https://doi.org/10.7840/kics.2025.50.5.835

A Comprehensive Study of IoT-Enabled Smart Umbrella Rental System Using Mobile Application

Mosalla Sophia◆, Akashi Kazunari*, Alao Chunghwan*, Jeong-Dong Kim*, Taek Lee*, Jung-Been Lee°

ABSTRACT

This study presents the development and evaluation of an IoT-based smart umbrella rental service designed to address urban challenges such as the inconvenience of carrying umbrellas and the inefficiencies in existing umbrella rental systems. Inspired by IoT-driven solutions like Ikasa and Retbella, this system enhances user convenience through real-time tracking, efficient management, and user-friendly mobile applications. The system architecture incorporates the NEO-6 GPS module for precise location tracking, ESP32 microcontrollers for seamless integration with reed switches, and Firebase for real-time data updates and secure management. Developed using React Native, the app offers cross-platform compatibility and QR code functionality for streamlined user interactions. Results demonstrate the system's ability to optimize umbrella management and rental processes, contributing to operational efficiency and environmental sustainability. Future directions include integrating advanced sensor technology and AI to elevate user adaptability and real-time monitoring.

Key Words : umbrella rental system, ESP32 microcontroller, mobile application, IoT

Ⅰ. Introduction

The integration of IoT technology into everyday objects has gained significant traction, transforming how we interact with our environment. The broader field of IoT applications spans various industries, enhancing daily activities through real-time data collection and intelligent decision-making[1]. The use of IoT in mobile applications has been recognized as valuable for various sectors, including consumer services, by providing real-time data and enhancing user experience[2]. The increasing ubiquity of IoT has the potential to drastically alter the way products and services are utilized, promoting efficiency and sustainability[3]. It has the potential to represent the broader societal implications of IoT integration, particularly its ability to transform how individuals interact with technology in various contexts, including mobility applications.

In a high traffic area like capital cities carrying a big umbrella is difficult. For example, if a person forgets their umbrella, they should either buy one or use an umbrella rental system. Additionally, most existing umbrella rental services lack the capability to track user activity and location. Furthermore, many of these services do not have a corresponding mobile application. This highlights the broader impact of

※ Enter acknowledgment here: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (RS-2023-00243114) and the MSIT(Ministry of Science, ICT), Korea, under the National Program for Excellence in SW), supervised by the IITP(Institute of Information & communications Technology Planning & Evaluation) in 2025(2024-0-00023).
◆ First Author : Sun Moon University Division of Computer Science and Engineering, sophiamosalla@sunmoon.ac.kr, 학생회원
° Corresponding Author : Sun Moon University Division of Computer Science and Engineering, jungbini@sunmoon.ac.kr, 정회원
* Sun Moon University Division of Computer Science and Engineering, horaciowa115@sunmoon.ac.kr; projoe00@sunmoon.ac.kr; kjd4u@sunmoon.ac.kr; comtaek76@sunmoon.ac.kr
논문번호 : 202411-267-E-RN, Received November 4, 2024; Revised December 16, 2024; Accepted December 19, 2024

IoT-enabled mobile apps in improving user convenience and accessibility, by tracking the location of umbrellas and usage tracking systems which are key principles driving the development of smart rental services.

Traditional umbrellas lack the technological integration needed to address modern user needs effectively. The primary problem is the frequent misplacement and underutilization of umbrellas, coupled with a lack of real-time tracking and accessibility features. This problem leads to inconvenience and increased waste[4]. To solve this problem, an IoT-based smart, on-demand umbrella rental system exemplifies a solution in the realm of global mobility, aligning with the broader trend of IoT applications. The comprehensive technological architecture of this system reflects a paradigm shift in how everyday objects are utilized, emphasizing user convenience, technological innovation, environmental sustainability, and increased accessibility[5].

This paper proposes an IoT-based smart umbrella rental system, a novel application that leverages IoT to improve umbrella usage. The main objectives of this research are to develop and evaluate the system's technological architecture and demonstrate its capabilities in real-time tracking and user convenience. The contributions of this study include the introduction of a IoT-based umbrella system, advancements in real-time data management using Firebase, and practical applications of ESP32 microcontrollers and GPS technology in consumer products.

The contributions of this paper are as follows:

- Advanced IoT-based umbrella rental system that enhances user convenience and operational efficiency
- Robust and scalable real-time data management techniques using Firebase
- Efficient and reliable communication system between devices using ESP32 microcontrollers
- Comprehensive analysis of user experience and accessibility improvements through IoT integration
- GPS tracking module for a umbrella rental box to provide precise real-time location data

The remainder of this paper is organized as follows. Section 2 introduces the related work about the existing similar umbrella rental services. Section 3 details our IoT-based umbrella rental system including the architecture, GPS tracking module, Google Map integration method, and hardware systems integrated with IoT-based modules. Section 4 provides the results of our system. Finally, in Section 5, we conclude our study and outline future research directions.

