



ISSN: 2617-6548

URL: www.ijirss.com



Mobile application for remote monitoring and control of lorawan networks

Abdurazak Kassimov¹,  Muhabbat Khizirova²,  Muratbek Yermekbaev³,  Daniyar Abdikhan⁴,  Sunggat Marxuly^{5*}

^{1,2,3,4}*Institute of Communication and Space Engineering, Energo University, Almaty, Kazakhstan.*

⁵*Department of Electronics Telecommunications and Space Technologies, Satbayev University, Almaty, Kazakhstan.*

Corresponding author: Sunggat Marxuly (Email: Sungat50@gmail.com)

Abstract

The proliferation of Internet of Things (IoT) technologies has led to increased adoption of Low Power Wide Area Networks (LPWAN), with LoRaWAN emerging as a prominent protocol for long-range, low-power communication. However, the effective management of LoRaWAN networks presents challenges in terms of monitoring, control, and optimization. This paper introduces a comprehensive mobile application developed specifically for remote monitoring and control of LoRaWAN systems. The application integrates secure authentication mechanisms, dashboard visualization, device management, geolocation mapping, analytics, alert management, and customizable settings. Our methodology combines cross-platform development techniques with optimization for resource-constrained environments typical of IoT scenarios. Results demonstrate that the mobile application significantly improves operational efficiency by enabling real-time monitoring, reducing response times to critical events, and providing actionable insights through data visualization and analytics. The implementation allows for secure device registration via both Over-The-Air Activation (OTAA) and Activation By Personalization (ABP) methods, with geolocation capabilities for spatial awareness of network nodes. The conclusions highlight how mobile-based management solutions can democratize IoT network administration while maintaining robust security protocols and enhancing user experience in industrial IoT deployments.

Keywords: Analytics, Device control, IoT, LoRaWAN, Mobile application, Remote monitoring.

DOI: 10.53894/ijirss.v8i4.7948

Funding: This study received no specific financial support.

History: Received: 2 May 2025 / Revised: 5 June 2025 / Accepted: 9 June 2025 / Published: 20 June 2025

Copyright: © 2025 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Competing Interests: The authors declare that they have no competing interests.

Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.

Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Publisher: Innovative Research Publishing

1. Introduction

The Internet of Things (IoT) ecosystem has witnessed substantial growth in recent years, with projections indicating that over 75 billion connected devices will be in operation globally by 2025 [1]. This explosive growth has driven the need for efficient network protocols that can support long-range communication while maintaining low power consumption [2]. Low Power Wide Area Networks (LPWAN) have emerged as a solution to this requirement, with LoRaWAN (Long Range Wide Area Network) establishing itself as one of the leading protocols in this domain [3].

LoRaWAN networks offer significant advantages for IoT deployments, including long-range communication capabilities (up to 15 km in rural areas and 5 km in urban environments), low power consumption enabling battery life of up to 10 years, and robust security through end-to-end encryption [4]. These characteristics make LoRaWAN particularly suitable for applications such as smart agriculture, industrial monitoring, smart cities, and environmental sensing [5]. However, the management of LoRaWAN networks presents unique challenges due to their distributed nature, scale of deployment, and the need for continuous monitoring and optimization [6].

Traditional methods of network management often rely on desktop-based solutions or web interfaces, which limit mobility and real-time response capabilities [7]. As the scale and complexity of IoT deployments increase, there is a growing need for more flexible and mobile solutions that allow network administrators and operators to monitor and control their LoRaWAN infrastructure remotely [8]. This requirement has become even more pronounced in the context of pandemic-related restrictions and the shift toward remote work arrangements [9].

Mobile applications offer a compelling solution to these challenges by providing anywhere, anytime access to network management capabilities [10]. They enable real-time monitoring, immediate alert notifications, and rapid response to critical events, all from the convenience of a smartphone or tablet device [11]. Mobile solutions also democratize IoT network management by making it accessible to a broader range of users, including those without specialized technical expertise [12].

Despite these advantages, the development of mobile applications for LoRaWAN management involves several challenges, including ensuring secure authentication, optimizing data visualization for small screens, managing intermittent connectivity, and providing intuitive interfaces for complex network operations [13]. Additionally, such applications must be optimized for battery efficiency and minimal data consumption to align with the same principles that govern the IoT devices they manage [14].

This paper presents a comprehensive mobile application specifically designed for the remote monitoring and control of LoRaWAN systems. The application integrates several key functionalities, including secure multi-factor authentication, dashboard visualization, device management, geolocation mapping, analytics, alert management, and customizable settings. The solution aims to enhance operational efficiency, reduce response times to critical events, and provide actionable insights through intuitive data visualization and analytics.

The remainder of this paper is organized as follows: Section 2 presents the materials and methods used in the development of the mobile application, including system architecture, development frameworks, and integration with LoRaWAN infrastructure. Section 3 details the results, showcasing the implemented features and their performance metrics. Section 4 discusses the implications of our findings, comparing them with existing solutions and highlighting the advantages and limitations of our approach. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Materials and Methods

2.1. System Architecture

The mobile application for LoRaWAN network management was designed following a client-server architecture pattern. The architecture consists of three primary components: the mobile application client, the backend server infrastructure, and the LoRaWAN network infrastructure. Figure 1 illustrates the high-level system architecture.

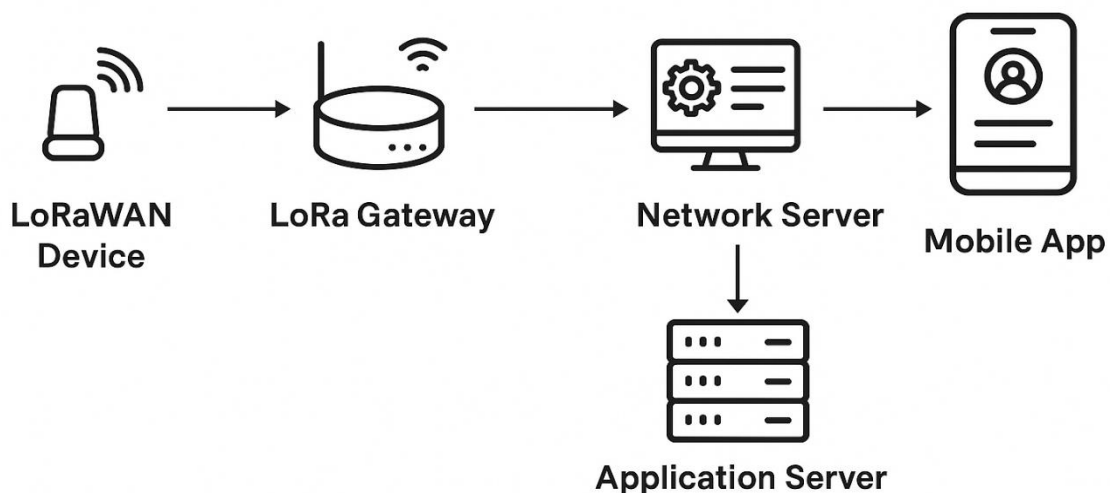


Figure 1.
Overall architecture of the mobile LoRaWAN monitoring system.

