

Internet of Things Driven Smart Parking System Design and Implementation (IoTSpS)

Nesnelerin İnterneti Tabanlı Akıllı Otopark Sistem Tasarımı ve Uygulaması (IoTSpS)

Abdullah ALAWAD¹, Vedat MARTTİN^{2*}

Abstract

The IoT-driven Smart Parking System is designed to enhance the efficiency of parking operations in metropolitan areas and high-traffic zones. By enabling drivers to swiftly identify available parking spaces, the system not only saves time but also alleviates traffic congestion. Furthermore, it promotes the optimal utilization of parking infrastructure, contributing to improved urban management. A dedicated IoT-integrated mobile application streamlines the process of locating vacant parking spots, thereby reducing search durations and enhancing user convenience. This initiative aims to mitigate urban parking challenges while fostering the development of sustainable transportation solutions. Within the scope of this study, a prototype system was developed, integrating hardware, embedded software, and a mobile application. Infrared (IR) sensors were employed for detecting parking space occupancy, while servo motors facilitated barrier control at entry and exit points. Arduino Mega served as the primary hardware controller, whereas NodeMCU, equipped with a Wi-Fi module, was utilized for data transmission via Firebase. The mobile application, developed using Flutter, provides functionalities such as user authentication, parking space monitoring, and reservation management. Additionally, the system allows for automated fee calculation based on usage duration, enabling payments through the application. To validate the system's performance, various operational scenarios were tested, successfully demonstrating the functionality of both the prototype and the mobile application.

Keywords: Internet of things (IoT), Mobile applications, Parking system.

Öz

Nesnelerin İnterneti (IoT) tabanlı Akıllı Park Sistemi, büyük şehirlerde ve yoğun bölgelerde park süreçlerinin daha verimli hale getirilmesini ve iyileştirilmesini hedeflemektedir. Sürücülere uygun park alanlarını hızlı bir şekilde bulma imkânı sunarak hem zaman tasarrufu sağlamakta hem de trafik yoğunluğunu azaltmaya katkıda bulunmaktadır. Bunun yanı sıra, sistem park alanlarının etkin kullanımını teşvik ederek kentsel yönetimin optimizasyonuna olanak tanımaktadır. Kullanıcılar için özel olarak geliştirilen IoT destekli mobil uygulama, boş park alanlarının tespit edilmesini hızlandırarak arama sürelerini en aza indirirken kullanım kolaylığı sağlamaktadır. Bu çalışma, kentsel park sorunlarını çözmeyi ve sürdürülebilir ulaşım sistemlerini teşvik etmeyi amaçlamaktadır. Bu bağlamda, IoT tabanlı Akıllı Park Sistemi, sürücüler için günlük yaşamı kolaylaştırırken kentsel trafik yönetimi ve çevresel sürdürülebilirliğin geliştirilmesi açısından önemli bir adım olarak değerlendirilmektedir. Bu çalışmada, donanım, gömülü yazılım ve mobil uygulamanın entegre edildiği bir prototip geliştirilmiştir. Park alanı tespiti için IR sensörleri kullanılmış, giriş ve çıkış bariyerlerinin kontrolü için servo motorlar entegre edilmiştir. Donanım kontrolü, Arduino Mega ile sağlanırken, Wi-Fi modülüne sahip NodeMCU kullanılarak Firebase ile veri iletişimi gerçekleştirilmiştir. Mobil uygulama, kullanıcı kimlik doğrulaması ve park yeri yönetimi işlevlerini içerecek şekilde Flutter ile geliştirilmiştir. Uygulama, park yeri uygunluk durumunu ve rezervasyon işlemlerini yönetmekte olup, kullanım süresine bağlı olarak ücretlendirme işlemleri mobil uygulama üzerinden gerçekleştirilebilmektedir. Ayrıca, sistemin işleyişini değerlendirmek amacıyla farklı senaryolar geliştirilmiş, bu senaryolar aracılığıyla hem prototipin hem de mobil uygulamanın işlevselliği başarıyla test edilmiştir.

Anahtar Kelimeler: Nesnelerin İnterneti (IoT), Mobil uygulamalar, Park sistemi.

^{1,2}Bilecik Şeyh Edebali University, Department of Computer Engineering, Engineering Faculty, Bilecik, Türkiye

*Corresponding Author/Sorumlu Yazar: vedat.marttin@bilecik.edu.tr

1. Introduction

Considering the increase in the world population, it is estimated that the number of cars will double to 2 billion by 2040. Therefore, it is clear that the need for parking spaces will increase (Smith, 2016). This situation causes a number of problems, such as drivers spending more time looking for a parking space and having a negative impact on traffic congestion and air pollution in closed parking areas (Li et al., 2014), (Vuković et al., 2014). Smart Parking Systems (SPS), proposed to solve such problems, offer a solution that reduces traffic congestion and emissions by reducing the time to search for a parking space (Surpris et al., 2014), (Zumma et al., 2023).

SPS optimize parking management in modern cities, enhancing user experience and providing many significant benefits. Users can quickly find and reserve available parking spaces through mobile applications, allowing them to park without wasting time. Because of this, the time spent searching for parking spots, especially in high-traffic areas, is significantly reduced, simplifying users' daily lives. SPS regulate city traffic by reducing congestion and improving overall efficiency. Thus, unnecessary vehicle circulation to find parking is prevented, leading to smoother traffic flow and more efficient urban transportation. More effective use of parking spaces allows more vehicles to park in the same area. Therefore, parking capacity increases and parking shortages in the city are alleviated. SPS minimize fuel consumption and air pollution by reducing unnecessary vehicle circulation. Because of this, they contribute to reducing environmental impacts and promote a more sustainable urban life. SPS also play a crucial role in enhancing security by better monitoring parking areas and increasing safety measures. All these benefits combined make urban life safer, more convenient, and more efficient. Because of this, SPS become an indispensable part of modern cities.

The main hypothesis of this study is: An IoT-based smart parking system developed using IR sensors, NodeMCU, and Firebase technologies can offer a successful solution that is more cost-effective than existing systems, provides fire safety integration, and improves user experience with cross-platform mobile application support.

When we look at the literature studies, within the scope of the keywords used in the article, SPS concept of previous studies has generally been about determining the availability of space. Microcontrollers such as Arduino, Raspberry and NodeMCU were used to display parking space availability via sensors' LCD screens and to control and communicate with the sensors. It helps the user to find suitable parking spaces faster through web pages or mobile applications. The studies carried out within the scope of the keywords of the article below are listed.

In their study, Khanna and Anand (2016) performed change control with Raspberry Pi by using ultrasonic sensors for each parking space. The sensors are connected to a system of lights that turn green, red or yellow depending on parking space availability. A cloud-based database was used and

supported by a mobile application to keep all records regarding parking areas and end users accessing the system.

In the study by Aydın et al (2017), the focus was on finding the nearest empty parking space using an algorithm-oriented and simulation-based genetic algorithm.

In the study by Qadir et al (2018), a system with ZigBee for energy efficiency, IR sensors, and GSM module for SMS notification was proposed for the parking system.

In their study, Rahman and Bhounik (2019) developed a smart parking system that uses ultrasonic sensors placed in each parking space to detect parking occupancy. The system also managed the gate at the entrance to the parking lot using Arduino Mega 2560. Information about the parking situation is linked to the website and is updated when the situation changes.