Ⅱ. Related Work

The Ikasa[1)] is a pioneering service of IoT-driven umbrella-sharing, operates extensively in Japan with a well-structured rental network. Primarily aimed at reducing the inconvenience of carrying umbrellas on rainy days, Ikasa deploys smart umbrella stands at high-traffic locations, including train stations, shopping centers, and commercial complexes. Each stand integrates IoT technology, allowing users to rent an umbrella with ease. Through the Ikasa mobile app, users can locate available umbrellas, reserve one, and track rental duration. The app provides real-time updates on umbrella availability at each stand, enabling a streamlined experience. Ikasa's approach centers on creating an efficient ecosystem with minimal environmental impact by encouraging umbrella reuse. Its emphasis on user convenience, combined with data analytics for tracking and optimizing stand locations, has made Ikasa a popular choice in urban areas where public transportation and walking are prevalent.

On the other hand, Retbella[2)] serves as a successful model for umbrella-sharing services across multiple European cities. This service differentiates itself by focusing on sustainability and user simplicity, aligning with eco-conscious consumer preferences. Retbella's IoT-enabled stands are strategically placed at urban centers, catering to pedestrians, tourists, and professionals in need of a quick, accessible solution for rain protection. Notably, Retbella incorporates a simplified rental system, allowing users to scan QR codes on each stand to initiate a rental without needing to download a dedicated app. This approach increases

1) https://www.i-kasa.com
2) https://www.rentbrella.com

accessibility, particularly for tourists or casual users who may be deterred by app installations. However, for frequent users, a Retbella mobile app is available, providing features such as subscription options, rental history, and loyalty rewards. Retbella has made a significant impact in promoting environmentally friendly practices through shared umbrella use, reducing waste from discarded umbrellas, and offering a convenient, app-optional experience.

Webrella[3] is a Korea-based umbrella rental service, emphasizing a highly accessible and data-optimized IoT solution, particularly appealing to densely populated cities. Each Webrella stand is equipped with sensors to monitor the number of umbrellas in real time and alert maintenance teams when stocks run low. This operational design ensures high availability during peak times, like sudden weather changes, and is supported by an intuitive mobile app that offers location tracking, reservation capabilities, and usage analytics. Webrella's app also enables users to monitor usage statistics, helping the company refine their offerings based on patterns in user demand. This approach not only enhances operational efficiency but also prioritizes user accessibility, as Webrella stations are integrated with public transit maps and routing apps for easy navigation. Additionally, the Webrella system highlights a commitment to efficiency and service reliability, making it a prime example of IoT adoption in smart urban solutions. Table 1 presents a comparative analysis of existing systems and our proposed system.

Ikasa, Retbella, and Webrella demonstrate that IoT technology can enhance user convenience, enable real-time tracking, and provide sustainable alternatives to traditional umbrella ownership. However, limitations exist in the current designs of these umbrella rental services. Most of these services employ a single input-output mechanism in their umbrella boxes, which is efficient for assembly but vulnerable to complete shutdown if one component malfunctions. Additionally, existing designs lack a specialized umbrella-detection system, permitting users to return items other than umbrellas, leading to potential system misuse and increased maintenance costs. Another notable shortfall is the limited focus on app integration; the absence of dedicated, fully featured mobile applications makes these services less user-friendly, especially for new users.

In this study, we aim to address these limitations through an enhanced box design equipped with multiple access points and an umbrella-specific detection system. Furthermore, we propose a more comprehensive and functional mobile application, providing users with seamless and intuitive experience to improve service adoption and efficiency.

3) https://webrella.co.kr

Table 1. Comparative Analysis of Umbrella Rental Systems

Aspect	IKasa	Rentbella	webrella	Our system
Region of Operation	Japan	European cities	South Korea	South Korea
IoT Features	Real-time updates	QR code rental, limited features	basic analytics	GPS tracking, QR code rental
Operational Efficiency	Optimized stand placement	Simplified rental process	Maintenance alerts for low stock	Advanced data management and location tracking
Mobile App	Essential for usage	Optional for casual users	Optional for casual users	Cross-platform, user-friendly
Box design	Single entry	separated entry	Single entry	separated entry

Ⅲ. System Implementation

Figure 1 shows the architecture of our proposed system.

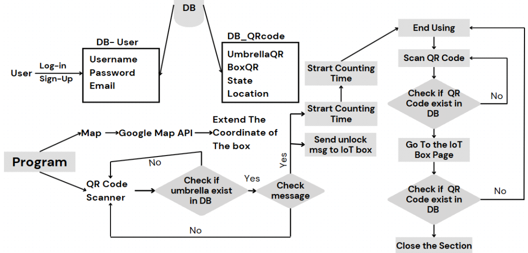


Fig. 1. Architecture diagram of the system.

