the system.

Use of Cloud Computing

Cloud computing has been instrumental in furthering fault tolerance by providing dynamic resource allocation and high availability. Studies by (20) showed that cloud-based indoor navigation systems could seamlessly handle component failures by reallocating tasks among available resources without impacting user experience.

2.3.3 Integration of Machine Learning and AI

Predictive Maintenance

Machine learning techniques are increasingly being employed to predict potential system failures before they occur. (21) discussed the use of predictive analytics to monitor system health and predict points of failure in real-time, thus preemptively addressing faults before they impact system performance.

Adaptive Algorithms

Adaptive algorithms can adjust their behaviour in response to faults in the system. This flexibility was highlighted in the work by (22), where algorithms dynamically adjusted the routing and guidance in response to detected faults in the navigation infrastructure, enhancing system robustness.

2.3.4 Real-World Applications and Case Studies

Hospital Navigation Systems

In critical environments like hospitals, fault tolerance is paramount. Research by (23) provided insights into how hospital indoor navigation systems utilise layered fault tolerance strategies to ensure that emergency services are always guided accurately, even when standard navigation aids fail.

Industrial Automation

In industrial settings, fault tolerance is crucial for safety and efficiency .(24) explored the integration of robust fault-tolerant designs in AGV used within factories, ensuring that navigation continues flawlessly even in the event of individual sensor or communication link failures.

Fault tolerance mechanisms in indoor navigation systems are vital for ensuring reliability and service continuity. The ongoing advancements in technology, particularly through the use of distributed systems, cloud computing, and AI, are enhancing the ability of these systems to withstand and quickly recover from failures.

Chapter 3

IMPLEMENTATION

3.1 System architecture

The BlueWay navigation system is meticulously crafted to offer users precise and reliable indoor navigation solutions leveraging the advanced capabilities of Bluetooth Low Energy (BLE) technology. This innovative system is engineered with a synergy of interconnected components, meticulously designed to harmonise and deliver an intuitive and efficient navigation experience within indoor environments.

At its core, BlueWay harnesses the power of BLE technology, a low-energy variant of Bluetooth, renowned for its ability to facilitate communication over short distances with minimal power consumption. By strategically deploying BLE beacons throughout indoor spaces, BlueWay establishes a robust network that serves as the foundation for accurate positioning and navigation.

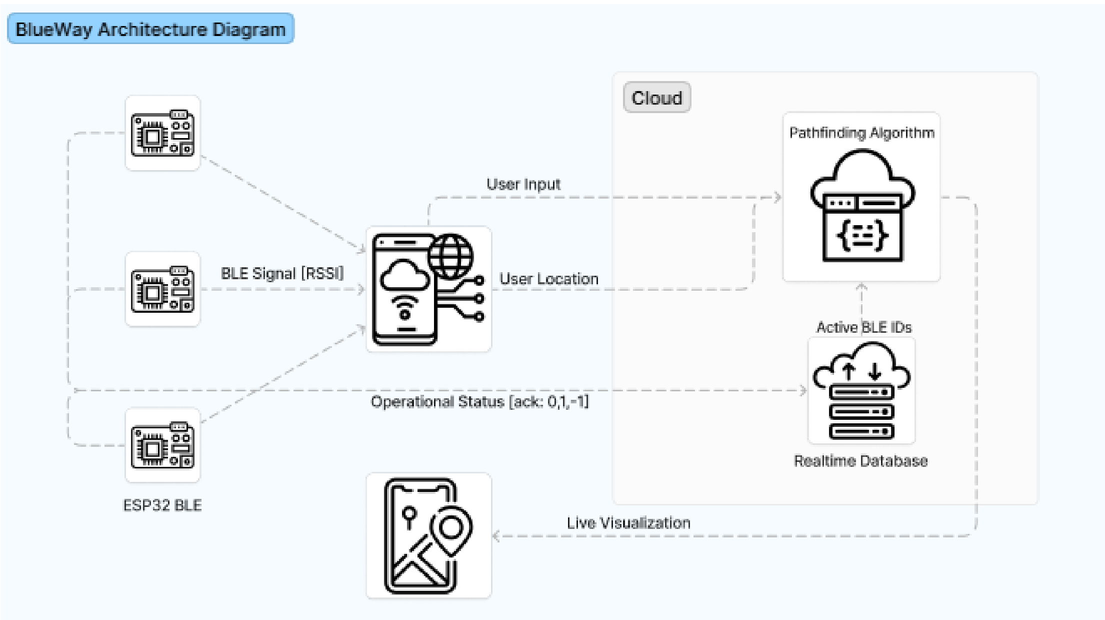

Figure 3.1: Architecture Diagram

The system's architecture revolves around several key components, each playing a pivotal role in ensuring seamless navigation:

3.1.1 RSSI Values Processing:

Received Signal Strength Indication (RSSI) values represent the strength of the signal received from BLE beacons. These values can vary depending on factors such as distance, obstacles, and interference. In the BlueWay system, RSSI values are processed to estimate the proximity of the user to nearby BLE beacons. By analysing RSSI values from multiple beacons, the system can triangulate the user's position and determine their location within the indoor environment. This information is essential for calculating the user's current location and generating optimal navigation routes.

3.1.2 2D Floor Plan Visualization:

The 2D floor plan visualisation feature within the Bluetooth app provides users with a graphical representation of the indoor environment. This visualisation overlay includes navigation paths, destination markers, and other relevant information, allowing users to easily understand their surroundings and navigate with confidence. By presenting navigation instructions in a clear and intuitive manner, the 2D floor plan visualisation enhances the user experience and facilitates seamless indoor navigation.

3.1.3 User Destination Selection:

User destination selection is a key interaction point within the Bluetooth app. Users can input their desired destination from a list of available options, such as specific rooms, points of interest, or designated areas within the indoor environment. This feature streamlines the navigation process by allowing users to quickly and easily specify their intended destination. Once a destination is selected, the app initiates the pathfinding process and generates an optimised route to guide the user to their chosen location.

3.1.4 Cloud-based Functionalities:

Cloud functions dedicated to shortest path estimation are integral to the BlueWay system's navigation capabilities. These functions receive user inputs, such as current location and selected destination, along with real-time data, including beacon signals and floor plan information. Leveraging advanced algorithms, such as Dijkstra's algorithm, these cloud functions compute the most efficient navigation routes within the indoor environment. By harnessing the scalability and computational power of cloud infrastructure, these functions can rapidly process complex data sets and generate optimised paths tailored to each user's specific requirements. The efficient execution of these cloud functions ensures that users receive accurate and reliable navigation guidance, enhancing their overall indoor navigation experience.

3.1.5 Fault detection

Fault detection mechanisms implemented in the cloud are essential for monitoring the operational status of ESP32 BLE beacons within the BlueWay system. These mechanisms involve periodic communication between the BLE beacons and cloud-based services, where the beacons send acknowledgment signals to indicate their operational readiness. Cloud-based algorithms analyse these acknowledgment signals in real-time, promptly identifying any anomalies or malfunctions within the beacon network. In the event of a fault or malfunction, such as a beacon outage or connectivity issue, the fault detection system triggers automated responses, such as alert notifications or beacon reinitialization procedures. By ensuring the reliability and uptime of the beacon network, these fault detection mechanisms contribute to the seamless operation of the BlueWay navigation system, minimising disruptions and enhancing user satisfaction.

By leveraging these components, the BlueWay navigation system delivers a comprehensive indoor navigation solution that combines advanced technologies with user-friendly interfaces, enhancing the accuracy, efficiency, and reliability of indoor navigation experiences.

The flow diagram in above outlines the structured approach to enhancing indoor navigation using BLE and RSSI, illustrating the integration of advanced pathfinding algorithms and robust fault tolerance mechanisms to ensure precise and reliable user guidance.

