

# Development of Smart Parking Security Application Using LoRa GPS Based IoT Geo-fencing

**Azad Affandi Ghadaffi Azhar<sup>1</sup>, Jong Siat Ling<sup>1\*</sup>**

<sup>1</sup> Faculty of Electrical and Electronic Engineering

Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, MALAYSIA

\*Corresponding Author: [sljong@uthm.edu.my](mailto:sljong@uthm.edu.my)

DOI: <https://doi.org/10.30880/eeee.2025.06.02.024>

## Article Info

Received: 19 June 2025

Accepted: 27 September 2025

Available online: 30 October 2025

## Keywords

Smart Parking, Geo-fencing, IoT, GPS Tracking, LoRa Communication, ESP32, Flutter Application

## Abstract

This paper presents the development of a smart parking security application using a LoRa GPS-based IoT geo-fencing system. The system improves vehicle monitoring and security by integrating real-time location tracking with instant alert notifications. An ESP32 microcontroller, equipped with a Neo-6M GPS module and SX1278 LoRa transceiver, transmits vehicle coordinates to a Firebase Realtime Database. A Flutter-based mobile application retrieves this data, visualizes the vehicle's position on Google Maps, and triggers notifications when the vehicle exits predefined geo-fencing zones. The system was tested at two different geographic boundaries and successfully detected unauthorized vehicle movements with immediate alerts. Experimental results demonstrate the effectiveness of LoRa communication for long-distance monitoring and the practical application of IoT in enhancing parking security. The system achieved an average detection accuracy of 97.5%, highlighting its strong real-time responsiveness and boundary detection precision.

## 1. Background of Study

In Malaysia, urban expansion and the increasing number of registered vehicles exceeding 33 million nationwide have intensified challenges related to parking availability and vehicle security. Metropolitan areas such as Kuala Lumpur, Johor Bahru, and Penang experience high traffic volumes and a constant struggle for parking spaces. This shortage affects not only large corporations but also small businesses, where limited parking infrastructure is shared among employees and visitors. Consequently, competition for parking leads to inefficiencies, delays, and rising levels of frustration.

The absence of intelligent parking management systems further aggravates the situation. Conventional parking practices, which rely heavily on manual supervision, are not only inefficient but also pose security risks. Unauthorized vehicle movement, theft, and the lack of real-time monitoring undermine user confidence in the safety of their parked vehicles. Additionally, prolonged vehicle idling while searching for parking contributes to environmental pollution and wasted fuel.

To address these concerns, recent developments have turned toward smart technologies such as Internet of Things (IoT) based parking systems, car-sharing platforms, and digital parking space management. However, these solutions primarily focus on optimizing parking availability rather than addressing vehicle security and real-time monitoring. There remains a critical need for a dedicated system that emphasizes parking safety and alerts users to potential vehicle breaches.

This paper introduces a smart parking security application that integrates geo-fencing and real-time GPS tracking using an IoT-enabled architecture. The system employs an ESP32 microcontroller as the core processing unit, replacing traditional Arduino or NodeMCU platforms, and incorporates a Neo-6M GPS module and SX1278

This is an open access article under the CC BY-NC-SA 4.0 license.



LoRa transceiver. Together, these components facilitate continuous vehicle tracking and long-range communication. The accompanying mobile application, developed using the Flutter framework and Firebase backend services, provides real-time updates, secure authentication, and instant alerts when a vehicle exits its designated zone.

This solution not only enhances vehicle security in corporate environments but also contributes to the broader objective of sustainable urban development. By leveraging affordable, scalable technologies, the proposed system offers a practical approach for improving parking management through intelligent monitoring and user-focused design.

## 1.1 Introduction

Despite advancements in parking management systems, many company parking facilities particularly in urban areas continue to face serious challenges in maintaining vehicle security and operational efficiency. Traditional parking setups typically rely on manual supervision, which is insufficient for addressing modern day concerns such as unauthorized vehicle movement and theft. These systems lack real-time monitoring and do not offer proactive responses to suspicious activities, leaving vehicles vulnerable and increasing anxiety among users.