Taylor et al. study (2021) uses the traffic congestion status of streets to inform users about parking availability. Thanks to the Google API integrated into the system, in addition to the recommended route that prevents traffic congestion, the location of the nearest parking lot is also presented according to their current location.

In the study of Allbadi et al. (2021), the identity cards of the users can be read by Radio Frequency Identification (RFID) at the parking gate from the proposed system. The sensors are connected to the Arduino Mega 2560, a programmable microcontroller with digital input and output pins. To transmit the data to the cloud, the exact location and number of available parking spots are obtained using the NodeMCU ESP8266 development board. The proposed system is compatible with Android Mobile Applications.

In the study proposed by Ahlam-Rauf et al. (2023), the IoT-based Smart Parking System consists of components such as Arduino Uno, Raspberry Pi 4 Model B, 74HC595 Multishift Register, HCSR04 Ultrasonic Sensor, RGB LED Lights, a resistor and an SG90 Micro Servo Motor. Flutter application without working the occupancy status of the parking area is shown via the interface. In the study, the prototype design was tried to be explained only with diagrams. The working of the developed prototype was not shown.

In this study by P. Kumar et al. (2023), a smart parking system model containing Arduino, ESP8266 and ultrasonic sensors is proposed. Arduino transmits the processed data (occupancy status) to WampServer via HTTP requests or other communication protocols. This information provides the status (occupied or empty) of each parking space. Within the scope of the literature study, the information given about the general operation of SPS attracts attention in the survey studies listed.

In this study by Gagan and Ravi (2024), a smart outdoor parking system that takes advantage of IoT technology, especially using embedded controllers (NodeMCU) and infrared (IR) sensors, is proposed. Entrance and exit gate management is also used in the study, allowing only users who have

reserved the parking space to reserve parking spaces and go to their destination. Android mobile application has been developed for the application communicates with firebase.

In this study by Sharmila et al. (2024), a Car Parking System prototype is presented with hardware tools such as IR sensor and servo motor, using Arduino Uno and IoT-connected mobile applications. Drivers can see whether parking spaces are available on the LCD screen before reaching the parking area. The mobile application part and user authorization are missing in the study.

This study by Visputa et al (2025), focusing on preventing unauthorized parking through the integration of IoT and deep learning (R-CNN and YOLO), presents an image processing-based approach.

In the study by Tomar et al (2025), an AI-focused system is proposed that detects parking spaces using YOLOv8 and Mask R-CNN algorithms and predicts demand using LSTM networks and Random Forest.

A comparison of the characteristics of similar studies is shown in Table 1.

In this study by Channamallu et al. (2023), an IoT-based cloud integrated smart parking system is presented. The proposed Smart Parking system consists of an IoT module used to monitor the occupancy status of each parking space. It allows the user to check the availability of parking space and make a reservation through the mobile application. This study by Alam et al. (2023) examines IoT- oriented smart parking management systems. Various approaches to the parking system, related technologies, commonly used components, communication standards, and related system security issues were evaluated. In their review, Channamallu et al. (2026) examined 124 academic papers and identified key limitations in smart parking systems. They proposed AI-powered integrated sensor frameworks for sensor technology, network architecture for communication networks, and blockchain technology for data management. Ndunda and Nicolas (2026) examined 25 smart parking applications worldwide and evaluated real-world applications in cities. They emphasized that sensor robustness, changes in political will, and funding issues affect the sustainability of these systems. Studies Channamallu et al.(2023,2026) ,Alam et al. (2023) and Ndunda and Nicolas (2026) are studies that describe the system in general terms but cannot classify it according to the features specified in the table.

Table 1. Comparison of similar studies

| Authors of the Study | Equipment Used in the Study | Microcontrollers | Mobile Application Status | Mockup or Prototype | Platform | Key Feature |
|-------------------------|---------------------------------|------------------------|---------------------------|---------------------|-------------|-----------------------------------|
| Khanna and Anand (2016) | Ultrasonic Sensors, Servo Motor | ESP8266 , Raspberry pi | Yes (Apache Cordova) | No | Web/Android | Cloud-based database |
| Aydin et al. (2017) | Navigation sensors | Not specified | No | No | Smart Phone | Genetic optimization +reservation |

| | | | | | | |
|----------------------------|--|------------------------------------|----------------------|-----------------|---------------|--|
| Qadir et al. (2018) | IR Sensor, GSM module | Arduino, ZigBee | Yes (SMS/App) | No | Android | ZigBee energy efficiency |
| Rahman and Bhoulmik (2019) | Ultrasonic Sensor | Servo Motor Arduino Uno | No | Yes (Mockup) | Not specified | Web-based information |
| Taylor et al. (2021) | Raspberry pi, Ultrasonic sensor | Raspberry pi, ESP8266 | Yes (Android Studio) | No | Android | Google API integration |
| Allbadi et al. (2021) | Servo motor, Ultrasonic sensor, Lcd, Rfid | Arduino Mega 2560, ESP8266 | Yes (Android Studio) | No | Android | RFID authentication |
| Ahlan-Rauf et al. (2023) | Servo motor, Ultrasonic sensor | Arduino Uno, Raspberry pi | Yes (Flutter) | No | IoS/Android | Diagrams only, no working prototype |
| P.Kumar et al. (2023) | Ultrasonic Sensor, LED, Servo Motor | Arduino Uno, ESP8266 | Yes (Apache Cordova) | No | Web/Android | HTTP with WampServer |
| Gagan and Ravi (2024) | Rfid, IR Sensor | ESP8266 | Yes (Android Studio) | Yes (Prototype) | Android | Entry only for reserved users |
| Sharmila et al. (2024) | IR Sensor, Servo Motor, Lcd | Arduino Uno | No | Yes (Prototype) | Not specified | LCD screen information |
| Vispute et al. (2025) | Ultrasonic Sensor, ESP32-CAM, YOLO | Not specified | No | No | Not specified | Unauthorized parking prevention with YOLO |
| Tomar et al. (2025) | Camera (YOLOv8) | Not specified | No | No | Not specified | AI-based demand prediction |
| This Study | IR Sensor, LCD, Servo Motors, Flame Sensor | Arduino Mega 2560, NodeMCU-ESP8266 | Yes (Flutter) | Yes (Mockup) | IoS/Android | Fire safety + cross-platform+ Cloud-based database |

Although the hardware materials used in the proposed SPS prototype are similar to other similar studies, the construction stages of the prototype, its explanation and the ease of use of the mobile application differ. Its different parts can be listed as follows:

- The IoTSpS prototype was designed as a model.
- In the IoTSpS model, parking lot entry-exit control is provided according to different scenarios. Vehicle location status is instantly queried via the Cloud.
- Vehicle exit evacuation is provided with the fire detection sensor (not found in the literature).
- IoTSpS mobile application is compatible for both IOS and Android.
- IoTSpS offers user information controls and subscription and pricing information from the mobile application.

By integrating with the mobile application, IoTSpS allows users to access parking information remotely as well as find parking spaces efficiently and save time. In addition, thanks to the mobile

application we developed, users can view the status of the parking spaces before arriving at the parking lot and choose the spot they want. These features significantly improve the user experience and effectively streamline parking management.

