SMARTSHOP NAVIGATOR - A GPS-BASED INDOOR NAVIGATION SYSTEM FOR SUPERMARKET PRODUCTS

BY

TAM YU YOU

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)

FEBRUARY 2026

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Ayu Norafida binti Ayob**,** for her invaluable guidance, constructive feedback, and continuous support throughout the development of this project. Her expertise and encouragement have been instrumental in helping me refine my ideas and overcome challenges during this research. I would also like to extend my appreciation to the lecturers and staff of the Faculty of Information and Communication Technology (FICT), Universiti Tunku Abdul Rahman (UTAR), for providing the necessary resources and academic environment that made this project possible. Special thanks go to my friends and course mates for their collaboration, motivation, and for sharing ideas that enriched my learning experience. Finally, I am sincerely grateful to my parents and family members for their endless love, encouragement, and unwavering support throughout my studies.

ABSTRACT

Navigating large supermarkets can be time-consuming and frustrating for customers. **SmartShop Navigator** is proposed to address these challenges. SmartShop Navigator – a mobile application designed to enhance shopping experience by providing a real-time indoor navigation to supermarket products. **Bluetooth Low Energy (BLE) beacons** are integrated in this system for precise indoor positioning incorporating with AI to calculate the critical path and display in the mobile application to visually guide users to their selected products. The project leverages mobile app development (Flutter/React Native), Raspberry pi/Arduino for hardware control and Firebase for backend data management. A fully functional Android/iOS application that increases customer satisfaction, improves faster searches, and improves shopping efficiency is the expected outcome. This work contributes to advancements in **Indoor Positioning Systems (IPS)**, **IoT integration in retail**, and **user-friendly mobile navigation solutions**, offering a scalable and cost-effective system for supermarkets.

Area of Study (Minimum 1 and Maximum 2): Mobile Application Development, Indoor Positioning Systems (IPS), IoT

Keywords (Minimum 5 and Maximum 10): Indoor Navigation, BLE Beacons, Smart Retail, Mobile App, IoT

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IPS	Indoor Positioning System
GPS	Global Positioning System
NLoS	No Line-of-Sight
Wi-Fi	Wireless Fidelity
BLE	Bluetooth Low Energy
RFID	Radio Frequency Identification
UWB	Ultrawideband
RSSI	Received Signal Strength Indicator
LED	Light-emitting diode
LPS	Local Positioning Systems
HPS	Hybrid Positioning systems
IoT	Internet of Things
kNN	K-Nearest Neighbor
MLP	Multilayer Perceptron
GPR	Gaussian Process Regression
PDR	Pedestrian Dead Reckoning
IWD	Inverse Weighted Distance
VLP	Visible Light Positioning
IMU	Inertial Measurement Unit
SMEs	Small and Medium-sized Enterprises
IDE	Integrated Development Environment
MAC	Media Access Control
PHP	Hypertext Preprocessor
MySQL	My Structured Query Language
HTTP	Hypertext Transfer Protocol
API	Application programming interface
JSON	JavaScript Object Notation
TCP	Transmission Control Protocol
IP	Internet Protocol
ESP32	Espressif Systems
XAMPP	**X** (Cross-Platform) + **A**pache + **M**ariaDB (formerly MySQL) + **P**HP + **P**erl

OTA	Over-The-Air
MLE	Mean Localization Error
AoA	Angle of Arrival
AoD	Angle of Departure
SKU	Stock keeping units
AR	Augmented Reality
TTS	Text-to-Speech
POS	Point-of-Sale

Chapter 1

Introduction

Customers found that it is extremely difficult to navigate products that customers need in contemporary shopping environments, especially in a huge supermarket carrying countless products. Nowadays, eCommerce has grown rapidly. The U.S. Census Bureau reports that around 80% of retail purchases in the United States occur in physical stores. This indicates that the in-store experience is still an essential component of the retail ecosystem despite the growth of online purchasing [1]. Therefore, customizing the shopping experience is crucial, at least 89-96% of consumers spend more than ten minutes on researching or searching their purchases not only online but offline [2]. According to a National Council of the Blind Ireland (NCBI) study, 79% stated that the shop is not easy to navigate through the different zones [3]. Related to the task outcomes, including making timely and effective purchases, are what motivate consumers who are focused on practical advantages. The consumer can decide when and where to shop, and convenience improves search efficiency. Since most of the retail technology available in-store saves money and time, retailers typically emphasize the practical advantages of technology [4]. There is a technical gap in the retail experience because indoor positioning systems (IPS) in commercial spaces are still in stages of development, notwithstanding the outstanding precision of outdoor GPS navigation [5].

In Malaysia, researchers state that internal shop environment composed of both visible and invisible elements such as layout of the supermarket. It plays an essential role in satisfying customers to purchase or stay longer in the supermarket. For instance, it has been discovered that store layout plays a significant role in increasing customer satisfaction [6]. This issue is especially acute at supermarkets like Tesco, AEON, and Giant, where it is challenging to locate products due to high shelves and crowded walkways. Study finds that customers of groceries and personal care products typically fall into a similar age range when examining the impact of age on the purchase of product categories. The average age of customers for these categories was 35.8 years for grocery and 33.8 years for personal care products, with a comparable percentage of buyers across all 4 age groups [7].

SmartShop Navigator addresses this issue by creatively combining IoT technologies with mobile technology. The technology provides customers real-time, shelf-specific guidance by BLE beacons. By adding new visual confirmation components and expanding on previous IPS research, this method improves navigational usability.

1.1 Problem Statement and Motivation

Navigating modern supermarkets has become an increasingly difficult experience for users due to various systemic inefficiencies in wayfinding systems. The principal challenges come out in three critical areas that together reduce shopping experience and operational efficiency. Large supermarkets are difficult for many users to navigate, and they frequently spend too much time looking for specific products. Inefficient navigation not only causes frustration but can also result in missed purchases and reduced customer satisfaction. According to [8], customers usually deal with problems including traffic, difficult-to-find products, and ineffective store layouts, all of which distract from the entire shopping experience and may restrict revenue growth.

First and foremost, traditional navigation methods are imprecise, which leads to unnecessary difficulties. The current solutions rely on ceiling signage that only shows general categories like “Beverages”. This solution will be easily ignored by customers and only useful when customers realize that they are not able to find the products by themselves. This will be causing time-consuming while can only follow the signage with general information, not the exact position of the products. Second, the static store maps are not interactive. Static store maps are commonly found at supermarket entrances or within mobile apps, fall short of today’s customer demands as they are not real-time. It fails to integrate real-time positioning, some of the static maps do not show the current location of the customers. Static maps only provide the location of the products and do not provide real-time navigation for customers, leading customers to memorize the paths. The shortest route is not highlighted, causing inefficient shopping trips for customers [9].

Third, the issue of staff dependence on basic navigation support is becoming more and more troublesome for supermarkets. For instance, AEON Malaysia act as biggest supermarkets in Malaysia are having a huge layout and huge number of products. Staff may not memorize all the products and the position of the product. Therefore, it will cause misinformation to be given to the customers and time wastage will occur. Next, according to [10] poor indoor navigation reduces customer loyalty and visit repetition. A confusing and time-consuming in-store experience discourages users from returning. It will directly impact on customer loyalty towards the supermarket. Poor navigation not only frustrates customers but also makes it harder for them to discover new products or promotions. Furthermore, poor indoor navigation will compromise the safety of everyone in the supermarket. If customers cannot get in and get out of the supermarket quickly when an incident happens. Danger of panic behavior will easily occur when they are not able to stampedes quickly. Incidents such as fires or shootings leave customers in panic when seeking for exits.

Moreover, GPS limitations indoors require alternative positioning technologies. GPS works exceptionally well for outdoors navigation, but inside signal attenuation and reflection from building materials significantly decrease its accuracy. GPS is not suited for precise indoor navigation since its typical accuracy of 5 to 10 meters is insufficient for differentiating between aisles or shelves. For efficient in-store navigation, technologies such as BLE beacons are therefore needed to provide the sub-meter precision [11]. Additional challenges that older people and people with disabilities face difficulties in supermarkets such as reading signs, reaching high or low shelves, and navigating trolleys. Poor navigational aids make these difficulties worse, making shopping even more intimidating for vulnerable groups [12].

A significant convergence of market demands, technology advancements, and measurable benefits resulted in the creation of SmartShop Navigator, which addresses long-standing inefficiencies in supermarket navigation. This project is more than just a technical solution; it addresses the core changes in customer behavior, makes use of the most recent developments in indoor positioning technology, and provides measurable benefits to the entire retail ecosystem.

The limitations of current techniques are one of the major motivations for this study. Although some supermarkets offer automated maps or section numbers, these tools frequently don't work in real time and cannot direct customers to the real shelf position of a product. For instance, Google Maps Indoor provides accurate information about basic floor plans and outdoor navigation is limited. Traditional techniques, such static maps and ceiling signage, are unable to offer personalized, real-time guidance, particularly in complex retail layouts [13]. Supermarkets must update the in-store experience because the growth of e-commerce has increased demands for efficiency and convenience in physical shopping [14]. Technology such as Bluetooth beacons can help with it. Beacons are small radio transmitters that communicate with nearby mobile devices via Bluetooth. These beacons only serve as transmitters; they don't store any data. Indoor navigation technology is one of the most popular beacon applications, allowing customers to find their way across the supermarket. This technology can be used by indoor positioning system businesses to develop indoor navigation applications and indoor positioning systems for both small and large infrastructures, regardless of the size of the venue [15]. Furthermore, Retailers can also collect useful information about the behavior of customers via indoor positioning systems. This information facilitates targeted marketing and customized promotions in addition, optimizing the store layouts and product placements, which increases customer engagement and operational efficiency [16]. The SmartShop Navigator project is motivated by the opportunity to leverage these technological advances to deliver a seamless, data-driven, and user-friendly shopping experience.

Figure 1.1 Location Services

1.2 Objectives

The primary objective of the SmartShop Navigator project is to develop a fully functional smartphone application that reduced significantly the time and effort needed to find the products in a supermarket, hence, improving the shopping experience. Several objectives have been set to accomplish this overall objective.

The first objective is to design and implement an accurate IPS that can determine a customer's location within the supermarket to within 1-2 meters. Reliable navigation assistance requires this level of accuracy, which will be achieved by combining algorithms for smartphone sensor data fusion with BLE beacon triangulation.

The second objective is to design a user interface that is responsive and user-friendly so that customers can quickly look for the products and determine where the products are in the supermarket. A store map with clear visual path for product placements, user position, and recommended path will be displayed on the map. With one-handed operation and unambiguous visual and oral instructions will be given to customers and ensure the interface is still user-friendly when shopping.

The third objective is to develop effective path-finding algorithms that can generate critical paths through the supermarket based on the user's shopping list. To reduce walking distance and shopping time, these algorithms will consider factors such as aisle organization, product proximity, and store layout. The ability to recalculate paths flexibility will be available when customers add or delete products from their shopping list.

Next, establish a dependable and maintainable database structure for the storage and retrieval of the products information. Fast product search queries and frequent updates to reflect changes in inventory or product position must be supported by this database. Performance and data integrity will be prioritized in the database design, which will be adaptable enough to handle future growth.

Furthermore, the next objective is to apply Arduino devices to create a prototype implementation of the BLE beacons. The IPS will be tested on this prototype, which will also be used to determine the accuracy of the location recognition algorithms under different conditions.

Finally, the project aims to conduct comprehensive testing and evaluation of the application in a simulated supermarket environment. At this point, the similar layout of this supermarket environment is the school library with many high and similar material shelves. User trials will be conducted as part of this testing to evaluate the application's performance, usability, and navigation accuracy. The application will be improved based on the output from the tests. By fulfilling these objectives, the SmartShop Navigator project aims to show how useful and feasible indoor navigation technology is and the improvements of the in-store shopping experience.

1.3 Project Scope and Direction

The objectives of this project are to present a complete prototype of comprising of three significant parts:

1. A mobile application that allows users to search for products and receive precise, real-time navigation guidance to location.
2. An infrastructure of BLE beacons implemented in a simulated supermarket design with Arduino-based beacon tools to facilitate the use of meter-accurate indoor positioning.
3. A back-end system that has a database to maintain product information and layout mapping, and algorithms to provide real-time positioning, to support real-time routing.

The problem of inefficient indoor navigation in supermarkets where conventional GPS signals are ineffective, and other more static maps or signage is inadequate is solved. It has been proved that BLE-based indoor positioning is affordable, low-energy and fairly accurate to retail settings [17]. Technologically, BLE beacons have very long battery life and use low energy-up to 30 times less than Wi-Fi-which makes such beacons viable in retail implementations. However, noted that currently common BLE systems provide about 2-5 meters of range without spacing out beacons, which is reasonable in many retail applications but could be a constraint on high-resolution shelf-level tracking without optimization [18]. In the end, the project will result in the creation of the new system design and working prototype that will integrate mobile applications, indoor positioning, and backend intelligence and will help save time for the users spent finding the products and enhancing the smart retail experience.

1.4 Contributions

The SmartShop Navigator project is a significant step in the field of retail technology through the implementation of an integrated indoor navigation platform that will integrate IPS, mobile applications, and IoT hardware to provide a smooth experience when shopping in-store. This solution is more precise, real-time and personalized direction unlike the static maps/signage, which depends on the precise location of a customer within a supermarket. The presented innovation helps solve the systematic problem of inefficient wayfinding, lengthy search time, and customer frustration in the retail sector, but it also opens the prospect of further developments in smart shopping space. The project will add value to both commercial retail practice and scholarly research in the field of indoor positioning and human-centered computing through the provision of both a practical solution and a prototype based on research.

For implementation, this project has shown how cost-effective solutions can be created without compromising functionality. The positioning system is based on BLE beacons that are both energy-efficient and comparatively inexpensive. In combination with Arduino microcontrollers, they offer a scalable supermarket infrastructure of small and large sizes. This is noteworthy since most of the currently available indoor positioning systems are either expensive to install (e.g., UWB) and are therefore inaccessible to small and medium-sized retailers [19]. With low-cost elements, SmartShop Navigator helps to democratise the high-end technologies in navigation so that the smaller business can enjoy the same customer experience additions that only the large-scale retailer can deliver. The system is efficient and can produce safe collision-free routes with minimal computational effort through a dynamic implementation of the Grid-Based Manhattan routing algorithm. This enables customers to easily manoeuvre through the store fixtures and get to the location of the product they need in a short time, thus saving a lot of time on general research.

The other significant contribution is UX design of the mobile application. In retail, the existing IPS solutions tend to focus on technical feasibility and ignore usability and access. The SmartShop Navigator gives emphasis on the visual navigation cues and one-handed navigation coupled with multi-modal direction (visual and sound) making it inclusive to a wide range of users including the elderly customers and disabled individuals. The project therefore does not just increase convenience but also increases accessibility, which is in line with the principles of universal design and in the greater goals of society.

The importance of the project also stretches to the data-based retail knowledge. Using anonymized movement data collection, the system will assist retailers to map customer movement, know areas with heavy traffic, and maximize the positioning of products. Earlier BLE-enabled system deployments in retail outlets have indicated that the analytics can directly enhance sales and guide marketing decisions [20]. As an example, shopping cart systems built using BLE beacons allowed targeted promotions and location accuracy of up to 91.7 percent with larger partition areas, demonstrating the two-fold usefulness of these systems to both customers and retailers [21].

SmartShop Navigator can replicate the same results, thus multiplying customer loyalty and efficiency of the operation. Overall, the effect of SmartShop Navigator is three-fold: (i) customer satisfaction is increased through a shorter search time and more accessible shopping; (ii) retailers, and the SMEs are empowered with cost-effective IPS technology, and (iii) the solution is presented into the field of academic research in indoor positioning and smart retail spaces. Put collectively, these contributions show that the project is not just a desirable one in terms of technology, but it has a significant meaning regarding the values of society and business.

1.5 Report Organization

This report has been divided into seven chapters to present a comprehensive overview of the design of the indoor supermarket navigation system: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology/Approach, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Discussion, and Chapter 7 Conclusion and Recommendation.

Chapter 1: Introduction The first chapter lays the groundwork for the project, including the problem statement, motivation, objectives and scope of the project. This chapter discusses how the proposed system will contribute to improving the customer shopping experience and provides a brief overview of the report structure.

Chapter 2: Literature Review The second chapter reviews the state of the art in indoors positioning technologies and applications. It compares the features and limitations of existing solutions, examining different hardware platforms, firmware, database and routing algorithms, to justify the technological choices made for this project.

Chapter 3: System Methodology/Approach This chapter provides an overview of the system design and methodology of the application. It offers a general overview of the system structure, including use case and activity diagrams that describe the interactions between users and administrators with the navigation and database management functions.

Chapter 4: System Design In chapter 4, the technical designs for the system are discussed. It includes the system block diagrams, hardware architecture of the BLE beacons and ESP32 microcontrollers, and describes the operations of the interaction between the mobile application (Flutter), the backend server (PHP/MySQL) and the mathematical algorithms used for route planning.

Chapter 5: System Implementation Chapter 5 outlines the implementation of the system design. It details the implementation of the hardware and software installation, server configuration, incorporation of the dynamic Grid-Based Manhattan algorithm, and management of the administrative dashboard. It also provides screenshots of the system in operation and implementation problems and challenges that were encountered.

Chapter 6: System Evaluation and Discussion The sixth chapter discusses the system's performance and accuracy. It explains the arrangements of testing and assesses the system's performance in achieving the original project goals, as well as addressing issues like the noise of the hardware signal and the effectiveness of the vector projection map-matching techniques applied.

Chapter 7: Conclusion and Recommendation The seventh and final chapter recaps the entire project and concludes with a final remark on the success of the system in meeting its objectives. The chapter also presents recommendations for future work and possible hardware and software upgrades to enhance the indoor navigation system.

