© NOV 2024 | IRE Journals | Volume 8 Issue 5 | ISSN: 2456-8880

Implementation of an Internet of Things Parking System: Case Study (Federal University of Technology Akure)

BOYINBODE OLUTAYO KEHINDE[1], DARAMOLA OLADUNNI ABOSEDE[2], IJAOLA JOSEPH BOLUWATIFE[3], ASHIRU TAOFEEK OLADAYO[4]
[1, 2] Department of Information Technology, Federal University of Technology, Akure, Nigeria

Abstract- The manual method of car parking system generally involves searching the whole parking area until an available parking spot is found which leads to time wasting and as well environmental pollution, this paper implemented an IoT-based parking management system at the Federal University of Technology Akure as case study. The system uses sensors to determine whether parking spaces are available and gives drivers real-time information, easing traffic congestion and improving the parking experience overall. The usefulness of the system in terms of speeding up parking turnover and decreasing the amount of time spent looking for locations is examined in the study. The study also investigates the prospects and difficulties of establishing an IoT-based parking management system in any setting. The results of the system show that, the system manages parking spaces effectively and efficiently and may be used in other situations that are similar.

Indexed Terms- Parking Spaces, Smart Car Parking System, Internet of Things, Message Queuing Telemetry Transport (MQTT), Flutter, Arduino Integrated Development Environment, NodeMCU.

I. INTRODUCTION

Internet of Things (IoT) provides the use of communication devices that could be tracked, controlled or monitored using remote computers connected through the internet. Rajani and Viegas (March 2017) in Smart Parking Space Detection Using MATLAB and Internet of Things reported that for the driver's convenience a system needs to be designed which can notify availability of parking spot in the area over the internet. One of the states of orderliness on the road and street are measured if vehicles are properly parked in the society. Federal University of Technology Akure as a case study has a vast and large portion of land which some or most buildings have their own parking lots, increase in the number of staff and cars give room for increase in demands of parking spaces. As a result of this, driver do not find it easy to locate available parking lot and thereby wasting time. Also, when drivers do not find parking lot in parking space, they result to parking in unauthorized spaces which can eventually create more problems. It is important to create an automated smart parking system in order to create an easy way for the drivers to navigate their ways out of confusion and any difficulty they might be facing on the campus.

Khanna and Anand (2016) reported that finding an available parking spot is always difficult for drivers, and it tends to become harder with ever increasing number of private car users. This situation can be seen as an opportunity for smart cities to undertake actions in order to enhance the efficiency of their parking resources thus leading to reduction in searching times, traffic congestion and road accidents. Problems pertaining to parking and traffic congestion can be solved if the drivers can be informed in advance about the availability of parking spaces around their intended destination. The design and development of such smart parking lot depends on the approach towards it: how drivers navigate their ways to the available parking lot, the actual location of the driver and the parking lot, allocated parking lot, the physical IoT sensor etc. because there are less resources for its effectiveness. This Paper focused on development of smart parking lot using IoT.

II. PARKING SPACES

Parking spaces are designated areas or facilities purposely created for temporary vehicle parking. They are a vital part of urban infrastructure and play a critical role in addressing the parking requirements of vehicles in different environments like cities, commercial zones, residential complexes, and public venues. Parking spaces offer convenient and well-organized parking choices, ensuring effective utilization of land and meeting the demand for parking.

The design and arrangement of parking spaces differ based on the particular demands and limitations of the area. They can encompass various forms such as open-air parking lots, multi-level parking structures, underground parking garages, or designated on-street parking areas. These diverse types of parking spaces can accommodate vehicles of varying sizes, ranging from cars, motorcycles, and bicycles to larger vehicles like buses or trucks.

The average size of parking spaces can vary depending on the location and jurisdiction. However, in general, a standard parking space for a car in a parking lot or garage is typically around 9 feet wide by 18 feet long (2.7 meters by 5.5 meters). This size allows for sufficient space to accommodate most passenger vehicles comfortably.