The first part of the article is the introduction, the next part is a general explanation about how the system works. The second part consists of an explanation of the hardware used in the construction of the system and the software information used. The third part describes the IoTSpS model design. In the fourth part, IoTSpS mobile application interfaces and user controls are explained. The fifth part describes sample scenario studies. In the last section conclusion is included.

2. Materials and Methods

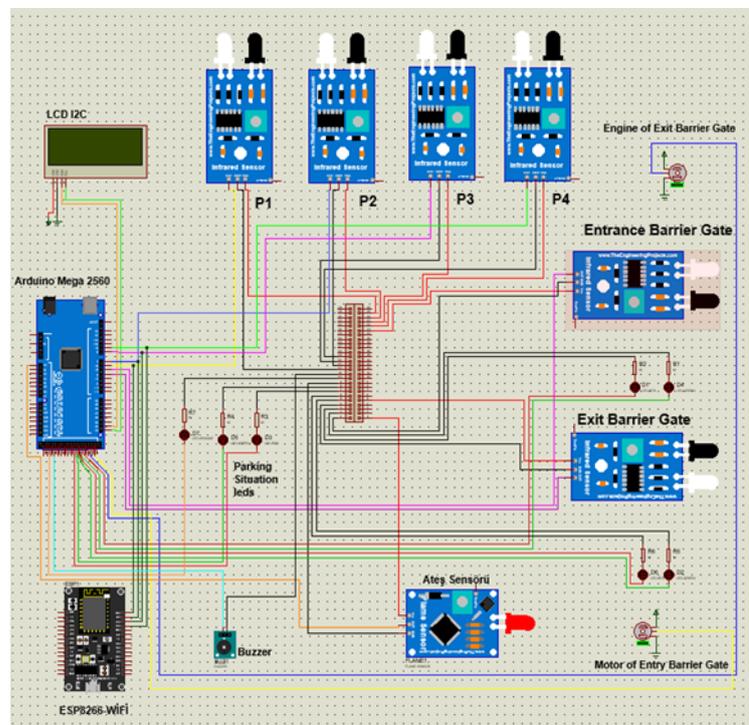


Figure 1. Structure of the System

The smart parking system consists of two main components: hardware and software. The structure in which the hardware part of the system is designed is shown in Figure 1. To develop the hardware part, programming based on C/C++ languages was done using Arduino IDE. Various libraries need to be installed to program and control hardware components in this language. For example, to communicate with Firebase on the ESP8266 microcontroller, the ESP8266 Firebase library must be used. This library provides functions such as sending and storing data to Firebase.

Development for the mobile application part was done using Flutter. Flutter is a tool used to develop modern and cross-platform applications. Dart programming language is used to write Flutter applications. Dart is the basic programming language used in Flutter applications and provides

integration with Flutter. In this way, applications can be developed for both Android and iOS platforms with a single code base.

2.1. Technologies Used to Develop the Hardware Part of the System

Hardware elements are shown in Figure.2 to used to develop the hardware part of the smart parking system. Arduino Integrated Development Environment (IDE) (Figure.2a) is developed for the Arduino platform and allows users to write, upload, and test code on Arduino boards(Arduino, 2015). It features a user-friendly text editor with syntax highlighting, making the process of writing and editing code easier.

Arduino IDE compiles the written code and uploads it to the board. The compilation process converts the code into machine language, allowing the Arduino board to execute it. Additionally, Arduino IDE includes pre-written libraries for sensors, motors, displays, and other components. These libraries facilitate working with specific hardware components and speed up the coding process.

Among the components used in the system is the Arduino MEGA 2560 R3, a version of the Arduino MEGA (Kumar et al., 2015) development board suitable for complex projects due to its extensive GPIO interface. The "R3" version provides a broad GPIO interface with more digital and analog input/output pins, enabling expanded connectivity and control capabilities. To transmit sensor data connected to Arduino to Firebase, the NodeMCU-ESP8266 WiFi module(Figure.2b) has been used. NodeMCU- ESP8266 (Parihar, 2019) is a development board used in IoT projects, based on the ESP8266 microcontroller module, facilitating WiFi connectivity and programmability through environments like Arduino IDE. An IR receiver-transmitter sensor module(Fig.2c) is employed to detect the presence of objects using infrared light. A flame detector sensor(Figure.2d) is utilized for fire detection, converting the presence of flames or fire into electrical signals, playing a significant role in fire detection systems.A servo motor(Figure.2e) is used to control entry and exit gates, designed to rotate to specific positions or angles (SG90 model typically rotates up to 180 degrees) (Pasquo, 2020). For users to view parking spaces at the entrance, a 16x2 LCD screen(Figure.2f) is used; these screens reduce connection complexity by connecting to microcontrollers via the I2C protocol (Clary, 2015) preferred in various embedded systems. LED lights are employed for illumination and signaling purposes; these semiconductor components are chosen for their energy efficiency and durability in various applications such as lighting, indicators, and displays(Figure.2g).An fire sensor alerts users with auditory signals upon detecting fire, utilizing a buzzer(Figure.2h) (Manurung et al., 2021) to convert electrical energy into mechanical vibrations for

audio alerts in various electronic devices and applications. Additionally a connection cable shown in Figure.2i.

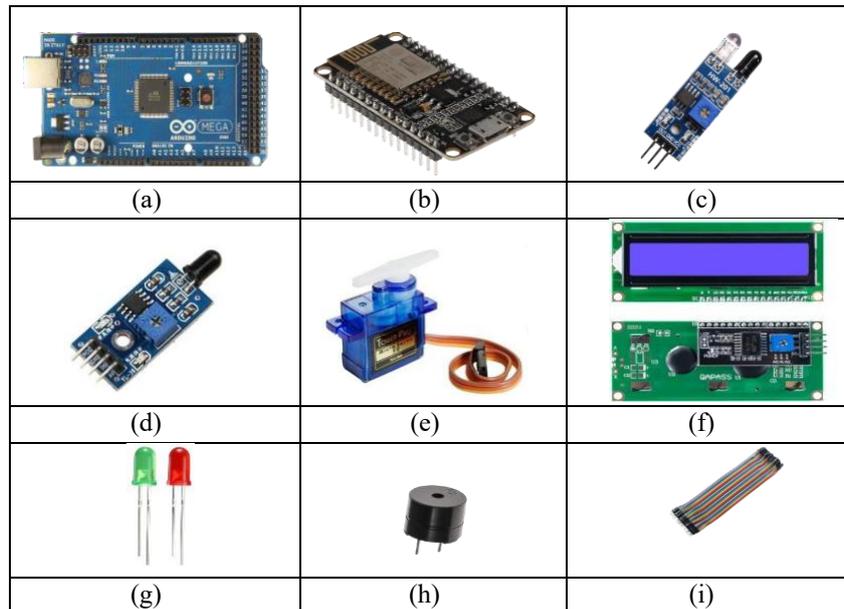


Figure 2. Hardware Elements Used in the System a) Arduino MEGA 2560 R3 b) NodeMCUESP8266 c) IR receiver-transmitter sensor module d) Flame sensor e) Servo motor SG90 model f) LCD display(16x2) g) LED (Light Emitting Diode) h) Buzzer i) Jumper wires

2.2. Technologies Used to Develop the Software Part of the System

The mobile application developed for users in the smart parking system is created using Android Studio IDE (Satyaputra, 2016) for developing Android applications. Provided by Google, this tool is designed to facilitate the coding, testing, and debugging processes of Android applications. Built on the IntelliJ IDEA platform, Android Studio supports programming languages such as Java, Kotlin, and C++.

