IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, April 2026
Impact Factor: 8.2

Software Design and Implementation of a Secure RC Delivery Robot with Live Monitoring using ESP32-CAM and LoRa Communication

Shivani Unde[1], Prof. Dr. A. U. Vikhe[2], Samiya Shaikh[3], Anupama Kahandal[4]
Asst. Prof., Department of Electronics &Telecommunication Engineering[1]
Students, Department of Electronics &Telecommunication Engineering[2,3,4]
Dr.Vithalrao Vikhe Patil College of Engineering, Ahilyanagar, Maharashtra,
Savitribai Phule Pune University, Pune

Abstract: *The rapid rise of e-commerce and contactless lo- gistics has created a demand for secure and intelligent de- livery systems. This paper presents the software design and implementation of an RC Delivery Robot equipped with theft detection and live monitoring features using ESP32-CAM and LoRa communication. The proposed software framework focuses on real-time video streaming, remote communication, event-based alert generation, password-based access control, and obstacle- aware system response.*

The system software is divided into two coordinated modules: the robot-side control unit and the remote monitoring unit. The robot-side software manages video transmission, sensor data acquisition, keypad authentication, theft detection, and actuator control. The remote-side software receives delivery status, battery alerts, and tamper notifications through LoRa communication and presents them to the operator. Event-driven logic is used to ensure timely response to security and safety conditions such as vibration detection, low battery, obstacle presence, and invalid password attempts.

The developed software system includes both embedded robotic control and a web-based application interface for user interaction, order management, and administrative supervision. The developed system demonstrates reliable real-time mon- itoring and secure delivery operation in a compact embedded environment. Experimental testing confirmed that the developed software modules performed reliably under delivery-related op- erating conditions. The software model is suitable for smart delivery, campus logistics, and low-cost robotic surveillance applications.

Keywords: RC delivery robot, ESP32-CAM, LoRa, live monitoring, theft detection, embedded software, secure delivery, IoT.

I. INTRODUCTION

The demand for automated delivery solutions has increased significantly due to the expansion of food delivery, courier services, and online retail systems [1]. Conventional delivery methods depend heavily on human intervention, which can lead to delayed service, package theft, limited tracking, and higher operational cost [2]. Robotic delivery platforms have emerged as a practical alternative for improving delivery efficiency and security [3].

In many delivery applications, the software layer is one of the most critical parts of the system because it coordinates sensing, communication, monitoring, and security decisions [4]. A delivery robot must not only move from one point to another, but also continuously monitor its surroundings, communicate alerts, and allow secure parcel access. These requirements make embedded software integration essential for safe and reliable operation [5].

This work focuses on the software implementation of an RC Delivery Robot with Theft Detection and Live Monitoring using ESP32-CAM and LoRa communication. The proposed software enables real-time visual surveillance

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, April 2026 Impact Factor: 8.2

using ESP32- CAM [6], long-range status communication through LoRa [7], password-protected parcel access, and automatic response to abnormal events.

In addition to embedded robotic control, modern delivery systems also require user-facing digital interfaces for order placement, authentication, and delivery management. There- fore, this work also includes the development of a full-stack web application that complements the robotic platform by enabling user interaction, order processing, and administrative control through a software-based online interface.

The primary contribution of this paper is the development of an integrated software workflow that combines monitoring, security, and communication functions into a single embedded robotic platform.

II. LITERATURE SURVEY

Recent developments in robotic delivery and IoT-based surveillance systems have shown strong potential for secure last-mile logistics [4]. Several systems have been designed for remote-controlled or semi-autonomous delivery, but many focus only on navigation or payload transport. Security, monitoring, and access control are often treated as secondary features [5].

IoT-enabled monitoring systems using camera modules have demonstrated that real-time visual observation can improve situational awareness in robotic applications [6]. ESP32-CAM- based implementations are especially attractive because they provide low-cost image capture and Wi-Fi-based streaming in compact form [1]. However, many existing systems depend only on Wi-Fi and therefore suffer from limited communication range in outdoor or semi-urban conditions.