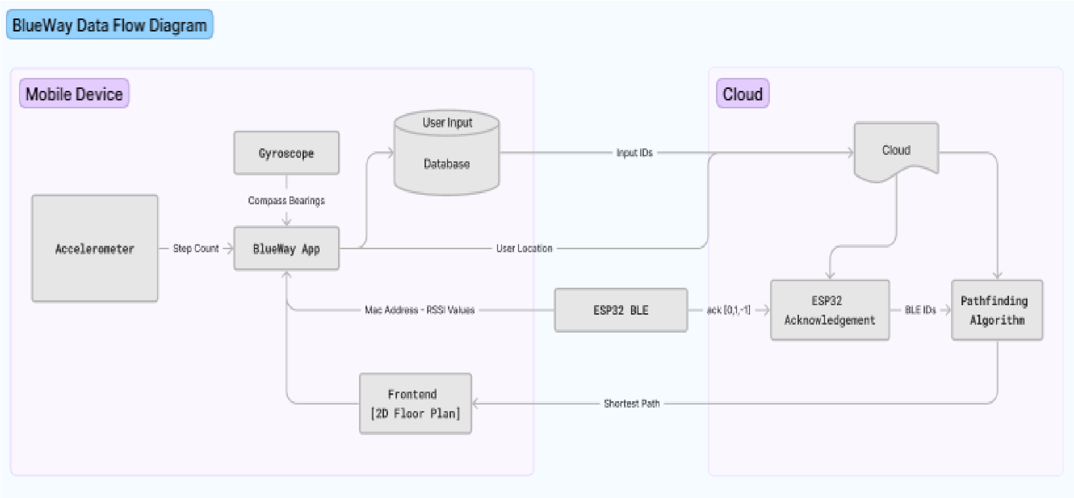

Figure 3.2: Dataflow Diagram

3.1.6 Sensors and Inputs:

The accelerometer is a sensor that detects motion, primarily used for counting steps, helping to gauge the user's activity level or mobility within a space. Simultaneously, the gyroscope and compass bearings provide essential orientation and direction data, crucial for determining the direction in which the device, and presumably the user, is moving. Together, these sensors offer a comprehensive understanding of the user's movement and orientation, which is vital for applications requiring real-time location and navigation assistance.

3.1.7 User Input and Database:

User Input and Database functionality are key components in applications involving navigation and personal settings. User Input allows individuals to provide various commands or data inputs, such as setting a destination or specifying preferences for navigation. Simultaneously, the Database serves as a critical backend feature that likely stores user data, settings, and potentially cached navigation data or user history. This storage capability not only enhances user experience by remembering preferences and frequent destinations but also improves the efficiency of the application by quickly retrieving necessary data without reprocessing it each time. Together, these features support a personalized and efficient navigation experience, adapting dynamically to the user's input and historical interactions.

3.1.8 BlueWay App Functionality:

The "Step Count" feature utilizes accelerometer data to calculate the number of steps taken by the user, providing insights into physical activity levels. Concurrently, "User Location" leverages a combination of accelerometer, gyroscope, and compass data to estimate the user's current location, which can be further enhanced by integrating with other location-determining technologies such as Wi-Fi or Bluetooth, though not explicitly mentioned, this is implied in the context of advanced data handling. Additionally, the "Mac Address - RSSI Values" functionality utilizes RSSI (Received Signal Strength Indicator) values linked to specific MAC addresses, likely those of Bluetooth beacons, to refine the user's location indoors where GPS signals may be unreliable. Together, these features form a robust system for tracking both movement and geographical positioning in various environments.The "Frontend [2D Floor Plan]" feature of the mobile app plays a crucial role in aiding navigation by displaying a 2D floor plan of the area. This component of the app is designed to interact seamlessly with the backend systems, retrieving and displaying appropriate map data tailored to the user's current location and inputs. This functionality not only enhances user orientation within a specific space but also ensures a smooth and interactive navigation experience, making it easier for users to visualize their route and understand their surroundings accurately.

3.1.9 Cloud

In the context of data handling and processing, Input IDs typically involve user identification or device/session IDs that are transmitted to the cloud for processing. The ESP32 BLE, a microcontroller endowed with Bluetooth Low Energy capabilities, plays a pivotal role in managing real-time location data through BLE beacons. Upon processing these inputs, the cloud system issues an acknowledgment back to the mobile device, confirming the successful reception and processing of the user's location and navigation queries. Subsequently, a sophisticated pathfinding algorithm in the cloud takes over, calculating the shortest or most efficient route based on the user's location, intended destination, and factors such as distance, accessibility, and potentially user preferences or prevailing environmental conditions like crowded areas or closed sections. The optimal route determined by this algorithm is then relayed back to the user’s device, updating the current route on the 2D floor plan displayed on the frontend, ensuring the navigation is both accurate and tailored to the current scenario.

Figure 3.3: The image depicts an ESP32 module connected to a power supply, demonstrating the setup phase for powering the device to enable its functionalities for IoT applications and indoor navigation tasks.

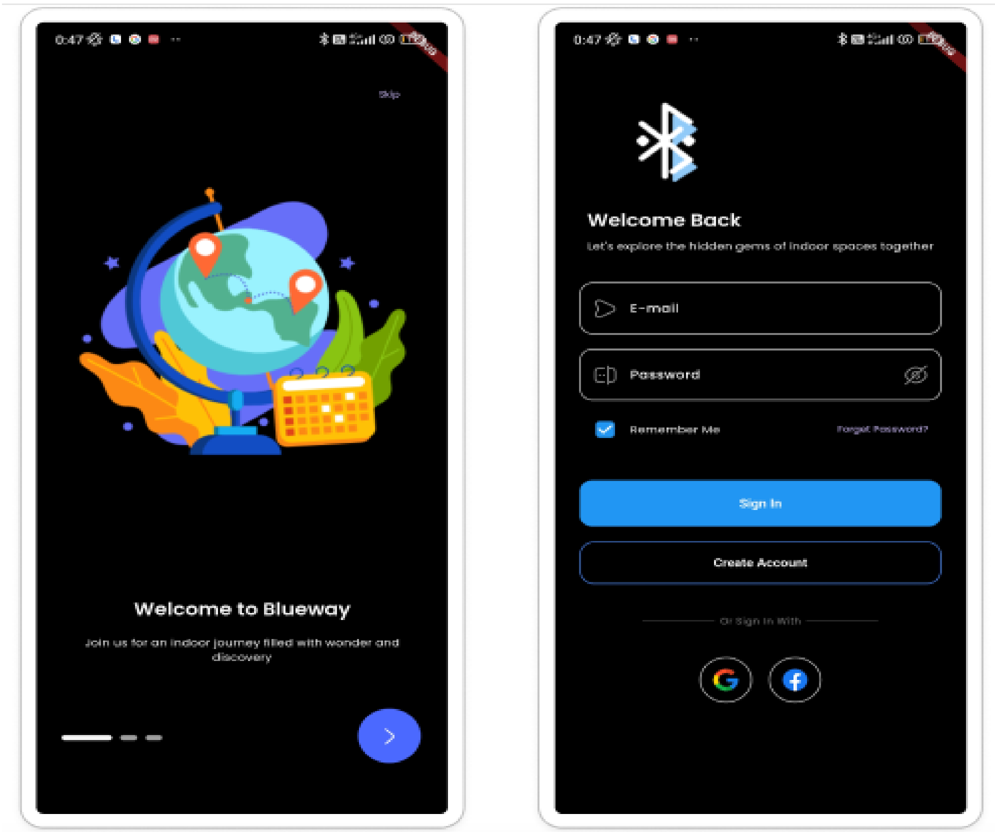

Figure 3.4: Login interface for the "Blueway" app, featuring options for email and password entry, account creation, and social media integration for streamlined user access.