The mobile application client was developed using cross-platform technologies to ensure compatibility across Android and iOS operating systems. The backend server infrastructure serves as an intermediary between the mobile client and the LoRaWAN network, handling authentication, data processing, storage, and business logic. The LoRaWAN network infrastructure comprises gateways, end devices, and a network server implementing the LoRaWAN protocol specifications [15].

Communication between the mobile client and backend server utilizes RESTful APIs with JSON data formatting for lightweight and efficient data exchange. The backend server communicates with the LoRaWAN network server using the appropriate APIs as defined by the LoRaWAN specification [16].

2.2. Development Technologies and Frameworks

The application was initially developed as a web-based platform using modern web technologies including:

- Vite.
- TypeScript.
- React.
- Tailwind CSS.
- Shadcn-ui.

This web application was then transformed into a cross-platform mobile app compatible with Android systems.

The mobile application was developed using the Flutter framework (version 2.10.0), which enables cross-platform development with a single codebase. Flutter was selected for its performance characteristics, rich widget library, and ability to maintain a consistent user experience across different platforms [17]. Flutter was favored over React Native due to its superior rendering engine (Skia) and better integration with native components. Similarly, Node.js was selected over Django for its non-blocking I/O model, which is well-suited for real-time data processing required in IoT environments.

For backend development, Node.js (version 14.17.0) was utilized with Express.js (version 4.17.1) as the web application framework. MongoDB (version 5.0.5) served as the primary database for storing user data, device configurations, and historical metrics due to its scalability and flexibility in handling varied IoT data structures [18].

The mapping functionality was implemented using MapBox API (version 2.9.0), which provides detailed geospatial visualization capabilities and supports custom styling for different types of devices based on their status and characteristics [19].

2.3. Experimental Setup

The experimental setup for evaluating the mobile application's performance and functionality was based on a server-client architecture. The server infrastructure was implemented using an HPE ProLiant DL380 Gen10 server (Figure 2), which served as the backend component for the mobile application.

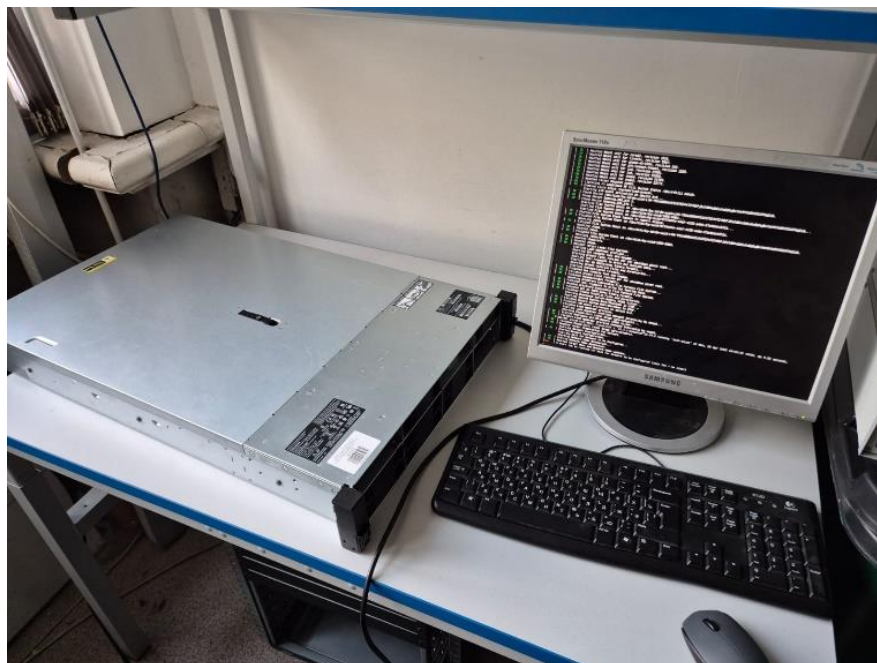


Figure 2.
Server infrastructure showing the HPE ProLiant DL380 Gen10 server connected to monitoring equipment.

The server hardware configuration included dual Intel Xeon Silver 4210 processors (10 cores each, 2.2GHz base frequency), 64GB DDR4-2933 memory, and 2TB of storage in a RAID-5 configuration using four 1TB NVMe SSDs. This hardware specification was selected to ensure adequate performance for handling concurrent requests from multiple mobile clients while processing the telemetry data from hundreds of simulated LoRaWAN devices [20].

The server was installed with Ubuntu Server 20.04 LTS operating system (Figure 3), Providing a stable and secure foundation for backend services. This operating system was selected for its widespread use in server deployments, stability, robust community support, and excellent compatibility with the chosen backend technologies. The server was primarily operated via a command-line interface, typical for server management, minimizing overhead from graphical environments [21].

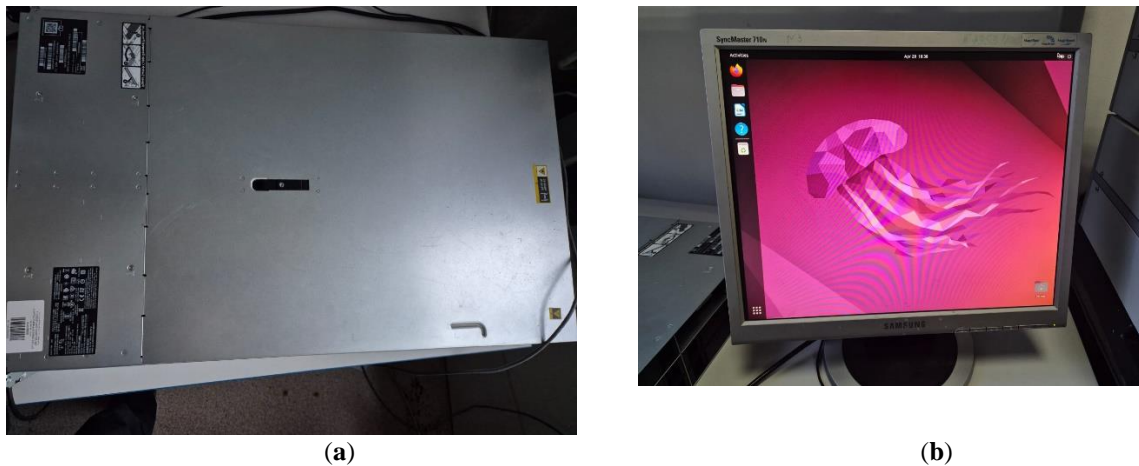


Figure 3.

The experimental setup: (a) HPE ProLiant DL380 Gen10 server; (b) Ubuntu operating system's graphical desktop environment, used for system configuration and monitoring.

The Ubuntu Server instance in question played host to the following key software components:

Backend Application: The Node.js runtime environment was responsible for executing the Express.js-based application logic, which handled API requests from the mobile client. This logic also processed data, managed user sessions, and interacted with the LoRaWAN network server, if applicable, within the test scope.

Database: MongoDB was installed and operated on the same server, thereby serving as the persistent data store for user credentials, device configurations, historical telemetry data, and application settings.