Long-range wireless communication technologies such as LoRa have become increasingly important in embedded monitoring applications [2]. LoRa offers low-power data exchange over long distances, making it suitable for alert transmission and status communication in delivery systems [7]. Although LoRa has been widely used in sensor networks and remote monitoring, its integration with live robotic surveillance and security-triggered communication remains limited.

Security features such as keypad authentication, tamper detection, and event-based alarms have been studied independently in locker systems and access-control devices [8]. However, fewer works combine these features into a mobile robotic delivery platform with real-time monitoring capability. Unlike many existing delivery or surveillance systems that focus only on communication or navigation, the proposed work combines embedded event handling, real-time monitor- ing, secure parcel access, and a user-facing web application into a single integrated software platform.

This paper addresses that gap by presenting a software framework that integrates video streaming, long-range alert transmission, password verification, theft detection, and event- driven control in a single robotic delivery system.

III. PROPOSED SOFTWARE ARCHITECTURE

A. Overview

The proposed smart delivery system is designed using a multi-layer software architecture that combines embedded robotic control, long-range communication, live monitoring, and a full-stack web application. The objective of this software architecture is to provide secure, observable, and manageable delivery operations through the integration of both hardware- level and application-level software components.

The proposed software framework is divided into four major functional layers:

- Robot-side embedded software
- Remote monitoring and communication software
- User-facing web application
- Administrative management interface

These software layers operate together to support end- to-end delivery workflow, including live monitoring, theft detection, user authentication, order placement, order tracking, and operational supervision.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, April 2026 Impact Factor: 8.2

B. Robot-Side Embedded Software

The robot-side embedded software is the core control layer of the delivery robot. It executes on the microcontroller- based hardware integrated into the robotic platform and is responsible for local sensing, event detection, communication, and hardware actuation.

At startup, the robot-side software initializes all connected modules including the ESP32-CAM, LoRa communication unit, keypad, vibration sensor, ultrasonic sensor, battery mon- itoring circuit, buzzer, warning indicators, and lock actuator. Once initialized, the software continuously monitors sensor inputs and user interactions while simultaneously managing event-based outputs.

The primary responsibilities of the robot-side software in- clude:

- Initializing sensors and communication modules
- Starting live camera streaming
- Monitoring keypad input for parcel authentication
- Detecting vibration and obstacle conditions
- Checking battery status
- Controlling buzzer, warning lights, and lock mechanism
- Sending event-based alert messages to the remote moni- toring unit

This layer acts as the intelligent operational engine of the robotic platform.

C. Remote Monitoring and Communication Software

The remote monitoring software acts as the communication and supervision endpoint of the robotic system. It receives status and alert information transmitted from the robot-side software through LoRa communication.

Its major functions include:

- Receiving LoRa packets from the robot
- Decoding system and alert messages
- Displaying operational and warning information
- Assisting the operator in remote supervision

This layer enables the operator to remain informed about theft events, low battery conditions, obstacle detection, pass- word validation, and delivery-related events without requiring direct physical access to the robot.

D. Web Application Integration

In addition to the embedded robotic software, the proposed system also incorporates a web-based software platform to enhance user interaction, delivery management, and adminis- trative control. The developed web application follows a full- stack architecture and was implemented using ReactJS for the frontend interface, NodeJS and ExpressJS for backend service handling, and MongoDB for database management.

The web application enables browsing of items, user regis- tration and login, cart management, order placement, payment handling, and order tracking. It also includes an admin panel for order supervision and status updates.

Fig.1 illustrates the operational workflow of the developed web application module. The process begins with the user interacting with the frontend interface developed using Re- actJS. The user browses available items, adds products to the cart, and proceeds through authentication using a secure login/signup process. Once authenticated, the order is placed and redirected to the payment gateway.

After successful payment, the order details are stored in the MongoDB database and become available to the admin panel for status updates. In the case of failed or cancelled payment, the order flow is terminated accordingly. The updated order information is then reflected in the user-side order tracking interface. This software workflow improves the practicality and completeness of the proposed smart delivery platform.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, April 2026
Impact Factor: 8.2

Fig. 1. Software Workflow of the Developed Web Application Module.