The mobile application developed for the smart parking system uses Firebase as its database. Firebase is a platform offering a wide range of services (Chougale, 2021). In this system, data from sensors is instantly recorded in Firebase Realtime Database (Maulana, 2020) through the mobile application, allowing users to view this data in real-time. Additionally, Firebase Authentication is used for user login and account creation, ensuring secure storage of user data on Firebase (Pramono and Javista, 2021). Among Firebase's other features are Firebase Hosting, which securely hosts web applications, Firebase Cloud Messaging (FCM) for sending notifications to applications, Firebase Analytics for tracking and analyzing user behavior, and Firebase Crashlytics for monitoring and reporting application errors. Firebase Cloud Functions enable serverless backend functions,

expanding the system's functionality. This allows the smart parking system to manage data securely and optimize user experience effectively.

The mobile application in the system has been developed using Flutter. Flutter is a user interface (UI) framework developed by Google for creating cross-platform mobile applications. It is based on the Dart programming language and aims to create fast, beautiful, and impressive user interfaces. Flutter allows you to develop applications for various platforms such as iOS, Android, and web using a single codebase, which accelerates development processes and reduces costs. With its widget-based architecture, Flutter facilitates creating reusable and flexible UI components. Additionally, it provides fast error correction, hot reload capabilities, and extensive library support, simplifying the development process. Flutter has become a preferred framework among startups, independent developers, and large companies, and it is used by many popular applications.

Dart is a programming language developed by Google. It offers broad platform support, making it suitable for web, mobile, desktop, and IoT applications. Dart supports object-oriented programming (OOP) paradigms and enables modular and reusable code writing (Bracha, 2015). It features a strong type system that allows for early error detection and offers fast execution on the Dart VM. Dart can also be compiled to JavaScript, making it usable on web browsers and mobile devices. With support for asynchronous programming, it ensures more responsive applications. Particularly when used with frameworks like Flutter, Dart is a powerful tool for developing modern and effective applications.

3. IoT Based Parking System Model

The IoT based parking system model section is presented in sub-sections covering hardware and application.

3.1. Hardware Part of the System

In Figure.3, the IoT-based Parking System Model features four parking spaces, entry and exit gates, and a control cabin. Inside the cabin, there is an LCD screen, a fire sensor, LEDs, and sensors for the entry and exit gates. These sensors are strategically placed to detect vehicles. All electrical connections for these components are made underneath the model. The system setup involves connecting all components to predefined pins on an Arduino board.



Figure 3. IoT Based Parking System Model

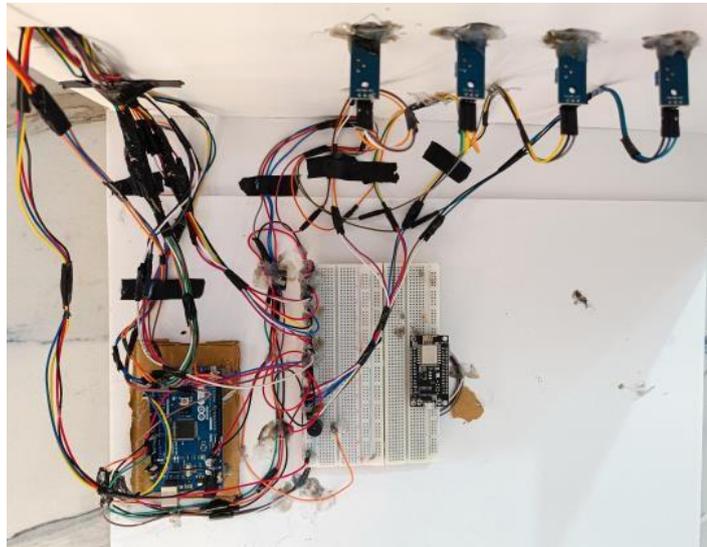


Figure 4. Hardware Connections

In Figure 4, shows the lower section of the model with electrical connections for all hardware elements. Power from Arduino is supplied to the breadboard, ensuring isolation between positive and negative terminals of sensors. Each sensor's output terminals are connected to Arduino pins as defined in the code. The ESP8266 WiFi module connects to Arduino for transmitting parking space statuses to a database, enabling real-time updates.

3.2. Application Part of the System

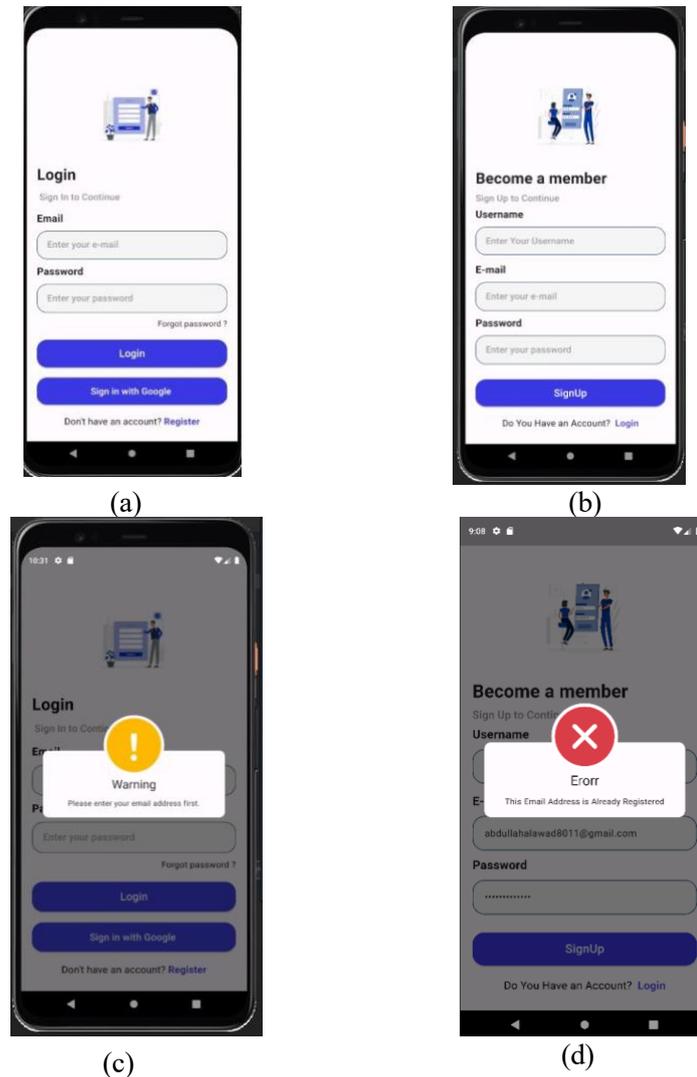


Figure 5. Parking Area Reserve Control

Figure.5 illustrates the interface elements for the mobile application's login and registration processes. Figure. 5a) shows the Login Page where users must log in with valid credentials; if credentials are incorrect, login will not proceed. Users can reset forgotten passwords via the "Forgot Password?" link or navigate to the registration page by clicking "Sign Up" if they lack an account. Figure. 5b) displays the Registration Page for creating new accounts, designed similarly to the Login Page. Figure. 5c) highlights the email control during login; if no email is entered, a warning "Enter your e-mail address" will appear. Figure. 5d) shows the registration email control where an error "This e-mail address is already registered" will be displayed if the email is already in use.