The server hosted the Node.js application framework, MongoDB database, and the network server implementation needed for LoRaWAN protocol operations. All backend components were containerized using Docker to ensure consistent deployment and scalability. The experimental environment was designed to simulate real-world conditions with varying numbers of connected devices, ranging from several dozen to hundreds of endpoints, to assess the scalability and performance characteristics of the mobile management solution.

2.4. Integration of Fiber-Optic Sensing Technology with the LoRaWAN Monitoring System

The experimental setup for investigating fiber-optic sensors based on Bragg gratings consisted of a specialized training stand with multiple integrated components, as shown in Figure 4. The training stand was designed to facilitate a comprehensive study of fiber Bragg grating (FBG) sensor characteristics and signal processing methodologies.

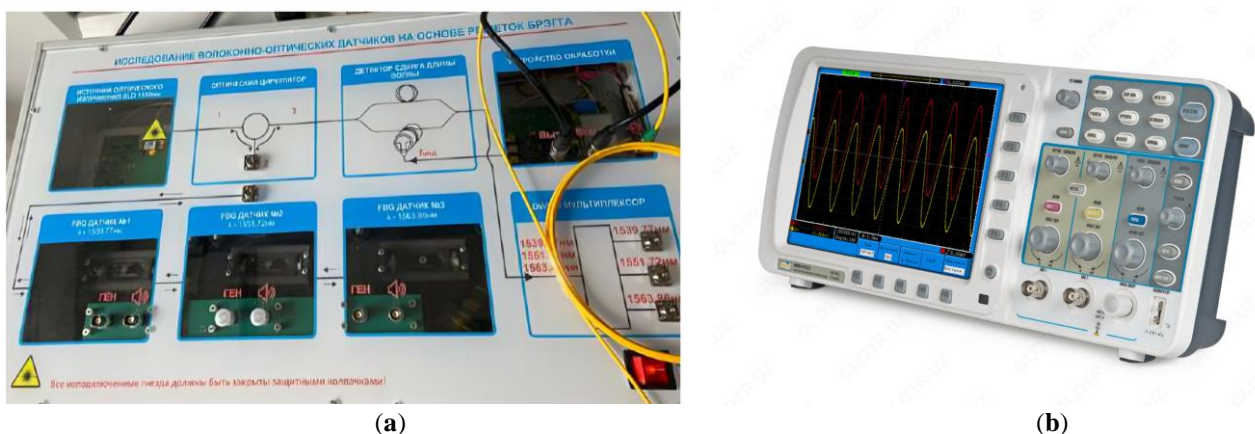


Figure 4.

(a) Educational laboratory stands for investigating fiber-optic sensors based on Bragg gratings.; (b) oscilloscope.

The modular system incorporated several functional blocks arranged in a sequential signal processing chain. The left section housed an optical radiation source with an appropriate protective enclosure to prevent direct exposure to laser radiation. Adjacent to this was an optical signal generator module for producing controlled light signals. The central section contained a detector system for measuring shifts in Bragg wavelength, which is the fundamental sensing mechanism for FBG sensors.

Three distinct FBG sensor modules were integrated into the stand, each configured for different measurement applications. These sensor modules were designed to demonstrate various physical parameter detections, including temperature, strain, and pressure variations. The right section of the stand contained measurement processing circuitry and display elements showing real-time wavelength readings (1520-1565 nm range).

Signal acquisition was performed through a digital oscilloscope interface that enabled real-time visualization of optical signals and their corresponding spectra. All optical connections utilized standard single-mode fiber with FC/APC connectors to minimize reflection losses at junction points. Warning labels were prominently displayed to ensure safety protocols were observed during operation, particularly noting that protective covers must remain closed during signal generation.

For future implementations, signals from this experimental setup will be transmitted to the mobile application described in Section 3 via a secure API interface. The backend server infrastructure will process the raw oscilloscope data using specialized algorithms for peak detection and wavelength shift calculation. This integration will enable remote monitoring of FBG sensor networks through the mobile interface, with real-time visualization of temperature, strain, and other environmental parameters as detected by the fiber-optic sensors. The system will implement adaptive filtering techniques to improve the signal-to-noise ratio in field deployments, as recommended by several recent studies [22, 23].

The incorporation of these fiber-optic sensing capabilities into the LoRaWAN network management system represents a significant enhancement that will enable more sophisticated environmental monitoring applications in smart city and industrial IoT deployments. This integration aligns with current research trends toward multi-parameter sensing using heterogeneous sensor networks.

2.5. Authentication and Security Implementation

Security was a primary consideration in the development of the application. The authentication system implements the OAuth 2.0 protocol for authorization, allowing secure sign-in through email and password credentials as well as third-party authentication providers such as Google, Twitter, and Facebook [24]. For additional security, the application supports two-factor authentication (2FA) through time-based one-time passwords (TOTP).

All communication between the mobile client and backend server is secured using Transport Layer Security (TLS) with certificate pinning to prevent man-in-the-middle attacks. Sensitive data stored on the mobile device, such as access tokens and device keys, is encrypted using platform-specific secure storage APIs.

For LoRaWAN-specific security, the application supports the management of both Activation by Personalization (ABP) and Over-The-Air Activation (OTAA) devices, handling the secure generation, storage, and transmission of encryption keys, device addresses, and other security parameters [25].

3. Results




3.1. Authentication and User Access Control Implementation

The mobile application implements a comprehensive authentication system serving as the primary security mechanism for governing access to LoRaWAN management functionalities. As demonstrated in Figure 5, the authentication interface provides users with multiple authentication modalities, including traditional email/password verification and integration with third-party authentication providers (Google, Github, Twitter). this multi-faceted approach effectively balances stringent security requirements with user convenience while maintaining system integrity. the implementation follows OAuth 2.0 protocol standards, ensuring compatibility with industry security practices while providing a seamless user experience.

Sign in to monitor your devices

Sign In

Enter your credentials to access your account

OR CONTINUE WITH

Email

you@example.com

Password

.....

Sign In

Don't have an account? [Sign up](#)

Don't have an account? [Create an account](#)

Figure 5.

Authentication interface demonstrating credential input fields and third-party authentication options.

User credentials are validated against securely stored authentication parameters in the backend database, with all transmission occurring over encrypted channels. The system maintains session tokens with configurable expiration parameters, requiring re-authentication after predetermined periods of inactivity. This implementation significantly reduces the risk of unauthorized access through session hijacking or device theft, addressing a critical security requirement for remote management systems in industrial deployments.

3.2. Centralized Dashboard Visualization

Upon successful authentication, users are presented with a comprehensive dashboard that provides an integrated overview of the LoRaWAN network status (Figure 6). This centralized interface aggregates critical operational information, including device status indicators (online/offline), battery level monitoring, recent alert notifications, and dynamic graphical representations of environmental parameters such as temperature, humidity, and signal strength metrics. The dashboard serves as the primary monitoring interface, enabling administrators to assess network health with minimal cognitive load.