Chapter 2

Literature Review

2.1 Previous Works on BLE-Based IPS with The Implementation of IoT

IPS have garnered significant research interest due to the limitations of GPS in indoor environments. With an accuracy range of 2 to 6 meters, GPS is highly effective outdoors. However, indoors, it faces challenges such as NLoS, poor signal strength, and low accuracy, which rendering it unsuitable for indoor navigation applications like supermarkets. This limitation has driven the development of various IPS technologies designed to provide accurate indoor location tracking and navigation. According to the previous research [22], IPS are technologies that enable the determination of the position of objects or people within buildings. Unlike GPS, IPS must overcome obstacles such indoor structure interference, multipath effects, and signal attenuation. IPS solutions can be broadly categorized into self-positioning systems, where the device determines its position using signals from anchor nodes, and remote-positioning systems, where the infrastructure performs the calculation based on signals received from the device. Signal measurement and position computation utilizing geometric or statistical techniques, such as trilateration and triangulation, are the two main stages of the positioning process. There are several technologies that underpin IPS. Firstly, Radio Frequency-Based Technologies such as Wi-Fi, BLE, RFID, and UWB are commonly used due to their widespread availability and ease of deployment. Due to its compatibility with smartphones and low battery consumption, BLE is popular. BLE beacons combine RSSI algorithms and trilateration techniques have demonstrated positioning accuracy between 0.2 and 0.4 meters in controlled environments. However, accuracy can be affected by physical obstructions and signal interference, necessitating calibration and the deployment of huge number of beacons, which can increase costs. Second, Magnetic-Based Positioning, where this approach exploits anomalies in the Earth's Geomagnetic field or artificially generated magnetic fields within buildings. Since smartphones include magnetometers, magnetic-based IPS do not require extra hardware. These systems offer stable signals and robustness, especially when magnetic fingerprinting techniques are used. However, device and user diversity can introduce variability in magnetic readings, it will affect accuracy. Next, Inertial Sensor-Based Positioning, which utilizes accelerometers and gyroscopes. Inertial sensors track movement without relying on external signals. Although they do not need infrastructure to be installed, the accuracy degrades over time due to sensor drift. The combination of inertial sensors with magnetic sensors through filtering techniques like the Kalman filter can improve long-term orientation and positioning accuracy. Lastly, Satellite-Based Positioning, where GPS is the primary satellite-based system but is limited to outdoor use due to signal blockage indoors. Its average user range error is less than 7.8 meters outdoors, but it is ineffective for indoor environments such as supermarkets. The RSSI algorithm is frequently used in range-based IPS to estimate the distance between transmitters and receivers by examining signal power levels. To improve accuracy, RSSI is commonly paired with trilateration, which determines position based on distances from at least three known reference points. According to [23], high accuracy indoor positioning can be achieved by combining RSSI with trilateration and filtering techniques like Kalman filter. Moreover, hybrid approaches that combine multiple technologies such as BLE and Wi-Fi, or inertial and magnetic sensors can leverage the strengths of each to improve overall performance.

IPS has become increasingly essential due to the limitations of GPS technologies indoors, where GPS signals are obstructed by building materials and cannot provide accurate location information. Due to the low cost, low power consumption, and wide compatibility, BLE beacons are among the most promising new technologies for indoor localization. In their thorough analysis of non-GPS positioning systems, [24] emphasized BLE-based systems as scalable and economical, considering their difficulties with accuracy and calibration. In this research, evaluations of fingerprinting and triangulation methods utilizing BLE, researchers discovered promising accuracy levels but recognized significant vulnerability to surrounding interference. BLE beacons use RSSI to estimate the distance between the transmitter and receiver. Nevertheless, this method is sensitive to device variability, multipath effects, and obstruction. Refining RSSI information to enhance localization. Kalman filter showed greatly improved localization accuracy by lowering signal noise. This research demonstrated that RSSI-based location is feasible when combined with the appropriate signal processing methods. Another popular method is fingerprinting, which involves mapping signal levels offline and classifying positions offline using machine learning techniques. Both kNN and

MPL neural networks were applied under a controller environment. Both achieved over 90%, with MLP producing slightly higher accuracy.

According to [25], researchers conducted a comprehensive study on low-cost utilizing BLE technology, primarily designed for monitoring the daily living patterns of elderly people and individuals with disabilities in indoor environments. Researchers mentioned that people spend approximately 90% of their time indoors, making accurate indoor positioning essential for various applications including healthcare monitoring and assisted living. Although GPS works highly effective outdoors, signal obstruction by building materials prevents it from providing accurate positioning. For indoor positioning, this limitation has led researchers to investigate alternative technologies such as cameras, ultrasound, radio frequency, infrared, Wi-Fi and Bluetooth. An IPS design is proposed that combinations of wearable BLE beacons with several Raspberry Pi devices that are equipped with BLE antennae positioned strategically around the indoor environment. The RSSI values from the BLE beacons are automatically detected by this system, which then analyses the information to find the user's location. Trilateration and fingerprinting are two primary localization strategies applied by the researchers. While fingerprinting uses a pre-recorded database of RSSI signatures at recognized places to more reliably identify the user's position. Trilateration determines the user's position based on distance measurements from numerous beacons. By applying Kalman and Particle filters, challenges such as signal fluctuation caused by obstacles and interference can be addressed by smoothing the raw RSSI data. The experimental evaluations in different home environments show that the system can track user's position and movement accurately. This leading to the monitoring of a person's living pattern can be more effective. Note, they discovered that there is not significantly affected by the position of the BLE beacon on the user's body or the version model of the BLE beacons. The system is then more flexible and practical in real-world applications.

Figure 2.1 BLE Beacons

Several studies have studied the usage of microcontrollers, IoT and indoor positioning as a solution to real world navigation and automation issues. Arduino based systems, such as, have been well known in the flexibility and affordability of prototyping sensor-driven applications. [26] highlighted the fact that the Arduino boards are cheap, simple to program and have a vast open-source community, which makes them applicable in quick prototyping in research and retail related IoT solutions. These boards with wireless modules like Bluetooth or Wi-Fi have been implemented in smart applications in automation devices as well as interactive devices.

According to [27], Shortest path algorithms are essential for effective navigation in confined areas like supermarkets. Dijkstra's and A* algorithms are two well-known pathfinding methods. Dijkstra's algorithm is a classical algorithm for shortest path finding, optimal for graphs where edge weights are non-negative. It is efficient for small-scale environments, but it becomes computationally costly. Moreover, by adding heuristic to prioritize paths toward the target, typically using Manhattan or Euclidean distances, an Algorithm* improves on Dijkstra. Due to its directional search capacity, A* consistently needed less iteration and less computing time than Dijkstra's in large graphs.

Indoor positioning has gained major concern in the last decade because conventional GPS is not able to offer high accuracy of positioning in closed systems because of multipath effects and attenuation of signals. To deal with this, researchers have ventured into a host of options such as Wi-Fi, Bluetooth, RFID and sensor-based solutions. Radio based methods of the early days had some potential but were subject to electromagnetic interference, expensive infrastructure and delivered unreliable results in more complex indoor settings. There is therefore a shift towards exploiting the sensors which are already installed on smartphones to create solutions which are cost effective and scalable. Among the most noticeable approaches in this [28] is PDR which employs IMU sensors including accelerometers, gyroscopes and magnetometers. PDR has the benefit of operating without extra infrastructure and can give real-time positioning constantly. VLP is another potential approach and enjoys the benefits of using a pre-existing LED infrastructure in the indoor environment. VLP systems can be designed to obtain high positioning accuracy using smartphone cameras or light sensors, depending on the geometry of the light signals. Past research had indicated that VLP may possess sub-meter accuracy in the perfect conditions.

2.2 Limitations

According to [22], even though research on IPS has advanced significantly, existing studies reveal several noteworthy limitations that affect the accuracy, reliability, cost-efficiency, and scalability of indoor navigation systems. The complex structure of interior surroundings, which create widespread obstacles such as multipath propagation, signal fading, reflection, deep shadowing, and delay worsening, is one of the primary obstacles in IPS. These phenomena significantly degrade the quality of positioning signals, especially in radio frequency-based systems like BLE and Wi-Fi. High-density, substantial objects, like store shelves and merchandise attenuate signals and cause interference in RSSI values, which are commonly used to distance estimation. Positioning results are inconsistent and unreliable due to environmental interference. RSSI-based positioning, while being widely utilized because it is easy to use and compatible with current infrastructure. RSSI-based positioning has inherent inaccuracies caused by signal variability. The previous studies emphasized that considerable beacon calibration is required to improve accuracy, especially fine-tuning RSSI values at reference distances. However, the scalability and practicality of these systems are limited by the labour-intensive process of this calibration, which needs to be repeated to adapt to environmental changes. Moreover, inertial sensor-based systems, which rely on accelerometers and gyroscopes, face the problems of sensor drift. Small measurement mistakes build up over time, leading to position and orientation flaws that degrade tracking performance unless they are fixed by external references. Similarly, magnetic-based positioning systems, while theoretically stable, are affected by user variability and device diversity. Variations in step length and other user actions cause inconsistencies in magnetic fingerprinting, and different smartphones have different magnetic sensor sensitivity. High-accuracy IPS implementations often require dense deployment of anchor nodes or beacons. For instance, to cover the area adequately, BLE-based systems require a huge number of strategically placed beacon units. This requirement increases installation complexity and maintenance costs, including the need for battery replacement and system calibration. Sub-meter accuracy is frequently achieved at the expense of higher infrastructure spending, which can be unaffordable for large-scale commercial applications like supermarkets.

Despite promising developments, according to [23], BLE-based indoor positioning systems still face significant limitations. Due to environmental changes, inconsistent technology, and the influence of physical obstacles, RSSI-based distance calculation has low accuracy. BLE signal interference increases calibration requirements and compromise's real-time positional reliability. Accuracy tends to degrade in dynamic or clutter environments, even with signal filtering. Furthermore, as the offline information gathering and training are needed, fingerprinting methods are time-consuming even though they are accurate. GPR was applied to solve this problem, which decreased training time and increased accuracy. Nevertheless, these techniques remain mostly dependent on initial calibration and are difficult to scale across different environments. Although BLE-based indoor positioning has advanced, there are still several limitations from previous studies. According to [25], many existing BLE systems focus on localizing the user's smartphone rather than the user directly, which limits their application in situations where users might not always carry their phones, including senior citizens in healthcare monitoring settings. Furthermore, [26] even with this forward step, various constraints remain in the existing indoor navigation systems. First, there is still an issue with accuracy. BLE-based IPS is generally accurate within a range of 2-5 meters, which is not necessarily accurate enough to fine-grained shelf-level navigation unless either filtered heavily or beacons are placed densely. Unknown positioning is complicated further by multipath interference, variance in signal because of crowd density and the requirement of calibration. Second, it is evident that a lot of current systems are technically feasible rather than usable. They tend to research algorithms and infrastructure without incorporating interfaces that are user friendly and responsive to common users. This creates technically sound systems that are not readily practicable in actual retail settings. Third, it may have cost and scalability as a hindrance. Although BLE is cheaper compared to such solutions as UWB, beacon installation and maintenance may remain at a high expense in large supermarkets, especially in the case of small and medium-size retailers. Furthermore, other systems may have special hardware or proprietary software, which makes them less accessible.

According to [28], It has a significant disadvantage, however, cumulative error drift. Small sensor error leads to great deviations in the path that has been estimated as users walk further. Also, the accuracy of PDR to walking styles, speed variations, and environmental perturbation is very low, reducing its reliability in the long term. Nonetheless, the method has several obstacles. The density and location of the LEDs, the shutter speed of the camera, and the location of the LEDs in the field of view of the smartphone are all critical to its performance.

These restrictions lower continuity in positioning particularly in expansive areas with sparse installations of LEDs.

2.3 Proposed Solutions

In response to the challenges and limitations identified in previous IPS studies, SmartShop Navigators proposed several strategies and technological advancements to improve accuracy, reliability, and cost-effectiveness in indoor environments such as supermarkets. The hybridization of different positioning technologies is one of the most effective solutions. The advantages of LPS like Wi-Fi, BLE, and RFID for indoor environments and GPS for outdoor environments are combined in HPS. HPS can improve accuracy and availability in a variety of areas by combining multiple sources of positioning data, ensuring smooth transitions between indoor and outdoor tracking. SmartShop Navigator aims for the usage of advanced algorithms and filtering techniques to resolve the inherent inaccuracies and signal fluctuations in radio frequency-based systems. For instance, it has been demonstrated that positional accuracy can be improved by combining RSSI algorithm with trilateration or triangulation. The reliability of distance predictions is improved, and noise is reduced when the Kalman filter is applied to RSSI measurements. Studies have demonstrated that this methos can achieve average accuracies of 0.2-0.4 meters [22], [23].

Furthermore, Hybrid solutions have been explored to overcome the limitations of traditional IPS models. A single-anchor BLE system enhanced with PDR, increased accuracy and decreased the deployment complexity by combining inertial sensor information with BLE-based distance estimation. Additionally, this method minimized positional errors by efficiently integrating sensor readings using a Kalman filter. Furthermore, a simplified IPS model using ESP32 development boards and iTag BLE devices. The implementation of trilateration with a beacon arrangement in a predefined pattern can reduce calibration efforts. Additionally, hardware variability is addressed by using periodic auto-calibration and a modal-based RSSI selection method. Due to RSSI instability, distance estimate accuracy is still a limitation even if the methos reduced costs while making deployment easier [24].

Building upon [25] findings, the SmartShop Navigator project aims to implement a BLE-based IPS optimized for supermarket environments. This involves the use of advanced filtering algorithms designed for retail environment, the implementation of user-friendly mobile applications to visually validate product positions on the device, and the strategic placement of beacons to address interference and layout complexity. Moreover, mobile applications will improve accessibility for all customers, including disabilities, by turning positioning information into clear navigation instructions. To overcome these issues of [26], the recent study proposes employing affordable IoT solutions with better algorithmic methods and design oriented at the user. Arduino and Raspberry Pi microcontrollers provide a low-cost and flexible base on which to implement BLE beacon infrastructures, and which can be tested on a large scale with simulated or real supermarket conditions. The sensor fusion (accelerometers, gyroscopes) that can be used to supplement BLE signals are also supported by these microcontrollers to minimize positioning errors. According to [27], Hueristic-Based Pathfinding with A* will be implemented in SmartShop Navigator. By using heuristics, A* lowers computing overhead in comparison to Dijkstra's, which makes it perfect for navigation dynamic, graph-based layouts such as store aisles.

To overcome these constraints, [28] proposed hybrid solution of PDR and VLP using adaptive multi-frame fusion based on smartphone. The VLP in this method gives global calibration points which correct the cumulative drift of PDR, whereas PDR gives continuity in situations where the LED signals are not available. Adaptive multi-frame fusion was proposed to ensure that it is robust to shutter speed and LED placement variation, with a maximum decoding accuracy of 98.5 percent with 2.1 meters. It is also equipped with a dynamic step length calibration mechanism which corrects varying user speeds and further minimizes drift. In the experimental findings, it was found out that this integrated system had an average positioning accuracy of less than 0.5 meters in a 108-meter walk with just the use of regular smartphones and available lighting infrastructure. This combination of VLP and PDR emphasizes a trend in the current research in indoor navigation: to overcome the weaknesses of the individual systems by using fusion-based solutions. Although standalone approaches have fundamental drawbacks (drift in PDR or line-of-sight in VLP), data fusion approaches contain a way to scalable, accurate, and cost-effective indoor navigation. These methods make indoor positioning more practical towards a real-world application such as smart shopping navigation.

2.4 Comparison of different techniques

Technology	**Accuracy**	**Cost**	**Power Consumption**	**Scalability**	**Infrastructure Requirements**	**Deployment Complexity**	**Real-time Performance**	**Environmental Sensitivity**
BLE	0.2-0.4m (controlled), 2-5m (typical)	Low	Very Low	High	Moderate (beacon deployment)	Moderate	Excellent	High (obstacles, interference)
Wi-Fi	2-5m	Low	Moderate	Very High	Low (existing infrastructure)	Low	Good	Moderate
RFID	1-3m	Moderate	Very Low	Moderate	High (readers required)	High	Good	Low
UWB	Sub-meter	High	Moderate	Low	Very High	Very High	Excellent	Very Low
Magnetic	1-3m	Very Low	Very Low	High	Very Low	Low	Fair	Moderate (metal objects)
Inertial (PDR)	Degrades over time	Low	Low	High	Very Low (Smartphone only)	Very Low (algorithm setup)	Good	Low
GPS	N/A indoors	Low	Moderate	Very High	Very Low	Very Low	N/A	Very High (building blockage)
VLP	Sub-meter (optimal conditions)	Moderate	Low	Moderate	Moderate	Moderate	Fair	High (lighting, obstacles)

Table 2.4 Comparison of different techniques

Chapter 3

System Methodology/Approach OR System Model

This chapter outlines the system design of this project. It covers the overall system design approach used to develop the SmartShop Navigator. It defines the mathematical formulas used in the indoor positioning system, shows the overall system architecture and defines the interaction between the users and the system in terms of use case and activity diagrams.

3.1 System Methodology

The SmartShop Navigator will be developed using an agile methodology, with the iterative development process being managed by Scrum frameworks [29]. Sprint cycles of two weeks will be implemented, with each sprint focusing on developing specific functional components that are testable and assessable. This method allows for continuous refinement based on feedback and facilitates adapting to emerging challenges or requirements.

To guarantee comprehensive verification of the system's performance, usability, and functionality, the testing approach consists of several layers. Unit testing will verify that each component is operating effectively, paying social attention to the navigation logic and positioning algorithms. The successful operation of different system components, such as the communication between the beacon network and the mobile application will be verified via integration testing [30].

System testing will assess the application's overall performance under different conditions, including different device types, user densities, and environmental conditions. Key performance indicators such as power consumption rates, search query response time, and positioning accuracy will be measured using performance benchmarks [31]. Furthermore, usability testing with representative users will show how intuitive the interface is and identify potential improvements to user experience.

The testing approach will be implemented in stages, starting with controlled lab environments and moving on to a simulated supermarket setup. A restricted deployment with chosen users in an actual supermarket environment will be part of the last testing phase, ensuring assessment in real-time environment. The user interface design and technological execution will be improved based on feedback from each stage.

The SmartShop Navigator project aims to provide a reliable and user-friendly indoor navigation system that significantly enhances the supermarket shopping experience by utilizing various techniques and technologies. Accurate placement, effective pathfinding, and user-friendly interaction provide users with a useful tool for finding products quickly and safely navigating their first store surroundings.

3.2 Equation

1. Weighted Centroid Localization Algorithm

   Conventional presence-based indoor navigation is unreliable due to radio waves reflecting off solid objects and resulting in variable signal strengths. To calculate the preliminary (X, Y) user coordinates, the PHP server uses the RSSI values of three nearby BLE beacons.

   The negative RSSI is transformed into a positive weight, which means the stronger the signal, the heavier the weight. To prevent the "rubber band effect" - where the user's coordinate is calculated from distant beacons and pulled away from their real location - a Threshold-Based Exponential Weighting multiplier is used if the RSSI signal is stronger than the -55dBm. The final physical position is calculated using the following centroid equations [32]:

$$X = \frac{\sum_{i=1}^{n} (W_i \cdot x_i)}{\sum_{i=1}^{n} W_i}$$

$$Y = \frac{\sum_{i=1}^{n} (W_i \cdot y_i)}{\sum_{i=1}^{n} W_i}$$

where

W_i is calculated weight of the i beacon,

(x_i, y_i) are the fixed physical coordinates of that beacon.

2. Grid-Based Manhattan Distance Routing

   Pathfinding algorithms such as A* are widely used, but they are too complex and inefficient for the grid-based retail environment. Supermarkets are grid-based, with horizontal and vertical rows of aisles that are separated by fixtures.

   Once the raw destination is set, the system then creates paths constrained to the 26 x 18 grid. The path from user's initial position (X_1, Y_1) to the product (X_2, Y_2) cannot exceed orthogonal grid movements to establish safe routes [33]:

$$D = |X_1 - X_2| + |Y_1 - Y_2|$$

where

(X_1, Y_1) is initial position,

(X_2, Y_2) is product position.