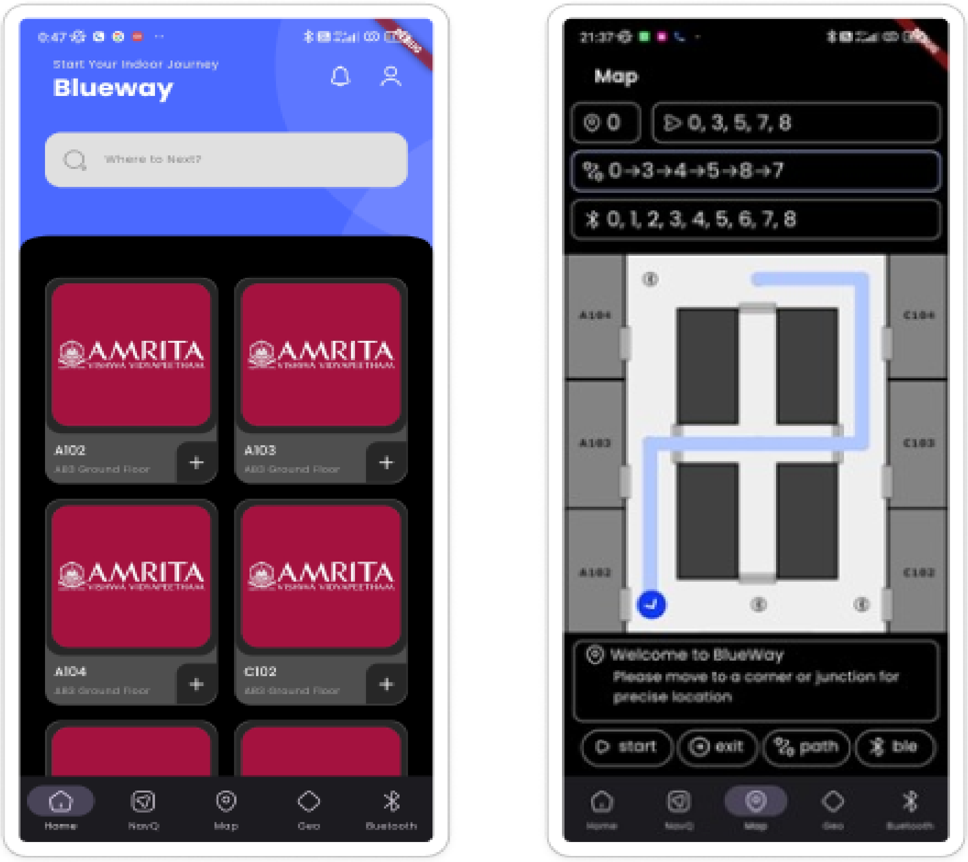

Figure 3.5: The application images showcase a clean and intuitive user interface, featuring a detailed map view that highlights navigational paths and key landmarks, accompanied by easy-to-follow directional cues for seamless indoor navigation.

3.2 Deployment of bluetooth beacons

The deployment of Bluetooth beacons is a critical step in setting up an indoor navigation system. It involves strategically placing ESP32 modules throughout the building to ensure comprehensive coverage and optimal signal reception for accurate location determination.

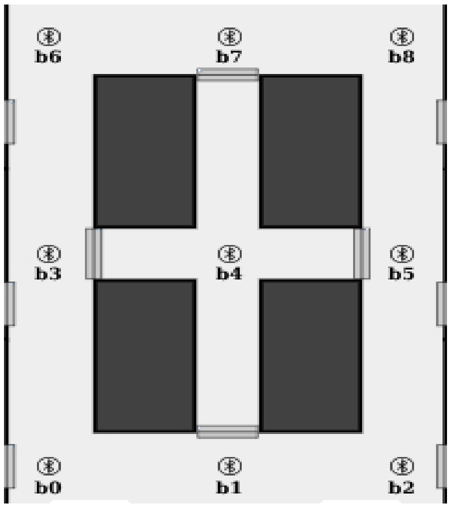

Figure 3.6: Floor plan showing the deployment of Bluetooth Low Energy (BLE) beacons (b0 to b8) for indoor positioning within a college building layout.

When deploying Bluetooth beacons for indoor navigation, it is crucial to strategically place them at key points within a building to maximize coverage and minimize interference, with common locations including entrances, exits, hallway intersections, and large open spaces. The distance between beacons typically spans up to 18 meters, depending on building architecture and material properties that could influence signal strength; thus, conducting signal strength tests across various building sections is essential to determine optimal beacon spacing. Beacons should be installed at a uniform height of about 3 meters above the floor to mitigate interference from daily activities and ensure unobstructed line of sight for receivers. For power considerations, beacons should be near power sources if not battery-operated, and the battery life and maintenance accessibility of battery-operated beacons must be considered. Additionally, beacon placement should prioritize security, not disrupt the aesthetic of the environment, nor inconvenience the building's occupants. Post-installation, thorough testing is imperative to validate the accuracy of location data, with potential adjustments for areas experiencing poor reception or interference. Documenting the precise location, beacon type, signal strength, and configuration details is also vital for ongoing maintenance and troubleshooting. This structured approach ensures the effective operation of indoor navigation systems and allows for adjustments based on unique building characteristics and initial testing results.

3.3 Pathfinding algorithm Implementation

3.3.1 Algorithm

The algorithm operates in several key phases, each designed to systematically find and optimize paths within a given graph represented as an adjacency matrix.

Below Dijkstra's algorithm is employed in this program to compute the shortest paths from a given start vertex to all other vertices within a graph represented as an adjacency matrix. Initially, all vertices are marked as unvisited, and the shortest distance to each vertex is set to infinity, except for the start vertex, which is set to zero. The algorithm proceeds by repeatedly finding the unvisited vertex with the smallest known distance from the start vertex, marking it as visited. For each newly visited vertex, the algorithm updates the shortest path distance for its adjacent vertices if a shorter path is discovered through the current vertex. This update is performed by comparing the current shortest distance to a vertex with the sum of the shortest distance to the newly visited vertex and the weight of the edge connecting them. The process repeats until all vertices have been visited or the shortest distances to all vertices are finalized. The result is a list of the shortest distances from the start vertex to all others, along with the paths themselves, reconstructed using a parent pointer array that tracks the immediate predecessor of each vertex on its shortest path.

Algorithm 1 Dijkstra's Algorithm for All-Pairs Shortest Paths

```
procedure DIJKSTRA(Adjacency_Matrix, Start_Vertex)
    n_vertices ← length of Adjacency_Matrix[0]
    shortest_distances ← list initialized to sys.maxsize of size n_vertices
    added ← list of boolean False of size n_vertices
    shortest_distances[Start_Vertex] ← 0
    parents ← list initialized to − 1 of size n_vertices
    parents[Start_Vertex] ← NO_PARENT
    for i ← 1 to n_vertices do
        nearest_vertex ← −1
        shortest_distance ← sys.maxsize
        for vertex_index ← 0 to n_vertices do
            if ¬added[vertex_index] and shortest_distances[vertex_index] < shortest_distance then
                nearest_vertex ← vertex_index
                shortest_distance ← shortest_distances[vertex_index]
            end if
        end for
        added[nearest_vertex] ← True
        for vertex_index ← 0 to n_vertices do
            edge_distance ← Adjacency_Matrix[nearest_vertex][vertex_index]
            if edge_distance > 0 and shortest_distance + edge_distance < shortest_distances[vertex_index] then
                parents[vertex_index] ← nearest_vertex
                shortest_distances[vertex_index] ← shortest_distance + edge_distance
            end if
        end for
    end for
    results ← empty list
    for vertex_index ← 0 to n_vertices do
        if vertex_index ≠ Start_Vertex then
            path ← call PrintPath with vertex_index and parents
            path_str ← convert path to string
            results.append([Start_Vertex, shortest_distances[vertex_index]])
        end if
    end for
    return results
end procedure
```

Algorithm 2 Finding paths covering specific nodes

procedure FINDPATHS($Results, Start\_Vertex, Vertices\_To\_Visit$)
  $current\_path \leftarrow Start\_Vertex$
  $current\_distance \leftarrow 0$
  $visited \leftarrow [Start\_Vertex]$
  $all\_paths \leftarrow$ empty list
  for each $entry$ in $Results$ **do**
    if $entry[0] = Start\_Vertex$ and $entry[1]$ in $Vertices\_To\_Visit$ and not in $visited$ **then**
      $new\_vertices\_to\_visit \leftarrow Vertices\_To\_Visit$ without $entry[1]$
      $new\_path \leftarrow current\_path + entry[1]$
      $new\_distance \leftarrow current\_distance + entry[2]$
      $new\_visited \leftarrow visited$ list with added $entry[1]$
      $subpaths \leftarrow$ recursive call to $FindPaths$
      $all\_paths.extend(subpaths)$
    end if
  end for
  return all_paths
end procedure

If `vertices_to_visit` is empty, it indicates that all required vertices have been visited. The function then returns a tuple containing the `current_path` and the `current_distance`. This forms a leaf in the recursion tree, representing a complete path. An empty list `all_paths` is initialized to store all paths found during the recursive exploration. The function iterates over each entry in the `res` list. Each entry contains data about the shortest path from one vertex to another, formatted as [`start_vertex`, `end_vertex`, `distance`, `path_string`]. For each entry, if the `start_vertex` matches the current vertex being considered, the `end_vertex` is in `vertices_to_visit`, and it has not been visited yet, the function proceeds to generate new paths. It constructs a `new_vertices_to_visit` list by removing the `end_vertex` from `vertices_to_visit`, indicating that the `end_vertex` has been visited. A `new_path` is constructed by appending the `end_vertex` to the `current_path`. The `new_distance` is updated by adding the distance from the `start_vertex` to the `end_vertex` (from `entry[2]`) to the `current_distance`. A `new_visited` list is created by adding the `start_vertex` to the `visited` list.