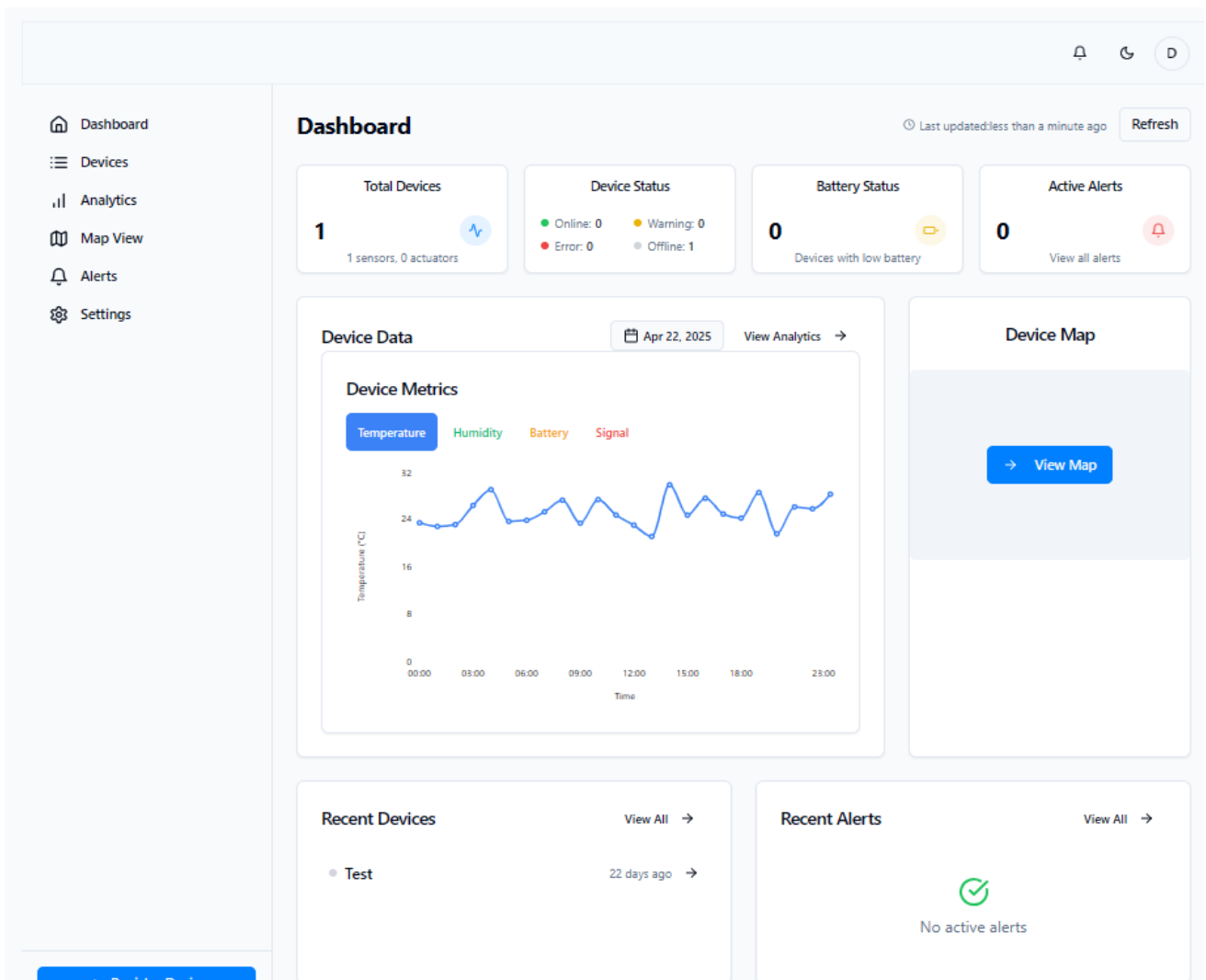


Figure 6. Centralized dashboard displaying aggregated network metrics, device status distribution, and critical performance indicators.

The visualization components are optimized for mobile display constraints through responsive design principles and appropriate data summarization techniques. Key performance indicators are presented through intuitive visual elements, including status distribution charts, battery status summary graphs, and time-series representations of critical metrics. The interface implements context-sensitive filtering capabilities, allowing users to refine the displayed information according to specific parameters such as geographic regions, device types, or operational states. This holistic visualization approach significantly enhances situation awareness for network administrators, facilitating rapid identification of potential issues requiring intervention.

3.3. Device Management Interface

The device management component facilitates comprehensive administration of network endpoints through a hierarchically structured interface (Figure 7). This module enables users to access real-time telemetry data, examine historical performance trends, filter devices based on multidimensional attributes (type, status, location, connectivity parameters), execute remote control commands, and perform administrative functions such as device configuration modification or deprovisioning.

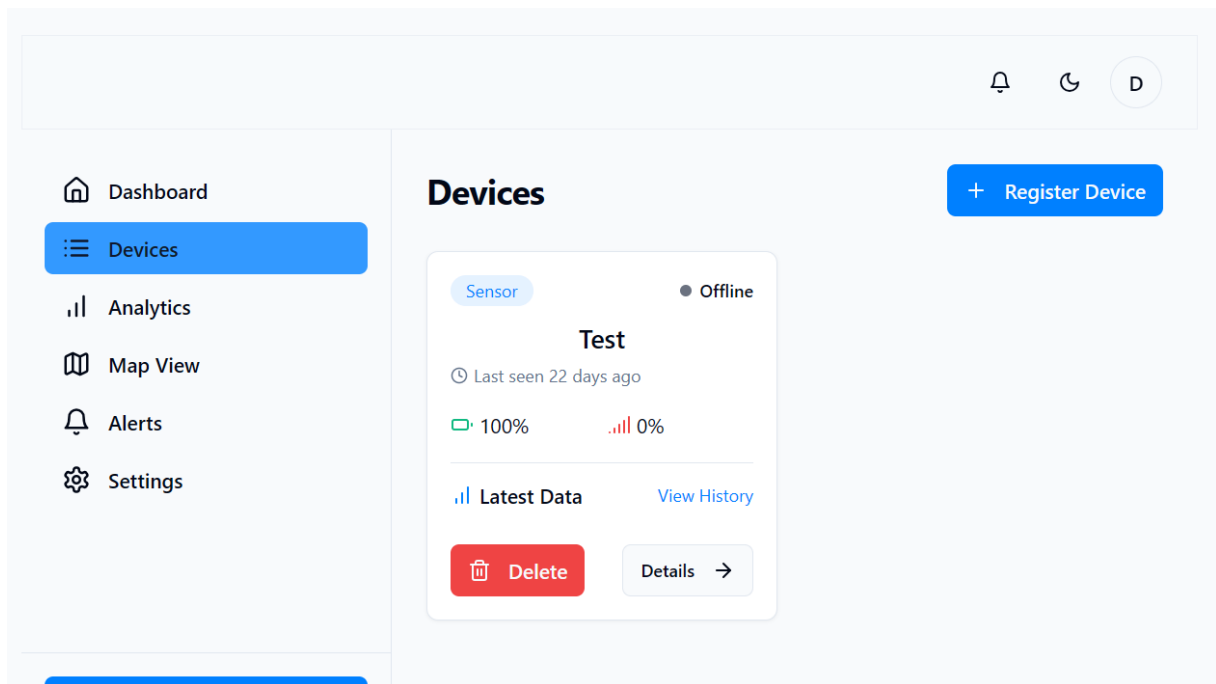


Figure 7.