Furthermore, the absence of automated tracking mechanisms and instant alerts limits the ability of existing systems to inform users when their vehicles exit defined boundaries. This delay in notification significantly reduces the chances of early intervention and recovery in case of vehicle loss. While some solutions have attempted to integrate GPS tracking, their implementation often falls short in delivering reliable, environment-agnostic performance especially in real-world conditions with signal limitations or varying coverage.

Moreover, the effectiveness of combining real-time GPS data with instant mobile notifications for improving parking security has not been extensively validated in corporate or campus environments. As such, there is a need for a robust, low-cost, and reliable smart parking solution that bridges this gap by enabling real-time vehicle location monitoring, intelligent geo-fencing detection, and prompt notification delivery to users.

The objectives of this work are threefold. First, to develop a LoRa GPS-based IoT geo-fencing system. Second, to implement a notification function that sends instant alerts when a vehicle exits a defined zone. Finally, to analyze the effectiveness of the proposed system in detecting unauthorized vehicle movements

## 1.2 Literature Review

Recent studies have explored smart parking and vehicle monitoring systems using various technologies such as IoT, GPS, LoRa, and RFID. Jaafar et al. [1] introduced an IoT-based smart parking system that effectively managed vehicle entry and exit but was limited to short communication ranges of less than 100 meters, making it unsuitable for large-scale outdoor areas. Similarly, Sinha et al. [2] proposed a GPS-based geo-fencing system that defined virtual boundaries for vehicle tracking. However, their approach required additional tools for real-time alerting, which reduced practicality for immediate security applications.

Other researchers have investigated long-range communication methods. Reddy et al. [3] highlighted LoRa communication as a promising technology for IoT applications due to its extended range and low power consumption, although performance could be affected in dense urban environments. In contrast, Sadaat and Shoukat [4] developed an RFID-based system for parking management, but the limited detection range of 1–5 meters made it unsuitable for outdoor security purposes.

More recent works have demonstrated the potential of combining IoT and mobile applications for vehicle monitoring. Li and Zhang [5] presented an IoT-based vehicle tracking system that offered real-time monitoring but relied heavily on constant internet connectivity, which could limit deployment in areas with unstable coverage. Zhu et al. [6] surveyed smart parking technologies and concluded that most existing systems focused on availability and payment convenience rather than vehicle security. Similarly, Parkopedia [7] and ParkMobile [8] have developed commercial platforms for digital parking management, but these emphasize booking and payment rather than theft prevention or boundary-based monitoring.

These findings highlight a research gap in the integration of long-range communication with geo-fencing and real-time notifications for parking security. Unlike previous studies, the proposed system combines an ESP32 with a Neo-6M GPS module and SX1278 LoRa transceiver, together with a Flutter-based mobile application, to deliver immediate alerts when vehicles breach designated zones. This approach directly addresses the limitations of prior works by prioritizing security, scalability, and independence from constant internet access. Table 1 summarizes previous works on parking system security methods.