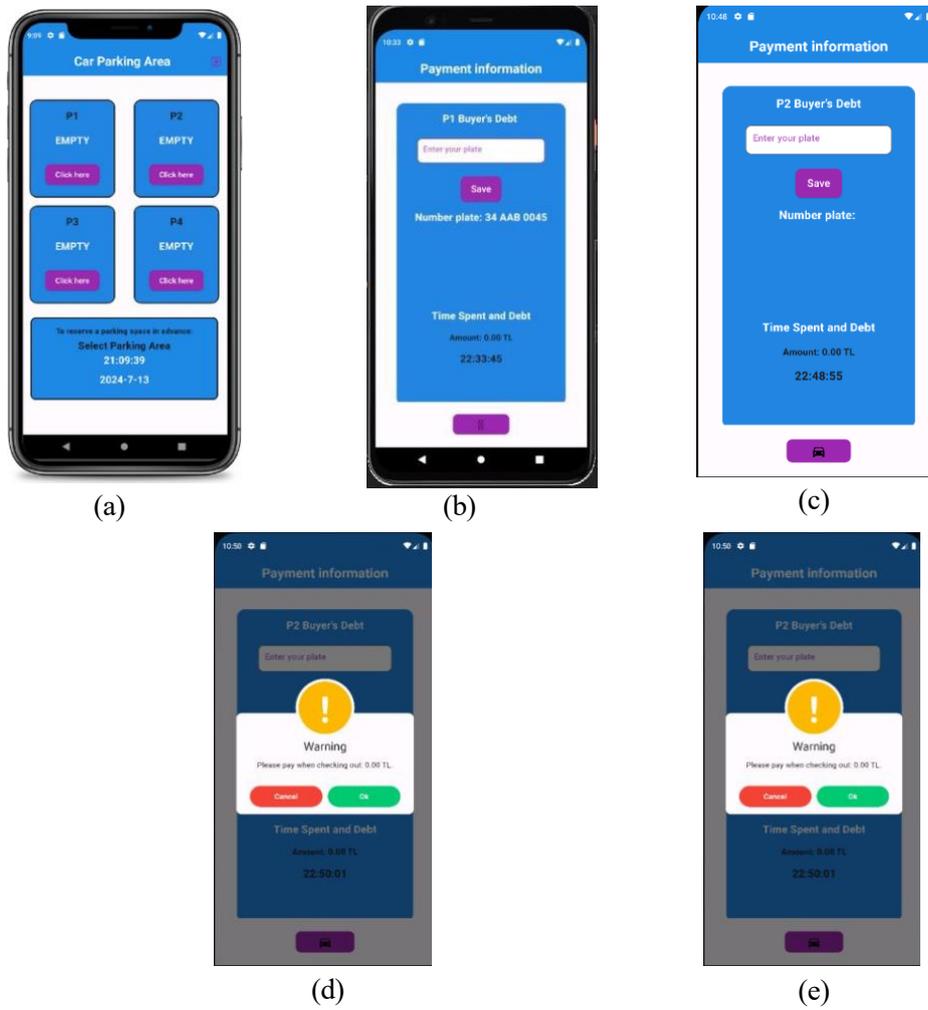


Figure 6. Home Page Interface and Parking Area Reserve Control

Figure. 6a) shows the Home Page interface of the mobile application, where users can view both available and occupied parking spaces, reserve spots in advance, and see hourly fee calculations, which enhances parking efficiency and saves time. Figure. 6b) details the park area interface, where users must enter their license plate after selecting a parking space to initiate fee calculation. If the license plate is not entered and "Save" is clicked, a warning message is displayed. All fee and plate information is updated in real-time in Firebase Firestore, and users receive payment details upon exit, with the parking space then marked as occupied. Figure. 6 presents the reservation control for parking areas. When reserving a space in advance, users are prompted with an interface where they must enter their license plate to start the fee calculation (Figure. 6c). If the plate is not entered and "Save" is clicked, a warning message "Please enter the license plate first" appears (Figure. 6d). Additionally, if the user attempts to leave after booking, a warning is displayed showing any outstanding fees and the amount due (Figure. 6e).

4. System Scenarios

The IoT-based parking system provides efficient entry, exit and occupancy management using sensors, servo motors and mobile application. The system informs users with real-time data updates, LCD screens and LED indicators, and shows the occupancy status of parking spaces. In case of fire, sensors warn with orange LED and buzzer, thus optimizing car park management and security.

4.1. Senario 1



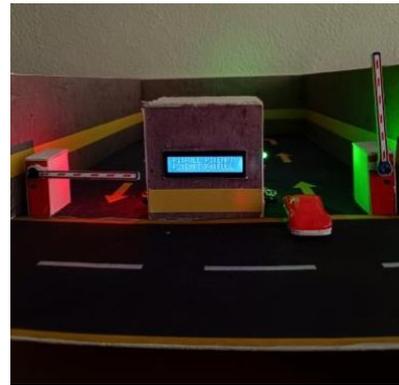
Figure 7. Scenario 1

Figure. 7 shows Scenario 1 that the situation where the entrance and exit doors are open. There are vehicles in the parking areas p1 and p4 in the parking lot (Figure. 7a). At the same time, a vehicle enters the parking lot at the entrance and a vehicle exits at the exit. The status of the vehicles parked in the parking lot is visible in the mobile application developed for the parking lot (Figure.7b). The LCD screen shows which area is empty or full(Figure.7c). The data coming from the sensors in the parking lot contains information that the p1 and p4 areas are full (Figure.7d). This data is updated in Realtime Firebase in real time in the mobile application and displayed to the user.

4.2. Senario 2



(a)



(b)

```

if (!lir_degeri1) {
  if (lir_p1 || lir_p2 || lir_p3 || lir_p4) {
    // Only open the entrance gate when the parking space is empty
    Giris_motor.write(aci_deger2);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, HIGH);
    delay(1000);
  } else {
    // If the parking area is full,
    // do not open the entrance door;
    // If a space is empty, open it
    Giris_motor.write(aci_deger1);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW);
  }
} else {
  Giris_motor.write(aci_deger1);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, LOW);
}

```

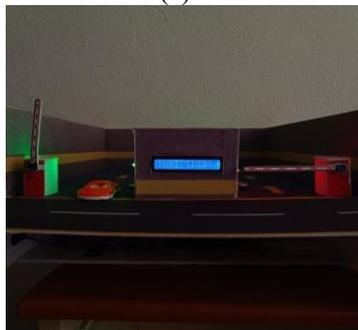
(c)

```

// Control of the servo motor at the exit door
digitalWrite(LED1, HIGH);
if (!lir_degeri2) {
  Cikis_motor.write(aci_deger1);
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, HIGH);
  delay(1000);
} else {
  Cikis_motor.write(aci_deger2);
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, LOW);
}

```

(d)



(e)

```

// Control of the servo motor at the exit door
digitalWrite(LED1, HIGH);
if (lir_degeri2) {
  Cikis_motor.write(aci_deger1);
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, HIGH);
  delay(1000);
} else {
  Cikis_motor.write(aci_deger2);
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, LOW);
}

```

(f)

Figure 8. Scenario 2

Figure. 8 illustrates Scenario 2, where the IR sensors at the parking system's entrance and exit doors detect vehicles, signaling the doors to open as vehicles enter or leave. The LCD screen displays available parking spaces, and the LEDs glow green indicating the doors are open (Figure. 8a). Figure. 8b shows the entrance door opens only when a vehicle is entering. The Arduino IDE interface depicted in Figure. 8c and Figure. 8d controls the servo motors for both doors, ensuring they open upon vehicle detection. In Scenario 2, shown in Figure. 8, the exit door opens when the IR sensor detects a vehicle, turning off the red LED and lighting up the green LED (Figure. 8e). The Arduino code shown in Figure. 8f confirms that the exit door will open upon vehicle detection.