The process flows with user authentication, where the user logs in by entering their username, password, and email, which are verified against the records in the DB-User database. Next, the user scans the QR code using a QR code scanner. The scanned QR code is checked against the DB-QR code to verify if it exists in the database. If the QR code corresponds to an umbrella and exists in the database, the system proceeds; otherwise, an error message is displayed. The system then uses the Google Map API to map the location of the box and extend the coordinates as needed. If the scanned QR code is verified, the system sends an unlock message to the IoT of the box, which then unlocks the box for the user.

A message is displayed to ensure the user is aware of the process. The system starts counting the time from the moment the box is unlocked to track the duration for which the umbrella is borrowed. When the user is done using the umbrella, they scan the BoxQR to start the return process. The system checks if the BoxQR is valid and exists in the database. If valid, the system goes to the IoT box page to send a close section message to the IoT device. The session is closed once the IoT device receives a valid message and confirms the return of the umbrella.

3.1 IOT Integration of Box

Each umbrella box is equipped with an ESP32 microcontroller, GPS NEO-6 module, and reed switches to monitor the status and location of umbrellas. Figure 2 shows our hardware system and Table 2 represents the circuit diagram.

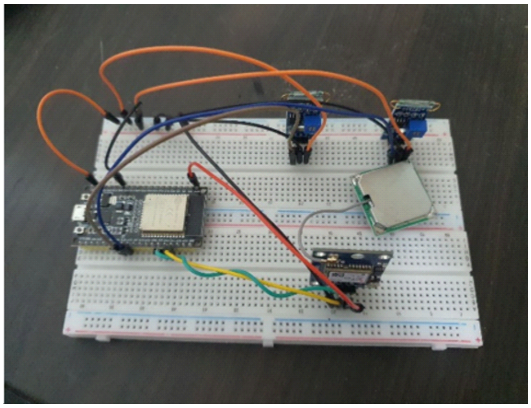


Fig. 2. Hardware System with Microcontroller, Reed Switches, and GPS Module.

Table 2. Circuit Diagram of System.

ESP32	GPS neo-6	Reed switch 1	Reed switch 2
5V	-	VCC	VCC
3V	VCC	-	-
GRD	GRD	GRD	GRD
G17	-	do	-
G18	-	-	-
G21	RX	-	do
G20	RX	-	-

Data is transmitted to Firebase for real-time updates and tracking. Within each container, an ESP32 microcontroller is intricately incorporated to facilitate the acquisition of data from both the GPS NEO-6 module and reed switches. ESP32 acts as the central intelligence unit, seamlessly orchestrating the integration of diverse components for a comprehensive tracking system.

An important component in this complex system is the application of reed switches. Reed switches represent a specialized category of electrical switches, uniquely responsive to magnetic fields for activation and deactivation. Typically, these switches comprise two slender, flexible ferromagnetic wires or blades known as reeds, strategically positioned with a slight separation within a hermetically sealed glass enclosure. This physical arrangement defines the reed switch contacts, with the change-over type featuring three reeds instead of the more common two. The specific configuration of reed switches in this context is purposeful: they are strategically arranged within the container to effectively detect the presence or absence of umbrellas. By leveraging the magnetism-driven actuation of reed switches, the system can discern whether an umbrella is positioned within the container, providing valuable data for tracking and inventory management.

Functionally, the ESP32 microcontroller seamlessly interfaces with both the reed switches and the GPS NEO-6 module. When reed switches detect the open or close status within the box, this information, coupled with GPS coordinates from the NEO-6 module, is diligently transmitted to a real-time database hosted on Firebase. This dynamic database becomes a repository of live information, constantly updated by the

ESP32 microcontroller as it receives and processes data from the reed switches and GPS module.

The ESP32, acting as the nerve center, not only manages the concurrent reception of data from reed switches and the GPS module but also orchestrates the seamless transmission of this information to the Firebase real-time database. This approach to data collection and transmission serves to enhance the efficiency and accuracy of the tracking system, making it a robust solution for real-time monitoring of the presence and location of umbrellas within the designated containers.

3.2 GPS Tracking

The system utilizes the NEO-6 GPS module for real-time tracking of umbrella boxes. This module provides accurate location data, essential for optimizing operational processes and improving resource allocation[6]. Tailored for battery-operated mobile devices with strict cost and space constraints, the NEO-6 modules excel in power efficiency and memory utilization[7]. Notably, the advanced technology of the NEO-6 series effectively counteracts jamming sources, mitigates multipath effects, and provides excellent navigation performance even in challenging environments[8]. Table 3 shows all specifications of the GPS Neo-6.

Table 3. Neo-6 GPS specification.