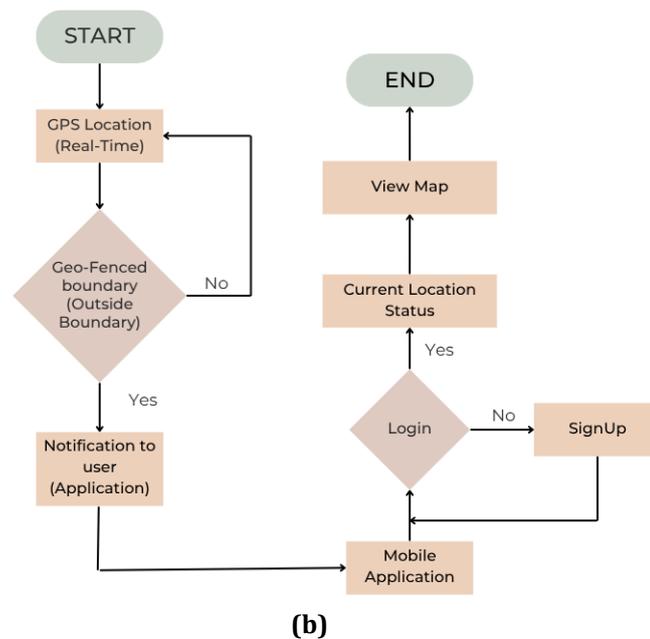
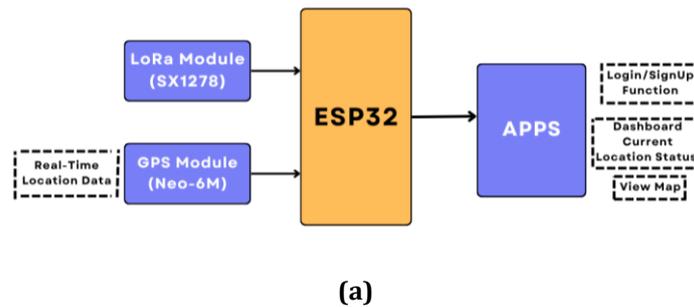
**Table 1** Comparison of Security Methods in Parking Systems

Author	Method	Range Covered	Weaknesses
Jaafar et al. [1]	IoT-based Smart System	50–100 m	Limited range; small parking lots only
Sinha et al. [2]	Geo-fencing + GPS	5–10 km	No real-time alerts; needs external tools
Reddy et al. [3]	LoRa Communication	Up to 15 km	Poor in dense urban areas
Sadaat & Shoukat [4]	RFID	1–5 m	Not suitable for long-range or outdoor use

## 2. Methodology

### 2.1 System Architecture

The Smart Parking Security System combines embedded hardware, wireless communication, and a mobile application to monitor vehicle location in real time and detect geo-fence violations. At its core, the ESP32 microcontroller acts as the central processing unit, interfacing with a Neo-6M GPS module to obtain latitude and longitude data and an SX1278 LoRa module for low-power, long-range communication with the Firebase cloud infrastructure, as illustrated in Fig. 1.



**Fig. 1** System Architecture (a) Block Diagram; (b) Flowchart

The system continuously tracks the vehicle’s position and determines whether it remains within the predefined geo-fencing boundary. When a boundary breach is detected, the ESP32 immediately triggers the LoRa module to transmit an alert to Firebase. A mobile application, developed using Flutter, retrieves this data, displays the vehicle’s position on Google Maps, and provides secure access through Firebase Authentication. The app also generates dual-mode notifications standard push alerts and call-style full-screen alerts to promptly inform users when a vehicle exits the designated zone.

The system begins operation once the ESP32 is powered, automatically initiating GPS tracking and maintaining real-time monitoring. It operates in a passive state while the vehicle remains within the defined boundary and activates an alert sequence when a geo-fence violation occurs.

## 2.2 Hardware and Software Development

The geo-fencing hardware circuit was designed to collect and transmit real-time vehicle location data using an ESP32 microcontroller, a Neo-6M GPS module, and an SX1278 LoRa transceiver. The ESP32 served as the main controller, processing GPS coordinates obtained from the Neo-6M module and transmitting them through the LoRa module. Operating in sub-GHz frequency bands, the LoRa transceiver enabled reliable long-range, low-power communication with the Firebase cloud, making the system suitable for outdoor deployment. The modular setup, built on a breadboard, provided flexibility for prototyping and supported future upgrades such as sensor integration and improved communication security. The complete circuit forms the foundation of the system's geo-fencing logic and alert mechanism, as shown in Fig. 2.