4.3. Senario 3

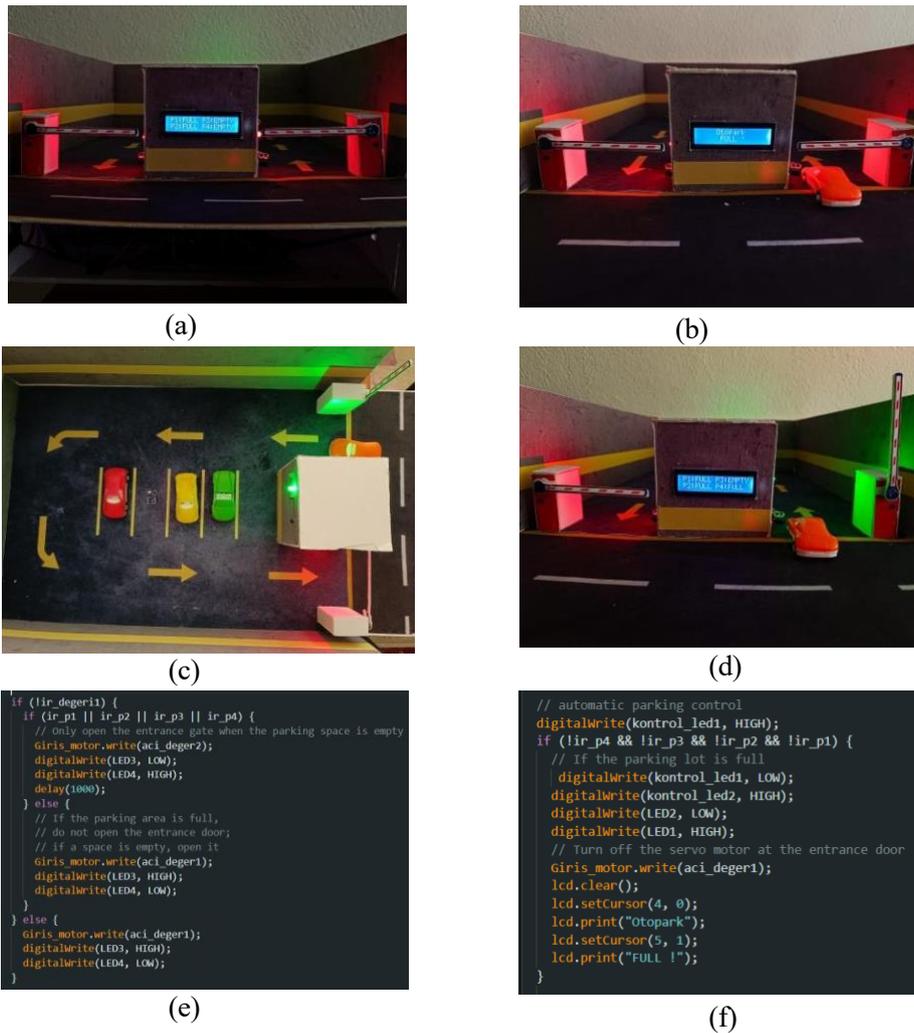


Figure 9. Scenario 3

Figure.9 shows Scenario 3. It is shown that the entrance and exit doors in the car park system are closed, and the LEDs there are glowing red because the entrance and exit doors of the car park are closed(Figure.9a). As seen on the LCD screen, when the parking lot is full and the sensor at the entrance door detects a vehicle, the servo motor will not open the entrance door(Figure.9b) Figure.9c and Figure.9d show that the entrance door is opened when any parking area is empty. As Figure.9e and Figure.9f are checked the parking area by Arduino codes evaluate whether it is full or not.

4.4. Senario 4

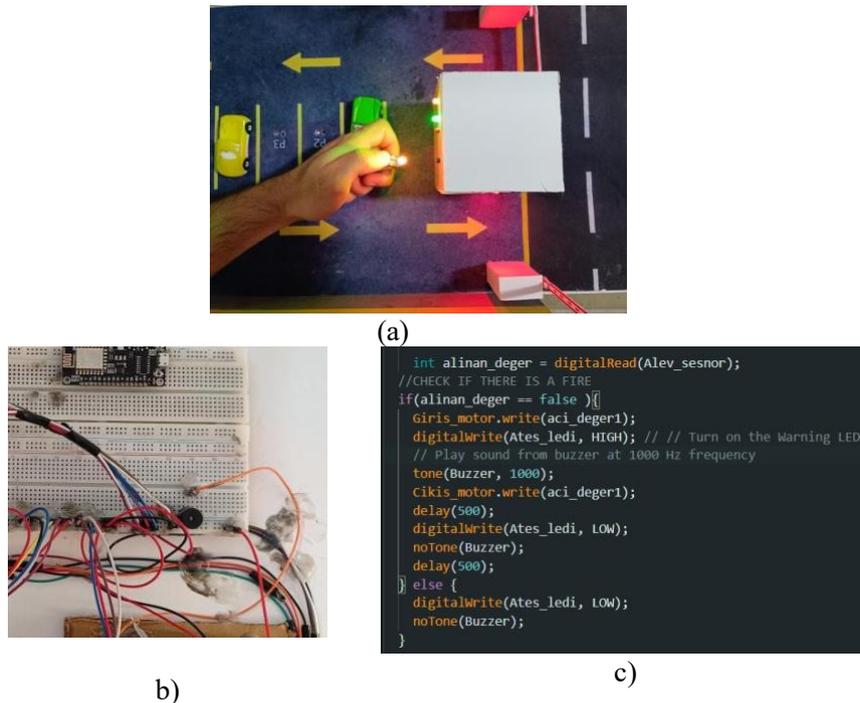


Figure 10. Scenario 4

Figure.10 shows Scenario 4. When the fire sensor in the parking lot detects a fire, the orange LED turns on (Figure.10a). The buzzer in the parking system informs the users with an audible warning when the fire sensor detects a fire(Figure.10b). In Arduino code, fire sensor and buzzer work together; When fire is detected, the buzzer sounds and the warning LED turns on (Figure.10c) You can review the general explanation of the project and the operation of the buzzer sound and LED warning in this scenario in video (URL-1).

5. Test Method and Findings

To calculate the system's accuracy rate, repeated tests were conducted within the scope of each scenario. The system's operation was recorded on video (URL-1) and analyzed in detail.