Parameter	Specification
Receiver Type	50 Channels
Frequency	GPS L1 frequency, C/A Code
SBAS	WAAS, EGNOS, MSAS
Tracking & Navigation (dBm)	NEO-6G/Q/T: -162, NEO-6M/V: -161, NEO-6P: -160
Reacquisition (dBm)	-160 for all
Horizontal Position Accuracy (m)	GPS: 2.5, SBAS: 2.0
Maximum Navigation Update Rate (Hz)	NEO-6G/Q/M/T: 5, NEO-6P/V: 1
Velocity Accuracy	0.1 m/s
Heading Accuracy	0.5 degrees
Altitude	50,000 m
Velocity	500 m/s

3.3 ESP32 Microcontroller and Read Switches

The ESP32 is a versatile 2.4 GHz Wi-Fi and Bluetooth combo chip, crafted using TSMC's low-power 40 nm technology. It performs in delivering optimal power and RF performance, demonstrating exceptional robustness, versatility, and reliability across a broad range of applications and power conditions. The ESP32 microcontroller manages data from the GPS module and reed switches, which detect the presence of umbrellas in the boxes.

A reed switch is a type of linear switch component that operates through magnetic signals, activated by a magnet. This "switch" refers to a dry reed tube, a passive electronic switch component known for its simple structure, compact size, and ease of control. One common issue with other umbrella rental systems is the detection of umbrellas. To address this, we installed reed switches at the bottom of each slot and placed magnets on top of each umbrella. This setup allows us to accurately monitor the insertion and removal of each umbrella. This setup ensures accurate tracking and inventory management, contributing to the overall efficiency of the system.

3.4 Google Map API Integration

The utilization of Google Maps API for dynamic location visualization has become a transformative force in online mapping service applications since its launch in 2005. Grounded in Asynchronous JavaScript and XML (AJAX), Google Maps introduced a novel client/server interaction, maintaining continuous connectivity between the client and server for the immediate downloading of additional map information[9].

This study delineates a comprehensive approach to real-time location tracking using the NEO-6 GPS module. Figure 3 process involves acquiring coordinates from the NEO-6 module and seamlessly transmitting this data to the Firebase Database for efficient storage and retrieval.

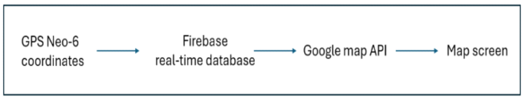


Fig. 3. Process of getting GPS coordinate on map.

The pivotal aspect of this location tracking system is the integration with the Google Maps platform, where the received coordinates are dynamically visualized. The integration with Google Maps is achieved through the utilization of JavaScript, facilitated by a dedicated library[10]. This dynamic visualization not only provides a real-time depiction of the tracked coordinates but also enhances the overall user experience by leveraging the powerful mapping capabilities of Google Maps[11].

The software architecture for the IoT-based umbrella rental system is built using the React Native framework. React Native is a popular choice for developing mobile applications due to its ability to create cross-platform apps with a single codebase. This framework ensures that the application works seamlessly on both iOS and Android devices. To further streamline the development process and enhance the user experience, Expo-Go is used. Expo-Go is a set of tools and services built around React Native, which simplifies the development and deployment processes.

Google Maps API plays a crucial role in this system by providing dynamic map visualization. This integration allows users to view their current location and the locations of umbrella boxes in real time. When a user needs to rent or return an umbrella, they can easily find the nearest box on the map. The map is dynamically updated, ensuring that users always have access to the latest information regarding box locations and availability.

3.5 QR code Scanner

The application includes a QR code scanner, which is implemented using JavaScript libraries. This feature enables users to interact with the umbrella rental system quickly and efficiently. When a user wants to rent or return an umbrella, they simply scan the QR code located on the umbrella or the box. The scanner reads the QR code and retrieves relevant information from the database.

After scanning a QR code, the system checks if the QR code corresponds to an umbrella or a box and verifies its existence in the database. If the QR code is valid and exists in the database, the system proceeds with the next steps, such as unlocking the box or ending the rental session.

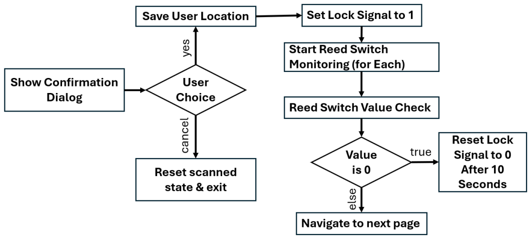


Fig. 4. Checking Reed switches and IOT config algorithm.

3.6 Checking Reed Switches and Confirmation on IoT Device Algorithm

Figure 4 shows the algorithm for checking reed switches and IoT config. This algorithm ensures the seamless integration of reed switch state monitoring and user interaction for IoT applications. It facilitates dynamic decision-making processes, including location saving, navigation, and lock signal management, leveraging Firebase Realtime Database and Firestore services. By implementing this design, IoT systems can provide real-time, user-driven responses, enhancing both functionality and user satisfaction.