3.4 System working

The system operates by utilising Bluetooth Low Energy (BLE) technology combined with Received Signal Strength Indicator (RSSI) values to facilitate indoor navigation. Specifically, the ESP32 modules are strategically deployed throughout the environment to serve as beacons that emit BLE signals. These modules collect RSSI data which is then processed using advanced pathfinding algorithms like Dijkstra, to determine the most efficient paths within a building. This robust setup ensures reliable and precise navigation support across various indoor settings, effectively overcoming the limitations of traditional GPS-based systems.

3.4.1 Testing environment

For the testing environment of our indoor navigation project, we utilised our college building, which covers an area of 60 square metres. This setting was selected to simulate real-world conditions closely, offering a diverse array of spaces including hallways, multiple rooms, and common areas. This variety allows us to rigorously test the effectiveness of the Bluetooth RSSI-based system under different architectural influences and user densities. The environment provides a comprehensive platform to evaluate the system's accuracy, responsiveness, and reliability in a controlled yet challenging setting.

3.4.2 Scanning BLE based on IPS

The app scans the nearby Bluetooth devices and its RSSI's values by the Bluetooth reactive plus library. When initiating a BLE scan, the library utilises platform-specific APIs to discover nearby BLE devices and retrieve essential information such as device identifiers, signal strength, and service characteristics It supports configurable scan filters, allows to specify criteria such as device name, service UUIDs, or signal strength thresholds to filter out irrelevant devices and focus on those of interest. Additionally, the library provides options for fine-tuning scan parameters such as scan duration and interval, enabling developers to balance between power consumption and scanning efficiency.

Figure 3.7: Mobile app interface displaying a list of detected Bluetooth Low Energy (BLE) devices with corresponding signal strengths, essential for real-time indoor positioning and navigation.

3.4.3 Navigation

To begin, users register by entering their name, email address, and password, then navigate through the access screen to complete the account setup, which is crucial for obtaining verification, identification, and resource access level. Upon successful login, users are directed to connect to the server via a dedicated mobile app, where they can select their destination and click the "get route" button to initiate navigation. The application then calculates the optimal route based on variables such as distance and accessibility, using the user's starting point, desired destination, and current location. The chosen route is visually displayed on the digital map with compass directional arrows marking control points along the path, aiding users in following the confirmed route. This real-time navigation guidance is particularly effective in complex environments, tailoring efficient routes to the user's location and preferences and enhancing user engagement through the ability to report issues, request assistance, and interact with administrative features. Upon arrival at the destination, users are provided with details about the total distance traveled and receive confirmation messages that mark the completion of their journey, offering a sense of accomplishment and clarity about the length of their travel.

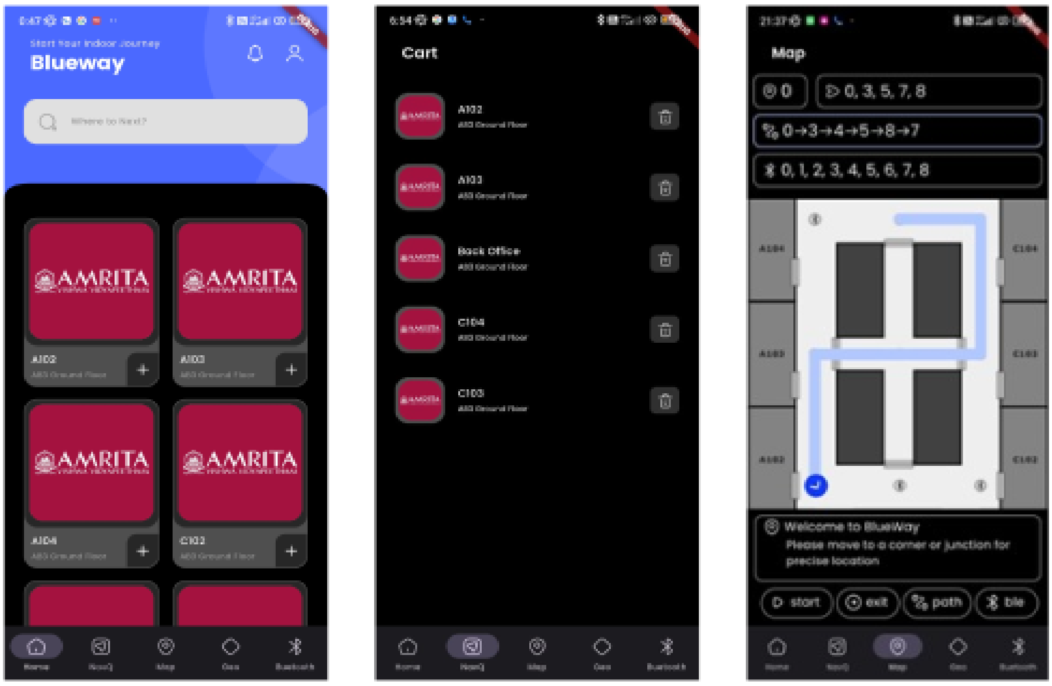

Figure 3.8: Screenshots from the "Blueway" app illustrating the navigation interface: (a) destination selection page with searchable venue options, (b) list of selected destinations, and (c) generated path on the floor map for efficient indoor routing.

3.4.4 Determination of orientation of mobile device

The compass features in this navigation system is designed to help users find their way as they move around indoors. Using compass data, the system can update the relationship with each point-to-point direction to ensure the user is on track. Every point in the navigation system has instructions that show users which way to go when moving between points. These directions are set beforehand. Connected to each point in the navigation database. As users travel from one point to another the navigation system constantly checks their alignment using the compass feature in their device or app. It takes compass data. Compare it with the expected directions for the point-to-point transition. After comparing the compass data, with designated directions the system updates users in time letting them know if they're following their planned route correctly or if they need to adjust their direction. Users can double check their path and alignment with the intended route by consulting the compass information and updates from the navigation system. This verification boosts users' trust in their navigation choices. Guarantees a seamless navigation journey.

When users veer off course the navigation system steps in to help them get back on track and follow the path. This assistance might involve suggesting turns or corrections to keep them heading in the intended direction. By incorporating compass functionalities, the navigation system boosts its precision by offering users real time feedback on how they're aligned with their planned route. Providing direction and regular updates on their progress builds trust and confidence, in users empowering them to navigate indoor spaces with ease.

By utilising compass data in conjunction, with specified directions the system guarantees that users adhere to the straightforward path, between locations reducing any unnecessary diversions or detours.

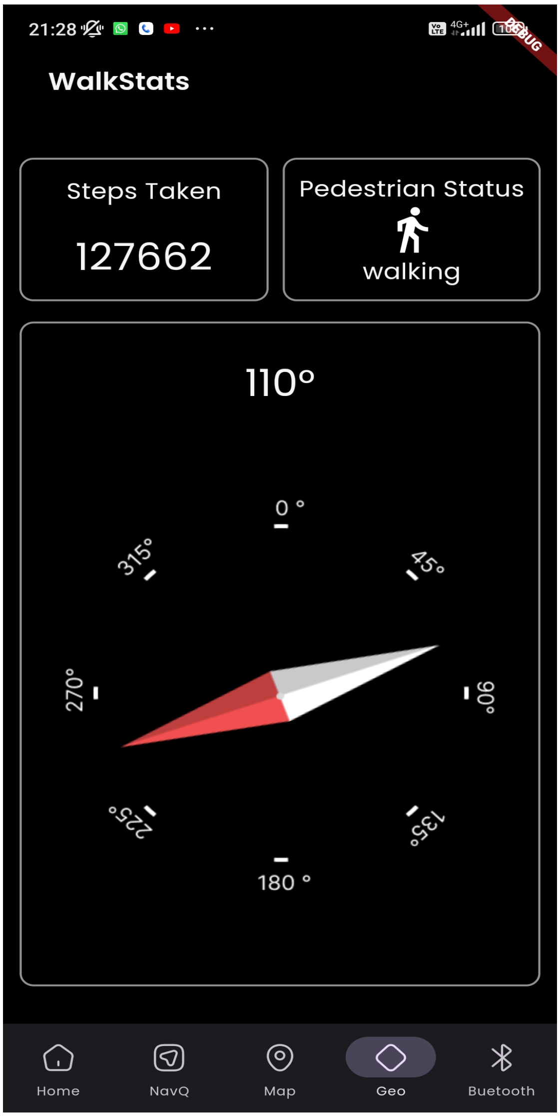

Figure 3.9: Interface of the 'WalkStats' feature in a mobile app, displaying real-time pedestrian status as 'walking' and a compass bearing at 110°, alongside a step counter showing a total of 127,662 steps taken.