The software development integrated embedded firmware, a cross-platform mobile application, and a cloud-based backend. The ESP32 was programmed using the Arduino IDE to acquire GPS coordinates, process location validity, and transmit alerts when boundary violations occurred. A mobile application, developed with Flutter, provided real-time vehicle tracking, secure authentication, and notification functions, including both standard push alerts and call-style full-screen alerts. Firebase supported the backend by offering authentication services, real-time database synchronization, and messaging functions to ensure seamless data flow between the hardware and the mobile interface.

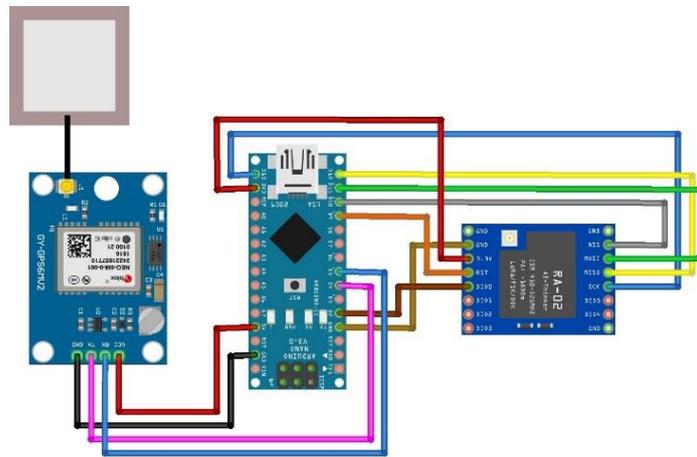


Fig. 2 LoRa GPS Geo-fencing Circuit

## 2.3 Geo-fencing Logic

The geo-fencing logic forms the core mechanism for boundary detection in the Smart Parking Security System. A virtual perimeter is defined using predefined GPS coordinates, which represent the key vertices of each monitored zone. These coordinates are embedded into the system firmware to enable real-time comparison with live GPS data from the Neo-6M module.

During operation, the ESP32 continuously monitors the vehicle's position and checks whether it lies within the defined boundary. If the vehicle remains inside, the system operates in normal tracking mode. When the GPS reading falls outside the designated area, the ESP32 immediately triggers an alert through the LoRa module to Firebase, which then prompts the mobile application to issue a notification to the user.

## 2.4 Application User Interface (UI) and Notification

The mobile application, developed using the Flutter framework, played a critical role in delivering a secure and user-friendly interface for real-time vehicle monitoring and notification. The application was evaluated across multiple functionality modules, including authentication, live data visualization, and alert delivery, all integrated with Firebase services for backend support.

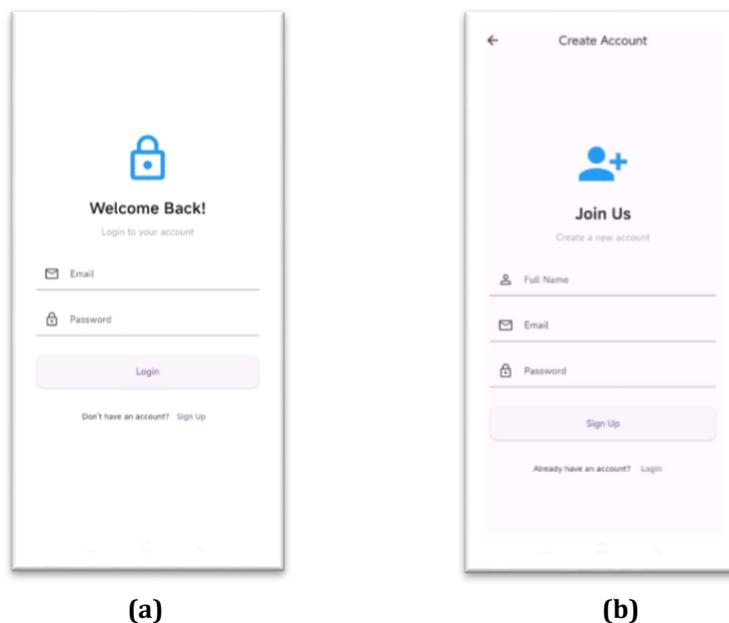
The application's login mechanism was implemented using Firebase Authentication, enabling secure verification of user credentials. During testing, the login system provided stable performance with swift redirection to the dashboard upon successful authentication. Invalid inputs generated appropriate error messages, improving usability and reducing the likelihood of user confusion. All communication between the application and Firebase was securely encrypted, ensuring data integrity, as illustrated in Fig. 3.