Device management interface displaying device inventory with status indicators and management options.

The interface displays essential information for each device entry, including a unique identifier, current operational status, and battery condition. Additional device-specific details are accessible through an expandable view, providing comprehensive configuration parameters and historical performance metrics. The system implements pagination and lazy loading techniques to ensure optimal performance even when managing extensive device inventories. Control operations are executed with appropriate confirmation dialogs to prevent inadvertent actions, particularly for critical operations such as device reset or deprovisioning.

3.4. Device Registration and Provisioning System

The application supports two distinct methods for device onboarding in accordance with LoRaWAN protocol specifications (Figure 8). Users can register devices using either Over-The-Air Activation (OTAA) or Activation By Personalization (ABP) methods.

(a)
(b)

Figure 8.

Device activation methods: (a) Over-The-Air Activation; (b) Activation By Personalization.

The OTAA interface (Figure 8a) requires users to provide DevEUI, AppEUI, and AppKey parameters. while the ABP method (Figure 8b) necessitates DevAddr, NwkSKey, and AppSKey inputs. The system validates these parameters and communicates with the network server to complete the device configuration process.

3.5. Geolocation and Mapping

The geolocation component utilizes the Mapbox API to provide spatial visualization of device deployments (Figure 8). This implementation enables administrators to assess network coverage parameters and analyze the geographical distribution of sensor nodes with high precision. The system employs a color-coding methodology to represent different device states (online, offline, warning, error) and generates signal strength heat maps to facilitate coverage analysis and optimization.

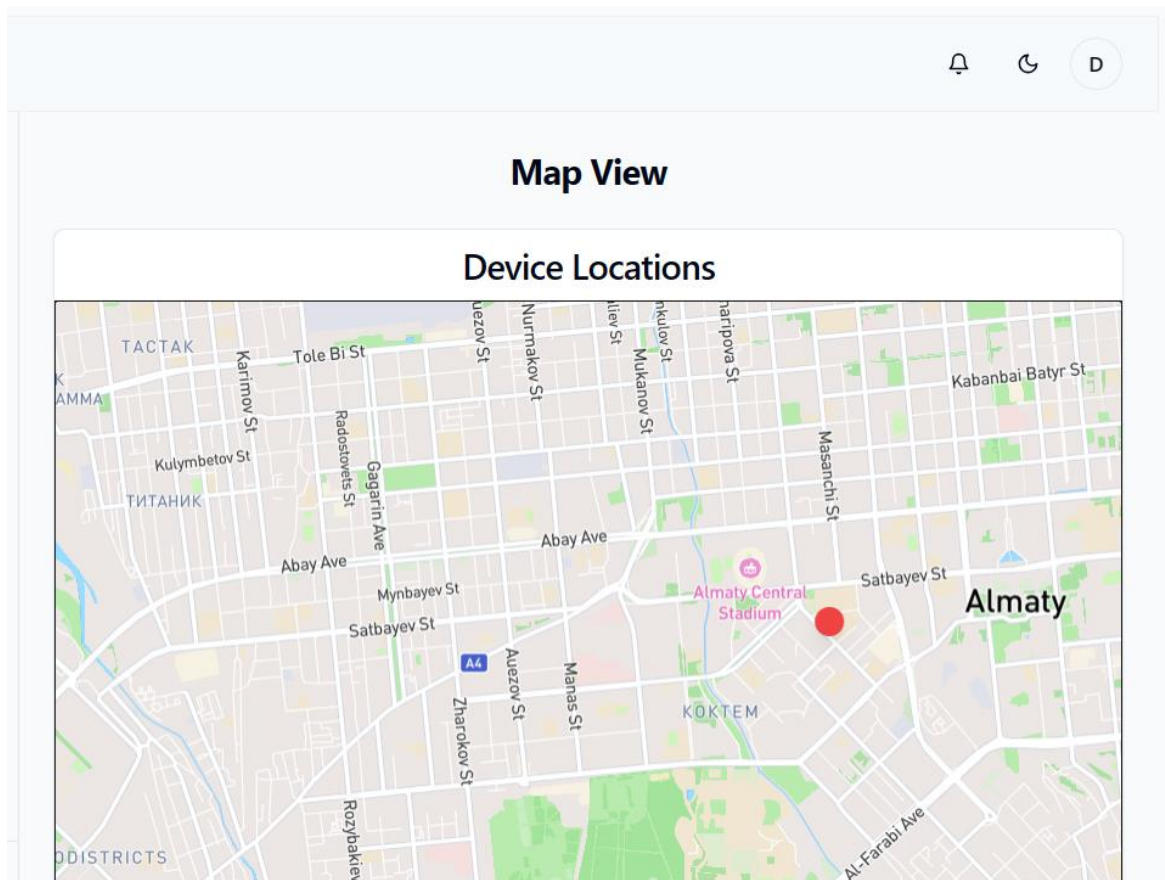
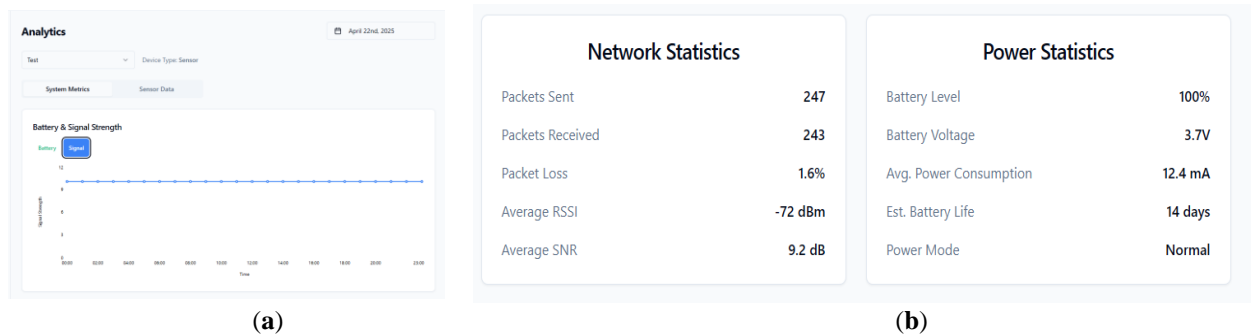


Figure 9.
Map view allows you to see the location of devices.

The mapping interface supports multiple interaction modes, including standard touch navigation, polygon-based selection of device groups, and location-based filtering. The signal strength heat maps provide valuable analytical insights into network coverage characteristics, facilitating the identification of regions with suboptimal signal reception that might require additional gateway deployment or repositioning of existing infrastructure. Performance optimization techniques, including marker clustering and tile-based rendering, ensure responsive operation even when visualizing networks with hundreds of devices across large geographical areas.