E. Software Operating Principle

The complete software system follows an event-driven and modular operating model. At the robot level, event-based logic is used to trigger theft alerts, authentication checks, battery warnings, and obstacle responses. At the application level, user requests such as login, order placement, and order tracking are processed through backend APIs and database transactions.

This layered architecture improves software modularity, maintainability, and future scalability while enabling the system to operate as a secure and intelligent smart delivery platform.

Fig.2 illustrates the overall software working flow of the proposed smart delivery robot system. The software operation begins with the initialization of all hardware and communication modules, including the ESP32-CAM, LoRa transceiver, vibration sensor, ultrasonic sensor, keypad, buzzer, lock actuator, and battery monitoring unit. After successful initialization, the ESP32-CAM establishes a Wi-Fi connection and starts live video streaming for real-time visual monitoring.

Once the system is ready, the software enters a continuous event-monitoring loop. In this loop, the robot repeatedly checks for theft-related vibration, nearby obstacles, battery status, and keypad-based parcel access requests. When abnormal events such as vibration, low battery, or obstacle detection occur, the system generates appropriate local actions and transmits alert messages through LoRa communication to the remote monitoring unit.

For parcel security, keypad-entered passwords are verified against stored credentials. If the entered password is valid, the parcel compartment is unlocked temporarily and automatically re-locked after a fixed access duration. In the case of repeated invalid attempts, a security alert is generated. The software also sends periodic status updates during

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, April 2026
Impact Factor: 8.2

operation. This event-driven control flow ensures reliable monitoring, secure access, and efficient software coordination within the proposed delivery robot platform.

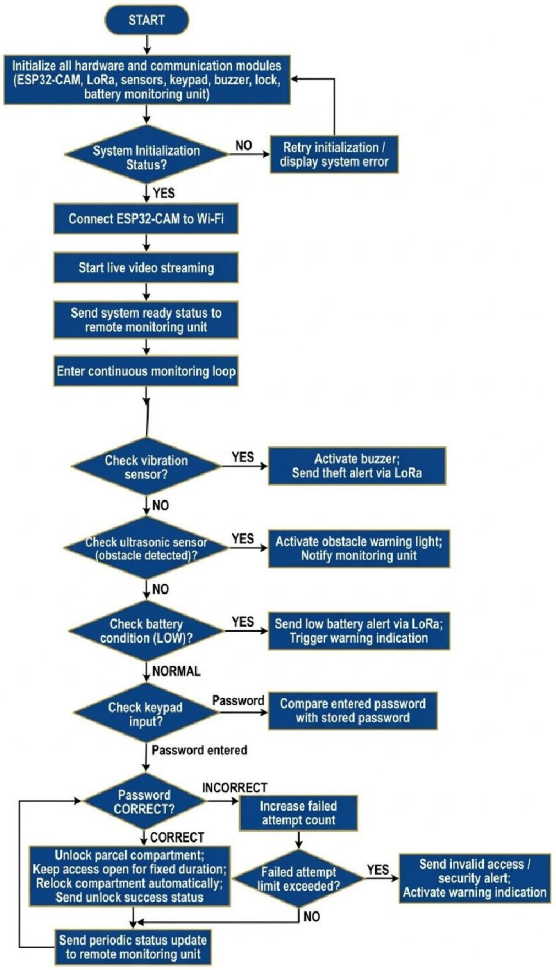

Fig. 2. Overall Software Working Flow of the Proposed RC Delivery Robot System.

IV. SOFTWARE MODULES

A. ESP32-CAM Live Monitoring Module

The ESP32-CAM live monitoring module is responsible for capturing and transmitting real-time video frames over a Wi-Fi network. This module enables the operator to visually inspect the surroundings of the robot during movement, parcel transport, and delivery operation. The integration of live visual feedback significantly improves situational awareness and sup- ports remote observation of environmental and security-related events.