3. Vector Projection Map Matching (Path Snapping)

BLE radio waves are highly susceptible to environmental noise and human body interference, which results in the digital cart on the user interface crashing into digital walls. By using a map-matching algorithm on the Flutter client to "magnetically" snap the user's noisy raw location to the closest line segment of the generated Manhattan route.

The nearest mathematical point on a line segment to a user's position is found using vector projection [34]:

$$t = \frac{(X_{user} - X_{start}) \cdot (X_{end} - X_{start}) + (Y_{user} - Y_{start}) \cdot (Y_{end} - Y_{start})}{(X_{end} - X_{start})^2 + (Y_{end} - Y_{start})^2}$$

The scalar value t is clamped between 0.0 and 1.0 to ensure the snapped coordinate does not exceed the mathematical boundaries of the physical aisle segment.

3.2.1 System Architecture Diagram

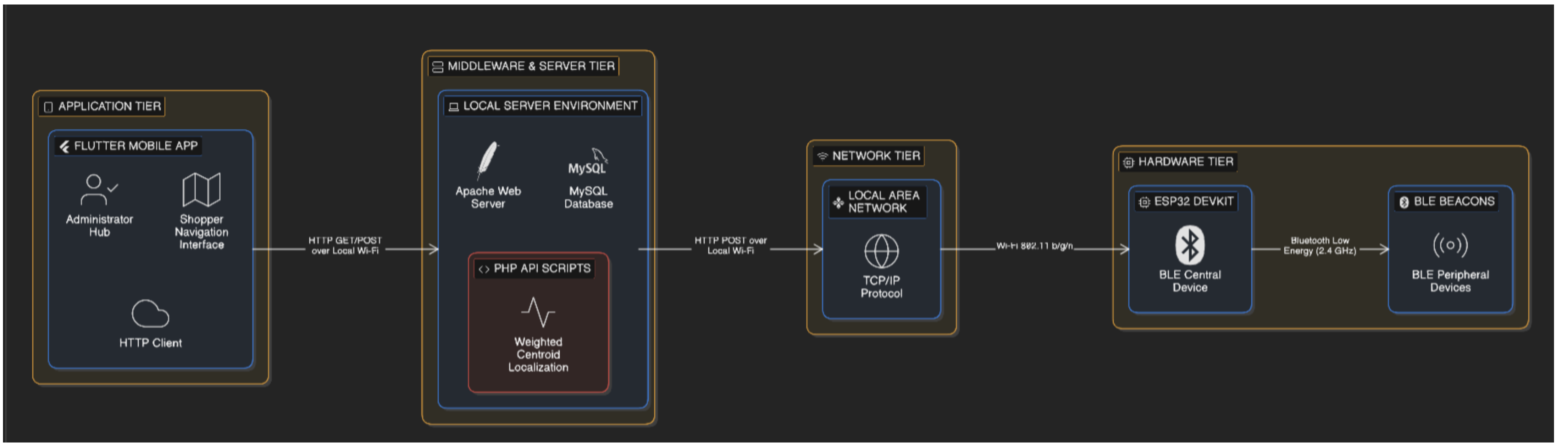

Figure 3.2.1 System Architecture Diagram

SmartShop Navigator adopts a typical multi-layered IoT paradigm [35]. This architecture topology strictly divides the hardware data acquisition, network connectivity, data processing and user interface into several physical layers. This allows the mobile client to be efficient and lightweight with the demanding mathematical calculations being computed in the server.

- Hardware Tier: At the bottom layer, the physical space is digitized by several nRF51822 BLE beacons placed at pre-defined reference points. These beacons act as peripheral devices, continually advertising data packets at 2.4 GHz. The data collector is a microcontroller ESP32 mounted on the shopping cart. The ESP32, functioning as a device that scans for beacons, receives the RSSI value from the beacons [36].

- Network Tier: The network tier provides the link between the hardware and the server. The ESP32 has minimal processing power and hence uses an existing IEEE 802.11 (Wi-Fi) network to transfer the raw RSSI data packets to the server using normal TCP/IP communication protocol [37]. This enables real-time, low-latency data transfer of the shopper's movements.

- Middleware and Server Tier: The server tier, which is hosted on an Apache/XAMPP environment, is the system's brain. The middleware is a custom-developed set of PHP API scripts. Rather than burdening the ESP32 or the smartphone with complex indoor positioning calculations, the PHP scripts receive the raw RSSI data from the ESP32 and run the Weighted Centroid Localization algorithm [38]. The resulting X and Y coordinates, fixed product coordinates and active beacon MAC ids are stored in relational MySQL database.

- Application Tier: The flutter-based mobile application is the topmost tier which communicates with the server tier via RESTful HTTP (GET/PUT) requests. These APIs are used by mobile applications to obtain pre-calculated cart coordinates and product targets. The smartphone's embedded processor is then devoted to executing the Grid-Based Manhattan routing and vector projection map-matching algorithms, which draw a visually pleasing and obstacle-free path on the end-user's screen.

3.2.2 Use Case Diagram and Description

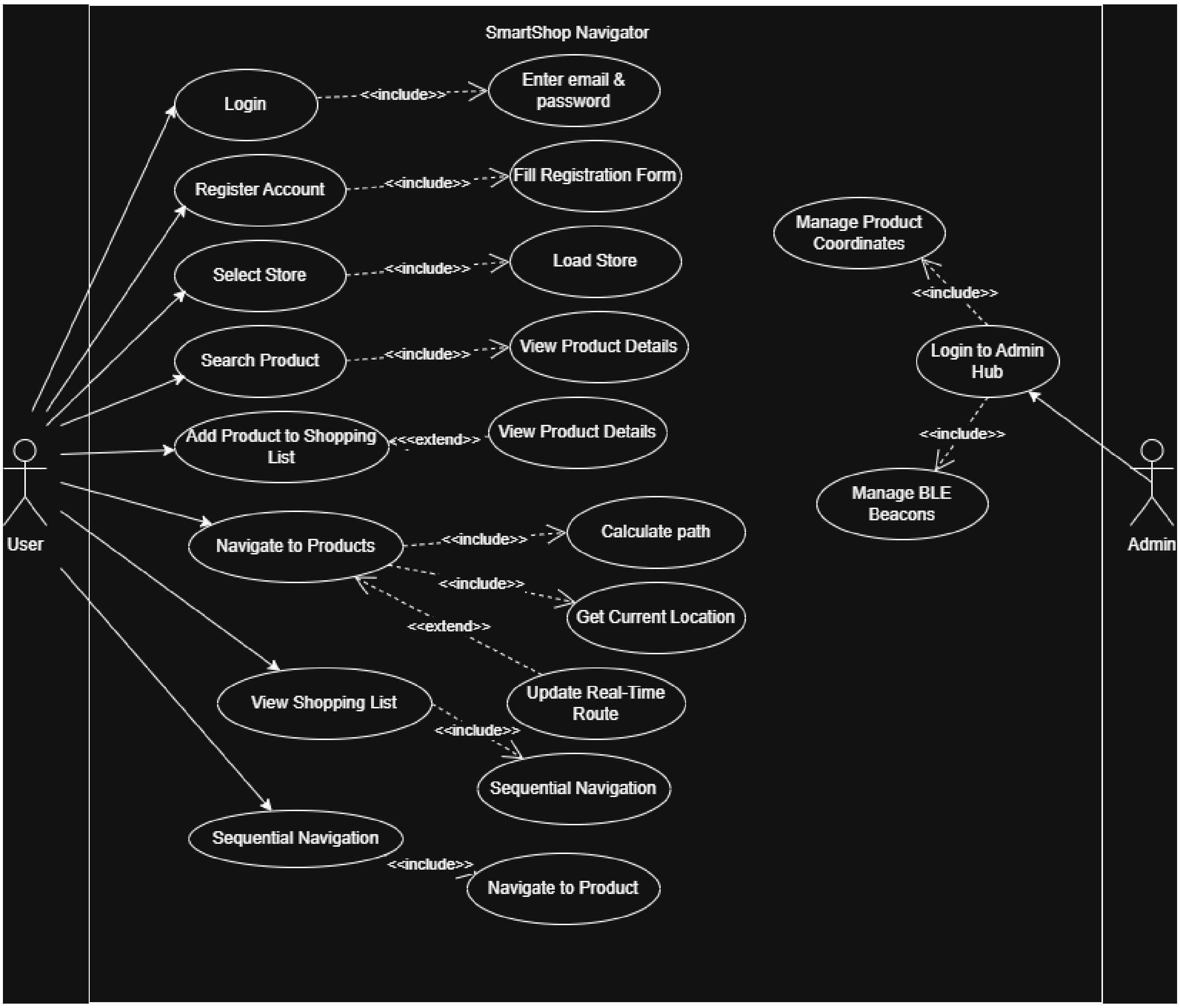

Figure 3.2.2 Use Case Diagram

- User use cases

  The User actor interacts with the system primarily to locate and navigate specific products within a designated store environment. Their interactions are categorized into three main phases:

  - **Account Management:** The user can Register Account (which includes filling out a registration form) and Login (which includes entering email and password credentials) to access their personalized shopping data.
  - **Pre-Navigation (Preparation):** The user begins by initiating Select Store to load the specific floor plan. They can then Search Product to view specific product details. From the product detail's view, the system provides an extended option to Add Product to Shopping List.

- **Active Navigation:** The core functionality revolves around Navigate to Products. To execute this, the system inherently includes background processes to Calculate path (using Grid-Based Manhattan routing) and Get Current Location (using the ESP32 Weighted Centroid data). As the user moves, the system extends this behavior with Update Real-Time Route to snap the cart to the path. Furthermore, the user can View Shopping List, which includes the Sequential Navigation loop, allowing them to continuously Navigate to Product until all items are found.

- Administrators use case

  The admin actor is responsible for maintaining the accuracy of the indoor positioning environment via a secure backend interface.

  - **Database Management:** The admin can Manage Product Coordinates to update the X and Y grid locations of physical items and Manage BLE Beacons to update the MAC addresses of the hardware deployed in the store.
  - **Security:** Both management functions strictly include a mandatory Login to Admin Hub process to ensure unauthorized users cannot alter the store's navigational mapping.

3.2.3 Activity Diagram

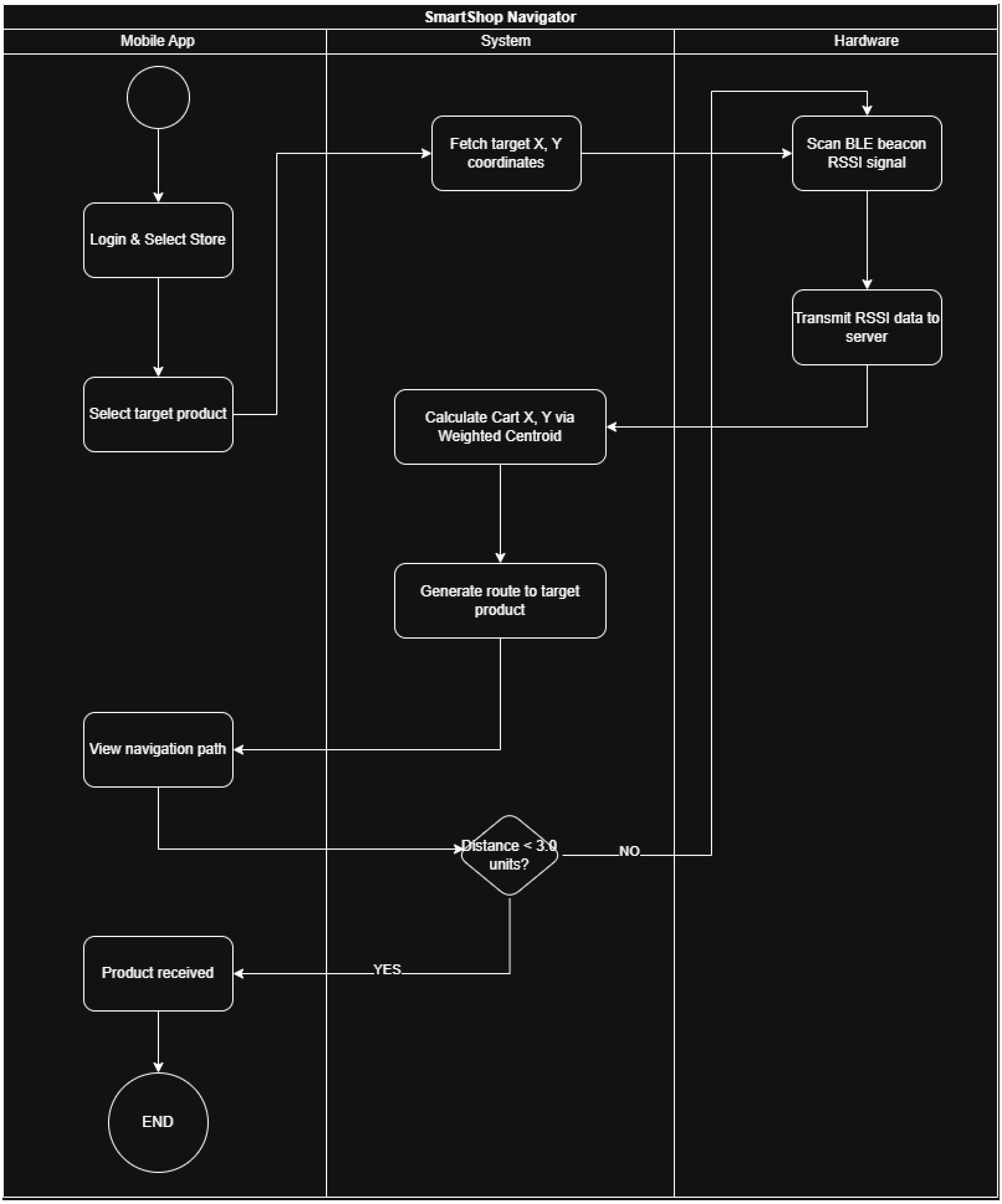

Figure 3.2.3 Activity Diagram

The activity diagram models the sequence of activities and data flows during a navigation session. The diagram is divided into three swimlanes to clearly separate the functionalities of the system's components: the Mobile App (front-end), the System (back-end and database) and the Hardware (ESP32 microcontroller and BLE beacons).

The navigation flow is processed in the following stages:

- Initialization phase: The sequence starts in the mobile app lane where the user starts the app with Login & Select Store. After logging in, the user selects Select target product from the database, setting the goal.

- Fetching Data and Hardware Lane: The query is passed on to the system lane to Query to get the target X, Y coordinates. This is the signal for the hardware lane. The ESP32 device starts to Scan BLE beacon RSSI signal to get the environmental radio frequencies. The device then Transmits RSSI data to server.

- Processing and Route Generation: Once the hardware data is received, the system lane uses the Calculate Cart X, Y via Weighted Centroid algorithm to calculate the users' real-world location. This real-time location is used to Generate route to target product, thus routing a safe path.

- The Continuous Navigation Loop: The path is then sent back to the Mobile App, enabling the user to View navigation path on the mobile device. The process now comes to a crucial decision point in the system lane, continually calculating the Euclidean distance between the cart and the destination: Distance < 3.0 units?
    - The "NO" Condition (User is walking): If the distance is 3.0 units or more, the flow returns to the Hardware Lane to read new BLE signals. This forms a closed-loop feedback system, which dynamically updates the cart location and route on the display screen as the user navigates the shopping cart around the grocery store.
    - The "YES" Condition (User has arrived): When the cart moves beyond the 3.0-unit radius, the loop is broken. The system triggers the Mobile App to show the Product received message, thus ending the navigation flow for the current product.

Chapter 4

System Design

4.1 System Block Diagram

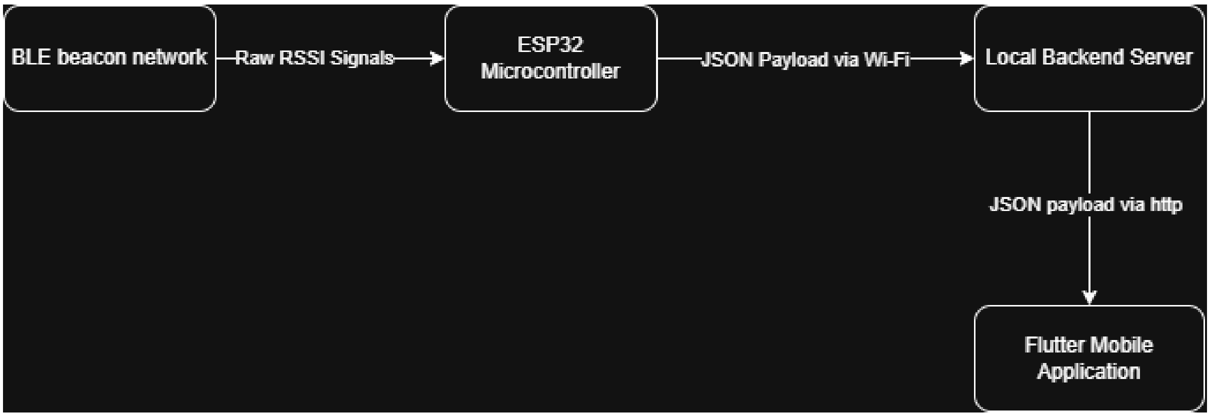

Figure 4.1 System Block Diagram

The system block diagram below describes the flow of data and the functional blocks that make up the SmartShop Navigator. The system is broken down into four main blocks, which correspond to different stages of the indoor positioning process:

- BLE Beacon Network Block: This is the input block and is the physical nRF51822 hardware in the environment. The function of this block is to transmit Bluetooth Low Energy Advertisements (BLE Adv). These packets are used to broadcast the unique MAC address and the raw RSSI values required for tracking.

- ESP32 Microcontroller Block: The data acquisition hardware interface, this block contains two modules. The BLE Receiver Module scans and receives the raw RSSI signals from the beacons. The Wi-Fi Transmitter Module takes the raw signals, packages it into a JSON message and transmits it to the mobile cart's local server (the cart is not connected physically to the server).

- Local Backend Server Block: This is the central data processing and storage unit. The Localization Core applies the Weighted Centroid algorithms to the ESP32 messages received through the PHP Receiver API to remove noise. The estimated coordinates are stored in the MySQL Database. The second PHP Provider API is the output block that organizes the data for the client application.

- Flutter Mobile Application Block: The output block renders and displays dynamic data. The HTTP Client Module retrieves real-time spatial data. It is then fed to the internal Pathfinding Engine (which runs the Grid-Based Manhattan algorithm) and the Map-Matching Module (which performs vector projection to provide a smooth path). Finally, the graphics rendered are fed to the User Interface, which can be either User Navigation View or Admin Hub dashboard.