It's important to note that parking space sizes can differ based on specific regulations and requirements set by local authorities, as well as the type of vehicles being accommodated. For example, parking spaces designated for larger vehicles or for accessible parking may have different size specifications to accommodate their unique needs.

There are several different types of parking spaces designed to accommodate various parking needs. Here are some common types:

1. Parallel Parking: Parallel parking spaces are typically found on the sides of streets. Vehicles are parked parallel to the curb, with each vehicle aligned in a straight line.
2. Angle Parking: Angle parking spaces are commonly seen in parking lots and garages. Vehicles are parked at an angle to the curb or the parking lot lines, allowing for easier entry and exit.
3. Perpendicular Parking: Perpendicular parking spaces are arranged at a 90-degree angle to the curb or the parking lot lines. This type of parking is commonly found in parking lots, shopping malls, and some residential areas.
4. Diagonal Parking: Diagonal parking spaces are similar to angle parking, but the vehicles are parked at a steeper angle. This type of parking is often seen in areas with limited space or narrow streets.
5. Head-in Parking: Head-in parking spaces require vehicles to enter the space by moving forward. This is the most common type of parking and can be found in various settings such as parking lots, residential areas, and commercial complexes.
6. Back-in Parking: Back-in parking spaces require vehicles to enter the space by reversing into it. This type of parking is sometimes used in specific areas where improved visibility is desired when exiting the space.
7. Stacked Parking: Stacked parking involves parking multiple vehicles vertically in a single space. This type of parking is commonly seen in parking garages or facilities with limited horizontal space.
8. Compact Parking: Compact parking spaces are designed for smaller vehicles, such as compact cars or motorcycles. These spaces are narrower than standard parking spaces and are often marked separately to accommodate smaller vehicles efficiently.

III. RELATED WORK

Khanna and Anand (2016) stated that as there are more vehicles on the road, it can be challenging for drivers to find parking spaces. This presents an opportunity for smart cities to take measures to improve the effectiveness of parking structures, which will reduce traffic jams, time wasted searching for spaces, and road accidents. The use of a mobile application linked to the cloud was implemented along with the way that different actors communicate with one another to assist users in learning the availability of parking spaces in real time. Parking sensors, a processing unit, a mobile application, and the cloud are the actors. The driver was compelled to pay a certain amount of money for each time they parked in a specific lot. The drivers are thus forced to pay for the service because they require it.

Just as Berlianty et al. (2018) said, a lack of parking spaces forces many drivers to cram their cars alongside the road. Their goal is to provide precise recommendations to increase parking space capacity using a system dynamic simulation approach. This system was created using a system dynamic simulation and developing simulation model, and it demonstrates that the scenario with the lowest error value.

According to Soni (2018), the system used the following for its execution in order to develop an automated smart parking management system that will make it simpler for vehicles to find parking places. This is a result of the sharp rise in the number of automobiles with inadequate parking infrastructure, poor management of the parking space supply, and clogged streets in metropolitan areas. an Arduino-based system architecture with IR sensors. Admin controls sensors, hardware (RFID card, GSM module, and IR sensors), and software utilizing cloud, server, and RFID card. An administrator of an ant mesh network cannot act on the network if a car is parked over a sensor. But enforcement is also a challenge.

Thagam et al. (2018) stated that it is disheartening if drivers do not always find it simple to locate a parking space to park their vehicle. This is clogging up traffic and using up the world's oil. To create a smart parking structure from the perspective of reservations using the Internet of Things (IoT), the system uses a mobile application as its interface to confirm the driver's identity to have access to the entry and exit of the parking space and to make reservations. The system cannot currently be used on iOS, but there is room for it in the future.

In line with Poonam Mangwani (2018), the primary issue in many supermarkets, halls, event halls, etc. is a shortage of enough parking spaces. This study suggests an Internet of Things (IoT)-based control for users to manage and watch over open parking spaces as well as watch over and secure parking spaces for vehicles. Drivers can reserve a place from any location thanks to a web application program used for parking cars. It consists of hardware implementation as well as software implementation (web development languages, domains, and servers) (SMPS-switch mode power supply, Wi-Fi Module, IR sensor). To book a parking spot, you need an additional internet connection to access the web application software.