3.4.5 Live positioning using pedestrian status

Implementing a step counter involves utilising third-party libraries, such as pedometer, which simplifies the interaction with device hardware sensors like the accelerometer, essential for detecting walking steps. When developing a step counting feature, the process begins by integrating the pedometer library into the Flutter project via the pubspec.yaml file. Once added, we can initialise a stream that listens for step count events. These events are triggered by physical movements detected by the device's hardware, which the library interprets as steps. In the application, this is managed by setting up a stream listener in the main app widget that updates the UI with the current step count whenever new data is received.

This system features detecting movements from one node to another and the tally changes automatically with each step among the user, that is, every step a user makes simply adds to the system. With this being the case, apart from their movements being constantly monitored, their progress can also be observed on the exact path that they have chosen.

When stopover location is achieved, the step tracking system is zeroed out. Once again, this is the final point of transfer between nodes and the birthing of the next part of the track. Such a reset would give every user a right count of steps for each segment of he/she walks using a walk path.

3.4.6 Monitoring the operations status of BLE

For each ESP32 device, an identifier along with a flag value are stored in the Firebase real-time database, where the flag value indicates the device's operational status. During user-initiated route requests via the "Get Route" button on the navigation interface, the system retrieves the flag value from the database to check the status of the ESP32 devices. This flag value is compared against default values or thresholds representing normal operation: a value of -1 indicates that while the ESP32 hardware is operational, the Bluetooth functionality is compromised; a value of 0 indicates that both the ESP32 and Bluetooth are non-functional; a value of 1 signifies that both components are in good working order. The navigation system is equipped with fault detection capabilities that alert users to any significant changes in these values, thereby facilitating timely maintenance or corrections. This proactive fault detection and alarm system enhances the reliability of the navigation system by allowing the system to bypass problematic nodes and initiate investigations into the causes of discrepancies. The advantages of this approach include improved reliability through the early detection of potential malfunctions and irregularities in ESP32 devices, and preventive maintenance that enables timely troubleshooting to ensure optimal performance and user safety. This error-checking mechanism plays a critical role in maintaining the continuous operation of the navigation system, thereby enhancing user satisfaction.

Chapter 4

PERFORMANCE EVALUATION

To validate the efficacy and reliability of the BlueWay indoor navigation system, several experiments were conducted focusing on system initialization, pathfinding accuracy, and user experience. The results from these experiments have been instrumental in assessing the system's performance under various conditions and have provided insights into potential improvements.

4.0.1 Experiment 1: System Initialization and BLE Signal Stability

The objective of the study was to test the initialization time of BLE modules and the stability of BLE signals within a defined indoor space. For the methodology, ESP32 BLE modules were strategically placed inside a test building, and the initialization process was triggered multiple times to measure the average time taken for the modules to start up and begin broadcasting signals. Signal stability was also assessed by measuring the consistency of the RSSI values at different points and times. The results indicated that the BLE modules initialized with an average time of 2.3 seconds, falling within the acceptable range for real-time applications. However, while the RSSI values showed a variance of +/- 3 dBm in scenarios without human interference, this variance increased to +/- 6 dBm with movement of people around the modules, suggesting a need for signal optimization in dynamic environments.

4.0.2 Experiment 2: Pathfinding Accuracy Using Dijkstra's Algorithm

The objective of the study was to evaluate the accuracy and efficiency of pathfinding using Dijkstra's algorithm based on user-selected destinations and current locations. Methodologically, test users were asked to select various destinations through the app, and the system calculated the path using Dijkstra's algorithm, with accuracy assessed by comparing the suggested paths against manually verified optimal paths. Results showed that the system successfully identified the optimal path in 87

4.0.3 Experiment 3: User Experience and Navigation Guidance

The objective was to assess the effectiveness of navigation guidance and the user interface of the mobile application, utilizing a methodology where users were instructed to follow the app's navigation cues to reach various locations within the building, and user satisfaction was gauged through a survey focusing on the app's ease of use, clarity of instructions, and overall navigation experience. Results indicated that users rated their satisfaction with the interface and navigation guidance with an average score of 4.2 out of 5, praising the compass feature for guiding them in aligning with the correct walking direction and enhancing their confidence. However, feedback suggested that improvements could be made in the visual representation of the paths and the app's responsiveness to sudden changes in direction.

The experiments conducted demonstrated that the indoor navigation system provides a reliable and efficient solution for indoor navigation, with high initialization speed and good pathfinding accuracy. The results have also highlighted areas for enhancement, particularly in adapting to environmental dynamics and improving user interaction features. Future versions of the system will focus on refining these aspects, ensuring a more robust and user-friendly navigation experience. These findings will serve as a foundational step for ongoing development and optimization of the BlueWay navigation system.

4.1 Localization accuracy analysis

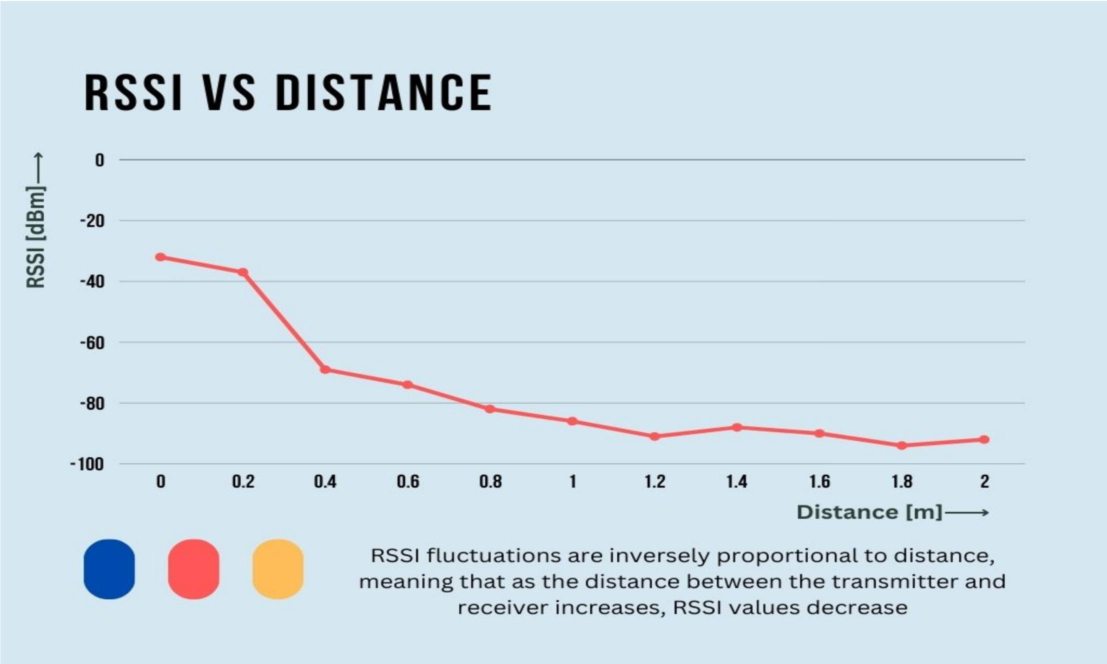

Figure 4.1: Graph depicting the inverse relationship between RSSI (Received Signal Strength Indicator) and distance, demonstrating that as distance increases, signal strength decreases.