The major functions of this module include:

- Camera initialization
- Wi-Fi connection establishment
- Local IP-based video server creation
- Continuous video frame streaming
- Frame refresh and session maintenance

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, April 2026 Impact Factor: 8.2

At system startup, the ESP32-CAM initializes its onboard camera and connects to the configured Wi-Fi network. Once connected, a local streaming server is created and the live feed becomes accessible through the generated IP address.

B. LoRa Communication Module

The LoRa communication module is responsible for trans- mitting low-bandwidth but critical system information between the robot-side unit and the monitoring unit over a long communication range.

The types of data transmitted through LoRa include:

- Delivery status
- Theft alerts
- Low-battery warnings
- Invalid password alerts
- Obstacle warnings
- System-ready notifications

The communication module performs initialization of the LoRa transceiver, encodes outgoing messages, transmits com- pact event packets, and supports communication reliability.

Typical event messages include:

- THIEF_ALERT
- LOW_BATTERY
- OBSTACLE_FOUND
- UNLOCK_SUCCESS
- WRONG_PASS

C. Theft Detection Module

The theft detection module is designed to identify unautho- rized movement, tampering, or suspicious physical interaction with the robot or parcel compartment using a vibration sensor as the primary trigger source.

The software continuously reads vibration input, filters unwanted noise where required, and triggers a security event when valid tampering is detected. On detection, the buzzer is activated and an alert is transmitted through LoRa to the monitoring side.

D. Password Authentication Module

The password authentication module is responsible for ensuring that only an authorized user can unlock the parcel compartment.

This module includes:

- Keypad input handling
- Password matching logic
- Unlock timing control
- Invalid attempt counting

If the entered password is correct, the solenoid lock is activated for a limited time and the compartment is unlocked. If the password is incorrect, the failed-attempt counter is in- cremented and a security alert may be generated after repeated invalid entries.

E. Obstacle Detection Response Module

The obstacle detection module processes ultrasonic sensor measurements to determine whether any object is present within a predefined safety range of the robot.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, April 2026 Impact Factor: 8.2

The software triggers an ultrasonic pulse, calculates distance using echo time, compares the value with a threshold, and activates a warning output when an obstacle is detected. This improves remote driving awareness and reduces the likelihood of accidental collision.

F. Battery Monitoring and Alert Module

The battery monitoring module continuously checks the battery condition to ensure uninterrupted delivery operation.

Its major tasks include:

- Reading battery-related sensor values
- Comparing readings with predefined threshold values
- Identifying low-battery condition
- Transmitting warning messages through LoRa

This module helps prevent unexpected shutdown and im- proves system reliability during delivery.

G. Web Application Module

The web application module was developed to provide a complete digital interface for user interaction, order manage- ment, and administrative supervision of the proposed delivery system.

1) User-Side Functionalities: The user-side interface sup- ports:

- User registration and login
- Browsing of available items
- Category-based product filtering
- Cart management
- Order placement
- Delivery address entry
- Payment handling
- Order tracking

The frontend interface was developed using ReactJS and follows a responsive component-based design suitable for both desktop and mobile use.

2) Authentication and Security: The application includes secure authentication using JSON Web Token (JWT)-based session handling. Password data is securely managed at the backend and tokens are used to maintain authorized user sessions.

3) Cart and Order Management: The cart management logic allows users to add, remove, and update item quantities before confirming an order. Once placed, order data is stored in the backend database and made available for tracking and status updates.

4) Admin Panel Functionality: A separate administrative interface was also developed. It supports:

- Adding new items
- Uploading item images
- Removing listed items
- Viewing customer orders
- Updating order status

5)Payment Integration: To support practical deployment, the web application also integrates a payment gateway work- flow. This enables secure online transaction handling and improves the real-world applicability of the delivery platform.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, April 2026
Impact Factor: 8.2

V. SOFTWARE IMPLEMENTATION TOOLS

A. *Arduino IDE*

The embedded robot-side software was developed and tested using Arduino IDE. This environment was selected because of its simplicity, hardware compatibility, and extensive library support for ESP32 and sensor-based embedded applications.