- Initialization of Reed Switch Monitoring

To monitor the reed switches effectively, an array of identifiers (switchNames) is initialized, corresponding to specific hardware components. Each reed switch is linked to its respective path in the Firebase Realtime Database using a reference (switchRef), ensuring accurate communication between the physical device and the database.

- Dynamic Reed Switch Monitoring

The algorithm leverages Firebase's onValue() listener to detect real-time changes in reed switch states. When a state change is detected, the system:

- Retrieves the updated state (value) from the database.
- Checks if the state indicates activation (value equals 0).

This algorithm's structured approach integrates efficient hardware monitoring with real-time database interactions, making it highly suitable for IoT-based systems requiring immediate user feedback and secure data handling.

3.7 Server and Database

The backend of the system consists of a server and a MySQL database. The server handles requests from the mobile application, processes data, and communicates with the database. User data, including usernames, passwords, and email addresses, is securely stored in the database. To ensure the security and privacy of user information, robust column-level encryption and password hashing mechanisms are employed.

Column-level encryption is a method of encrypting data at the level of individual columns within a database table. This technique allows sensitive data in specific columns, such as credit card numbers, social security numbers, or personal health information, to be encrypted without affecting other columns. By encrypting data at the column level, organizations can achieve a higher level of security granularity. This means that even if someone gains unauthorized access to the database, the encrypted columns remain protected. Implementing column-level encryption typically involves using encryption algorithms like AES (Advanced Encryption Standard). The encrypted data is stored in its encrypted form and only decrypted when accessed by authorized users or applications with the appropriate decryption keys. This approach ensures that sensitive information is safeguarded from unauthorized access, both during storage and transmission.

And Password hashing is a process used to transform a plain text password into a fixed-size string of characters, which is typically a hash value. This process ensures that the original password cannot be easily retrieved from the hash. Hashing functions, such as bcrypt, SHA-256, or Argon2, are designed to be one-way functions, meaning they are computationally infeasible to reverse. When a user creates or updates a password, the password is hashed and the hash value is stored in the database, not the plain text password. When the user attempts to log in, the provided password is hashed again, and the resulting hash is compared to the stored hash. If they match, access is granted. This method protects user passwords from being exposed, even if the database is compromised. Additionally, modern hashing techniques often incorporate salting, which involves adding a random value to the password before hashing. This ensures that even identical passwords result in different hashes, further enhancing security. The detail of the database is shown in Figure 5.

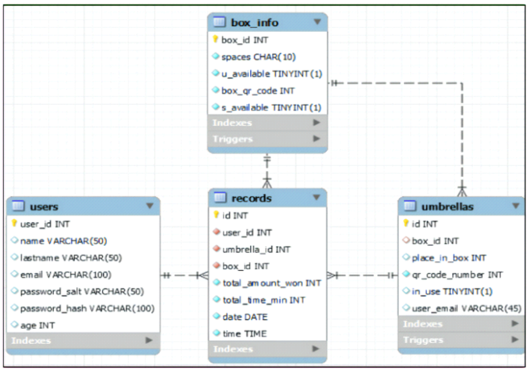


Fig. 5. Database diagram.

This database diagram represents a relational database schema designed to manage information related to users, umbrellas, boxes, and records of usage. The *users* table includes columns such as *user_id*, which is the primary key and a unique identifier for each user, *name* for the first name, *lastname* for the last name, *email* for the user's email address, *password_salt* for the salt used in password hashing, *password_hash* for the hashed password, and *age* for the user's age.

The *box_info* table includes *box_id* as the primary key, *spaces* for information about the spaces in the box, *u_available* to indicate umbrella availability (1 for available, 0 for not available), *box_qr_code* for the QR code associated with the box, and *s_avabile* to indicate slot availability (1 for available, 0 for not available).

The *records* table includes *id* as the primary key, *user_id* as a foreign key referencing *user_id* of *users* table, indicating the user associated with the record, *umbrella_id* as a foreign key referencing *id* of *umbrellas* table, indicating the umbrella involved, *box_id* as a foreign key referencing *box_id* of *box_info* table, indicating the box involved, *total_amount_won* for the total amount won by the user, *total_time_min* for the total usage time in minutes, *date* for the record date, and *time* for the record time.

The *umbrellas* table includes *id* as the primary key, *box_id* as a foreign key referencing *box_id* of *box_info* table, indicating the box in which the umbrella is stored, *place_in_box* for the slot number in the box, *qr_code_number* for the umbrella's QR code, *in_use* to indicate if the umbrella is in use (1 for in use, 0 for not in use), and *user_email* for the email of the user associated with the umbrella.