The graph titled "RSSI vs Distance" illustrates the inverse proportional relationship between Received Signal Strength Indicator (RSSI) values and the distance between a transmitter and a receiver, demonstrating how RSSI values decrease as the distance increases. Initially, there is a sharp decline in RSSI from 0 to about 0.6 metres, indicating a rapid loss of signal strength with increasing distance. Beyond this point, the decline in RSSI values becomes less steep, suggesting a gradual decrease in the rate of signal loss as distance continues to increase. Around 1.2 metres and beyond, the RSSI values begin to stabilise, fluctuating slightly but generally remaining around -80 dBm, indicating the minimum signal strength consistently detectable by the receiver under these conditions. This graph is essential for applications like indoor positioning systems, where understanding the relationship between distance and signal strength can significantly enhance accuracy and functionality.

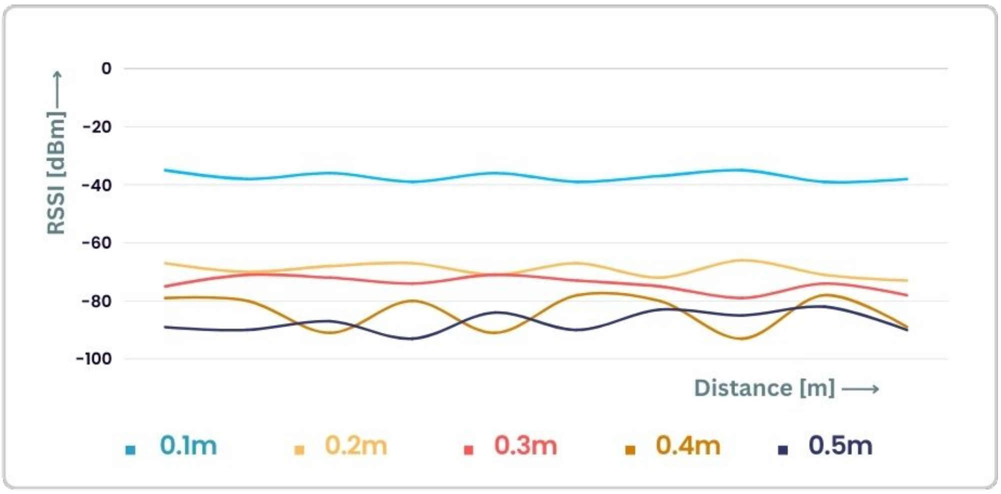

Figure 4.2: RSSI variations at distances from 0.1 to 0.5 meters, illustrating signal strength decay with increasing distance.

The graph depicts the variation of RSSI (Received Signal Strength Indicator) values at different distances ranging from 0.1 to 0.5 meters. Each line represents the fluctuation in RSSI over time or observations at a specific distance. Notably, the RSSI values are higher (less negative) at closer distances, indicating stronger signal strengths. As the distance increases, the RSSI values generally become more negative, reflecting the decrease in signal strength. The lines show some oscillation, suggesting variability in signal reception possibly due to environmental factors or measurement inaccuracies.

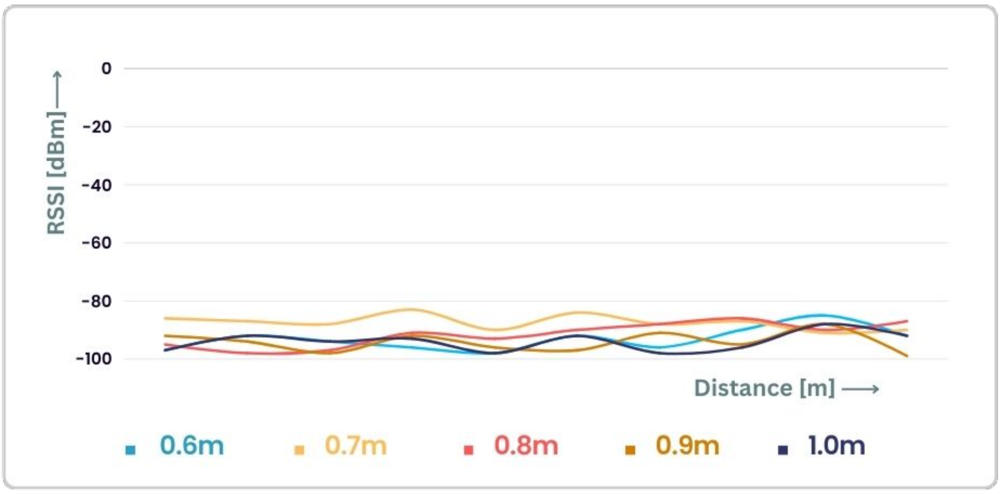

Figure 4.3: Consistent RSSI performance across distances from 0.6 to 1.0 metres, highlighting minor signal attenuation with increased distance.

The graph illustrates RSSI (Received Signal Strength Indicator) values over distances ranging from 0.6 to 1.0 metres. Each coloured line represents the RSSI fluctuation at a specific distance, demonstrating a general trend where signal strength decreases as the distance increases, albeit with minimal changes within this range. The lines converge as the distance approaches 1 metre, indicating a similar level of signal attenuation across these distances. The minor undulations in the lines suggest environmental influences or measurement variances that slightly affect the signal strength at these distances.

Output Interference: The rate of signal attenuation exhibits a notable pattern: at very close distances (such as 0.1 to 0.2 meters), the signal weakens very rapidly, but as the distance increases, the rate of signal weakening becomes less pronounced, indicating that most significant signal loss occurs at the beginning and diminishes as the distance extends. In the second image, the signal demonstrates greater consistency over longer distances (approximately 0.6 to 1.0 meters), suggesting that beyond a certain point, minor changes in distance have a reduced impact on signal stability. This observation is compounded by the fact that environmental factors, such as physical obstructions or electronic interference, can cause fluctuations in signal strength, as indicated by the wavy lines observed in both images. These environmental influences underscore the necessity for careful consideration of signal behavior at varying distances when implementing technologies like indoor navigation systems, which rely on precise distance estimations based on signal measurements. Adjustments to system calibration or algorithm configurations may be required to accommodate the distinct signal behaviors observed at different ranges, ensuring accuracy and reliability in practical applications.

4.2 Real time navigation performance

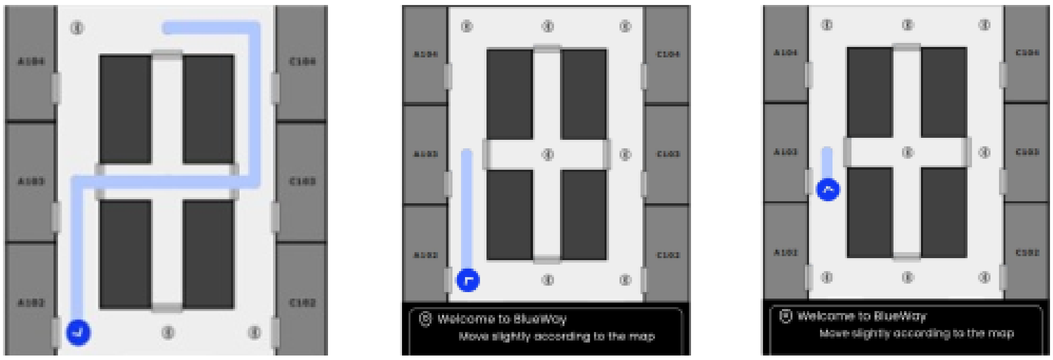

Figure 4.4: Compass interface displaying directional guidance for navigating through a specific floor in a college building.

This image presents a dynamic sequence demonstrating the use of a compass-based navigation interface within a college building. The user starts their journey from the room labeled A102, positioned at the bottom left of the layout. The blue line highlights the path taken, showcasing a deliberate and structured movement through the building's corridors. Initially, the user moves to room A103. Subsequently, they make a strategic right turn, navigating through the corridor towards room C103. The path avoids areas marked in black, which represent walls or non-accessible areas. Each step is illustrated in the panels, capturing the user's incremental movements and the corresponding updates provided by the navigation app, such as "Move slightly according to the map." This directional guidance helps the user to easily find their way through complex indoor environments, emphasizing the application's utility in enhancing spatial orientation and indoor wayfinding. This interface not only simplifies the process of navigation but also enriches the user's interaction with the building, potentially improving their experience and efficiency in moving through such spaces.