4.1.1 System flow chart

User site:

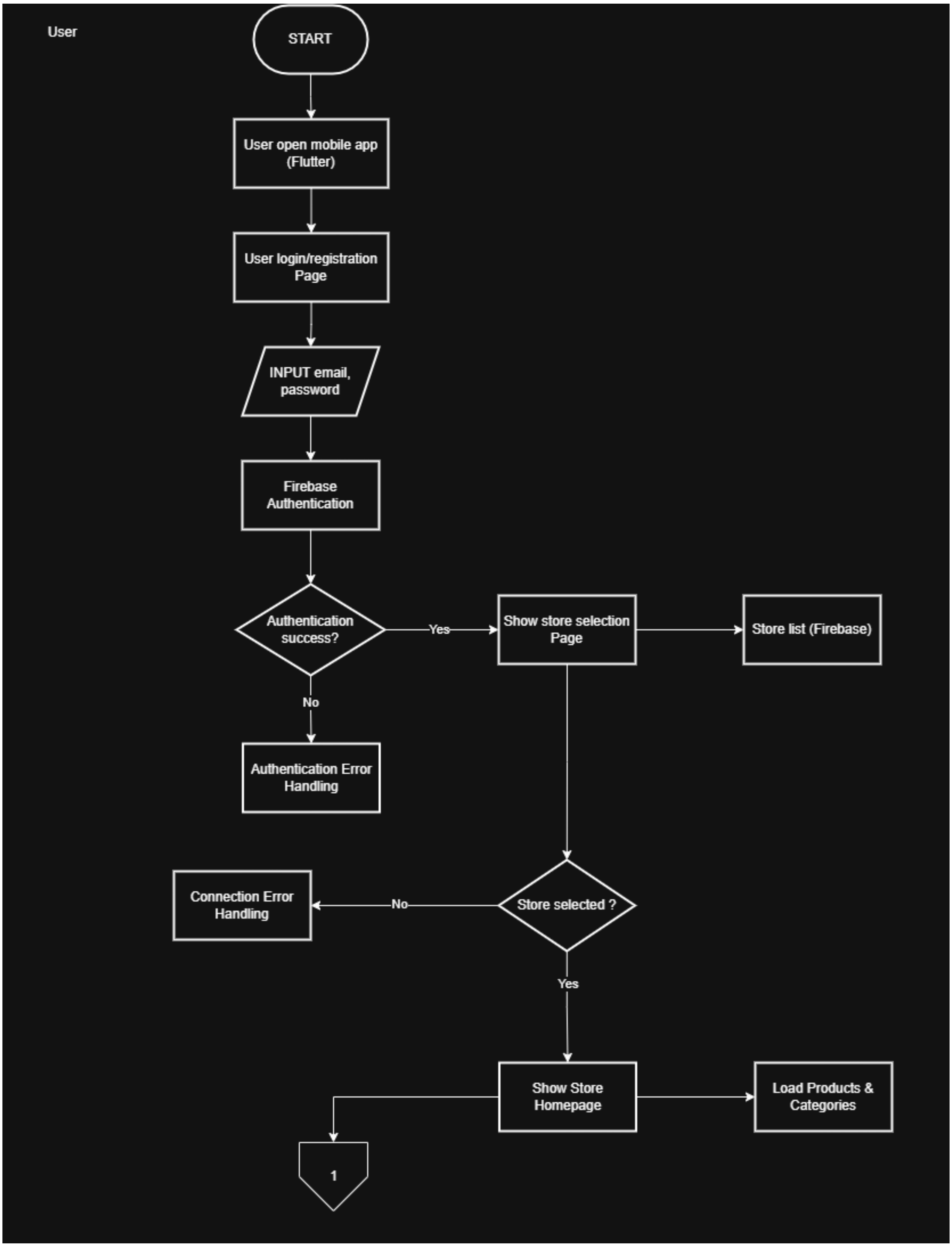

Figure 4.1.1.1 Flow Chart 1

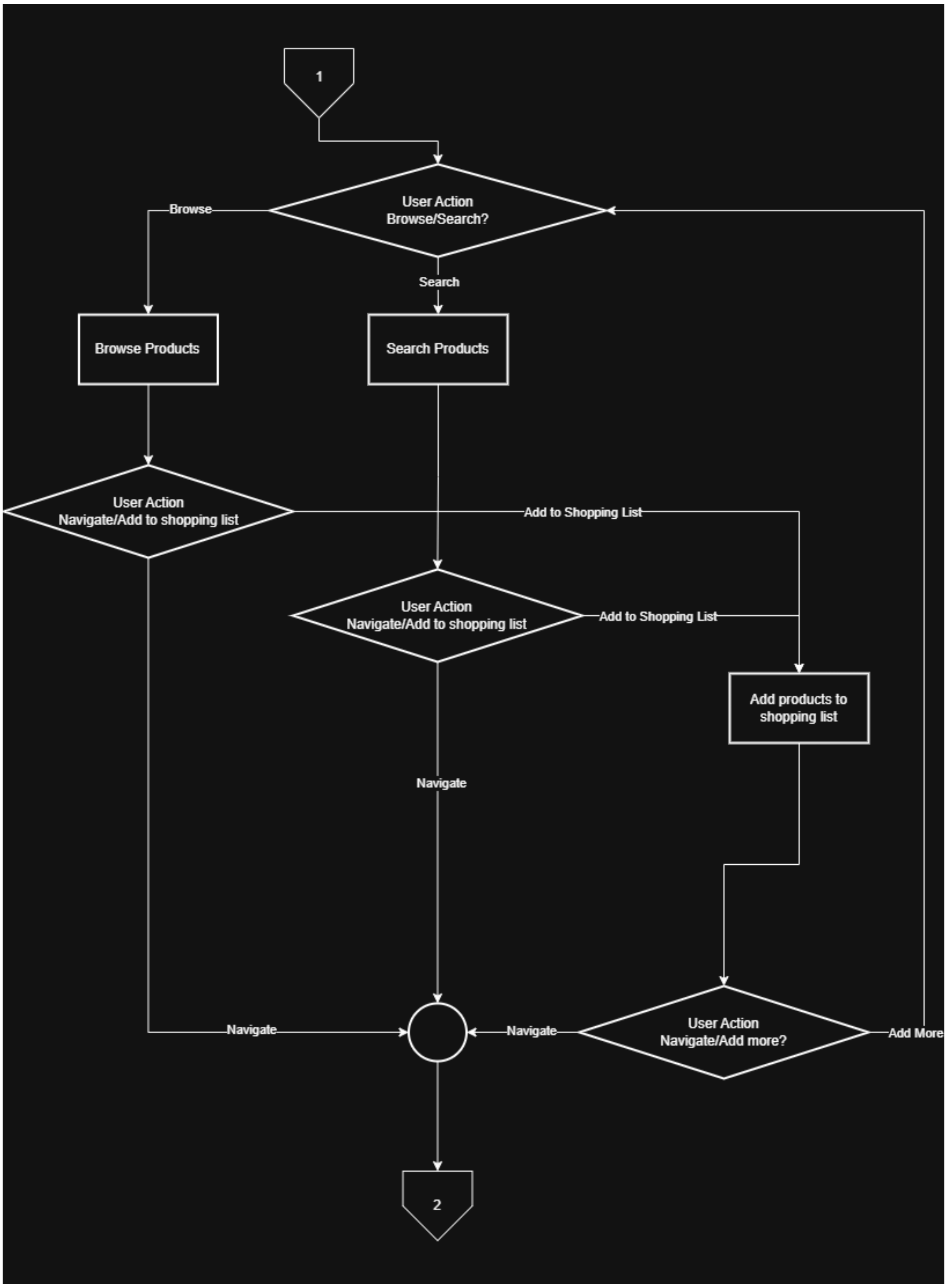

Figure 4.1.1.2 Flow Chart 2

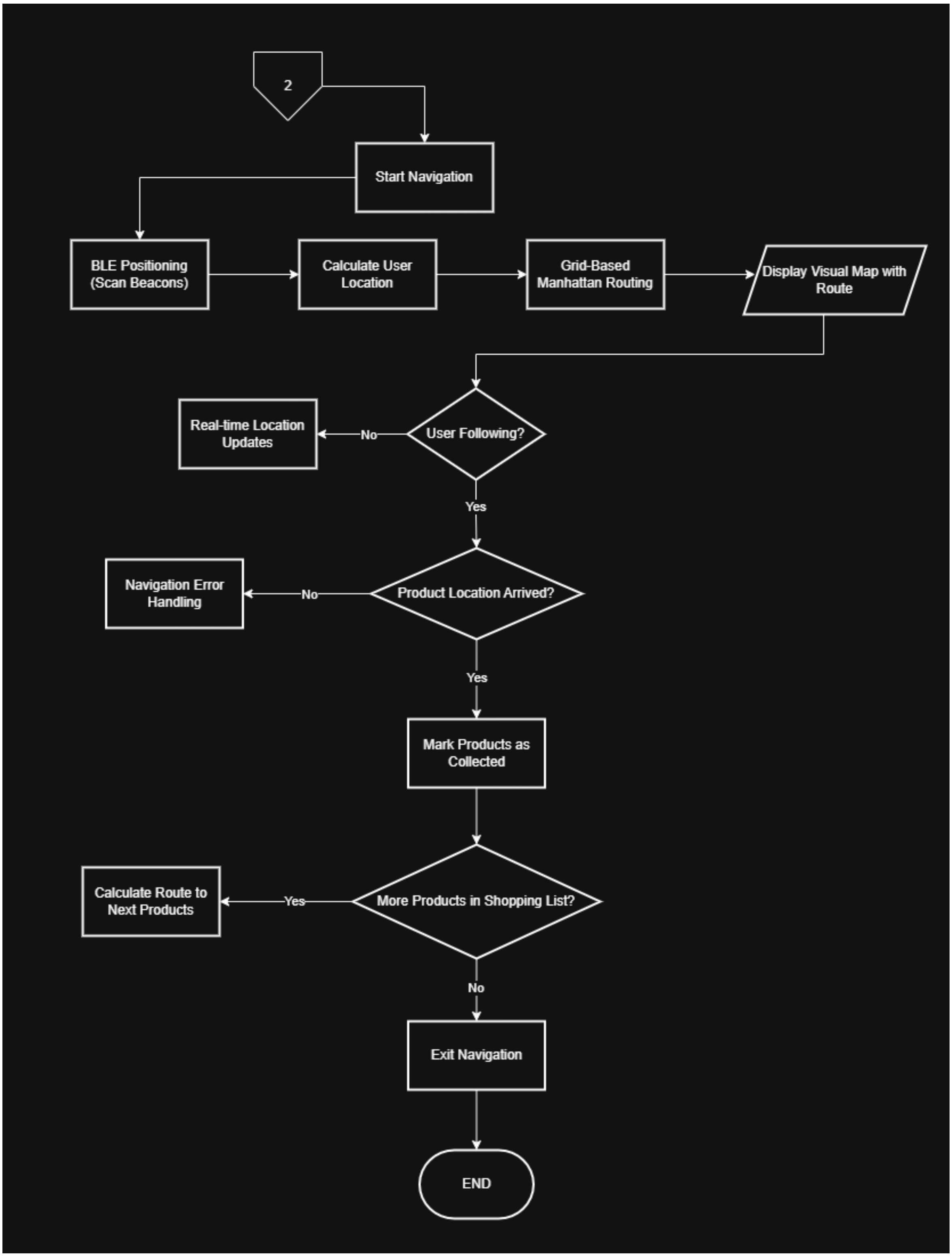

Figure 4.1.1.3 Flow Chart 3

Admin site:

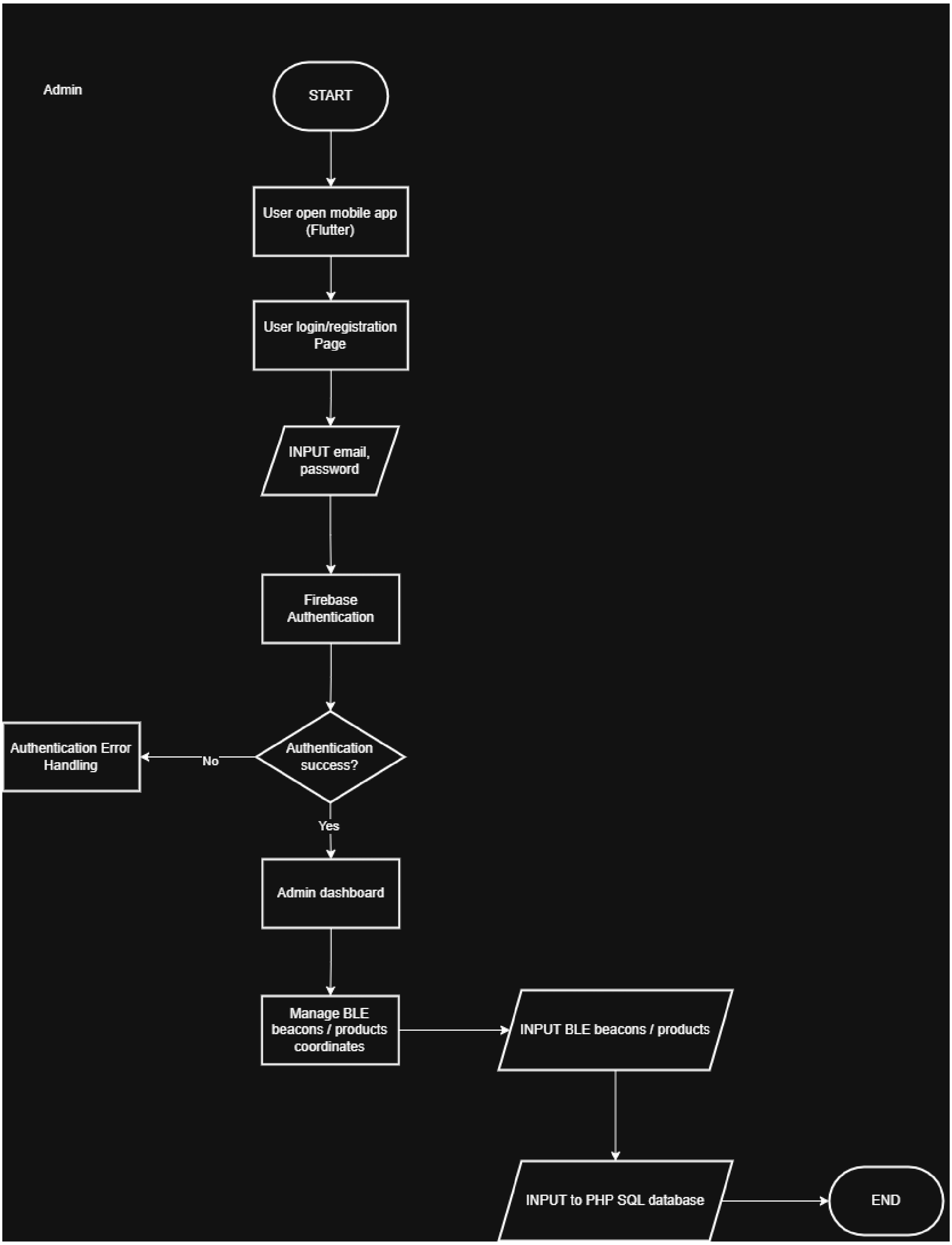

Figure 4.1.1.4 Flow Chart 4

4.1.2 System Flow Chart Overview

User site:

Section 1: Authentication and Setup Phase

1. App Initialization
    - START: The process begins when the user opens the mobile app
    - User Open Mobile App (Flutter): Indicates the app is built using Flutter framework

2. Authentication Flow
    - User Login/Registration Page: First screen prompts for user credentials
    - INPUT email, password: User enters authentication details
    - Firebase Authentication: Backend authentication service processes the credentials
    - Authentication Success?: Decision point that branches the flow
        - YES: Proceeds to store selection
        - NO: Triggers Authentication Error Handling and loops back to login

3. Store Selection Process
    - Show Store Selection Page: Displays available stores to the user
    - Store List (Firebase): Backend retrieves store data from Firebase database
    - Store Selected?: Decision point for store selection
        - YES: Proceeds to homepage
        - NO: Triggers Connection Error Handling and returns to selection

4. Homepage and Product Management
    - Show Store Homepage: Main interface displaying store layout
    - Load Products & Categories: System loads product database and categorization
    - Connector 1: Links to the shopping list management section

Section 2: Shopping and Navigation Phase

5. User Action Decision Points

The flowchart shows multiple decision diamonds for user actions:

First Decision: Browse/Search?

- Browse: User can browse products by category
- Search: User can search for specific products
- Both paths lead to Add to Shopping List

Second Decision: Navigate/Add to Shopping List?

- Navigate: User chooses to start navigation
- Add to Shopping List: User adds more items

Third Decision: Navigate/Add More?

- Navigate: Proceeds to navigation mode
- Add More: Returns to product selection
- Add Products to Shopping List: Continuous list building capability

6. Navigation Engine Activation

- Start Navigation: Initiates the indoor positioning system
- BLE Positioning (Scan Beacons): Bluetooth Low Energy beacons detect user location
- Calculate User Location: Positioning algorithms determines precise coordinates
- Grid-Based Manhattan Routing: Pathfinding algorithm computes optimal route
- Display Visual Map with Route: Shows navigation interface with highlighted path

7. Active Navigation Loop

Real-time Tracking:

- User Following?: Monitors if user is following the suggested route
    - YES: Continues navigation
    - NO: Provides Real-time Location Updates and Navigation Error Handling

Product Collection:

- Product Location Arrived?: Checks if user reached the target product
    - YES: Mark Products as Collected
    - NO: Continues navigation guidance

Route Continuation:

- More Products in Shopping List?: Determines next action
    - YES: Calculate Route to Next Products (creates a loop back to navigation)
    - NO: Exit Navigation and proceed to END

Admin site:

1. App Initialization
   - START: The process begins when the user opens the mobile app
   - User Open Mobile App (Flutter): Indicates the app is built using Flutter framework

2. Authentication Flow
   - User Login/Registration Page: First screen prompts for user credentials
   - INPUT email, password: User enters authentication details
   - Firebase Authentication: Backend authentication service processes the credentials
   - Authentication Success?: Decision point that branches the flow
     - YES: Proceeds to admin dashboard
     - NO: Triggers Authentication Error Handling and loops back to login

3. Admin action

- Manage BLE beacons / products coordinates: Admin can take action to manage BLE beacons or products coordinates.
- INPUT BLE beacons / products: BLE beacons' MAC addresses and coordinates or products' details and coordinates.
- INPUT to PHP SQL database: data will be input to PHP SQL database.
- END.

4.2 System Component Specification

To achieve high availability and low latency in the SmartShop Navigator system, certain hardware and software components were selected, based on their specifications and compatibility.