The database tracks the status and location of each umbrella and box, which is crucial for maintaining accurate and real-time information, allowing users to find available umbrellas and return them to the correct locations. When an umbrella is rented or returned, the system updates the status and location in the database accordingly. Indexes are set on primary keys and foreign keys to optimize database performance and ensure quick data retrieval. Although the diagram suggests the presence of triggers, it does not provide specifics; triggers typically enforce business rules or maintain data integrity by automatically invoking database procedures upon certain actions or events.

Ⅳ. Results

This section details the outcomes of integrating map features, implementing QR code scanner functionality, enhancing app performance, and utilizing the IoT capabilities of the box.

4.1 Map integration

The Google Maps JavaScript API provides a seamless integration for displaying real-time locations of users and umbrella boxes. This process begins with the installation of requisite libraries such as expo-task-manager, expo-location, and react-native-maps. These commands facilitate the incorporation of essential libraries, paving the way for the seamless integration of Google Maps functionality into our mobile app, 'Umber'. By following these steps, the system ensures a robust foundation for the development of an interactive and visually intuitive

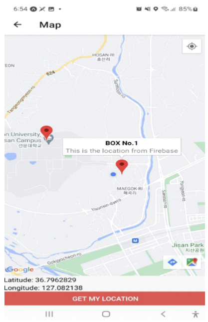


Fig. 6. Mapping User and Umbrella Box Locations in App.

map interface, providing real-time location insights. The result of the application is shown in Figure 6.

4.2 QR Code Scanner

To ensure optimal identification and interpretation by users, tailored QR codes are generated for each box and umbrella, with embedded specific text values for efficient storage and retrieval. The implementation of this QR code system necessitates camera permissions to enable users to scan the QR codes seamlessly. The QR scanning functionality is realized through the integration of JavaScript libraries, notably the jsQR library[12]. This library enables the device to scan QR codes using its camera, facilitating a streamlined process, as shown in Figure 7.

To initiate the rental process, scan the umbrella's QR code. Each QR code is stored in our database, allowing the application to recognize it and begin tracking the rental period. To complete the rental process, scan the QR code on the return box.

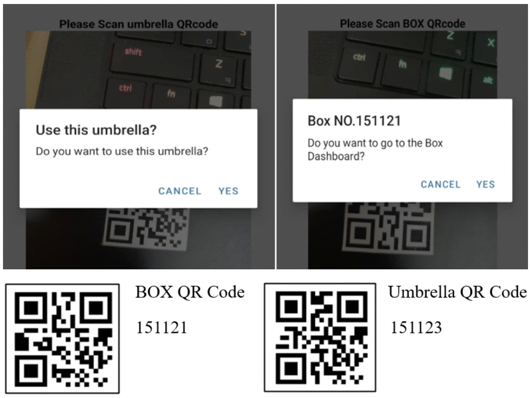


Fig. 7. QR code scanning function using jsQR library.

4.3 Box IoT Dashboard

Each umbrella in the system is equipped with a magnet that serves to activate the corresponding reed switch. Simultaneously, a GPS module, such as the neo-6, captures precise location data. This location information is then transmitted to Firebase through the integrated Wi-Fi module of the ESP32 microcontroller. The JavaScript code, interacting with Firebase, fetches real-time data to ascertain the status of each hole in the umbrella box—whether open or closed. Leveraging this information, the code dynamically adjusts the color representation for each hole, providing users with an intuitive visual cue.

During the project, a transition was made from employing *HardwareSerial* to *SoftwareSerial* due to its enhanced convenience in retrieving data from the GPS module. Adopting *SoftwareSerial* necessitates the installation of the *SoftwareSerial* library specifically tailored for ESP32. This adjustment streamlines the data retrieval process from the GPS, contributing to the overall efficiency and functionality of the system. Figure 8 shows the dashboard of the app. *A* displays the state when an umbrella is inserted, and B shows the state when there is no umbrella inside the box.

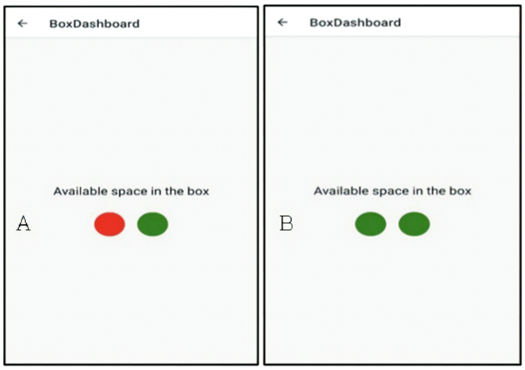


Fig. 8. Dashboard to show available space in the box.

4.4 Communication Performance

The performance of the system was evaluated in terms of latency, data throughput, and GPS accuracy to ensure seamless operation during the umbrella rental process as shown in Table 4.