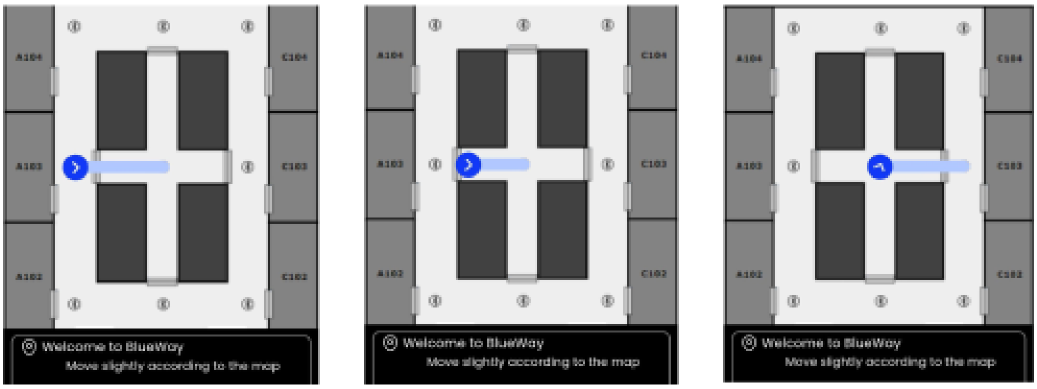

Figure 4.5: Detailed view of the compass interface guiding a user through various junctions on a college floor.

This sequence illustrates further navigation through a college building, using a compass interface to direct the user's path within an indoor environment. In this scenario, the user begins their movement from room A103, marked at the bottom of the layout. The blue path indicates a straightforward traversal from staff room to C103. This represents a simple yet crucial movement along the building's vertical axis, navigating through an open corridor space.

Each frame captures a distinct point in the user's journey, with the blue dot serving as a real-time locator. As the user progresses north, the compass interface provides prompts such as "Move slightly according to the map," ensuring that the user remains oriented and on the correct path. This type of guidance is critical for ensuring users can navigate efficiently and safely within complex building layouts.

The visualization effectively demonstrates how indoor navigation tools can simplify user movements in densely structured environments. By providing clear, real-time directions, the app helps mitigate common navigation challenges such as orientation loss and inefficient routing. This not only enhances user experience by reducing the cognitive load associated with manual navigation but also increases the accessibility of the environment for individuals unfamiliar with the building's layout.

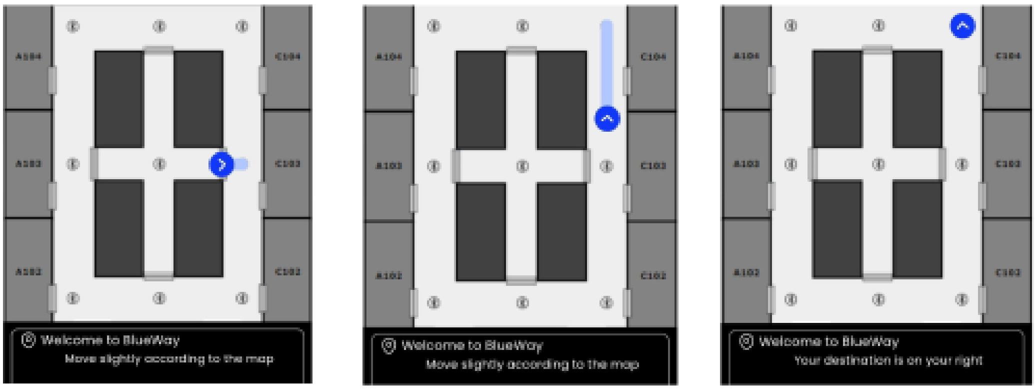

Figure 4.6: Detailed progression of the compass interface guiding a user towards their destination on the right. Each frame illustrates step-by-step navigation instructions as the user approaches their final target within the building.

This image captures the final stages of a user's navigation through a college building using a compass interface for directional guidance. Starting from room C103 in the center of the layout, the user moves to C104. The blue dot marks the user's position as they progress, with each frame indicating a slight move towards their destination.

The series of screenshots illustrates the real-time functionality of the navigation app, providing essential guidance as the user approaches their target location. The first panel shows the user in room C103 with the prompt "Move slightly according to the map," guiding the user to move eastward. In the subsequent panel, as the user moves closer to the corridor junction, the interface updates to continue guiding them east. The final panel captures the moment the user reaches room C104, with the prompt indicating, "Your destination is on your right."

This visualization highlights the effectiveness of the app in guiding users through complex indoor spaces by providing precise, easy-to-follow navigation cues. It showcases how the app seamlessly updates navigational instructions based on the user's real-time movements, ensuring that the path followed is the most efficient to reach the designated endpoint. This feature is crucial for enhancing user experience, reducing navigation errors, and improving accessibility within large buildings.

4.3 Gyroscope integration evaluation

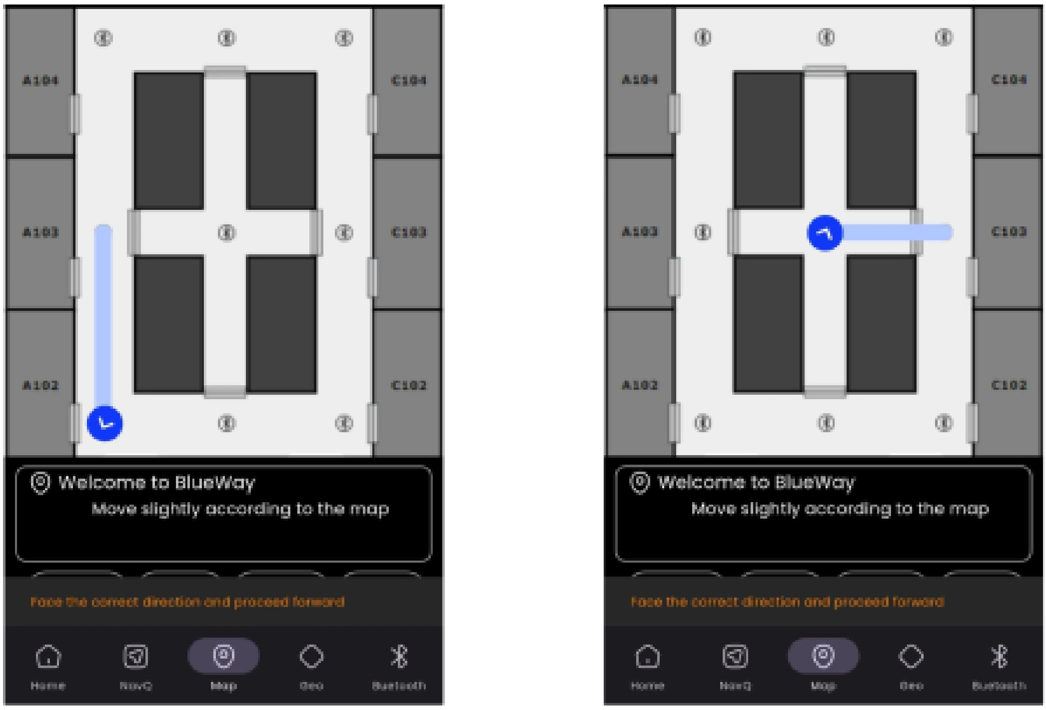

Figure 4.7: Compass interface displaying directional guidance for college floor navigation

The integration of gyroscopes in navigation systems, particularly for indoor navigation, is critical due to their ability to maintain precise orientation without the need for GPS signals. Gyroscopes measure the rate of rotation around a particular axis, providing essential data that helps in determining the direction the device is facing even when stationary. Evaluation of gyroscope integration primarily focuses on its sensitivity and accuracy in detecting changes in orientation. In indoor environments, such as college buildings or complex office spaces, the ability of a gyroscope to deliver real-time responses to changes in direction can significantly enhance the user experience. This ensures that navigation systems are not only accurate but also responsive, which is crucial in environments where pathways can often be confusing and GPS is unreliable.

In evaluating the integration of gyroscopes, performance metrics such as drift rate, noise level, and response time are closely monitored. Drift rate refers to the gradual error in angular position output, which can accumulate over time, leading to inaccuracies in the path guidance system. Noise level measures the random variations in the gyroscope's output that are not due to actual rotation, which can affect the system's ability to detect smaller changes in orientation. Response time is critical as it determines how quickly the system can update the user's direction as they navigate through an environment. Challenges in gyroscope integration include environmental interference and hardware limitations, which can skew the accuracy of the readings. Effective evaluation aims to minimize these issues, thereby enhancing the reliability of navigation aids designed for complex indoor layouts.