The sign-up feature offered a streamlined registration process requiring only a valid email address and password. New users were able to access their dashboards immediately after registration, without delays or secondary verification steps. Firebase's backend ensured real-time data synchronization, and no service interruptions or data inconsistencies were observed throughout the testing phase, as shown in Fig. 3.

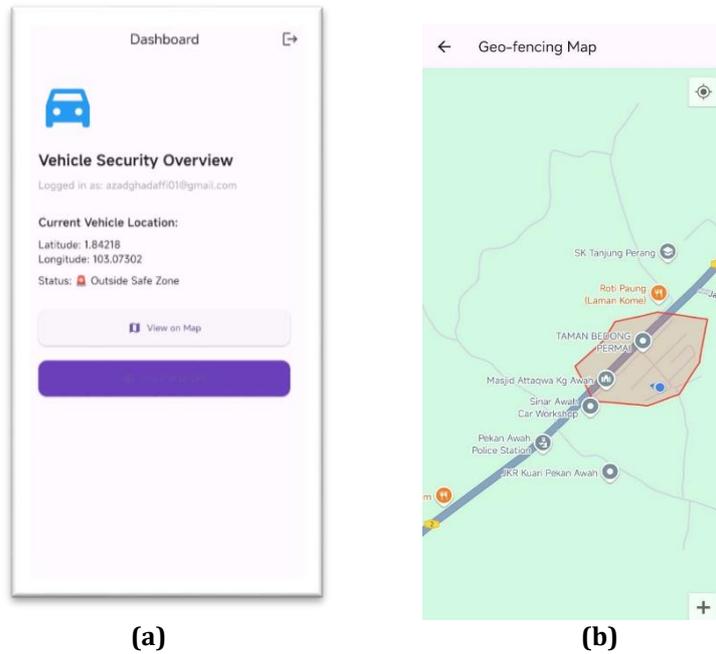
The dashboard served as the central control interface, displaying up-to-date GPS coordinates and the current geo-fencing status of the monitored vehicle. Data was fetched directly from Firebase Realtime Database, and the interface successfully reflected updates with minimal latency. The dashboard also featured interactive elements such as status refresh and direct navigation to the live map, enhancing overall usability. Testing across different internet speeds confirmed that the dashboard consistently maintained near real-time data synchronization, as illustrated in Fig. 4.

The application's map view, powered by the Google Maps API, accurately displayed the vehicle's location with high precision. The interface overlaid geo-fencing boundaries on the map, allowing users to visualize the vehicle's position relative to the defined safe zones. Cross-validation between hardware-based and app-based geo-fencing logic provided redundancy and increased system reliability. During movement tests and under variable GPS signal strength, the map continued to update smoothly, confirming its robustness and responsiveness, as shown in Fig. 4.