As stated by Upendra kumar tiwari (2020), drivers in urban areas constantly struggle to find parking spaces and deal with traffic congestion. For this reason, the architecture is divided into three sections: the frontend interface, the backend event listener, the database records of the vehicles that are currently available or parked in the parking area, and other components (Arduino, IR sensors etc.). Due to the components and sensors used in the development of an IoT project, carrying out the implementation of an IoT project is expensive.

According to Gómez-Ruz et al. (2021), the rise in the number of private vehicles has made travel more challenging and created high parking demand, necessitating the adaptation of automation within a vehicular parking system. The project's prototype development was completed in a series of stages, each of which had to be completed before moving on to the next. These stages included geometric design, vehicle capacity, machine vision and video processing, object-oriented programming (OOP) in MATLAB, and web development. The systems often have substantial installation and operation expenses.

Mustafa (2021) claims that the requirement for parking is growing in tandem with the number of university employees working there, despite the fact that the number of parking spaces has remained constant at 50. For the realization of this project, hardware and software components are used. Software instructions carried out in accordance with a working program code within a closed and unending programming cycle control the operation of hardware components. Hardware equipment description: The equipment consists of an Arduino board, "Jumper" cables, a 16x2 LCD (liquid crystal display), and a "PUSH BUTTON" switch with C++ programming code built in the Arduino development environment for the digitization of university. Due to the stress of having to pass through the barrier at the entrance and departure of the parking places, drivers tend to park carelessly.

According to Balfaqih et al (2021), drivers can find a parking spot on their own by searching every spot until one becomes vacant. Time is wasted as a result of this. The smart packaging system uses cloud computing, fog computing nodes, and data acquisition to operate. It is pricey.

IV. SYSTEM ARCHITECTURE

This smart car parking system is divided into three main sections which are all connected to each other, through the use of IoT (internet of things) technology; the three main components exchange data between one and another. The system consists of a NodeMCU (microcontroller) which serves as the processing center of the system, it retrieves the data from the sensors and carries out preprogrammed instruction on the data received and then proceeds to disburse the needed information to the mobile application through the use MQTT (message queuing telemetry transport). The figure1 below gives an overview of the system architecture.

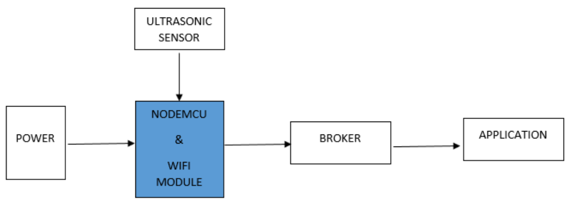

Figure 1: Overview of The System's Architecture

V. IMPLEMENTATION AND RESULT

The Implementation of this project requires the development of an IoT device and a mobile application;

The IoT device was created using a NodeMCU microcontroller, it serves as a processor in which other devices are connected to such as the sensor, bulk converter, and battery heads. The NodeMCU receives signals from the sensors and then uses its Wi-Fi module to upload information through the use of the MQTT (message queuing telemetry transport) protocol, the MQTT protocol uses a publish and subscribe messaging transport system So, the data received would be sent using the MQTT protocol to the mobile applications.

The mobile application will be built using Flutter which is a cross platform for building mobile application, it is based on dart programming language. The mobile application will receive data from the NodeMCU through the MQTT protocol and then display the information to the user.

A. HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE COMPONENTS:

The following are the minimum requirements required for the creation of the IoT component NodeMCU, Ultrasonic sensor, Bulk converter, 9V battery. Figure 2 shows the hardware circuit.