The system's communication performance was evaluated based on key metrics such as latency, data throughput, and GPS accuracy. Latency, defined as the average time required for the GPS module to transmit location data to the server via the ESP32 and Firebase, was consistently low, averaging 150 ms. This ensures a smooth user experience by providing near real-time updates on umbrella box locations and statuses. The system demonstrated a data throughput of 500 bytes per second, which is sufficient for transmitting lightweight data such as QR codes and GPS coordinates. Additionally, the GPS module exhibited reliable accuracy, achieving a precision of ±5 meters in urban environments and ±10 meters in rural areas. This level of performance effectively supports the system's operational requirements, enabling users to locate umbrella boxes with ease while ensuring scalability and reliability. This concise addition is simple yet effective in providing quantitative evidence for the system's reliability.

Table 4. Communication Performance.

Metric	Value	Explanation
Latency	150ms	Ensures smooth real-time updates.
Throughput	500 bytes/s	Handles QR code and GPS data efficiently.
GPS Accuracy	±5meters (urban), ±10 meters (rural)	Reliable navigation; better in urban areas.

4.5 Server and Database Management

In the database, all umbrella QR codes, box states, and locations are systematically recorded based on user email. As users engage with umbrellas, the state of each umbrella in the database dynamically updates, facilitating accurate tracking of the umbrella's status throughout its usage. Figure 9 shows the example of

id	user_id	umbrella_id	box_id	total_amount_won	total_time_min	date	time
1	1	1	1	20	2	2023-12-11	15:00:01
2	1	1	1	70	7	2023-12-11	16:33:44
3	1	1	1	40	4	2023-12-11	16:38:31
4	1	1	1	40	4	2023-12-11	16:44:21
5	1	1	1	20	2	2023-12-11	16:46:24
6	1	1	1	50	5	2023-12-11	16:49:15
7	1	1	1	50	5	2023-12-11	16:49:15
8	1	1	1	20	2	2023-12-11	16:53:58
9	1	1	1	20	2	2023-12-11	16:54:03
10	1	1	1	30	3	2023-12-11	16:55:20
11	1	1	1	30	3	2023-12-11	16:55:24
12	1	1	1	30	3	2023-12-11	16:56:46
13	1	1	1	90	9	2023-12-11	16:59:23
14	1	1	1	30	3	2023-12-11	17:00:35
15	1	1	1	10	1	2023-12-11	17:02:36
16	1	1	1	10	1	2023-12-11	17:10:47

Fig. 9. Store all data, time and amount in DB

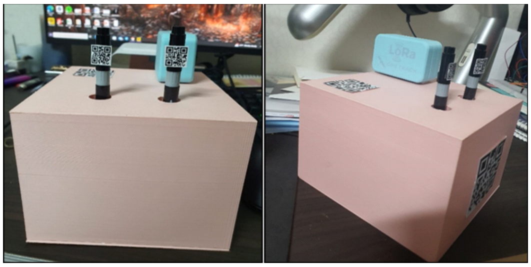


Fig. 10. Umbrella box prototype printed using a PLA 3D Printer.

all the data, time and amount in the database.

For this study, we created a prototype using 3D-printed PLA. The design includes a box to hold the umbrellas, with the GPS module positioned at the top to ensure optimal signal reception. Inside the box, we integrated the ESP32 microcontroller and red switches. Two pens serve as placeholders to represent umbrellas. Additionally, the ESP32 is housed in a separate, waterproof compartment to prevent any damage from moisture.

Ⅴ. Conclusions

This study has explored the integration of IoT technology into everyday objects, such as umbrellas, through the development of a smart rental service. Motivated by challenges faced in high-traffic urban areas, the proposed IoT-enabled system aims to enhance user convenience, operational efficiency, and environmental sustainability by reducing waste from misplaced or underutilized umbrellas.

Key contributions of this work include the design and implementation of an innovative IoT-based umbrella rental system, featuring a GPS tracking module, robust data management using Firebase, and ESP32 microcontrollers for seamless device communication. The findings demonstrate the system's potential to improve user experiences while addressing urban challenges, setting a foundation for future advancements.

However, practical implementation poses challenges, particularly in constructing a durable, weather-resistant, and cost-effective umbrella box. Factors such as theft prevention, ease of maintenance, and aesthetic appeal also require careful consideration. Future research will focus on developing modular hardware designs and guidelines to overcome these limitations, ensuring the system is scalable and robust for real-world use.

Furthermore, detection accuracy, communication performance, and power consumption remain unexplored aspects of the system. The impact of these factors, including the distance from access points and hardware constraints, will be studied in future work to optimize system performance.

By emphasizing the practicality of IoT in consumer applications, this study encourages broader adoption and improvement of IoT-enabled technologies, paving the way for future innovations.