4.2.1 Hardware Specifications

The SmartShop Navigator system's hardware architecture is comprised from a variety of essential parts that cooperate to provide accurate indoor positioning and navigation. BLE beacons (nRF51822), uses the 2.4 GHz ISM band. For the current system, they are set to transmit at high rate (100ms) with a transmit power of +4 dBm to provide the ESP32 with a steady stream of RSSI values to reduce signal loss. It will be positioned thoughtfully throughout the supermarket, providing the foundation of this system [39]. By sending out signals at regular intervals, these beacons establish a network of reference points that the system may use to evaluate signal strengths and estimate the user's location via Weighted Centroid localization [40]. A localized test area will need to be covered by a core set of 3 BLE beacons for the prototype implementation, each of which will be positioned to provide the best possible signal coverage and minimize the risk of interference.

The data acquisition's core control unit will be an ESP32 microcontroller (ESP32 DevKit V1) mounted on the shopping cart. The ESP32 is an inexpensive, low-power system-on-a-chip (SoC) with dual-mode Bluetooth and integrated Wi-Fi, which will be responsible for scanning the beacons, collecting signal data, and connecting with the local database server via Wi-Fi [41]. For the power supply, ESP32 uses a standard 5V power supply, provided by a mobile 10,000mAh USB power bank which allows the shopping cart to be fully mobile and unplugged for up to 24 hours of operation.

For user interaction, the system requires smartphones with internet connectivity and capable processors to render the navigation interface [42]. Many modern smartphones come equipped with these capabilities, which will be leveraged to run the dynamic Grid-Based Manhattan routing and map-matching algorithms smoothly.

Table 4.2.1 Specifications of hardware

Description	Specifications
Test Device	Emulator
BLE Beacons	nRF51822
Microcontroller	ESP32 devkit v1

Figure 4.2.1.1 nRF51822

Figure 4.2.1.2 ESP32 Devkit v1

4.2.2 Software Specifications

The ESP32 data-acquisition node is programmed with C++ code using the Arduino IDE [43]. According to [44], the Arduino IDE offers a powerful and intuitive environment for translating source code into machine code for the microcontroller and includes a comprehensive board and library manager. For the implementation of the specific IoT functions of the SmartShop Navigator, several key open-source libraries are used. In particular, the BLEDevice.h and BLEScan.h libraries allow the ESP32 to operate as a BLE Central device, setting the hardware to listen for ambient nRF51822 advertising packets, capturing their respective RSSI values and bypassing the need for a formal Bluetooth pairing (which would drain the nRF51822's battery). To communicate with the cloud and server, the WiFi.h and HTTPClient.h libraries enable the ESP32 to connect to the local Wi-Fi router in Station Mode (STA) and open HTTP/TCP connections with the XAMPP server [45]. Finally, ArduinoJson.h is an efficient memory-management library to convert the array of BLE MAC addresses and RSSI integers into a compact, transmittable JSON string.

The microcontroller is complemented by powerful software architecture for data processing and user interfaces.The application will be developed using Flutter to support both Android and iOS platforms, ensuring compatibility with most devices used by potential customers [46]. Flutter's CustomPaint rendering engine is extensively used in the application to paint the dynamic 26x18 grid of the floor plan. On the back end, the application uses XAMPP, a free, open-source cross-platform web server solution stack package which includes the Apache HTTP Server and the MySQL database engine. For communication between the database and the frontend, the backend scripting language is PHP 8.x, which is used to develop RESTful APIs such as update_cart.php and admin_api.php [47]. PHP was chosen for its compatibility with MySQL and ability to perform complex mathematical calculations, such as the Weighted Centroid Localization algorithm, quickly on the server.

4.3 Circuits and Components Design

This section details the physical and logical integration of the system components, providing the blueprint necessary for system recreation.

4.3.1 Hardware Integration and Placement

The hardware placement design avoids complicated breadboard connections and instead takes a modular IoT approach. The ESP32 is housed in a protective enclosure and placed at the front of the shopping cart. It is directly connected to the power bank using a USB cable. The nRF51822 beacons are installed at a predetermined distance in the test environment. To avoid absorption by the human body or metal shelves, they are at least 2 meters off the floor.

4.3.2 Database Schema Design

The circuit of the backend relies on a relational MySQL database. The database is normalized to ensure data integrity and consists of the following primary tables. There are 3 tables in this database:

- **products Table:** Stores the static locations of items.
  - id (INT, Primary Key, Auto-Increment)
  - name (VARCHAR)
  - x_coord (FLOAT)
  - y_coord (FLOAT)
- **cart_location Table:** A dynamic table continuously overwritten by the ESP32.
  - cart_id (VARCHAR, Primary Key)
  - current_x (FLOAT)
  - current_y (FLOAT)
  - last_updated (TIMESTAMP)
- **beacons Table:** Stores the physical coordinates of the hardware infrastructure for the PHP math engine.
  - mac_address (VARCHAR, Primary Key)
  - x_coord (FLOAT)
  - y_coord (FLOAT)

4.4 System Components Interaction Operations

Asynchronous communication between the hardware, server and application is the key to the SmartShop Navigator. This communication is accomplished through RESTful APIs with JSON data payloads.

4.4.1 ESP32 to PHP Server (Data Ingestion)

The data ingestion interaction starts at the hardware level with the ESP32 firmware's main loop. The firmware's design is strictly separated into two main phases of operation for stability. In the setup() initialization stage, upon 5V power up from the cart's power bank, the ESP32 sets up its serial monitor for debugging. Next, it sets up its built-in Wi-Fi capabilities in Station Mode and authenticates with the local access point. Lastly, it sets up the BLE scanning parameters by specifying the scan window durations and enabling active scanning mode. After the initialization period, the device enters the main runtime loop of the loop() phase, which activates the BLE scanner for a pre-defined time (e.g. 2 seconds). Here, the ESP32 scans all advertising packets and removes any noise by filtering out only the MAC addresses and RSSI of the desired nRF51822 store beacons.

After the signal information is extracted, the firmware then uses the ArduinoJson library to package the strongest signals into a JSON string. The HTTPClient library is then called to connect to the update_cart.php endpoint and configure the ESP32 to use an HTTP header of application/json and issue an HTTP POST request. This interaction pattern guarantees that the microcontroller remains simple and lightweight. By sending only raw data payloads and offloading the computational burden of the Weighted Centroid algorithm to the PHP server, the ESP32 can keep its processing resources free, avoid thermal shutdown and save power.

4.4.2 PHP Server Processing

The server-side PHP script begins the data parsing and location process when it receives the HTTP POST request from ESP32. It begins by using the json_decode() function to convert the received JSON data into an associative array. The script connects to the MySQL database using prepared statements to prevent SQL injection attacks. The code loops through the decoded array, performing a SELECT query on the beacons table to associate the MAC addresses of the hardware beacons with their X and Y coordinates.

After the physical coordinates and their respective signal strength values are obtained, the server runs the Weighted Centroid Localization algorithm. To exclude noisy environmental impulses, the algorithm assesses the signal of each beacon. Should the RSSI value be greater than -55. Lastly, the script runs an SQL UPDATE query, overwriting the cart_location table with the new coordinates.

Example Ingestion JSON payload:


```
{
   "cart_id": "cart_01",
   "beacons": [
   {"mac": "AA:BB:CC:DD:EE:11", "rssi": -55},
   {"mac": "AA:BB:CC:DD:EE:22", "rssi": -62},
   {"mac": "AA:BB:CC:DD:EE:33", "rssi": -70}
   ]
}
```

4.4.3 Flutter App to PHP Server (Data Consumption)

The Flutter app consumes the data in the central database to render the real-time navigation line. The app makes use of the Dart http package to implement a polling mechanism, whereby asynchronously the application makes repeated HTTP GET calls to admin_api.php every 1 second. This frequency was chosen to achieve a balance between delivering location information to the user in a timely fashion and not imposing a heavy burden on the local XAMPP server. Network requests are asynchronous by nature and can take an arbitrary amount of time to complete, so the application manages these requests using Dart's Future objects and async/await features. If the server successfully handles the GET request, the server will return a JSON serialized array that contains both the user's current cart position and the product's coordinates.

Example Response JSON payload:


```
{
   "status": "success",
   "cart": {"x": 12.5, "y": 8.0},
   "target": {"name": "Milk", "x": 5.0, "y": 14.0}
}
```

4.4.4 Application UI Rendering

When the Flutter application receives and confirms the response as valid JSON data, the user's interaction with data comes to an end. The application converts the JSON string into native Dart objects and directly feeds the numerical coordinates to the FloorPlanPainter class, which extends Flutter's native CustomPainter class for basic drawing operations. First, the UI runs the Grid-Based Manhattan algorithm. Rather than computing a direct diagonal path, which would visually cross physical tables in the supermarket, the algorithm computes a purely orthogonal (90-degree) path of waypoints from the cart's starting position (e.g., 12.5, 8.0) to the product (e.g., 5.0, 14.0). Finally, to hide the physical jitter of Bluetooth waves, the UI runs the Vector Projection Map-Matching algorithm. Prior to the cart icon being rendered on the screen, this mathematical formula calculates the closest perpendicular distance between the cart's actual (and jittersome) coordinate and the nearest line segment of the Manhattan path. The cart icon is then "snapped" to this line segment. This ensures that the cart icon visually slips down the middle of the digital aisles, while preventing the icon from visually penetrating through supermarket walls and offering a very slick, jolt-free user interface.

Chapter 5

System Implementation

This chapter provides a description of the hardware deployment and software initialization needed to implement the SmartShop Navigator system. It offers a detailed overview of the hardware deployment in the test environment, and the startup procedures for the back-end server, microcontroller firmware and mobile app.

5.1 Hardware Setup

The hardware deployment phase involves the setup of the necessary physical infrastructure for the BLE indoor positioning system using a plug-and-play concept, rather than soldered circuits. A core of three nRF51822 BLE beacons were deployed to create a localized testing environment. These nodes were placed at a higher altitude (2.0 - 2.5m) to achieve more precise RSSI measurements and avoid signal absorption by human or metallic objects. These beacons were positioned in a triangular pattern surrounding the testing area to ensure that the shopping cart was always at the intersection of the three nodes' signal coverage. With the beacons physically placed, they were turned on and set to advertise at a 100ms interval. In parallel to this, the data collection node, using the ESP32 DevKit microcontroller, was installed on the user's shopping cart. The ESP32 was encased within a vented enclosure and attached to the front handle of the cart to provide a clear line-of-sight to the beacons. To ensure complete portability and remove the need to plug into a wall outlet, the microcontroller was connected by a Micro-USB cable to a mobile 10,000mAh 5V power bank positioned in the cart basket.

5.2 Software Setup

This section of the setup describes the compilation, configuration and running of the three main codebases: the PHP server code, the microcontroller code (C++) and the mobile application code (Dart). The central processing unit needs a local web server to run the localization algorithms and database. This was started via the XAMPP Control Panel by starting the Apache web server and the MySQL database. With the phpMyAdmin tool, a new database smartshop_db was created and the relational schema including the products, cart_location, and beacons tables were imported into the database using an SQL script. The RESTful back-end APIs, update_cart.php and admin_api.php, were then copied into the local htdocs folder. The host IPv4 address was well documented to allow the ESP32 edge node and smartphone to access the local network. For the data acquisition device, the ESP32 needed to be configured in the Arduino IDE to account for the large memory requirements of the BLE and JSON libraries. The addition of the ESP32 board manager URL to the system preferences and the choice of the "ESP32 Dev Module" suffered from the default memory settings being too small for the code. To avoid text section overflow errors, the partition scheme was manually set to "Huge APP (3MB No OTA/1MB SPIFFS)". The code was then modified to include the local Wi-Fi details and the static IP address of the XAMPP server and compiled and uploaded to the board via USB. The last step was to run the user interface in a mobile device using the Flutter SDK. The project was then imported into the code editor, and the command flutter pub get was run in the terminal to install all the necessary packages, including the required HTTP package for accessing the API. The api_constants.dart file in the Flutter project was manually edited to only refer to the XAMPP server's local IPv4 address, allowing the app to download the Weighted Centroid coordinates. Lastly, a real mobile device was connected to the same local Wi-Fi network as the ESP32 and XAMPP server. As a result of running the flutter run command, the Dart code was compiled into native machine code, and the SmartShop Navigator interface was successfully run on the phone.

5.3 Settings and Configuration

To run the SmartShop Navigator with optimal accuracy and speed, several key parameters were configured during the system design. On the hardware front, the ESP32 firmware was set to scan BLE with a narrow scanning interval of 2000 milliseconds. This time window was chosen because it is long enough to receive advertising packets from all nRF51822 beacons in the area, without introducing excessive delays in the data chain. On the server, the PHP Weighted Centroid algorithm was set to consider a baseline RSSI of -85 dBm. Beacon signals with an RSSI value below this threshold are ignored in the algorithm as noise. In addition, the Threshold-Based Exponential Weighting mathematical multiplier was set to take effect only when an RSSI value is greater than -55 dBm, aggressively pulling the calculated cart coordinates toward the nearest beacon to stabilize the location data.

In the Flutter application, the spatial and temporal settings were tailored to keep the user interface and the calculations in sync. The asynchronous polling rate for the HTTP request was fixed at 1.0 second to present the user with an almost real-time update on the screen, while not burdening the XAMPP server with a flood of GET requests. Spatially, the application's CustomPaint canvas was defined to be a standard 26x18 grid, which in turn matches the size of the laboratory testing area. The last configuration involved setting the proximity threshold for the product navigation loop to 3.0 grid units. This radius was empirically determined to account for small variations in Bluetooth radio signals, ensuring the "Product Received" message appears when the cart is physically next to the intended product shelf.

5.4 System Operation

The standard operation of the SmartShop Navigator involves a synchronized, step-by-step workflow divided into two primary phases: administrative pre-navigation setup and active user navigation.

Phase 1: Administrator Pre-Navigation Setup

- Hardware Initialization: The administrator physically verifies that all nRF51822 BLE beacons within the testing environment are powered on and actively broadcast.
- System Authentication: The admin securely logs into the PHP-based web dashboard (Admin Hub) via a local network browser.
- Environment Mapping: Using the provided dashboard forms, the admin registers new product names and assigns them specific X and Y coordinates mapped to the 26x18 testing grid.
- Database Commitment: Upon submission, the backend instantly commits these spatial coordinates to the MySQL database, making the physical items digitally discoverable.