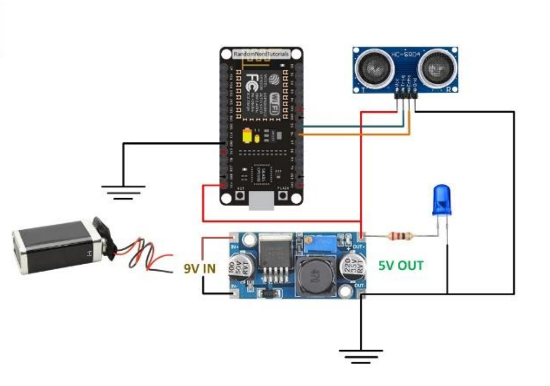

Figure 2: Circuit diagram of the Hardware Component

From figure 2 above, the necessary connections for the Node is shown. The ultrasonic sensor was interfaced with the NodeMCU using several jumper wires, a buck converter was also interfaced with the NodeMCU module to reduce the incoming voltage from the power source, the ultrasonic sensors are connected through the I2c pins, the power for the sensor is connected to the VIN pins and the buck converter in order to regulate the voltage being sent to the microcontroller and sensor. All the components used were coupled together to perform their different task on the Vero board with some basic code. The ultrasonic sensor was connected to the NodeMCU to send data gotten from car object. The power supply for the sensor, which is typically 5V, is provided by the Vcc pin on the Ultrasonic sensor, which are connected to the positive terminal of the power supply. The NodeMCU is commonly supplied with a consistent power source using a bulk converter. The ultrasonic pulse is initiated by the Trig pin. The Trig pin of the sensor is connected to a digital pin of the NodeMCU.
The duration until the pulse returns and is detected is output by the Echo pin. The Echo pin of the sensor is connected to another digital pin of the NodeMCU. The GND pin of the sensor is connected to the GND pin of the MCU and the NodeMCU's GND pin is connected to the power supply's negative terminal. The sensor's

ground is connected to the system ground using the GND pin. Figure 3 shows the system setup; Figure 4 shows the readings from MQTT and Figure 5 shows the output of the Park Monitoring App.

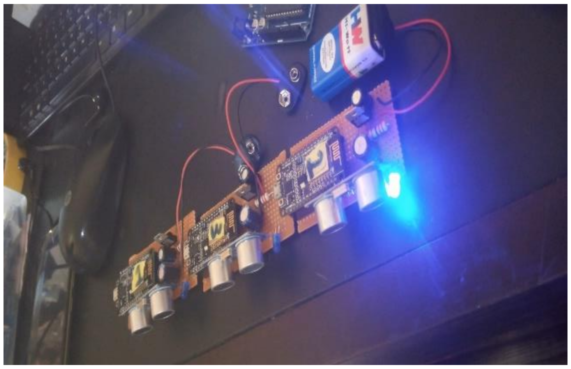

Figure 3: System setup

SYSTEM PROGRAMMING:

i. ARDUINO IDE:

Once the assembling of the components is completed, we move to the programming of the NodeMCU, firstly we connect the microcontroller to the MQTT broker over a Wi-Fi, the code implemented makes use of the asyncMqttclient libraries, which provide functions for connecting to Wi-Fi networks and the MQTT broker respectively. The Wi-Fi network credentials (SSID and password) are defined as constants at the beginning of the code. The MQTT broker host and port are also defined as constants.

ii. FLUTTER:

The user interface for an IoT management system for parking spaces can be built using Flutter as an IDE (Integrated Development Environment). The Flutter framework offers a quick and cutting-edge method to create dynamic, responsive, and animated user interfaces, making it ideal for creating user-friendly interfaces for IoT devices. The parking space management system's specifications may be met by developers using Flutter, and the framework's compatibility for various platforms enables the same code to run on both iOS and Android devices.

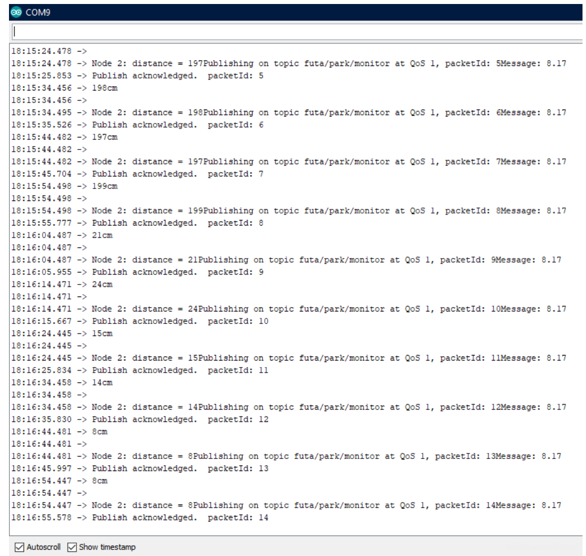

Figure 4: Readings from the sensor published on MQTT

The MqttServerClient library's four MQTT broker-related functions are defined in the given code.