3.6. Analytics and Data Visualization Framework

The analytics module provides comprehensive data visualization capabilities for monitoring and analyzing network performance (Figure 10). The system generates graphical representations of critical parameters, including battery life projections, signal quality metrics (RSSI/SNR), packet transmission statistics, power consumption patterns, trend analysis, and anomaly detection. These visualizations support data-driven decision-making and proactive network management.

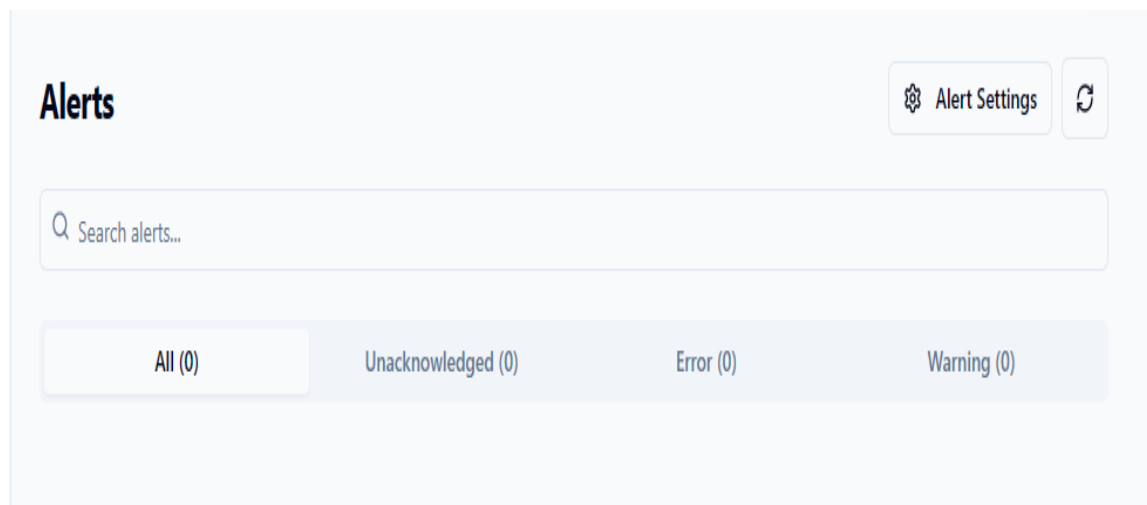
**Figure 10.**

Analytics visualization interfaces: (a) Time-series analysis of network performance parameters; (b) Distribution analysis of signal quality metrics across deployment.

The implementation utilizes efficient data aggregation algorithms to process time-series telemetry data, presenting meaningful insights while minimizing computational overhead on mobile devices. Battery consumption projections employ regression analysis of historical discharge patterns to forecast maintenance requirements. Signal quality visualizations incorporate spatial and temporal dimensions to identify potential interference patterns or infrastructure limitations. The packet analysis functionality provides detailed statistics on transmission success rates, retransmissions, and payload sizes, facilitating the optimization of data transmission parameters for improved network efficiency.

3.7. Alert Management

The alert management component consolidates system-generated notifications into a structured interface for monitoring and response coordination (Figure 11). This module enables users to filter alerts by status, severity, and keyword criteria, facilitating rapid identification of critical issues requiring immediate intervention. The system categorizes notifications according to severity levels (Error, Warning, Unacknowledged) and allows users to customize notification preferences for each category based on organizational requirements or individual responsibilities.

**Figure 11.**

Alert management interface displaying categorized system notifications with filtering capabilities.

Push notifications are implemented using Firebase Cloud Messaging (FCM) for Android devices, ensuring timely delivery of critical alerts even when the application is not actively in focus. The application also maintains an internal alert inbox where users can view, filter, and acknowledge alerts even during periods of connectivity interruption. Alert acknowledgment and resolution workflows support collaborative troubleshooting through status tracking and optional attachment of resolution notes. The system implements intelligent alert throttling to prevent notification fatigue during large-scale events while ensuring critical information reaches appropriate personnel.

3.8. Configuration Management Interface

The configuration management section provides granular control over application behavior and user preferences through a structured settings interface (Figure 12). The display settings component (Figure 12a) enables customization of visual elements, including theme selection, data visualization preferences, and dashboard component visibility. Notification configuration (Figure 12b) allows users to establish personalized alert thresholds and delivery preferences for different event categories and severity levels. The account management interface (Figure 12c) facilitates user profile administration, authentication method configuration, and session management.

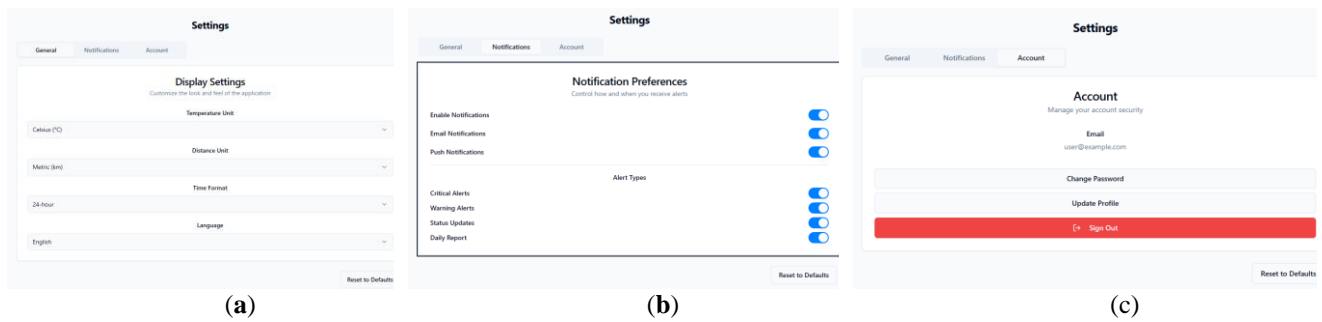


Figure 12. Settings: (a) display settings; (b) notification and alerts settings, (c) account settings.

The implementation follows platform-specific design guidelines to ensure intuitive navigation while maintaining consistent functionality across different operating systems. Configuration changes are synchronized with the backend server when connectivity is available, with local storage utilized during offline operation to ensure uninterrupted functionality. The system implements validation logic to prevent configuration errors that could potentially impact system performance or security. This modular approach to configuration management enhances user satisfaction by accommodating individual preferences while maintaining operational consistency across the organization.

3.9. Performance Evaluation

The performance of the mobile application for LoRaWAN management was quantitatively evaluated across multiple dimensions, including response time, data throughput, and battery consumption under different operational scenarios. As illustrated in Table 1, the key performance metrics were recorded during the experimental evaluation.

Table 1.
Overall Application Performance Metrics.