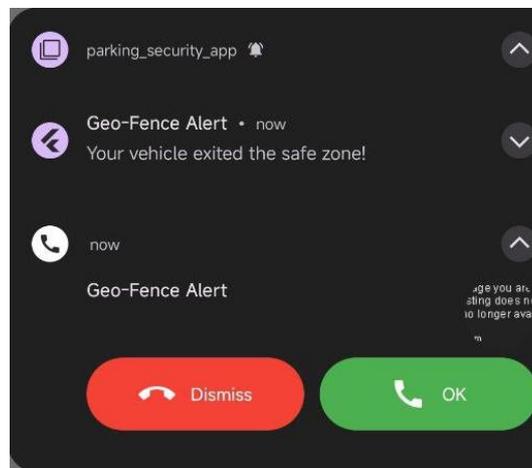
One of the most critical features of the mobile application was its dual-mode notification system, which ensured timely alerts in response to geo-fence violations. Standard local push notifications were implemented using the *flutter local notifications* package, enabling reliable alerts while the app was running in either the foreground or background, as illustrated in Fig. 5.



**Fig. 3** UI Interface (a) login; (b) sign-up



**Fig. 4** UI Interface (a) Dashboard; (b) Map View



**Fig. 5** Notification

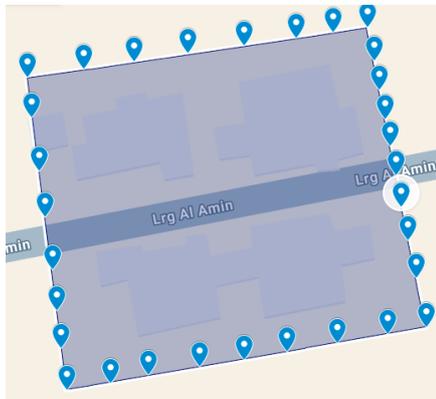
## 2.5 Analysis of Effectiveness of the System

To evaluate the practical effectiveness of the proposed geo-fencing system, extensive testing was conducted using two distinct boundary configurations, each representing different deployment scenarios. The boundaries were visually defined using the Google Maps API and integrated into the mobile application, forming the virtual perimeters used for real-time monitoring. Fig. 6 shows the first defined geo-fencing boundary used for static testing, while Fig. 7 illustrates the second boundary applied during dynamic testing.

In both scenarios, a vehicle equipped with the ESP32, Neo-6M GPS, and SX1278 LoRa modules was moved systematically inside and outside the defined geo-fenced areas. Real-time GPS coordinates were continuously transmitted to the Firebase Realtime Database, from which the mobile application retrieved and processed the data for visual and logical evaluation.

In the first test, a static coordinate-based method was adopted. A total of 20 GPS locations were manually selected: 10 within the defined geo-fence, and 10 outside. The vehicle was positioned at each point sequentially. For all 10 in-bound coordinates, the system correctly identified the vehicle as inside the safe zone, with no notifications triggered. Conversely, for all 10 out-of-bound points, the system immediately detected the breach and sent alerts via both popup notifications and call-style full-screen alerts. The system achieved 100% detection accuracy for this controlled test, demonstrating precise execution of the geo-fencing algorithm under static conditions and validating the correctness of the boundary logic and notification triggers.

The second test involved continuous dynamic testing to simulate real-world vehicle movement. The vehicle was driven along a route that repeatedly entered and exited the geo-fenced area. During this process, 20 GPS positions were recorded, 10 from inside and 10 from outside the boundary, selected during moments of low-speed travel or when the vehicle was stationary to ensure data accuracy. In this scenario, the system maintained strong performance, correctly identifying 19 out of 20 positions, resulting in a 95% detection accuracy. The one misclassification occurred due to a minor delay in GPS signal stabilization, which briefly delayed alert generation. Despite this, notification latency remained within acceptable bounds (typically within a few seconds), and the system quickly resumed normal operation. Table 2 summarizes the performance assessment.