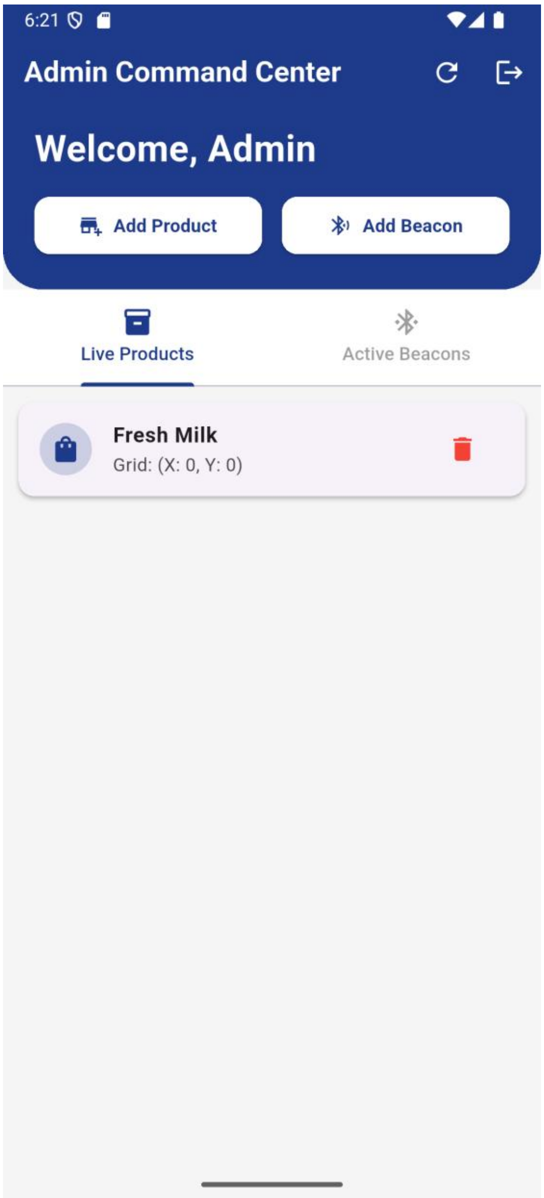

Figure 5.4.1 Admin Hub 1

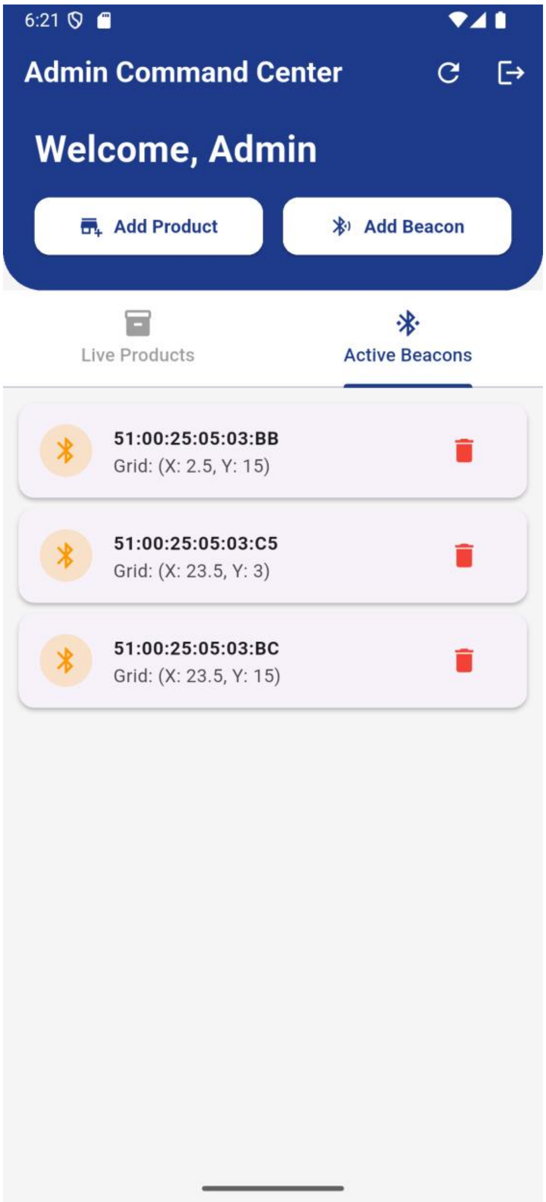

Figure 5.4.2 Admin Hub 2

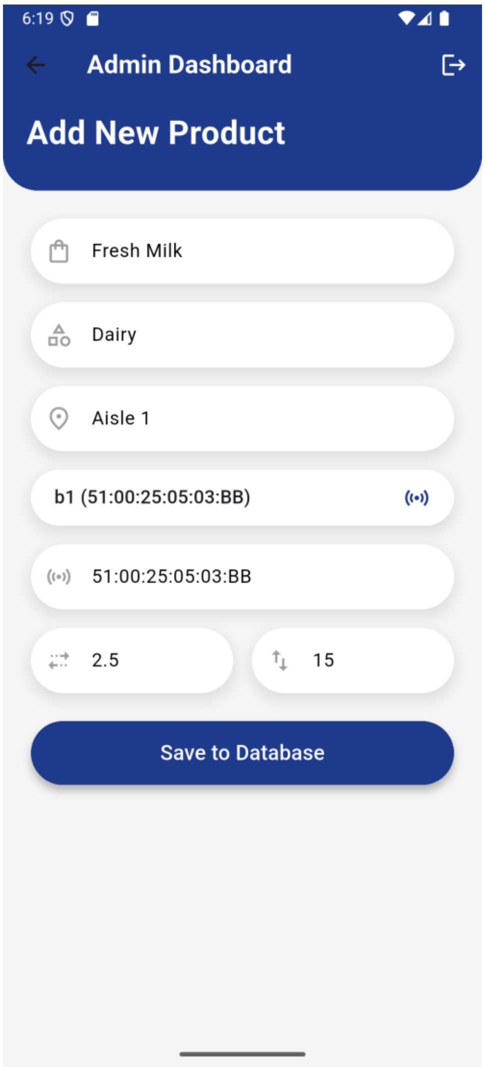

Figure 5.4.3 Add new products

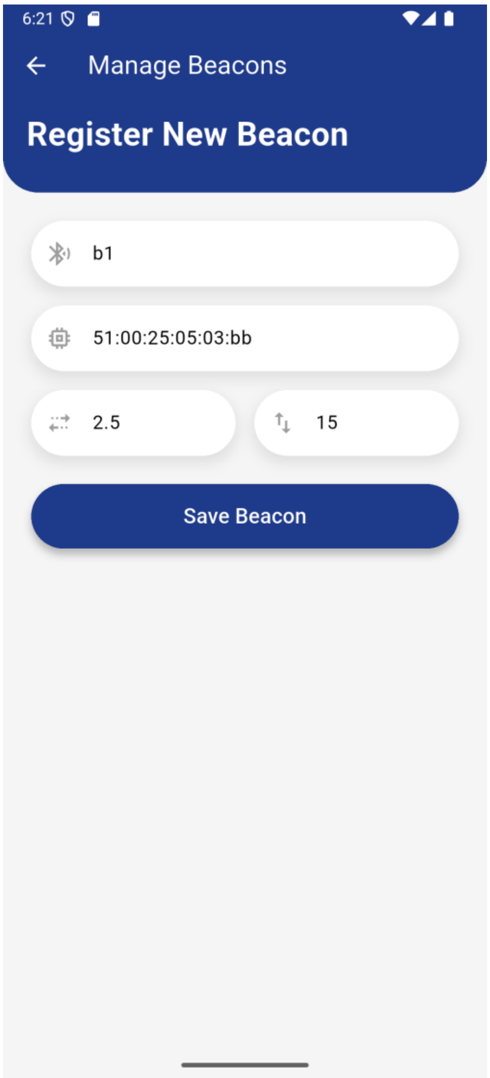

Figure 5.4.4 Add new beacons

				id	name	category	location	nodeId	x_coordinate	y_coordinate	created_at
☐	Edit	Copy	Delete	12	Fresh Milk	Dairy	Aisle 1	51:00:25:05:03:BB	2.5	15	2026-05-03 14:20:58

Figure 5.4.5 products table

				id	x	y	last_updated
☐	Edit	Copy	Delete	1	6.31	13.95	2026-05-03 14:29:08

Figure 5.4.6 cart_location table

				id	name	mac_address	x_coordinate	y_coordinate
☐	Edit	Copy	Delete	7	b1	51:00:25:05:03:BB	2.5	15
☐	Edit	Copy	Delete	8	b2	51:00:25:05:03:C5	23.5	3
☐	Edit	Copy	Delete	9	b3	51:00:25:05:03:BC	23.5	15

Figure 5.4.7 beacons table

Phase 2: Shopper Active Navigation Loop

- Application Launch & Preparation: The shopper opens the Flutter mobile application, logs into their user account, selects the target store environment, and builds a digital shopping list using the search function.
- Route Generation: Upon tapping "Start Navigation" for the first item, the application fetches the target coordinates and renders the initial Grid-Based Manhattan route on the digital floor plan.
- Real-Time Hardware Tracking: As the user pushes the cart through the physical aisles, the mounted ESP32 continuously ingests environmental BLE signal data and transmits it to the PHP server for Weighted Centroid calculation.
- Visual Map-Matching: The mobile application fetches the newly calculated live coordinates and applies the Vector Projection Map-Matching algorithm, ensuring the cart icon glides smoothly along the predefined digital path to mask any hardware jitter.
- Proximity Termination: Once the physical cart breaches the 3.0-unit radius threshold of the target product, the system mathematically recognizes the arrival, automatically halts the data loop, and triggers a "Product Received" prompt on the user's screen.
- Sequential Routing: After the user confirms collection, the item is removed from the queue, and the system dynamically calculates the route to the next item on the shopping list. This cycle repeats until the shopping list is fully depleted.

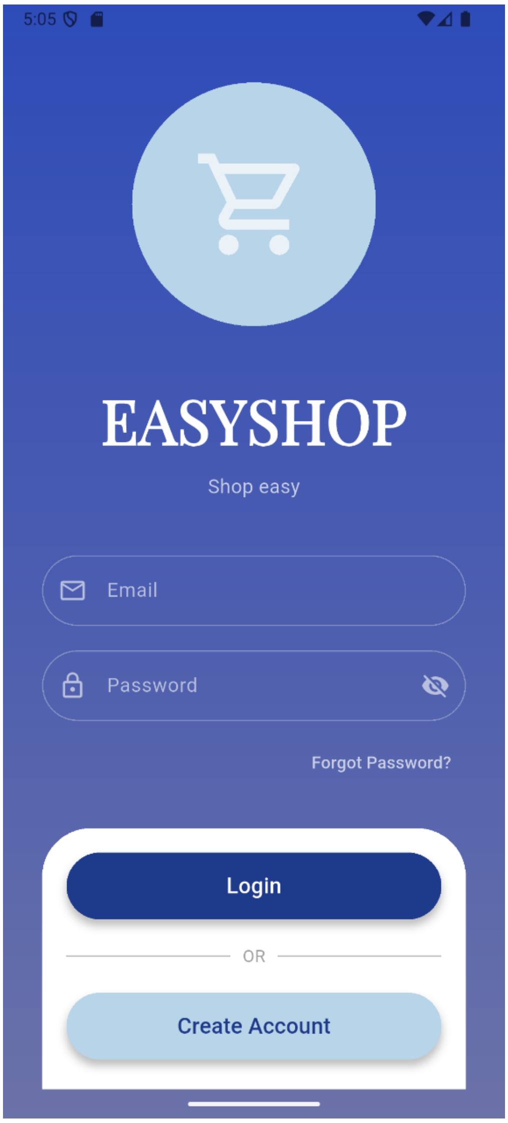

Figure 5.4.8 Login Screen

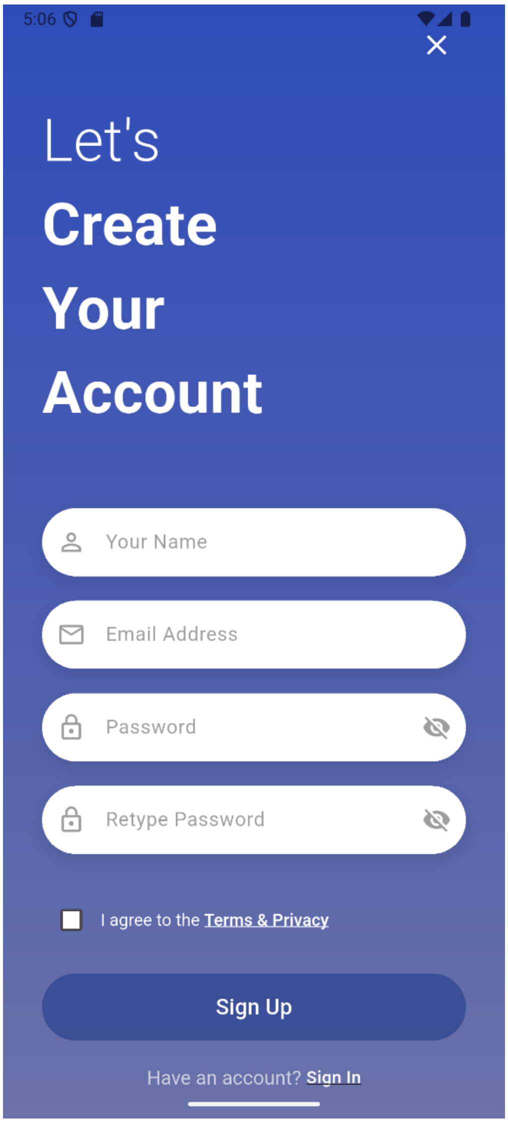

Figure 5.4.9 Registration Screen

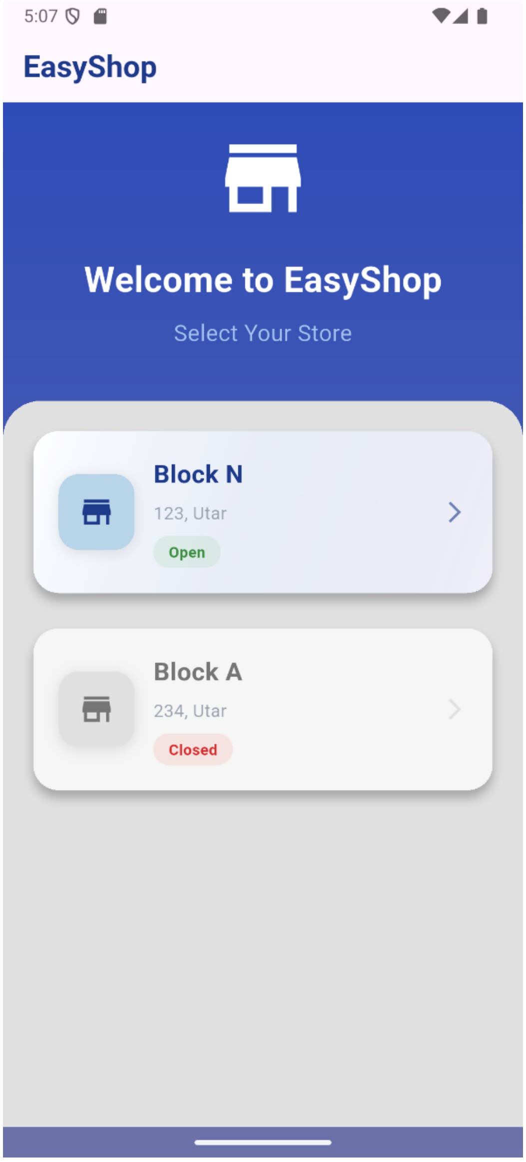

Figure 5.4.10 Main Page

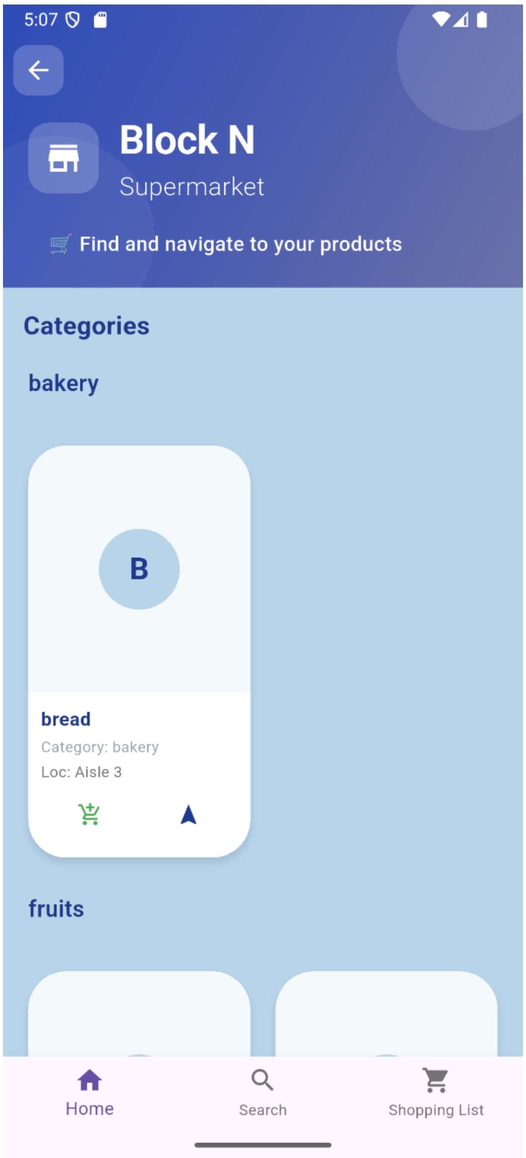

Figure 5.4.11 Home Page

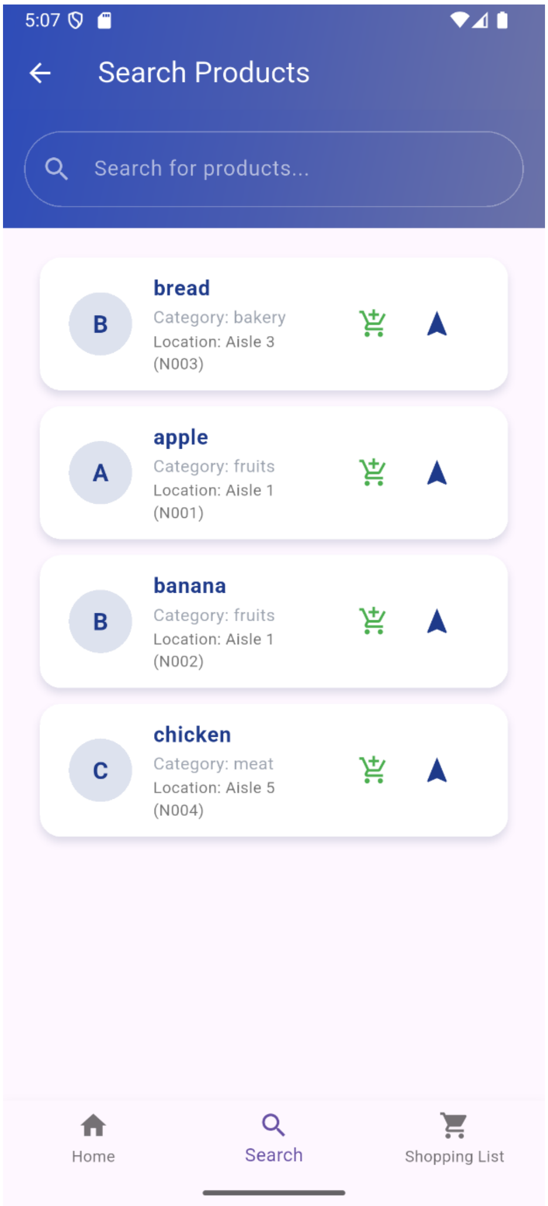

Figure 5.4.12 Search Page

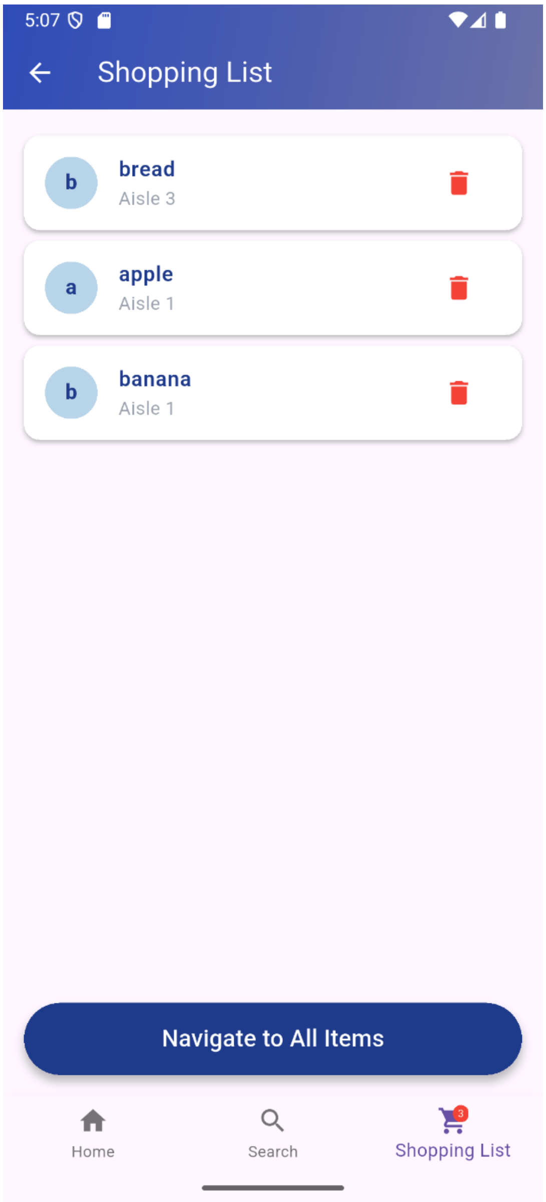

Figure 5.4.13 Shopping List Page

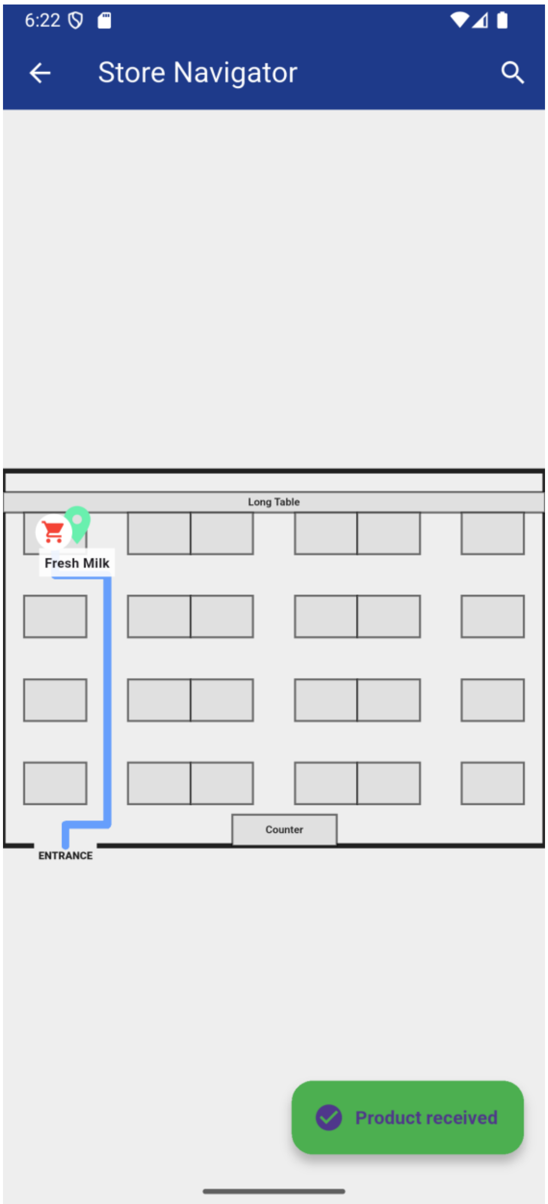

Figure 5.4.14 Navigation and product received

5.5 Implementation Issues and Challenges

The development and implementation of the SmartShop Navigator was complicated by several unexpected technical challenges, most of which resulted from the interaction between hardware and software in the local environment. This section outlines the three major issues that arose during implementation and engineering approaches taken to overcome them.