The following tasks were implemented using Arduino IDE:

- Sensor monitoring logic
- Keypad verification logic
- LoRa communication routines
- Output control for buzzer, LEDs, and lock actuator
- Event-based debugging and testing

B. *Embedded C / Arduino Programming*

The robot-side embedded software was developed using Embedded C with Arduino-style syntax. The code was organized into modular functions for initialization, monitoring, communication, event handling, and actuator control.

The programming approach followed:

- Modular function-based design
- Continuous loop-based execution
- Condition-driven event detection
- Compact packet encoding for LoRa communication

C. *Frontend and Backend Web Development Tools*

In addition to embedded programming, the proposed system also involved the implementation of a full-stack web application.

1) Frontend Development: The frontend was developed using ReactJS. It was used to create the user interface and administrative dashboard of the software system.

The frontend implementation included:

- Responsive user interface design
- Routing and navigation
- Cart and order pages
- Authentication forms
- Order tracking interface
- Admin dashboard pages

Additional frontend tools used include:

- **React Router DOM** for routing
- **Axios** for API communication
- **React Toastify** for notifications
- **Context API** for global state management

2) Backend Development: The backend was developed using NodeJS and ExpressJS. It was responsible for API development, request handling, authentication, order processing, image upload handling, and database interaction.

Backend implementation included:

- User registration and login APIs
- Cart management APIs
- Item management APIs

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, April 2026 Impact Factor: 8.2

- Order placement and tracking APIs
- Admin order update functionality

3) Database and Supporting Technologies: MongoDB was used as the primary database for storing user records, item details, cart information, and order history. Mongoose was used for schema modeling and data interaction.

Supporting technologies used include:

- **JWT** for authentication token handling
- **bcrypt** for password hashing
- **multer** for image upload handling
- **Stripe** for payment gateway integration

These tools collectively enabled the development of a complete software ecosystem for the proposed delivery platform.

VI. RESULTS AND DISCUSSION

The developed software system was tested under multiple delivery-related operating scenarios to evaluate its monitoring, communication, authentication, and management performance. The system demonstrated stable execution across both the embedded robotic modules and the full-stack web application. The ESP32-CAM live monitoring module successfully established a real-time video stream, enabling the operator to observe the surroundings of the robot during movement and delivery operation. The live feed remained functional within the available Wi-Fi coverage area and supported visual
supervision of the delivery path.

The LoRa communication module reliably transmitted event-based alert and status messages between the robot-side unit and the monitoring unit. Messages related to theft detection, low battery condition, invalid password attempts, and obstacle detection were successfully transmitted and received during testing.

The theft detection module responded correctly to vibration- based tampering events by activating the buzzer and generating an alert. Similarly, the password authentication module correctly handled both valid and invalid access attempts, thereby ensuring controlled access to the parcel compartment.

The obstacle detection module also functioned correctly by identifying nearby objects and activating a warning indication. The battery monitoring module successfully detected low- power conditions and generated timely warning messages.

In addition to the embedded software, the developed web application was also tested successfully under both user-side and admin-side operational scenarios. The user interface sup- ported responsive browsing, authentication, cart interaction, order placement, and order tracking without major functional issues.

The backend APIs correctly handled user registration, login verification, item retrieval, cart synchronization, and order storage. The administrative panel also functioned effectively by supporting item addition, item deletion, and order status updates. Payment integration workflow was also tested as part of the order placement process.

Overall, the results indicate that the proposed software architecture is effective in integrating embedded robotic control, real-time monitoring, long-range communication, user inter- action, and administrative management into a unified smart delivery platform.