Table 2. System Performance Test Results

| Test Category | Number of Tests | Successful | Unsuccessful (0) | Accuracy Rate (100%) |
|--|-----------------|------------|------------------|----------------------|
| IR sensor vehicle entry detection | 3 | 3 | ✓ | ✓ |
| IR sensor vehicle exit detection | 2 | 2 | ✓ | ✓ |
| IR sensor parking occupancy detection | 4 | 4 | ✓ | ✓ |
| Servo motor barrier opening | 5 | 5 | ✓ | ✓ |
| Servo motor barrier closing | 5 | 5 | ✓ | ✓ |
| Barrier staying closed when parking full | 1 | 1 | ✓ | ✓ |
| LCD screen update | 3 | 3 | ✓ | ✓ |

| | | | | |
|---|-----------|-----------|----------|-------------|
| Mobile application data synchronization | 2 | 2 | ✓ | ✓ |
| Fire sensor flame detection | 1 | 1 | ✓ | ✓ |
| Buzzer fire warning | 1 | 1 | ✓ | ✓ |
| LED fire warning | 1 | 1 | ✓ | ✓ |
| TOTAL | 28 | 28 | 0 | 100% |

As shown in Table 2, a total of 28 tests were conducted across all scenarios, including vehicle entry/exit detection, parking occupancy monitoring, full-parking condition management, and fire safety warnings. The system achieved a 100% success rate in all tested categories, demonstrating reliable operation under controlled laboratory conditions. The IR sensors accurately detected vehicle presence in all 9 tests (3 entry, 2 exit, 4 occupancy), the servo motors correctly operated the barriers in all 10 tests (5 opening, 5 closing), and the fire detection system successfully responded to flame presence in all 3 tests (sensor detection, LED activation, buzzer activation). This comprehensive testing confirms the functional integrity of the IoTSpS prototype and our hypothesis.

6. Discussion

Scalability: The IoT-based smart parking system presented in this study, while encompassing only four parking spaces at the prototype level, possesses a highly scalable design due to its fundamental architectural structure. The IR sensors, NodeMCU-ESP8266 module, and Firebase-based cloud infrastructure at the heart of the system have been selected to maintain functionality even with an increase in the number of parking spaces. The horizontally scalable nature of Firebase Realtime Database allows for low-latency data flow management even with a large-scale increase in the number of sensors. However, certain architectural improvements may be necessary when aiming for a city-scale application. In this context, modular sensor layouts that allow each parking space to be added to the system as independent modules can be used; and in cases where sensor density increases, regional ESP8266/ESP32 microcontroller clusters can be preferred instead of a single control board to balance data traffic. In large parking lots with high vehicle density, the use of low-latency communication protocols such as MQTT together with the Firebase infrastructure will improve real-time performance. Similarly, the processing load on the mobile application interface will increase with an increase in the number of parking spaces; Dynamic listing, asynchronous data processing, and lightweight client-side architectures can be addressed. Based on these assessments, the proposed system appears adaptable to large-scale applications in terms of both hardware and software.

Robustness and Potential Failure Modes: When the proposed smart parking system is to be implemented in real-world conditions, it is likely to encounter various failure modes, ranging from sensor performance and communication infrastructure to hardware components and data integrity. Therefore, a comprehensive assessment of the system's robustness is critical. Sensor-related errors, particularly IR sensors, can lead to false positive or false negative results due to the effects of ambient light, weather conditions, or physical obstacles. Similarly, flame sensors can give misleading warnings due to changes in ambient temperature and light. Therefore, it is evident that the success rate may drop below 100% in real-world conditions (long-term use, varying lighting conditions, temperature changes, etc.). These types of problems can be significantly reduced by using hybrid detection with ultrasonic or magnetic sensors and employing multi-sensor verification mechanisms.

In ESP8266-based modules, Wi-Fi disconnections can cause delays in data updates, while network-related errors on the Firebase side can disrupt the processing of real-time notifications. Therefore, the integration of local buffering, reconnection policies, and error handling strategies is crucial. At the hardware level, servo motors may experience mechanical fatigue due to continuous operation, or power distribution instability may occur with an increasing number of sensors. Such problems can be addressed by using industrial-grade motors, implementing regular maintenance procedures, and opting for regulated power supplies.

Risks related to software and data consistency can also limit system reliability. Since data collisions may occur during simultaneous reservations or status updates, the use of Firebase's transaction mechanisms is recommended. Firebase Authentication and token-based protection measures against security threats such as unauthorized access or data manipulation should be integrated into the system. This comprehensive assessment reveals areas for improvement to make the system more robust in long-term and large-scale real-world applications.

7. Conclusions and Recommendations

This study presents an IoT-based smart parking system that integrates sensor-driven space monitoring, automated gate control, and a cloud-connected mobile application for user interaction. The system successfully demonstrates real-time detection of parking availability through IR sensors, automated barrier operation via servo motors, and cloud-based data synchronization using the Firebase Realtime Database. The accompanying mobile application enables slot reservation, user authentication, and payment management, offering a functional prototype aligned with modern smart-city requirements.

A key contribution of this work is the development of a scalable and modular system architecture. Although the prototype includes only four parking slots, both the hardware and cloud software

infrastructure are designed to support large-scale deployments. The modular sensor layout, the use of ESP8266-based wireless communication, and Firebase's horizontally scalable database structure collectively enable the system to operate effectively even as the number of parking spaces increases. These design characteristics position the proposed solution as a viable foundation for multi-level parking facilities and city-wide smart parking frameworks. Future iterations can further enhance scalability through the integration of message-queue protocols such as MQTT and distributed microcontroller clusters to balance data flow in high-density environments.

In addition to scalability, the study highlights several robustness considerations necessary for real-world operation. Possible sources of system failure such as sensor inaccuracies, network delays, hardware degradation, and data consistency issues were identified, and mitigation strategies were proposed. Incorporating hybrid sensor mechanisms, local caching, error-recovery procedures, industrial-grade actuator components, and enhanced authentication methods will significantly improve system reliability and operational continuity.

Overall, the developed system demonstrates the feasibility of a lightweight yet extensible IoT architecture for smart parking management. With future improvements in sensor reliability, communication resilience, and large-scale architectural optimization, the proposed framework has the potential to be deployed in real urban environments, contributing to improved traffic flow, reduced congestion, and more efficient use of parking resources.

Authors' Contributions

The planning, methodology, design, and organization of the study were done by Vedat Marttin. The model design and development were done by Abdullah Alawad. Data collection and analysis were a collaborative effort by authors Vedat Marttin and Abdullah Alawad, who subsequently wrote the article by interpreting the results of the study.

Statement of Conflicts of Interest

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The author declares that this study complies with Research and Publication Ethics.