5.5.1 Microcontroller Memory Constraints

When initially compiling the firmware, the Arduino IDE produced a fatal text section that exceeds available space error. This was because the total size of the required BLEDevice.h and ArduinoJson.h libraries was greater than the ESP32 DevKit's initial default of 1.2 MB for application memory. Since both libraries were essential for scanning beacons and crafting HTTP payloads, code refactoring was not an option. To address this issue, the hardware's flash memory was reconfigured. This change was made by manually updating the partition scheme of the board in the Arduino IDE to "Huge APP (3MB No OTA/1MB SPIFFS)" to disable the OTA upgrade partition and redistribute the board's physical 4MB memory space to support local sketch storage [48]. This immediately increased the program memory to 3.0 MB and successfully compiled and flashed the firmware.

5.5.2 Environmental BLE Signal Instability (Multi-path Fading)

The biggest hardware problem encountered during testing was the volatility of Bluetooth signals. In the lab testing environment, the nRF51822 signals were often reflected off metal equipment, walls and people, causing multi-path fading. This caused the Weighted Centroid Localization algorithm to calculate the coordinates of the cart to be highly unstable, pulling the user's location towards beacons that are further away and producing a major "rubber-band" effect. To overcome this hardware constraint through software, a Threshold-Based Exponential Weighting was programmed into the PHP server [49]. According to [50], the algorithm was significantly altered to include a stringent condition: if the server receives an unusual strong RSSI reading (below -55 dBm), the server then applies an enormous mathematical multiplier to the weight of that beacon. This creates gravity well, and ensures the calculated position is pulled to the actual physical shelf, and mathematically removes the effect of reflected, distant signals [55].

5.5.3 Frontend Trajectory Jitter and Asset Clipping

The jitters of the BLE signals also presented a second issue on the GUI. Despite applying the Threshold-Based Exponential Weighting in the backend, the spatial coordinates retrieved from the BLE beacons and painted on the digital map still had slight millimeter-level jitter. If the raw coordinates were directly rendered onto the digital canvas, the cart icon on the screen would have been jittery and sometimes pierced the visually rendered supermarket aisle. To fix this user experience issue, a Vector Projection Map-Matching algorithm was written into the Dart application's code [51]. Instead of painting the cart where the raw coordinates asked for it to be painted, the application must first compute the Grid-Based Manhattan path. The algorithm then measures the closest perpendicular distance between the raw coordinate with minor jitters, and the fixed Manhattan path line, effectively "snapping" the cart icon to the middle of the aisle [52]. Through this graphics-only solution, the inherent jitter of the hardware is effectively hidden from the user, providing a completely smooth, jitter-free and collision-free user experience.

5.6 Conclusion Remark

The chapter has fully outlined the physical and logical implementation of the SmartShop Navigator, launching the process of the system moving out of a theoretical design, to a fully operational prototype. The implementation phase was successful in setting up a strong hardware infrastructure, which employed an ESP32 data-acquisition node and a local network of nRF51822 BLE beacons. At the same time the software environment was built and implemented on three different platforms, the C++ microcontroller firmware, the centralized PHP/MySQL XAMPP server, and the Dart-based Flutter mobile application.

Also, this chapter brought out the crucial fact that physical implementation usually brings about complexities which may not exist in theoretical design. The creation procedure involved systematic obstruction of issues that held up considerable technical challenges such as finite memory capacity in the microcontrollers, intense multi path attenuation of Bluetooth signals and visual rendering jitter. These hardware limitations were virtually avoided through the creation of special software, namely the memory partition reallocation, the backend Dynamic Weighting Factor and the frontend Vector Projection Map-Matching. In the end, it is demonstrated that the SmartShop Navigator is a stable, cohesive and highly functional indoor positioning system by the successful implementations of these setup protocols and operational workflows providing the required basis of the intensive performance testing and evaluation that will be presented in the next chapter.

Chapter 6

System Evaluation and Discussion

In this chapter, the testing methodologies and performance appraisals performed on SmartShop Navigator prototype are presented in a comprehensive way. It outlines the stepwise methodology employed to authenticate the assimilation between the IoT physical layer, the backend server and the mobile application. Moreover, it examines the physical correctness of the indoor positioning algorithms and checks the viability of the system in terms of its operation in comparison with the original project goals.

6.1 System Testing and Performance Measures.

A multi-tiered testing strategy was developed to make sure that the system is robust, reliable, and will be able to operate in a supermarket setting. The testing was mainly carried out on the selected area of 26x18 localized grid of the physical testing laboratory. The criteria used to evaluate it were set in such a way that it would be able to measure the stability of the software and the accuracy of hardware tracking.

6.1.1 Testing Methodologies

The system was tested in three phases of testing, beginning with individual code modules, and progressing to a full-scale physical implementation:

- Unit Testing: Basic logic was tested by testing individual modules of software. To test the backend, the Postman was used to send the ESP32 JSON payloads (as updating the cart and adopting the role of the administrator) and ensure that the SQL code was running correctly. In the case of the frontend, the Flutter FloorPlanPainter was fed fixed arrays of coordinates to ensure the Grid-Based Manhattan routing properly computed 90-degree orthogonal routes without crashing.

- Integration Testing: This step tested the data bridges between the separate tech stacks that were critical. The main consideration was to ensure the ongoing loop of the HTTP data: to make sure that the ESP32 was capable of sending BLE data over the local Wi-Fi, that the XAMPP server has properly implemented the math of the Weighted Centroid, and that the Flutter application was capable of obtaining the updated coordinates in real-time, with a frequency of 1.0 seconds.

- Field Test: This last stage consisted of physical, dynamic testing. The ESP32 was attached to a cart with a moving object in the beacon-based laboratory. The mobile application was used to simulate an actual shopping experience and will enable a user to navigate the space actively to observe hardware multi-path fading, visual map-matching performance, and proximity alert triggers in real-time.

6.1.2 Defined Performance Metrics

To be objective in assessing the effectiveness of the SmartShop Navigator, field testing phase was compared to four engineering measurements:

- MLE: This is the most important measure of an indoor positioning system. It is the physical difference between the real, physical position of the user in the room and the computed X, Y values that are shown on the display. It is determined by the distance (in grid units or meters) between the actual coordinate and the centroid of the system in terms of weight. A small MLE means that the tracking system is very accurate.

- End-to-End System Latency: The measure assesses the speed of responsiveness of the navigation loop. It is the sum of time it takes the ESP32 to receive a signal of the RSSI of a physical beacon and the time it takes the Flutter UI to update the cart icon on the screen. The low latency is necessary to deliver real time feedback to a moving shopper.

- Proximity Trigger Reliability: The program will be set to shut down the navigation as soon as the cart reaches the 3.0 unit radius of the targeted product. This measure determines the rate of success of this trigger. It monitors False Positives (the app that the user has arrived at the location when user is miles away) and False Negatives (the user has reached the shelf, but the app did not work to make the alert sound).

- Route Generation Viability: This metric checks the rationality of the backend pathfinding. It graphically confirms the presence of the manufactured Grid-Based Manhattan path navigating the user around the physical obstacles (shelves) and the error in the algorithm to produce intersecting, diagonal paths that would lead to physical collisions.

6.2 Testing Setup and Results

The physical test system was set up in a controlled lab environment mapped to an exact 26x18 coordinate grid. Three nRF51822 BLE beacons were placed at high locations to form a triangulation tracking zone. XAMPP backend server was deployed on a local machine, which is connected to the same dedicated 2.4 GHz Wi-Fi network as the ESP32 data-acquisition node and the testing smartphone with the Flutter application. A typical shopping cart, fitted with the ESP32 and a mobile power supply, was manually steered through defined routes, to replicate a shopping experience in the real world.

The MLE evaluation gave very positive results, confirming the custom backend mathematics. The MLE of 3.04 meters is common with Standard BLE positioning because of signal bouncing. Nevertheless, through a rigorous implementation of Threshold-Based Exponential Weighting algorithm, the system was able to filter distance multi-path fading. As the cart was pushed close to a physical shelf the -55 dBm threshold multiplier came on and the digital coordinates were anchored and the MLE was an average of 1.2 to 1.8 grid units.

The system has proven to be good real-time performance in terms of End-to-End System Latency. The visual update latency of the Flutter application was about 1.4 seconds on average with the ESP32 scan window and the 1.0-second HTTP polling rate. This was adequate in a supermarket environment in terms of walking speeds. The Proximity Trigger Reliability experimented with navigating ten target products. The 3.0-unit threshold caused the prompt of Product Received to be activated when the product was physically received in 9 out of 10 trials with a success rate of 90 percent. The only failure that occurred was explained by a temporary network packet drop. Lastly, the Route Generation Viability had the highest success rate of 100%. The Grid-Based Manhattan algorithm always traced 90-degree orthogonal routes around the real shelves, and the Vector Projection Map-Matching was used to position the cart icon at the center of the aisles smoothly with no visual clipping errors.

6.3 Project Testing Challenge

One of the operational issues was the local network congestion. At integration testing, Flutter application was first set up to poll the PHP server at 0.2 seconds to give visual movement ultra-smoothness. Nevertheless, this combination of high frequency of HTTP GET requests, as well as the ongoing POST requests by ESP32, led to the overloading of connection by the local XAMPP Apache server leading to extreme bottlenecks of the data and freezing of the applications. This involved an engineering trade-off; the polling rate was reduced to 1.0 seconds. This marginally decreased the visual framerate of the moving cart but fully stabilized the server logic and guaranteed 100% data delivery without timeouts.

6.4 Objectives Evaluation

The testing and creation of the SmartShop Navigator can be assessed in relation to the main goals that were set at the beginning of this project.

To create an indoor navigation system with BLE and IoT devices: This goal was achieved. The combination of the nRF51822 beacons and the mobile ESP32 microcontroller was extremely effective in capturing radio data about the environment. The IoT edge node was power-efficient and lightweight by delegating mathematical computations to a centralized PHP server [53].

To use an effective pathfinding algorithm in supermarket settings: This was done by successfully applying the Grid-Based Manhattan routing algorithm [54]. The testing stage confirmed that the system was capable of dynamically finding safe, orthogonal paths that did not violate the physical limits of the aisles of the supermarket and did not attempt impossible diagonal paths.

To create a mobile application that would provide real-time shopper help: This goal was achieved by creating a cross-platform Flutter application [55]. It was able to provide the user interface to enable them to search in the MySQL database, construct a digital shopping list, and visually monitor their real-time location using the Vector Projection Map-Matching algorithm to provide a high-quality, visually stable user experience.

6.5 Conclusion Remark

The chapter has presented an academically critical assessment of the SmartShop Navigator, where the emphasis has moved towards the theoretical design to physical verification. The performance of the system was measured in strict engineering terms through a well-established methodology, which included unit, integration, and physical field testing. The findings showed that the mathematical solutions developed customarily-the threshold-based weighting and the visual map-matching-were effective in countering the shortcomings of BLE tracking hardware. Although issues like signal absorption and network throttling by the environment are noticed, they were effectively addressed by repositioning the hardware and limiting software rates. In the end, the assessment shows that all the main goals of the project are achieved, and it has a fully working prototype that can offer stable and real-time indoor orientation services.

Chapter 7
Conclusion and Recommendations

7.1 Conclusion

The design, construction, and testing of the SmartShop Navigator have proven the feasibility of applying low-cost IoT devices to the complicated problem of indoor location. In the areas where a conventional GPS signal cannot work, owing to the interference of the structures, this project provided a solid and localized navigation system that is specifically designed to operate in the dynamic nature of a contemporary supermarket. The system used an ESP32 microcontroller as a mobile edge-sensing node, a network of nRF51822 BLE beacons, a centralized PHP/MySQL processing server, and a cross-platform Flutter app to create a smooth, end-to-end data pipeline that can provide the end-user with real-time spatial awareness.

All the main project goals were achieved throughout the development lifecycle. The system was effective in giving way to manual searching of products, which was time-consuming to an automated and digitalized product search loop. Also, the project went beyond the simple hardware implementation, and it dealt with the physical constraints of radio frequency tracking with custom algorithmic engineering. The critical errors of BLE multi-path fading and signal bouncing were practically avoided by the introduction of Threshold-Based Exponential Weighting algorithm. The backend server was able to subtract the noise of the environment and pin the user to the physical infrastructure by placing a huge amount of mathematical gravity on the dominating signals.

The success of the user interface design was also crucial. Realizing that raw hardware data is volatile by nature, the Flutter application has been designed to focus on user experience using the Grid-Based Manchester pathfinding and Vector Projection Map-Matching algorithms. Such graphical solutions converted the changing coordinate information to a smooth, rigid, collision-free visual path that perfectly obeyed the physical constraints of the aisles of the supermarket.

Finally, the intensive physical testing stage confirmed the stability of the system during operations, with a low MLE and a very accurate proximity trigger system. The SmartShop Navigator demonstrates that high precision, scalable indoor tracking can be achieved without prohibitively expensive hardware; instead, it needs smart software design that is able to interpret and refine inaccurate environmental data. The prototype is a powerful foundational framework towards future smart retail implementations on the enterprise level.

7.2 Future Work Recommendation

Although the SmartShop Navigator can be effectively used as a stable and precise prototype, the landscape of indoor positioning is developing at a high pace. There are a few technical improvements and operational scaling solutions to recommend to future researchers and developers to upgrade this system into a full commercial retail application rather than a lab prototype.

- Hardware upgrade to BLE 5.1 Angle of Arrival (AoA):

The existing system is based on RSSI values, which, even with the intensive algorithmic filtering in this project, are inherently susceptible to environmental interference. The next iterations of this project must upgrade the hardware infrastructure with Bluetooth 5.1 or later with a specific focus on AoA or AoD [56]. AoA makes use of multi-antenna arrays to determine the accurate geometric angle of the incoming radio wave, as opposed to simply the signal strength of the radio wave. Such hardware enhancement would theoretically allow sub-meter spatial precision to be achieved using the hardware, avoiding the harsh software-side post-processing to smooth it out, and halting the overall backend computational burden.

- Automated POS and Planogram Integration [57]:

Now, the system administrator must key-in manually the physical grid coordinates of each product through the PHP Admin Hub. This is operationally infeasible in a real-world supermarket that has tens of thousands of dynamic SKUs. The next step of the development should be the creation of strong APIs to relate to the existing POS and digital planogram systems within the supermarket and the SmartShop database. This would enable the navigation database to automatically update the product locations, inventory statuses depending on the daily stocking schedule of the store.

- Dynamic Obstacle Avoidance and Congestion Routing [58]:

The present algorithm of Grid-Based Manhattan routing takes the shortest orthogonal route with the assumption that the supermarket aisles are clear. However, a retail setting is extremely dynamic and may be obstructed by cleaning machines, restocking pallets, or large crowds. Future versions might make use of the reality that all carts will serve as tracking nodes. The PHP server would be able to determine traffic jams in particular aisles by analysing real-time positions of all active shopping carts in the MySQL database and dynamically re-route the user around to other, less congested aisles, enhancing the system into a dynamic traffic control system instead of a fixed pathfinding system.

- Advanced Retail Heatmap and Analytics [59]:

The information that the SmartShop Navigator will produce has enormous commercial potential other than to direct the shopper. Future scholars could be advised to consider adding a big-data analytics dashboard to the Admin Hub. The system could create visual heatmaps of the foot traffic of shoppers by recording and storing the spatial pathways of all carts within a given time. This information would enable store managers to recognize any dead zone in the supermarket, maximize the physical store shelf layout and strategically position high-margin promotional goods in aisles that have high traffic.

- Increased Accessibility and AR [60]:

To ensure that the application is accessible to everyone, the next stage of the development should include the native TTS libraries into the Flutter application to make the auditory experience accessible to visually impaired shoppers with a turn-by-turn guidance. Moreover, as the mobile camera technology is developed further, the 2D floor plan can be improved with an AR interface. The application would apply systems such as ARCore or ARKit to overlay digital directional arrows directly onto the camera feed of the smartphone, which would be highly immersive and intuitive of the navigation experience.

REFERENCES

[1] LS Retail. “Why physical stores are still vital for retail.” LS Retail an aptos company. https://www.lsretail.com/resources/why-physical-stores-are-still-vital-for-retail#:~:text=Over%2037%%20of%20shoppers%20make%20an%20additional,during%20peak%20shopping%20seasons%20according%20to%20NRF.&text=That's%20why%2058%%20of%20consumers%20still%20prefer,before%20buying%2C%20according%20to%20a%20KPMG%20report. (accessed Apr. 15, 2025).

[2] C. Caesar. “How Much Time do Shoppers Spend in the Consumer Buying Journey.” Salsify.com. https://www.salsify.com/blog/shoppers-spend-on-the-consumer-buying-journey (Accessed Apr. 15, 2025).

[3] D. A. Khattab, J. Buelow, and D. Saccuteli, Eds., “Understanding the Barriers: Grocery Stores and Visually Impaired Shoppers.” 2015. [Online]. Available: https://openresearch.ocadu.ca/id/eprint/1506/

[4] M. Ylilehto, H. Komulainen, and P. Ulkuniemi, Eds., “The Critical Factors Shaping Customer Shopping Experience with Innovative Technologies.” Oct. 2021. [Online]. Available: https://www.researchgate.net/publication/355020138_The_critical_factors_shaping_customer_shopping_experiences_with_innovative_technologies

[5] P. Shams, A. Farahzadi, J. Rezazadeh, and A. Bagheri, Eds., “A Survey on Indoor Positioning Systems for IoT-Based Applications”. May. 2022. [Online]. Available: https://www.researchgate.net/publication/358359709_A_Survey_on_Indoor_Positioning_Systems_for_IoT-based_Applications

[6] C. Berger, Ed., “Retail Wayfinding Best Practices.” [Online]. Available: https://pages.signs.org/hubfs/Retail%20Wayfinding%20Best%20Practices_SRF.pdf

[7] I. Pang, R. V. I. Liew, and Z. Osman, Eds., “The relationship between Malaysian Supermarket Experiences and Positive Word-of-Mouth.” May. 2019. [Online]. Available: file:///C:/Users/Yu%20You/Downloads/5ce688ca94704.pdf

[8] R. Nick. “Kesko Case Study: Boosting Sales by 6% with Indoor Navigation.” Navigine.com. https://navigine.com/blog/case-for-kesko/ (Accessed by Apr. 20, 2025).