VII. CONCLUSION

This paper presented the software design and implementation of an RC Delivery Robot with Theft Detection and Live Monitoring using ESP32-CAM and LoRa communication, integrated with a full-stack web application for digital delivery workflow management. The developed software framework successfully combines embedded robotic control, live video monitoring, long-range alert communication, password-based parcel access, theft detection, obstacle warning, battery monitoring, user interaction, order management, and administrative supervision.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, April 2026 Impact Factor: 8.2

The proposed software architecture demonstrates that a low-cost embedded robotic platform can be effectively extended through modern web technologies to support secure, observable, and manageable delivery services. The system is suitable for educational, prototype, and small-scale smart logistics applications and provides a strong foundation for future intelligent robotic delivery solutions.

VIII. FUTURE SCOPE

The proposed system can be further enhanced through the following improvements:

- Mobile application-based monitoring and control
- GPS-based real-time robot location tracking
- Cloud-based storage of alerts and delivery history
- Face recognition-based parcel authentication
- Autonomous path planning and navigation
- AI-based obstacle classification and avoidance
- Encrypted communication for higher security
- Integration with smart city and IoT logistics infrastructure

IX. ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Prof. Dr. A.U. Vikhe for his valuable guidance, continuous encourage- ment, and insightful suggestions throughout the course of this project. His support played a crucial role in shaping the quality and direction of this work. We also thank the faculty members of the Department of Electronics and Telecommunication Engineering, Padmashri Dr. Vitthalrao Vikhe Patil College of Engineering, Ahilyanagar, for providing the necessary re- sources and academic support.

REFERENCES

[1] Espressif Systems, “ESP32-CAM Technical Documentation,” Espressif Systems, 2023.
[2] Semtech Corporation, “LoRa Technology Overview,” Semtech Wireless and Sensing Products, 2022.
[3] Arduino, “Arduino IDE Documentation,” Arduino Official Documenta- tion, 2023.
[4] A. Author, B. Author, and C. Author, “IoT-based robotic monitoring system,” International Journal of Embedded Systems, vol. 10, no. 2, pp. 45–52, 2023.
[5] X. Author and Y. Author, “Secure delivery robot using wireless communication,” Journal of Smart Automation, vol. 8, no. 1, pp. 12–18, 2022.
[6] M. Author and N. Author, “Camera-based IoT surveillance system for real-time monitoring,” International Journal of Smart Systems, vol. 6, no. 3, pp. 77–84, 2021.
[7] P. Author and Q. Author, “LoRa-enabled embedded monitoring for long- range applications,” Journal of Wireless Embedded Communication, vol. 9, no. 4, pp. 91–99, 2022.
[8] R. Author and S. Author, “Smart access control and tamper detection system using embedded sensors,” International Journal of Security Electronics, vol. 5, no. 1, pp. 15–22, 2020.
[9] Arduino, “Arduino IDE Documentation,” Arduino, 2025. [Online]. Available: https://docs.arduino.cc/software/ide/
[10] M. A. Rahman, S. H. M. Ali, and T. Hasan, “IoT-Based Smart Moni- toring and Surveillance Systems: A Review,” Sensors, vol. 22, no. 14, pp. 1–24, 2022.
[11] A. Koubaa, B. Qureshi, M. Sriti, Y. Javed, and E. Tovar, “A Wireless Sensor Network for Real-Time Indoor Surveillance and Monitoring Applications,” International Journal of Distributed Sensor Networks, vol. 10, no. 5, 2014.
[12] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low Power Wide Area Networks: An Overview,” IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 855–873, 2017.
[13] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia- Segui, and T. Watteyne, “Understanding the Limits of LoRaWAN,” IEEE Communications Magazine, vol. 55, no. 9, pp. 34–40, Sept. 2017.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, April 2026
Impact Factor: 8.2

[14] J. Chen, K. Li, and Y. Zhou, "Design and Implementation of an Intelli- gent Parcel Locker System with Secure Access Control," International Journal of Embedded Systems, vol. 13, no. 3, pp. 201–210, 2021.
[15] Node.js, "Node.js Documentation," OpenJS Foundation, 2025. [Online]. Available: https://nodejs.org/en/docs
[16] Express.js, "Express Web Framework Documentation," OpenJS Foun- dation, 2025. [Online]. Available: https://expressjs.com/