1. mqttSubscribe: This function subscribes to the "futa/park/monitor" topic with a "atMostOnce" Quality of Service (QoS) level. The process prints a message stating that the client has subscribed to the topic.
2. mqttPublish: This command publishes a string message to the "futa/park/monitor" topic and accepts a string message as an argument. The function first clears any previous payload data using the "clear" method of the "MqttClientPayloadBuilder" object. Following that, the "addString" method is used to add the message to the payload. After that, the function publishes the message to the topic with a "atMostOnce" QoS level.
3. mqttUnsubscribe: This function unsubscribes from the "futa/park/monitor" topic.
4. mqttDisconnect: This function disconnects the client from the MQTT broker.

These functions provide a convenient interface for working with the MQTT broker and performing common tasks such as subscribing, publishing, unsubscribing, and disconnecting from the broker.

B. RESULT

The mobile app receives response and output the result on the user interface based on the information it gets from the broker and the following describes different stages of response getting from the physical sensor to its equivalent space on the mobile application i.e. the physical sensor on Node 1 represents space 1 on the mobile app, the physical sensor on Node 2 represents space 2 on the mobile app, the physical sensor on Node 3 represents space 3 on the mobile app, which makes each sensor to publish data to the mobile app independently

A CAR REPRESENTATION

In this case, there is a representation of a parking space occupied in space 1 of the mobile application getting a message from Node 1 sensor, showing that there is an occupied space on Node 1 while no responds from Node 2 to space 2 and no message from Node 3 to space 3 which make space 2 and space 3 to be unoccupied.

Fig 5: A Car representation at the parking spaces (School of Engineering parking lot)

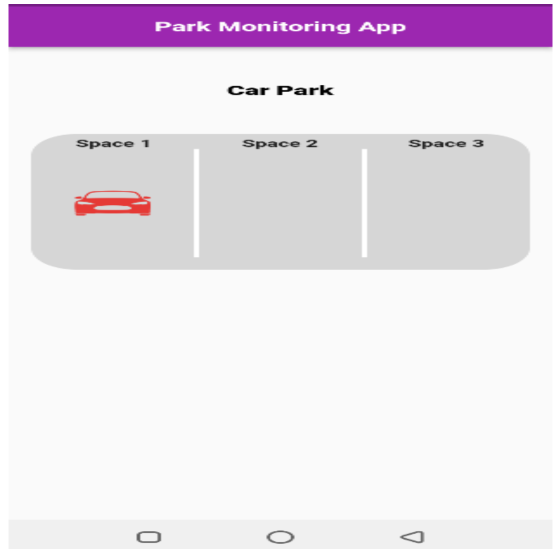

Figure 6: A Car Representation from mobile app.

TWO CARS REPRESENTATION

In this case, there are representations of parking spaces occupied in space 1 and space 2 of the mobile application getting messages from Node 1 sensor and Node 2 sensor respectively, showing that there is an occupied space on Node 1 and Node 2 while no message from Node 3 to space 3 which makes space 3 to be unoccupied.