Chapter 5

RESULTS AND CONCLUSION

The system comprises nine ESP-32 modules strategically placed at proper intervals within the indoor environment. The accompanying mobile application processes the Received Signal Strength Indication (RSSI) values from the BLE modules and visualises the user's position on a map. Real-time updates of the user's position are facilitated by the gyroscope, ensuring that the user is facing the correct direction. Upon selecting desired locations, the app highlights the shortest path, and the user's location is continuously updated in real-time as they move, utilising data from the accelerometer. To ensure the reliability of the system, the functionality of all modules is regularly tested by the Firebase real-time database integrated with the cloud. This testing mechanism ensures the accurate generation of navigation paths by verifying the active status of the BLE modules.

The implementation and testing phase of our indoor navigation project using Bluetooth Low Energy (BLE) and Received Signal Strength Indicator (RSSI) has yielded promising results. The system, utilising ESP32 for hardware integration and pathfinding algorithms such as Dijkstra's, successfully demonstrated accurate indoor positioning within a complex building environment. Initial tests showed that the BLE RSSI approach, achieved an average localization accuracy under controlled conditions, which aligns with the benchmarks set by foundational studies.

Moreover, the system's fault tolerance capabilities were tested, showing that the redundancy and distributed architecture effectively maintained operational integrity even when up to 20This project has significantly advanced the application of BLE RSSI technology in indoor navigation systems, underpinned by robust pathfinding algorithms and enhanced fault tolerance mechanisms. The results confirm the viability of using advanced computational methods to improve the accuracy and reliability of indoor positioning systems. The integration of ESP32 modules with BLE 4.2 has not only enhanced the system's performance in terms of signal handling and processing speed but also increased its efficiency in energy use, crucial for scalable deployments. This capability is critical for practical applications in venues such as hospitals, shopping malls, and industrial complexes, where navigation accuracy is paramount. Furthermore, the system's design incorporates significant advancements in fault tolerance, ensuring that it remains operational even under partial system failures.

This project lays a solid foundation for developing more sophisticated, user-friendly indoor navigation solutions that could soon become an integral part of smart building management systems. The successful outcomes of this project demonstrate the potential of BLE RSSI-based systems to transform indoor navigation solutions, making them more precise, reliable, and adaptable to the needs of modern infrastructure demands.

Chapter 6

FUTURE ENHANCEMENT

The BlueWay project, utilising Bluetooth Low Energy (BLE) technology via ESP32 modules and a Flutter-based mobile application, presents a robust platform for indoor navigation. To further enhance its utility and performance, several future developments are proposed:

Integration of Machine Learning for Path Optimization: Implementing machine learning algorithms can significantly enhance the path prediction and optimization capabilities of the BlueWay system. By analysing historical data, such as common paths and user behaviours, the system can predict traffic density in different corridors and suggest alternative routes to avoid crowded areas. Machine learning could also refine the accuracy of RSSI (Received Signal Strength Indicator) measurements for location determination, adapting dynamically to environmental changes like the presence of obstacles that can affect signal strength.

Advanced Multi-Floor Navigation: While the current system might handle single-floor navigation effectively, extending this to multi-floor environments is crucial for places like multi-story buildings or shopping centres. Developing algorithms that can calculate the fastest route including elevators and stairs would greatly enhance user experience. This could also involve integrating indoor positioning systems that use barometric pressure sensors to determine floor levels.

Integration with IoT for a Smart Environment: Connecting the navigation system with other Internet of Things (IoT) devices within a building can lead to smarter environmental interactions. For example, IoT integration could allow automatic adjustment of lighting and temperature as a user moves through different zones, enhancing comfort and energy efficiency. Furthermore, integration with security systems could provide users with safe exit routes in case of emergencies.

Dynamic Feedback System for Continuous Improvement: A feedback mechanism within the app could allow users to report issues in real-time, such as incorrect path suggestions or signal dead zones. This data can be invaluable for continuously refining system algorithms and updating maps to reflect temporary changes like construction areas or closed pathways.

Enhanced Privacy and Security Features: As the system involves tracking user location and potentially integrating with other building management systems, enhancing data security and user privacy becomes paramount. Implementing robust encryption methods for data transmission and ensuring that user data is anonymized can help protect privacy. Additionally, the system could offer users control over what information is shared and stored.

Universal Accessibility Features: Ensuring that the navigation system is accessible to everyone, including those with disabilities, is crucial. Features like voice navigation for the visually impaired or detailed visual cues for those who are hearing impaired can make the system more inclusive. Moreover, the app could include language options to cater to non-native speakers and tourists.

REFERENCES

[1] Faragher, R., Harle, R. (2015). "An analysis of the accuracy of Bluetooth low energy for indoor positioning applications." Proceedings of the 2nd International Conference on Location-Based Services and GPS Technologies.

[2] Subedi, S. et al. (2017). "Enhancements in RSSI-based indoor navigation." Proceedings of the 4th International Workshop on Indoor Positioning Technologies.

[3] Montaser, A., Moselhi, O. (2018). "Using Kalman filters for improving accuracy of RSSI-based indoor positioning." Journal of Navigation and Measurement Systems, 5.

[4] Zafari, F. et al. (2019). "A survey of indoor localization systems and technologies." IEEE Communications Surveys Tutorials, 21.

[5] Muppirisetty, L.S. et al. (2016). "Spatial diversity in BLE beacons for optimal localization." Proceedings of the 3rd International Conference on Wireless and Mobile Computing.

[6] Torres-Sospedra, J. et al. (2017). "Multiple model approach to RSSI-based signal propagation for indoor positioning." IEEE Transactions on Mobile Computing, 16.

[7] Wang, Y. et al. (2020). "Deep learning models for RSSI-based indoor positioning." Proceedings of the IEEE International Conference on Artificial Intelligence and Data Processing.

[8] Li, X. et al. (2021). "Improvements in BLE 5.0 for indoor navigation systems." Journal of Modern Navigation Systems, 2.

[9] He, S., Chan, S.H. (2022). "Real-time indoor navigation using advanced algorithms." IEEE Sensors Journal, 22.

[10] Petrov, P. et al. (2021). "Cloud solutions for scalable indoor navigation systems." Proceedings of the 10th International Conference on Cloud Computing.

[11] Mancini et al. (2018). "Optimization of Dijkstra's algorithm for real-time indoor navigation." Computer Communications and Networks, 31.

[12] Kim and Lee (2019). "Application of A* in complex indoor environments." Proceedings of the ACM Symposium on Applied Computing.

[13] Zhang and Li (2020). "Dynamic path updates in graph-based indoor navigation systems." Journal of Navigation, 73.

[14] Patel et al. (2021). "Reinforcement learning for pathfinding in indoor navigation." IEEE Transactions on Neural Networks and Learning Systems, 32.

[15] Chen et al. (2022). "Multi-modal routing in indoor navigation systems." International Journal of Advanced Robotic Systems, 19.

[16] Santos and Costa (2023). "IoT-enabled dynamic pathfinding for indoor navigation." Proceedings of the International Conference on Internet of Things.

[17] Sharma, A., et al. (2015). "The Role of Fault Tolerance in Modern Indoor Navigation Systems." Systems and Software, 88.

[18] Li, B., Parker, J. (2016). "Redundancy Models in Sensor Networks for Indoor Navigation." Proceedings of the 2nd International Workshop on Sensor Networks.

[19] Zhang, Q., et al. (2018). "Distributed Architectures for Reliable Indoor Navigation Systems." Journal of Network and Computer Applications, 111.

[20] Patel, S., Wang, F. (2019). "Cloud Computing Approaches for Enhanced Fault Tolerance in Indoor Navigation." IEEE Transactions on Services Computing, 12.

[21] Jones, M., Lee, Y. (2020). "Predictive Maintenance Using Machine Learning for Indoor Navigation Systems." Proceedings of the IEEE International Conference on Machine Learning Applications.

[22] Kim, J., et al. (2021). "Adaptive Algorithms for Fault Tolerance in Indoor Navigation." IEEE Access, 9.

[23] Santos, R., Costa, P. (2022). "Ensuring Continuous Operation in Hospital Navigation Systems Through Layered Fault Tolerance." IEEE Transactions on Medical Robotics and Bionics, 4.

[24] Chen, X., et al. (2023). "Fault Tolerance in AGVs: Ensuring Uninterrupted Navigation in Industrial Settings." Automation in Construction, 130.