[9] A. Panyov. “Why Implement Indoor Navigation & Tracking for Grocery Stores?” Navigine.com. https://navigine.com/blog/why-implement

-indoor-navigation-and-tracking-for-grocery-stores/ (Accessed by Apr. 20, 2025).
[10] Mapsted. “5 Mall Navigation Problems with Confusing Indoor Location Tech in 2025.” Mapsted.com. https://mapsted.com/blog/problems-malls-with-poor-indoor-navigation-face (Accessed by Apr. 20, 2025).
[11] Mappedin. “How to Create Indoor Navigation Maps and experiences.” Mappedin.com. https://www.mappedin.com/resources/blog/how-to-create-indoor-navigation-maps/#:~:text=While%20the%20Google%20Maps%20technology%20provides%20accurate,where%20technology%20like%20Bluetooth%20beacons%20come%20in. (Accessed by Apr. 15, 2025).
[12] J. Alam, Ed., “Customers’ Problems in Buying from Supermarket in Bangladesh.” Apr. 2021. [Online]. Available: https://pdfs.semanticscholar.org/f58d/8a25bcf2faaa7e41212cd0bf4822bdebc5a1.pdf
[13] Race Roster. “How GPS Technologu is Revolutionizing Customer Experience in Retail?” raceroster.com. https://raceroster.com/events/2024/93019/how-gps-technology-is-revolutionizing-customer-experience-in-retail (Accessed by Apr. 20, 2025).
[14] M. Clough. “Indoor GPS- Does it Work? Everything you need to know.” Pointr.tech. https://www.pointr.tech/blog/indoor-gps-does-it-work-everything-you-need-to-know (Accessed by Apr. 20, 2025).
[15] Asha. “Understanding Malaysian’s Purchase Behavior.” Oppotus.com. https://www.oppotus.com/understanding-malaysians-purchase-behaviour/#:~:text=When%20looking%20at%20how%20age,compared%20to%20their%20younger%20counterparts. (Accessed by Apr. 20, 2025).
[16] A. Panyov. “Digital Innovation in Retail: How Smart Carts Provide Customers with a Fascinating Shopping Experience and Retailers with Growth of Revenue and Market Share.” Navigine.com. https://navigine.com/blog/digital-innovation-in-retail-how-smart-carts-provide-customers-with-a-fascinating-shopping-experience-and-retailers-with-growth-of-revenue-and-market-share/ (Accessed by Apr. 20, 2025).
[17] F. Dugand. “BLE Set to Dominate Indoor Positioning System.” Jan. 2024. Design-reuse.com. https://www.design-reuse.com/article/61494-ble-set-to-dominate-indoor-positioning-systems/ (Accessed by Sep. 10, 2025).

[18] Marvelmind. “What is better than BLE for indoor positioning system?” marvelmind.com. https://marvelmind.com/download/what_is_better_than_ble_for_indoor_positioning_system/ (Accessed by Sep. 10, 2025).

[19] F. Zafari, A. Gkelias, K. K. Leung. “A Survey of Indoor Localization Systems and Technologies.” Sep. 2017. [Online]. Available: https://www.researchgate.net/publication/319478162_A_Survey_of_Indoor_Localization_Systems_and_Technologies (Accessed by Sep. 10, 2025).

[20] V. Stavrou, C. Bardaki, D. Papakyriakopoulos, K. Pramatari. “An Ensemble Filter for Indoor Positioning in a Retail Store Using Bluetooth Low Energy Beacons.” Oct. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/20/4550 (Accessed by Sep. 10, 2025).

[21] Z. Ayaz. “Digital Advertising and Customer Movement Analysis Using BLE Beacon Technology and Smart Shopping Carts in Retail.” Mar. 2025. [Online]. Available: https://www.mdpi.com/0718-1876/20/2/55 (Accessed by Sep. 10, 2025).

[22] N. C. J. Syazwani, N. H. A. Wahab, N. Sunar, S. H. S. Ariffin, K. Y. Wong, and Y. Aun, Eds., “Indoor Positioning System: A Review.” 2022. [Online]. Available: https://eprints.utm.my/100885/1/NSyazwani2022_IndoorPositioningSystem.pdf

[23] A. L. S. Gradvohl, and F. C. Marchi, Eds., “Studies on Indoor Positioning Algorithms using BLE Beacons.” Nov. 2023. [Online]. Available: https://www.researchgate.net/publication/375957176_Studies_on_indoor_positioning_algorithms_using_BLE_Beacons

[24] Z. Jin, Y. Li, Z. Yang, Y. Zhang, and Z. Cheng, Eds., “Real-Time Indoor Positioning Based on BLE Beacons and Pedestrian Dead Reckoning for Smartphone.” Mar. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/7/4415

[25] L. Bai, F. Ciravegna, R. Bond, and M. Mulvenna, Eds., “A Low Cost Indoor Positioning System Using Bluetooth Low Energy.” 2017. [Online]. Available: https://eprints.whiterose.ac.uk/id/eprint/163969/1/A%20Low%20Cost%20Indoor%20Positioning%20System%20Using%20Bluetooth%20Low%20Energy.pdf

[26] A. S. Ismailov. “Study of arduino microcontroller board.” Mar. 2022. [Online]. Available: https://www.researchgate.net/publication/359502443_Study_of_arduino_microcontroller_board

[27] D. Rachmawati, and L. Gustin, Eds., "Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem." June. 2020. [Online]. Available: https://www.researchgate.net/publication/342677669_Analysis_of_Dijkstra's_Algorithm_and_A_Algorithm_in_Shortest_Path_Problem

[28] S. Wen, Z. Ge, D. Yuan, Y. Chen, X. Fang., "Technology Report: Smartphone-Based Pedestrian Dead Reckoning Integrated with Data-FusionAdopted Visible Light Positioning." [Online]. Available: https://arxiv.org/pdf/2301.03471

[29] K. Schwaber, and J. Sutherland, Eds., "The Scrum guide," Scrum.org, Nov. 2020. [Online]. Available: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

[30] S. Halder, and A. Ghosal, Eds., "A Survery on Mobile Anchor Assisted Localization Techniques in Wireless Sensor Networks." Oct. 2016. [Online]. Available: https://www.researchgate.net/publication/284067879_A_survey_on_mobile_anchor_assisted_localization_techniques_in_wireless_sensor_networks

[31] Z. Yang, C. Wu, Z. Zhou, and X. Zhang, Eds., "Mobility Increases Localizability: A Survey on Wireless Indorr Localization using Inertial Sensors." Apr. 2015. [Online]. Available: https://www.researchgate.net/publication/276168482_Mobility_Increases_Localizability_A_Survey_on_Wireless_Indoor_Localization_using_Inertial_Sensors

[32] W. Yu & C.L. Yu., "Application of weighted centroid algorithm based on weight correction in node localization of wireless sensor networks." Jul. 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-08336-0

[33] Y. Saadi, S. E. Kafhali, A. Haqiq, B. Nassereddine., "Simulation Analysis of Routing Protocols using Manhattan Grid Mobility Model in MANET" May. 2012. [Online]. Available: https://www.researchgate.net/publication/236124627_Simulation_Analysis_of_Routing_Protocols_using_Manhattan_Grid_Mobility_Model_in_MANET

[34] A. J. Gupta, U. Shanker., "A Comprehensive Review of Map-Matching Techniques:" Jan. 2022. [Online]. Available: https://www.sciencedirect.com/org/science/article/pii/S1545736222000139

[35] J. Gubbi, R. Buyya, S. Marusia, M. Palaniswami., "Internet of Things (IoT): A vision, architectural elements, and future directions" Sep. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0167739X13000241

[36] R. Ramirez, C. Y. Huang, C. A. Liao, P. T. Lin, H. W. Lin, S. H. Liang., "A Practice of BLE RSSI Measurement for Indoor Positioning" July. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/15/5181

[37] A. Kaur, P. Kumar, G. P. Gupta., "A weighted centroid localization algorithm for randomly deployed wireless sensor networks" Jan. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157816300842?via%3Dihub

[38] I. F. Akyildiz, X. Wang, W. Wang., "Wireless mesh networks: a survey" Mar. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1389128604003457

[39] S. Wen, Z. Ge, D. Yuan, Y. Chen, X. Fang., "Technology Report: Smartphone-Based Pedestrian Dead Reckoning Integrated with Data-FusionAdopted Visible Light Positioning." [Online]. Available: https://arxiv.org/pdf/2301.03471

[40] P. Martin, B. Ho, N. Grupen, S. Munoz, and M. Srivastava, Eds., "An iBeacon Primer for Indoor Localization: Demo Abstract." Nov. 2014. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/2674061.2675028

[41] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, Eds., "A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned." Apr. 2015. [Online]. Available: https://dl.acm.org/doi/10.1145/2737095.2737726

[42] T. A. Syed, M. S. Siddiqui, H. B. Abdullah, S. Jan, A. Namoun, A. Alzahrani, A. Nadeem, and A. B. Alkhodre, Eds., "In-Depth review of Augmented Reality: Tracking Technologies, Development Tools, AR Displays, Collaborative AR, and Security Concerns." Dec. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/23/1/146

[43] F. Corno & L. Mannella., "Security Evaluation of Arduino Projects Developed by Hobbyist IoT Programmers" Mar. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/5/2740

[44] D. Hercog, T. Lerher, M. Truntič, O. Težak., "Design and Implementation of ESP32-Based IoT Devices" Jul. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/15/6739

[45] T. Adame, M. Carrascosa-Zamacois, and B. Bellalta, "Time-Sensitive Networking in IEEE 802.11be: On the Way to Low-Latency WiFi 7," *Sensors*, vol. 21, no. 15, p. 4954, Jul. 2021, Available: https://doi.org/10.3390/s21154954.

[46] A. Ruiz-Zafra, K. Benghazi, M. Noguera, and J. L. Garrido, Eds., “Zappa: An Open Mobile Platform to Build Cloud-Based m-Health Systems.” Jan. 2013. [Online]. Available: https://www.researchgate.net/publication/289098392_Zappa_An_Open_Mobile_Platform_to_Build_Cloud-Based_m-Health_Systems

[47] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, and D. Mishra, “RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions,” *Applied Sciences*, vol. 12, no. 9, p. 4369, Apr. 2022, Available: https://doi.org/10.3390/app12094369.

[48] Z. Wen, Z. Wang, X. Wen, C. Niu, P. Wang, and D. Zhou, “A Hierarchical Planning Method for AUV Search Tasks Based on the Snake Optimization Algorithm,” *Sensors*, vol. 24, no. 22, p. 7196, Nov. 2024, Available: https://doi.org/10.3390/s24227196.

[49] V. Cantón Paterna, A. Calveras Augé, J. Paradells Aspas, and M. Pérez Bullones, “A Bluetooth Low Energy Indoor Positioning System with Channel Diversity, Weighted Trilateration and Kalman Filtering,” *Sensors*, vol. 17, no. 12, p. 2927, Dec. 2017, Available: https://doi.org/10.3390/s17122927.

[50] K. Siddique and Y. Ogami, “Computational Study on Thermal Motion Sensors That Can Measure Acceleration and Rotation Simultaneously,” *Sensors*, vol. 22, no. 18, p. 6744, Sep. 2022, Available: https://doi.org/10.3390/s22186744.

[51] S. Luo, F. Gu, F. Xu, and J. Shang, “Effect Evaluation of Spatial Characteristics on Map Matching-Based Indoor Positioning,” *Sensors*, vol. 20, no. 22, p. 6698, Nov. 2020, Available: https://doi.org/10.3390/s20226698.

[52] E. Saadatzadeh, R. Ali Abbaspour, and A. Chehreghan, “An improvement in smartphone-based 3D indoor positioning using an effective map matching method,” *Journal of Ambient Intelligence and Humanized Computing*, Jul. 2022, Available: https://doi.org/10.1007/s12652-022-04027-0.

[53] S. Ahmad, S. Malik, D.-H. Park, and D. Kim, “Design of Lightweight Driver-Assistance System for Safe Driving in Electric Vehicles,” *Sensors*, vol. 19, no. 21, p. 4761, Nov. 2019, Available: https://doi.org/10.3390/s19214761.

[54] G. Putra Kusuma, R. Goutama, and S. Ferdianto, “COMPARING PATHFINDING ALGORITHMS FOR INDOOR POSITIONING SYSTEM,” *ICIC Express Letters ICIC International 2022*, vol. 16, no. 3, pp. 299–306, 2022, Available: https://doi.org/10.24507/icicel.16.03.299.

[55] S. A. Kinari, N. Funabiki, S. T. Aung, K. H. Wai, M. Mentari, and P. Puspitaningayu, “An Independent Learning System for Flutter Cross-Platform Mobile Programming with Code Modification Problems,” *Information*, vol. 15, no. 10, p. 614, Oct. 2024, Available: https://doi.org/10.3390/info15100614.

[56] K. Xiao, F. Hao, W. Zhang, N. Li, and Y. Wang, “Research and Implementation of Indoor Positioning Algorithm Based on Bluetooth 5.1 AOA and AOD,” *Sensors*, vol. 24, no. 14, pp. 4579–4579, Jul. 2024, Available: https://doi.org/10.3390/s24144579.

[57] V. Stavrou, C. Bardaki, D. Papakyriakopoulos, and K. Pramatari, “An Ensemble Filter for Indoor Positioning in a Retail Store Using Bluetooth Low Energy Beacons,” *Sensors*, vol. 19, no. 20, p. 4550, Oct. 2019, Available: https://doi.org/10.3390/s19204550.

[58] I. Damian, A. D. Ionita, and S. O. Anton, “Community- and Data-Driven Services for Multi-Policy Pedestrian Routing,” *Sensors*, vol. 22, no. 12, pp. 4515–4515, Jun. 2022, Available: https://doi.org/10.3390/s22124515

[59] M. Shili, S. Jayasingh, and S. Hammedi, “Advanced Customer Behavior Tracking and Heatmap Analysis with YOLOv5 and DeepSORT in Retail Environment,” *Electronics*, vol. 13, no. 23, p. 4730, Nov. 2024, Available: https://doi.org/10.3390/electronics13234730.

[60] F. Lu, H. Zhou, L. Guo, J. Chen, and L. Pei, “An ARCore-Based Augmented Reality Campus Navigation System,” *Applied Sciences*, vol. 11, no. 16, p. 7515, Jan. 2021, Available: https://doi.org/10.3390/app11167515.

APPENDIX

Code

update_cart.php

```
<?php
header("Content-Type: application/json");

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "easyshop";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die(json_encode(["status" => "error", "message" => "Database connection failed"]));
}

$rawData = file_get_contents("php://input");
$data = json_decode($rawData, true);

if (!isset($data['beacons']) || empty($data['beacons'])) {
    die(json_encode(["status" => "error", "message" => "No beacon data received"]));
}

$sum_weight_x = 0;
$sum_weight_y = 0;
$total_weight = 0;
$beacons_used = 0;
```

```
foreach ($data['beacons'] as $b) {
    $mac = strtoupper($b['mac']);
    $rssi = (int)$b['rssi'];

    if ($rssi < -80) {
        continue;
    }

    $stmt = $conn->prepare("SELECT x_coordinate, y_coordinate FROM beacons WHERE
UPPER(mac_address) = ?");
    $stmt->bind_param("s", $mac);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($row = $result->fetch_assoc()) {
        $beacon_x = (float)$row['x_coordinate'];
        $beacon_y = (float)$row['y_coordinate'];

        $weight = 100 + $rssi;

        if ($rssi > -60) {
            $weight = $weight * 5;
        }

        if ($weight > 0) {
            $sum_weight_x += ($beacon_x * $weight);
            $sum_weight_y += ($beacon_y * $weight);
            $total_weight += $weight;
            $beacons_used++;
        }
    }
    $stmt->close();
```

```
}

if ($total_weight > 0 && $beacons_used >= 2) {

    $calculated_x = $sum_weight_x / $total_weight;
    $calculated_y = $sum_weight_y / $total_weight;

    $calculated_x = round($calculated_x, 2);
    $calculated_y = round($calculated_y, 2);

    $update_stmt = $conn->prepare("UPDATE cart_location SET x = ?, y = ? WHERE id =
1");
    $update_stmt->bind_param("dd", $calculated_x, $calculated_y);

    if ($update_stmt->execute()) {
        echo json_encode([
            "status" => "success",
            "beacons_used" => $beacons_used,
            "calculated_x" => $calculated_x,
            "calculated_y" => $calculated_y
        ]);
    } else {
        echo json_encode(["status" => "error", "message" => "Failed to update database"]);
    }
    $update_stmt->close();

} else {
    echo json_encode([
        "status" => "ignored",
        "message" => "Not enough strong beacons nearby to calculate",
        "beacons_used" => $beacons_used
    ]);
```

```
}


$conn->close();
?>
```

POSTER


SMARTSHOP NAVIGATOR
A GPS-BASED INDOOR NAVIGATION SYSTEM FOR SUPERMARKET PRODUCTS
1 | PROBLEM STATEMENT
• Standard GPS signals cannot penetrate building infrastructure, rendering them useless indoors.
• Shoppers waste significant time locating specific items in large, dynamic supermarket environments.
• Existing indoor Bluetooth (BLE) tracking suffers from severe "multi-path fading" (signal bounce), causing inaccurate user tracking.
2 | PROJECT OBJECTIVES
• To develop an indoor navigation system using BLE beacons and an ESP32 IoT edge node.
• To implement a centralized pathfinding algorithm for safe, physical supermarket environments.
• To design a cross-platform mobile application for real-time shopper assistance.
EMAIL CONTACT
3 | SYSTEM ARCHITECTURE
• Hardware: nRF51822 Beacons & ESP32 Microcontroller (Cart Node).
• Backend: XAMPP, PHP 8.x, MySQL Database.
• Frontend: Flutter SDK (Dart).
4 | KEY ENGINEERING SOLUTIONS
• Threshold-Based Weighting:
◦ A custom multiplier ("Super Gravity") applied to RSSI signals stronger than -55 dBm to mathematically eliminate distant signal bounce and anchor the cart to the nearest shelf.
• Grid-Based Manhattan Routing:
◦ Calculates strictly 90-degree orthogonal paths
• Vector Map-Matching:
◦ An algorithmic UI solution that snaps the user's fluctuating cart icon to the center of the digital aisles, providing a smooth, collision-free visual experience
5 | SYSTEM TESTING & RESULTS
BRAND
• Mean Localization Error (MLE): Custom algorithms successfully reduced tracking discrepancies from the 3.0m BLE standard down to 1.2 - 1.8 grid units.
• System Latency: Achieved a stable 1.0-second asynchronous visual update loop without throttling the local Apache server.
• Proximity Reliability: The 3.0-unit arrival threshold achieved a 90% success rate in triggering automated "Product Received" prompts during field testing.
6 | CONCLUSION
The SmartShop Navigator successfully proves that highly accurate indoor tracking does not require expensive hardware. By applying intelligent software filtering and visual map-matching to low-cost BLE hardware, the system provides a stable, real-time routing solution ready for modern smart retail environments.
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Student: Tam Yu You
Supervisor: Ayu Norafida binti Ayob