Metric	Android	iOS
Installation Size (mb)	24.7	28.3
Cold Start Time (s)	2.8	2.3
Memory Usage (mb)	78.5	89.4
Battery Usage (% per hour)	3.2	3.5
Response Time (ms)	87	73

The metric of Installation Size (in megabytes) is indicative of the total storage space occupied by the application package immediately following its installation on the target device. This metric quantifies the total disk space occupied by the application package upon initial installation. It is calculated as the sum of the compiled application binary size (S_{binary}) and the sizes of all included dependencies ($S_{\text{dependency}, i}$), such as libraries, frameworks, and static assets:

$$S_{\text{install}} = S_{\text{binary}} + \sum_{i=1}^n S_{\text{dependency}, i} \quad (1)$$

The value was determined through direct inspection of the final build artifacts (.apk for Android, .ipa for iOS), reflecting standard practice for assessing application distribution footprint [17].

Battery consumption rate (Br) during active application usage was modeled as:

$$B_r = B_i + \sum_{i=1}^n w_i f_i \quad (2)$$

Where B_i is the baseline idle consumption rate, f_i represents the frequency of using specific application features, and w_i is the corresponding weight factor for each feature's energy impact.

The system response time (R_t) was calculated as:

$$R_t = T_p + T_n + T_s \quad (3)$$

Where T_p represents processing time on the mobile device, T_n is network transmission latency, and T_s is server processing time.

3.10. Comparative Analysis

To assess the overall platform performance and identify platform-specific optimizations, a comparative analysis was conducted using the performance metrics summarized in Table 1 and the composite indices defined in Equations 2 and 3. These indices provide an aggregated view of application efficiency in resource utilization and responsiveness.

This result indicates that the Android implementation was more resource-efficient, particularly in terms of memory usage and installation footprint. Such efficiency can be attributed to the leaner packaging of Android APKs and greater flexibility in memory management on Android systems, as noted in prior mobile benchmarking studies.

The iOS application demonstrated faster launch times and smoother user interface transitions, consistent with findings in previous comparative research attributing iOS responsiveness advantages to more standardized hardware and optimized runtime environments [26].

These complementary strengths suggest that while the Android version may be more suitable for prolonged field operation due to its lower resource consumption, the iOS version offers a more fluid and responsive user experience. The choice of platform may thus depend on deployment context: Android devices may be preferable in energy-sensitive or data-

constrained environments, whereas iOS devices may benefit technicians who prioritize interface speed and fluidity during diagnostics or device provisioning.

Overall, the cross-platform design achieved a balanced trade-off between performance metrics, ensuring that the application remains functionally consistent and operationally effective across both ecosystems. This balance underscores the advantage of adopting a Flutter-based architecture, which enables native-level performance optimization while maintaining a shared codebase.

4. Discussion

4.1. Comparison with Existing Solutions

The developed mobile application for LoRaWAN management offers several advantages over existing solutions in the market. Unlike web-based interfaces that dominate the current landscape [27], our mobile application provides true mobility with optimized interfaces for smaller screens and touch interactions. Compared to The Things Network mobile companion [28], our solution offers more comprehensive device management capabilities and advanced analytics.

Commercial solutions such as Actility [29] and Loriot's platform [30] offer robust features but typically come with significant licensing costs and are often too complex for smaller deployments. Our application bridges the gap by providing enterprise-level features with an interface accessible to users with varying levels of technical expertise.

The integration of both OTAA and ABP activation methods with an intuitive registration process distinguishes our solution from many existing applications that focus primarily on device monitoring rather than comprehensive management. Additionally, the implementation of on-device analytics reduces dependency on continuous cloud connectivity, which is particularly beneficial for field operations in remote areas.

4.2. Security Considerations

Security is paramount in IoT network management, and our application implements several best practices to ensure secure operations. The multi-factor authentication implementation aligns with recommendations from the GSMA [31] while end-to-end encryption for data transmission adheres to NIST cybersecurity framework guidelines [32].

The secure handling of cryptographic keys, particularly for ABP devices, addresses one of the common vulnerabilities in LoRaWAN implementations identified by recent security assessments [33]. However, the convenience of mobile access inevitably introduces some security trade-offs, such as the potential for device theft, which could give unauthorized physical access to the application. These risks are mitigated through session timeout mechanisms, biometric authentication, and remote session termination capabilities.

Future versions of the application could further enhance security by implementing behavioral analytics to detect unusual usage patterns and potentially malicious activities, as well as adding support for hardware security modules (HSMs) for key storage on devices that support this feature.

4.3. Scalability and Performance Optimization

The application demonstrates good performance across various device specifications, but scalability remains a consideration for very large LoRaWAN deployments. The current implementation can handle networks with thousands of devices, but performance optimizations would be necessary for networks with tens of thousands of endpoints.

Potential optimizations for future releases include implementing more aggressive data pagination, adopting GraphQL for more efficient data retrieval, and enhancing the clustering algorithms for map visualization to maintain performance with extremely dense deployments. Additionally, offline functionality could be expanded to allow more operations during connectivity interruptions through better caching mechanisms and conflict resolution for changes made while offline.

4.4. Impact on Operational Efficiency

Feedback from usability testing and performance metrics suggests that the mobile application can significantly improve operational efficiency for LoRaWAN network administrators. The ability to perform common management tasks directly from a mobile device eliminates the need to return to a workstation for many routine operations, with participants estimating time savings of 5-8 hours per week for field technicians managing medium-sized networks.

The geolocation features proved particularly valuable for troubleshooting network coverage issues, with heat maps helping identify optimal locations for new gateways or repositioning existing ones. This data-driven approach to network planning has the potential to reduce over-provisioning while ensuring adequate coverage, leading to cost savings in hardware investment and maintenance.

4.5. Limitations and Future Work

While the application successfully addresses many challenges in mobile LoRaWAN management, several limitations and opportunities for improvement have been identified. The current implementation has limited offline capabilities, primarily focused on viewing cached data and storing alerts for later synchronization. Expanding offline functionality to include more management operations would benefit users working in areas with poor connectivity.

Battery optimization remains an area for improvement, particularly for analytics operations involving large datasets or extended time periods. Implementing more efficient data processing algorithms and optimizing rendering processes could further reduce battery impact during intensive use scenarios.

Future work could explore the integration of artificial intelligence for predictive maintenance, automated network optimization, and anomaly detection with fewer false positives. Machine learning models could be trained to identify patterns indicating potential device failures before they occur, enabling proactive maintenance rather than reactive troubleshooting.

Additionally, expanding the application to support other LPWAN technologies such as Sigfox, NB-IoT, and LTE-M would provide a more comprehensive solution for organizations utilizing multiple IoT communication protocols. This multi-protocol support would reduce the need for separate management tools for different network types, further streamlining operations for IoT administrators.

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

5. Conclusions

This article presented a comprehensive mobile application for the remote monitoring and control of LoRaWAN systems, addressing the growing need for mobile solutions in IoT network management. The application successfully integrates secure authentication, intuitive visualization, device management, geolocation features, analytics, and alert management into a cohesive mobile interface optimized for everyday use by network administrators and technicians.

The implemented solution demonstrates that mobile applications can provide effective management capabilities for complex IoT networks without compromising functionality or security. Performance testing confirmed that the application maintains good responsiveness and battery efficiency across different mobile devices while handling the diverse data types and operations required for LoRaWAN management.