References

[1] M. N. Khan, M. R. H. Ontor, T. Ahmed Shuvo, and N. K. A. Rahman, "The internet of things (IoT): Applications, investments, and challenges for enterprises," *Int. J. Multidisciplinary Research (IJFMR)*, vol. 6, no. 1, pp. 1-10, 2024. (https://doi.org/gzwpkz)

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Syst.*, vol. 29, no. 7, pp. 1645-1660, Sep. 2013. (https://doi.org/10.1016/j.future.2013.01.010)

[3] R. Hassan, "Internet of things (IoT) in smart cities: Enhancing urban living through technology," *J. Eng. Technol.*, vol. 5, no. 1, pp. 1-16, 2023.

[4] J.-H. Moon, B. Peng, J.-H. Kwon, and T.-K. Kim, "Implementation of smart umbrella stand based on IoT," *J. Internet of Things and Convergence*, vol. 9, no. 1, pp. 57-64, Feb. 2023. (https://doi.org/10.20465/KIOTS.2023.9.1.057)

[5] G. Yang, et al., "A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," *IEEE Trans. Industrial Inf.*, vol. 10, no. 4, pp. 2180-2191, Nov. 2014. (https://doi.org/10.1109/TII.2014.2307795)

[6] J.-H. Kim and G. I. Kim, “Extended Kalman filter based mobile robot localization in indoor fire environments,” *Int. J. Mechanical Eng. and Robotics Res.*, vol. 5, no. 1, 2016.

[7] S. Hu, “Online map application development using google maps API, SQL database, and ASP.NET,” *Int. J. Inf. and Commun. Technol. Res.*, vol. 3, no. 3, 2013.

[8] u-blox, “NEO-6 u-blox 6 GPS modules.”

[9] M. P. Peterson, “International perspectives on maps and the Internet: An introduction,” in *Int. Perspectives on Maps and the Internet*, pp. 3-10, Berlin, Germany, Springer, 2008. (https://doi.org/10.1007/978-3-540-72029-4_1)

[10] S. Udell, *Beginning Google Maps Mashups with Mapplets, KML, and GeoRSS*, New York, NY: Apress, 2009.

[11] J. Sturm, et al., “Model learning and real-time tracking using multi-resolution surfel maps,” in *Proc. 2013 AAAI Conf. Artificial Intell.*, 2013.

[12] M. J. B. Ghamari, A. Ghanbarzadeh, and S. Hakak, “Google maps JavaScript API,” 2016.

Mosalla Sophia

Mar. 2021-Current : B.Eng., Div. of Computer Science and Engineering, Sun Moon University.

<Research Interests> AIoT, Mobile IoT, Big data, Signal processing, Deep Learning.

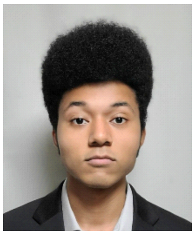
[ORCID:0009-0002-2541-7913]

Akashi Kazunari

Aug. 2024 : B.Eng., Division of Computer Science and Engineering, Sun Moon University.

<Research Interests> AI, Web development, Cybersecurity, IoT, Computer Vision.

Alao Chunghwan



Mar. 2021-Current : B.Eng., Division of Computer Science and Engineering, Sun Moon University.

<Research Interests> AI, IoT, Blockchain, Deep Learning.

[ORCID:0009-0001-3198-5696]

Jeong-Dong Kim

Feb. 2008 : M.S. degree, Computer Science and Engineering, Korea University.

Aug. 2012 : Ph.D. degree, Computer Science and Engineering, Korea University.

Mar. 2016-Current : Associate Professor in the Department of Computer Science and Engineering, Sun Moon University.

<Research Interests> Deep Learning, Healthcare, Software & Data Engineering, and Bioinformatics.

[ORCID:0000-0002-5113-221X]

Taek Lee

Sep. 2010-Feb. 2011 : Visiting Researchers, Department of Computer Science, Hong Kong University of Science and Technology.

Mar. 2012-Aug. 2017 : Lecturer, Seoul Women's University, Korea University, Duksung Women's University, Sungshin Women's University.

Mar. 2016-Aug.2017 : Research Professor, Research Division, Korea University.

Sep. 2017-Feb. 2023 : Assistant Professor, Department of Convergence Security Engineering, Sungshin Women's University.

Mar. 2023-Current : Assistant Professor, Division of Computer Science and Engineering, Sun Moon University.

<Research Interest> information security, software engineering, ICT & healthcare convergence research, wearable device and sensor data fusion, machine learning and data mining, probabilistic risk estimation and simulation

[ORCID:0000-0003-2277-8211]

Jung-Been Lee

Feb. 2011 : M.S. degree, Department of Computer and Radio Communications Engineering, Korea University.

Aug. 2020 : Ph.D. degree, Department of Computer Science and Engineering, Korea University.

Sept. 2020-Feb. 2023 : Research Prof., Chronobiology Institute, Korea University.

Mar. 2023-Current : Assistant Prof., Division of Computer Science and Engineering, Sun Moon University.

<Research Interests> Digital healthcare informatics, Sensor data mining, AI, Software Engineering

[ORCID:0000-0002-8208-0387]