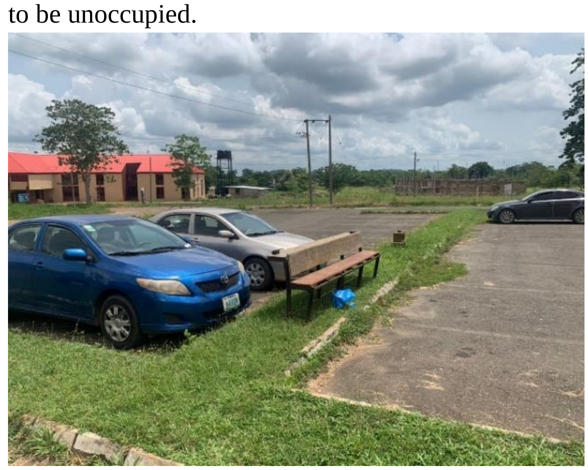

Fig 7: Two Cars representation at the parking spaces (School of Computing parking lot)

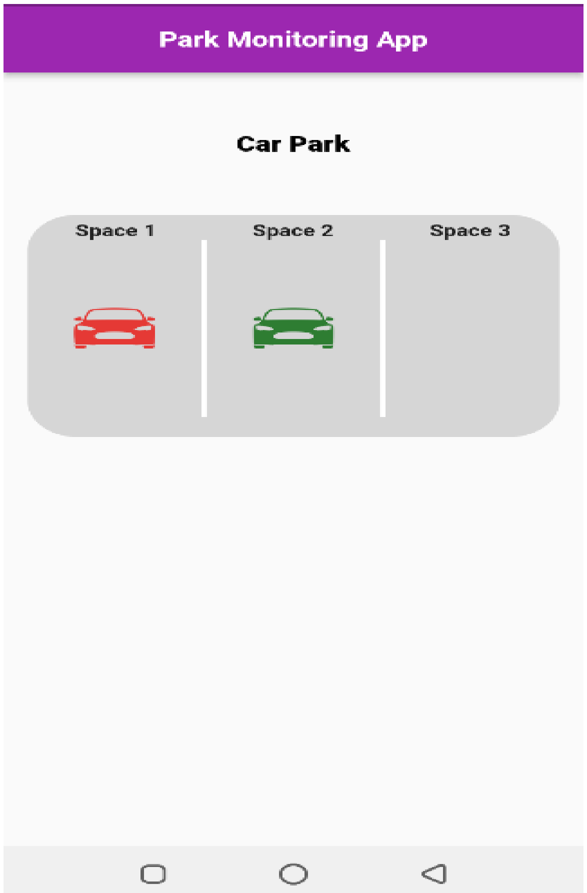

Figure 8: Two Cars Representation from mobile app.

THREE CARS REPRESENTATION

In this case, there are representations of parking spaces occupied in space 1, space 2 and space 3 of the mobile application getting messages from Node 1 sensor, Node 2 sensor and Node 2 sensor respectively, showing that there is an occupied space on Node 1, Node 2 and Node 3 which means that spaces are all occupied.

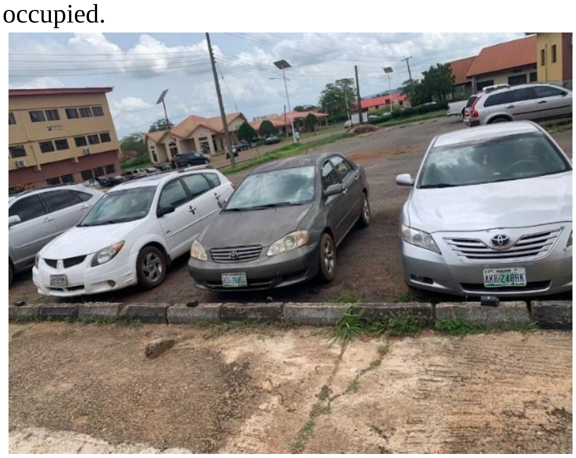

Fig 9: Three Cars representation at the parking spaces (School of Engineering parking lot)

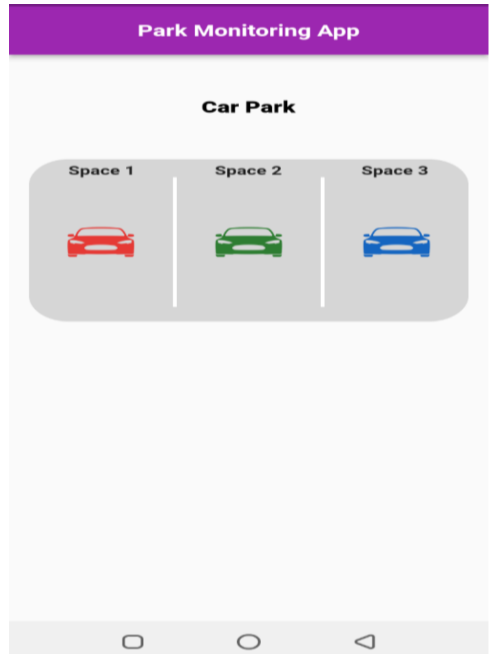

Figure 10: Three Cars Representation from mobile app.