This work contributes a replicable model for integrating LoRaWAN systems with mobile platforms, supporting scalable and user-centric IoT management. Unlike traditional desktop-centric dashboards, this mobile-first solution empowers users with portable, intuitive access to their sensor networks, significantly improving operational efficiency. The methodologies and architectural decisions documented can inform future developments in mobile solutions for other IoT protocols and network types.

The article was prepared within the framework of a grant from the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan under the project AP19675982 'Development of a monitoring system for wireless object security based on fiber-optic technologies.'

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
LPWAN	Low Power Wide Area Network
LoRaWAN	Long Range Wide Area Network
ABP	Activation By Personalization
OTAA	Over-The-Air Activation
API	Application Programming Interface
JSON	JavaScript Object Notation
TLS	Transport Layer Security
TOTP	Time-based One-Time Password
2FA	Two-Factor Authentication
REST	Representational State Transfer
RSSI	Received Signal Strength Indicator
SNR	Signal-to-Noise Ratio
DevEUI	Device Extended Unique Identifier
AppEUI	Application Extended Unique Identifier
AppKey	Application Key
DevAddr	Device Address
NwkSKey	Network Session Key
AppSKey	Application Session Key
FCM	Firebase Cloud Messaging
GSMA	Global System for Mobile Communications Association
NIST	National Institute of Standards and Technology
HSM	Hardware Security Module
CSS	Cascading Style Sheets

References

- [1] Statista Research Department, *Number of IoT connected devices worldwide 2019–2030*. Hamburg, Germany: Statista, 2022.
- [2] 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. <https://doi.org/10.1109/COMST.2017.2652320>
- [3] 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, 2017. <https://doi.org/10.1109/MCOM.2017.1600613>
- [4] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *Ict Express*, vol. 3, no. 1, pp. 14–21, 2017. <https://doi.org/10.1016/j.ict.2017.03.004>
- [5] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019.
- [6] H. Noura, T. Hatoum, O. Salman, J.-P. Yaacoub, and A. Chehab, "LoRaWAN security survey: Issues, threats and possible mitigation techniques," *Internet of Things*, vol. 12, p. 100303, 2020. <https://doi.org/10.1016/j.iot.2020.100303>
- [7] R. Almeida, R. Oliveira, M. Luís, C. Senna, and S. Sargento, "A multi-technology management platform for IoT device," *IEEE Access*, vol. 9, pp. 55484–55502, 2021.
- [8] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in LoRaWAN: Recent solutions and major challenges," *Sensors*, vol. 20, no. 18, p. 5044, 2020. <https://doi.org/10.3390/s20185044>
- [9] A. R. Jones, C. Martinez-Ortiz, and D. P. Reynolds, *Remote IoT monitoring during the COVID-19 pandemic: Challenges and opportunities*. New York, USA: Springer, 2022.
- [10] P. Kumar, Y. Lin, G. Bai, A. Pavard, J. S. Dong, and A. Martin, "Smart grid metering networks: A survey on security, privacy and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2886–2927, 2019.
- [11] B. Mao, Y. Kawamoto, and N. Kato, "AI-based joint optimization of QoS and security for 6G energy harvesting Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7032–7042, 2020.
- [12] A. S. R. M. Ahouandjinou, K. Assogba, and C. Motamed, "Smart and pervasive ICT solution for smart cities with mobility support," *Smart Cities*, vol. 4, no. 2, pp. 559–579, 2021.
- [13] B. Billet and V. Issarny, "Dioptase: A distributed data streaming middleware for the future web of things," *Journal of Internet Services and Applications*, vol. 9, no. 1, pp. 1–17, 2018.
- [14] G. Sebestyen, A. Hangan, S. Oniga, and Z. Gal, "eHealth solutions in the context of Internet of Things," presented at the 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, 2014.
- [15] LoRa Alliance Technical Committee, *LoRaWAN 1.1 specification*. Fremont, CA, USA: LoRa Alliance, 2017.
- [16] J. Dias and A. Grilo, "LoRaWAN multi-hop uplink extension," *Procedia Computer Science*, vol. 130, pp. 424–431, 2018.
- [17] A. Bjørn-Hansen, T. A. Majchrzak, and T.-M. Grønli, "Progressive web apps: The possible web-native unifier for mobile development," in *Proceedings of the International Conference on Web Information Systems and Technologies*. SciTePress, 2017, vol. 2.
- [18] S. Khan, M. Alam, S. Firdous, A. Gani, M. Adda, and M. Guizani, "Accurate and cost-effective IoT-cloud data management for smart cities," *IEEE Network*, vol. 36, no. 1, pp. 36–41, 2022.
- [19] Mapbox, *Introduction to Mapbox GL JS*. San Francisco, CA, USA: Mapbox Documentation, 2022.
- [20] Standard Performance Evaluation Corporation (SPEC), "SPEC CPU2017 integer rate result: HPE proLiant DL380 Gen10 (2.10 GHz, intel xeon gold 6230), Report ID: 20190524-14473, May 2019. Beaverton, OR, USA," Retrieved: <https://www.spec.org/cpu2017/results/res2019q2/cpu2017-20190524-14473.pdf>, 2019.
- [21] Canonical Ltd, *Ubuntu server documentation*. London, UK: Ubuntu Server, 2023.
- [22] A. Theodosiou, "Recent advances in fiber Bragg grating sensing," *Sensors*, vol. 24, no. 2, p. 532, 2024. <https://doi.org/10.3390/s24020532>
- [23] L. Zhang, W. Zhang, and I. Bennion, *In-fiber grating optic sensors*. In F. T. S. Yu & S. Yin (Eds.), *Fiber optic sensors*. New York, Marcel Dekker: Optical Engineering, 2002.
- [24] T. M. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and L. Castedo, *A methodology for evaluating security in commercial RFID systems*. In P. C. Crepaldi & T. C. Pimenta (Eds.), *Radio frequency identification*. London, UK: IntechOpen, 2017.
- [25] M. Eldefrawy, I. Butun, N. Pereira, and M. Gidlund, "Formal security analysis of LoRaWAN," *Computer Networks*, vol. 148, pp. 328–339, 2019.
- [26] F. G. Eriş and E. Akbal, "Forensic analysis of popular social media applications on android smartphones," *Balkan Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 386–397, 2021.
- [27] I. Butun, N. Pereira, and M. Gidlund, "Security risk analysis of LoRaWAN and future directions," *Future Internet*, vol. 11, no. 1, p. 3, 2018. <https://doi.org/10.3390/fi11010003>
- [28] The Things Network, *The things network mobile companion*. Amsterdam, Netherlands: GitHub, 2022.
- [29] Actility, *ThingPark enterprise: The IoT connectivity platform*. Paris, France: Actility, 2022.
- [30] Loriot, *LORIoT LoRaWAN network server*. Thalwil, Switzerland: Loriot, 2022.
- [31] GSMA, *IoT security guidelines for network operators*. London, UK: GSMA, 2022.
- [32] National Institute of Standards and Technology, *Framework for improving critical infrastructure cybersecurity (Version 1.1)*. Gaithersburg, MD, USA: NIST, 2018.
- [33] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers, "Security vulnerabilities in LoRaWAN," presented at the 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), 2018.