**Fig. 6** First Geo-fencing Boundary



**Fig. 7** Second Geo-fencing Boundary

**Table 2** Summary and Performance Assessment

Test Type	In-Bound Accuracy	Out-of-Bound Accuracy	Overall Detection Accuracy
Static Boundary	10/10	10/10	100%
Dynamic Boundary	9/10	10/10	95%

### 3. Conclusion

This project successfully developed and validated a complete IoT-based geo-fencing vehicle monitoring system utilizing an ESP32 microcontroller, Neo-6M GPS, and SX1278 LoRa module for low-power, long-range tracking, integrated with a Flutter-based mobile application and Firebase cloud services. The system reliably captured and transmitted real-time GPS data, visualized it through an interactive mobile interface, and triggered dual mode notifications popup and full-screen call-style alerts upon geo-fence breaches. Through static and dynamic testing involving 40 total GPS coordinates, the system achieved an average detection accuracy of 97.5%, demonstrating strong real-time responsiveness and boundary detection precision. The implementation highlights the potential of combining open-source tools and affordable components to address security needs in parking and vehicle tracking scenarios. Moreover, the project provided valuable hands-on experience in embedded systems, mobile app development, and real-time cloud integration, laying the groundwork for future enhancements such as multi-device support, historical route logging, and emergency contact features.

### 4. Recommendation

To enhance the performance and expand the capabilities of the proposed geo-fencing system, several improvements are recommended for future development. First, integrating SMS and email alert services alongside the existing mobile notification system would increase user awareness in scenarios where mobile data or app access is unavailable. Second, implementing a route history logging feature would enable users to review previous movements, which is useful for audit trails, fleet management, and investigation purposes. Third, expanding the system to support multiple devices and vehicles simultaneously would make it scalable for commercial applications such as delivery tracking or campus security. Additionally, integrating automatic emergency contact alerts upon boundary violations could significantly enhance safety, especially in theft or unauthorized use scenarios. Finally, optimizing the system for low-power sleep cycles and solar-powered operation could increase deployment feasibility in remote or off-grid environments, aligning with sustainable smart city initiatives.

## Acknowledgement

The authors would like to express the deepest gratitude to the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, for the logistical support.

## Conflict of Interest

The author declares that there is no conflict of interest regarding the publication of this article.

## Author Contribution

*Azad Affandi Ghadaffi Azhar is the original author and developer of this project. He was solely responsible for the complete system design and implementation, including hardware integration, GPS data collection, geo-fencing configuration, mobile application development, Firebase integration, and system testing. He also authored and compiled the entire manuscript. Jong Siat Ling served as the project supervisor, providing continuous technical guidance and support throughout the development process. She contributed by reviewing relevant literature, refining the project scope, and offering critical feedback to strengthen the system architecture and research direction.*

## References

- [1] J. A. Abdulsahab, R. M. Nafea, W. A. M. Al-Jawher, and M. L. Hayyawi, "IoT based smart parking system," J. Port Sci. Res., vol. 7, no. 3, pp. 196–203, 2024.
- [2] A. Sinha, A. Singh, and A. Baghel, "Smart GPS geo-fencing system: A review," Int. J. Eng. Appl. Sci. Technol., vol. 7, pp. 166–171, 2023.
- [3] B. S. Reddy, H. Gupta, R. Jha, and S. S. Hashmi, "LoRa interference issues and solution approaches in dense IoT networks: A review," Telecommun. Syst., vol. 87, pp. 517–539, 2024.
- [4] H. Sadaat and O. Shoukat, "RFID-based smart parking," unpublished, 2022.
- [5] Y. Li and M. Zhang, "IoT-based vehicle tracking and monitoring system," Int. J. Comput. Sci. Inf. Secur., vol. 17, no. 5, pp. 45–50, 2019.
- [6] L. Zhu, X. Zhang, and X. Li, "Smart parking systems: A survey," J. Traffic Transp. Eng., vol. 5, no. 3, pp. 154–160, 2018.
- [7] Parkopedia, "Smart Parking Solutions," Parkopedia, 2023. [Online]. Available: <https://www.parkopedia.com>
- [8] ParkMobile, "Find Parking with ParkMobile," ParkMobile, 2023. [Online]. Available: <https://www.parkmobile.io>