References

- Ahlam-Rauf, A., Bee, K. S. and Thang, W. R. (2023). Architecture Design of an IoT-based Smart Parking System. *10th International Conference on Future Internet of Things and Cloud (FiCloud)*, Marrakesh, Morocco, (pp. 136-141). IEEE. doi: 10.1109/FiCloud58648.2023.00028
- Alam, M. R., Saha, S., Bostami, M. B., Islam, M. S., Aadeeb, M. S. and Islam, A. M. (2023). A Survey on IoT Driven Smart Parking Management System: Approaches, Limitations and Future Research Agenda. *IEEE Access*, 11, 119523-119543.
- Allbadi, Y., Shehab, J. N. and Jasim, M. M. (2021). The smart parking system using ultrasonic control sensors. *In IOP Conference Series: Materials Science and Engineering* (Vol. 1076, No. 1, p. 012064). IOP Publishing. doi: 10.1088/1757-899X/1076/1/012064
- Aydin, I., Karakose, M., & Karakose, E. (2017). A navigation and reservation based smart parking platform using genetic optimization for smart cities. 2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG), 120-124. IEEE. <https://doi.org/10.1109/SGCF.2017.7947615>
- Arduino, S. A. (2015). Arduino. Arduino LLC , 372. [PDF document]. Retrieved from <https://schoolpress.sch.gr/stream/files/2024/01/Arduino.pdf>
- Bracha G. (2015) *The Dart programming language*. Addison-Wesley Professional. ISBN:978-0-321-92770-5
- Channamallu, S. S., Kermanshachi, S., Rosenberger, J. M. and Pamidimukkala, A. (2023). A review of smart parking systems. *Transportation Research Procedia*, 73, 289-296. doi:10.1016/j.trpro.2023.11.920
- Channamallu, S. S., Kermanshachi, S., Rosenberger, J. M., & Pamidimukkala, A. (2026). Smart parking systems: A comprehensive review of digitalization of parking services. *Green Energy and Intelligent Transportation*, 5(1), Article 100293. <https://doi.org/10.1016/j.geits.2025.100293>
- Chougale, P., Yadav, V., Gaikwad, A., & Vidyapeeth, B. (2021). Firebase-overview and usage. *International Research Journal of Modernization in Engineering Technology and Science*, 3(12), 1178-1183.
- Clary, M. (2015). Interfacing to an LCD screen using an Arduino. College of Engineering. [PDF document]. Retrieved from https://www.egr.msu.edu/classes/ece480/capstone/spring15/group05/uploads/4/7/5/1/47515639/ece_480_app_note_matt_clary.pdf
- Gagan, S. and Ravi, H. K. (2024). ParkNest-The Smart Parking System Using IoT. *International Conference on Smart Systems for applications in Electrical Sciences (ICSSES)*, Tumakuru, India (pp. 1-6). IEEE. doi: 10.1109/ICSSES62373.2024.10561362.
- Khanna, A. and Anand, R. (2016). IoT based smart parking system. *International conference on internet of things and applications (IOTA)*, Pune, India, (pp. 266-270). IEEE. doi: 10.1109/IOTA.2016.7562735.
- Kumar, R. H., Roopa, A. U. and Sathiya, D. P. (2015). Arduino ATMEGA-328 microcontroller. *Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng*, 3(4), 27-29.
- Li, H., Zeng, X., Zhang, D., Chen, P., Yu, Z., Sheng, G. and Fu, J. (2014). Occurrence and carcinogenic potential of airborne polycyclic aromatic hydrocarbons in some large-scale enclosed/semi-enclosed vehicle parking areas. *Journal of hazardous materials*, 274, 279-286. doi: 10.1016/j.jhazmat.2014.04.016
- Manurung, M. J., Poningsih, P., Andani, S. R., Safii, M., and Irawan, I. (2021). Door Security Design Using Fingerprint and Buzzer Alarm Based on Arduino. *Journal of Computer Networks, Architecture and High Performance Computing*, 3(1), 42-51.
- Maulana, I. F. (2020). Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android. *Jurnal RESTI, Rekayasa Sistem dan Teknologi Informasi*, 4(5), 854-863.
- Ndunda, E., & Nicolas, A. (2026). Smart on-street parking: Survey of actual implementations in cities and insights from practitioners. *HAL Open Science*. <https://hal.science/hal-05494857>
- Parihar, Y. S. (2019). Internet of things and nodemcu. *Journal of Emerging Technologies and Innovative Research*, 6(6), 1085.
- Pasquo, G. (2020). SG90 Servo Characterization. Aeronautical Engineering Student at Universidad Tecnológica Nacional-Regional Haedo-Argentina.
- P. Kumar V, Gandham, A., Emmanuel, G. and Reddy, J. K. (2023). Smart Parking System using IoT. *7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* Kirtipur, Nepal, (pp. 1169-1176). IEEE. doi: 10.1109/I-SMAC58438.2023.10290386.
- Pramono, L. H., and Javista, Y. K. Y. (2021). Firebase Authentication Cloud Service for RESTful API Security on Employee Presence System. *4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 1-6). IEEE.

- Qadir, Z., Al-Turjman, F., Khan, M. A., & Nesimoglu, T. (2018). ZIGBEE based time and energy efficient smart parking system using IOT. 2018 *18th Mediterranean Microwave Symposium (MMS)*, 295-298. IEEE. <https://doi.org/10.1109/MMS.2018.8611810>
- Rahman, S. and Bhoumik, P. (2019). IoT based smart parking system. *International Journal of Advances in Computer and Electronics Engineering*, 4(1), 11-16.
- Satyaputra, A., Aritonang, E. M., and Kom, S. (2016). Lets Build Your Android Apps with Android Studio. Elex Media Komputindo
- Sharmila, P., Rohinth, P., Sarvesh, G., Priyadharshan, P., Balasakthishwaran, M. and Gandhi, S. (2024). Next-Gen Parking Solutions. *International Conference on Communication, Computing and Internet of Things (IC3IoT)*, Chennai, India (pp. 1-6). IEEE. doi:10.1109/IC3IoT60841.2024.10550402
- Smith, M. N. (2016). The number of cars worldwide is set to double by 2040. In *World Economic Forum*, Geneva.
- Surpris, G., Liu, D. and Vincenzi, D. (2014). How much can a smart parking system save you?. *Ergonomics in design*, 22(4), 15-20.
- Taylor, O., Ezekiel, P. S. and Emmah, V. T. (2021). Smart Vehicle Parking System Using Computer Vision and Internet of Things (IoT). *European Journal of Information Technologies and Computer Science*, 1(2), 11-16.
- Tomar, V., Shrestha, I., & Jain, A. (2025). Analyzing parking solutions with AI and computer vision: A smart parking framework. *Proceedings of International Conference on Computer Science and Communication Engineering (ICCSCE 2025)*, 1643-1654. Atlantis Press. https://doi.org/10.2991/978-94-6463-858-5_133
- URL-1: <https://www.youtube.com/watch?v=hLb6LVU-3yc> (Date Accessed: 7 February 2025).
- Vispute, S., Potdar, M., Pawara, P., Patil, S., Patil, S., & Ahirrao, D. B. (2025). Intelligent parking system to prevent unauthorized parking of vehicles using IoT and R-CNN with YOLO. 2025 *3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, 378-389. IEEE. <https://doi.org/10.1109/IDCIOT64235.2025.10914920>
- Vuković, G., Urošević, M. A., Razumenić, I., Kuzmanoski, M., Pergal, M., Škrivanj, S. and Popović, A. (2014). Air quality in urban parking garages (PM10, major and trace elements, PAHs): Instrumental measurements vs. active moss biomonitoring. *Atmospheric environment*, 85, 31-40.
- Zumma, M. T., Tahmiduzzaman, K. M., Khan, R., Rahman, M., Suhaeel, A. A. and Ahmed, M. F. (2023). An approach to Arduino Uno-based smart car parking system. *14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India, (pp. 1-6). IEEE. doi: 10.1109/ICCCNT56998.2023.10308179.