CONCLUSION

The Federal University of Technology Akure's IoT-based parking management system has a number of advantages over conventional parking management systems. It is possible to lessen traffic congestion, cut down on the amount of time drivers spend looking for parking spaces, and boost parking space turnover by using sensors to detect parking availability and to provide real-time information to drivers.

Parking spaces may be managed very effectively and efficiently by the Internet of Things using ultrasonic sensors, buck converters, batteries, Flutter for UI, and NodeMCU. The Flutter-created user interface analyses and presents the real-time data gathered by the ultrasonic sensors on the occupancy status of each parking place. The buck converters help in efficiently managing the power supply to the sensors, while the batteries ensure that the system continues to function even when there is no electricity. The Node MCU, the central component of the system, enables communication between the sensors and the database.

The system also offers insightful information and statistics on parking usage and patterns, which may be used to guide future planning choices and parking policy decisions. The technology has the ability to lessen the environmental effect of parking by cutting down on the amount of time cars are left idle while looking for parking spaces. Using an IoT-based parking management system has advantages, but there are also some disadvantages, such as the need for a significant initial investment, ongoing maintenance, and the potential for technical difficulties. These problems need to be carefully examined and fixed in order to guarantee the system's efficient deployment and long-term performance.

This case study demonstrates how IoT-based parking management systems can reduce traffic congestion, enhance the parking experience for drivers, and provide useful information for planning and policy decisions.

RECOMMENDATIONS

To increase the system's functionality, it would be wise to include more features:

1. It would be possible for users to pay for parking through the system directly if a payment gateway was included. This creates a form of steady revenue to further help fund improvements for the overall system.
2. By diverting cars to available parking spaces, connecting the system with traffic management systems may help to ease traffic congestion and create a geography feature to lead drivers to available spots.
3. The optimization of the parking management process and increased accuracy of occupancy status data may be achieved by applying machine learning techniques.

REFERENCES

[1] Balfaqih, M., Jabbar, W., Khayyat, M., & Hassan, R. (2021, December). Design of Smart Parking System Based on Fog Computing and Internet of Things.

[2] Berlianty, I., Astanti, Y. D., Soejanto, I., & Febriani S., E. (2018). Determining of Parking Lot Area Policy Using System Dynamic Simulation Approach. http://dx.doi.org/10.2991/amca18.2018.198 Retrieved 15th November 2022

[3] Gómez-Ruíz, D. L., Espejel-García, D., Ramírez-Alonso, G., Espejel-García, V. V., & Villalobos-Aragón, A. (2021). Implementation of an Available Parking Space Detection System in Hectic Parking Lots.

[4] Khanna, A., & Anand, R. (2016). IoT based smart parking system. University of petroleum & energy studies. https://researchgate/net/publication/30382610_IoT_based_smart_parking_system Retrieved 7th December 2022

[5] Mangwani, P. (2018). Smart Parking System Based on Internet of Things. https://www.ripublication.com/ijaer18/ijaerv13n12_24.pdf Retrieved 13th November 2022

[6] Rajani, Z., & Araujo e Viegas, M. (2017, March). Smart Parking Space Detection Using MATLAB and Internet of Things. UG Student, Head of Department Electronics and Telecommunication Don Bosco College of Engineering, Fatorda, Margao, Goa, India.

[7] Soni, V. D. (2018). IoT Based Parking Lot. https://www.researchgate.net/publication/343106785_IOT_BASED_PARKING_LOT Retrieved 4th December 2022.