Revolutionizing Urban Mobility: An Integrated Smart Car Parking System Utilizing Mobile Technology And Iot Solutions

Suresh Kumar M[1], Nawin Narayan S[2]

[1, 2] Dept of Electronics and Communication

[1, 2] 2Chennai Institute of Technology (Autonomous), Kundrathur, Chennai 600 069

Abstract- *Parking used to be straightforward, but now it often feels complicated and time-wasting. Many old parking systems need a lot of people to manage them, making it hard for drivers to find a spot quickly. These outdated systems also use up a lot of energy and make drivers spend extra time looking for places to park. This research introduces a solution: a smart parking system accessed through a simple phone app. Users start by downloading this app and setting up an account. They then take pictures of their car's license plate and registration to get verified. At the parking entrance, users use the app to scan a special code, allowing them entry. Once inside, the app lets them pick which floor they'd like to park on. A clever system then finds them a parking spot based on their preference and what's available. Each parking spot has sensors that quickly tell if it's free or taken, updating the system regularly. When it's time to leave, users use the app to pay and get out by scanning another code. This smart system reduces the need for many workers and also makes the parking area look better. Overall, this system focuses on making parking easier and saving people's time..*

Keywords- Application Programming Interface, Car Parking, Internet of Things, Mobile Applications, Sensor Node.

I. INTRODUCTION

The Indian economy is thriving, leading to an increase in purchasing power. Consequently, there has been a significant rise in the number of newly registered vehicles in our congested cities. According to the Ministry of Road Transport and Highways [1], the number of registered vehicles increased from 295K to 327K in just one year. This surge in vehicles has created a demand for extensive parking areas in places like malls, hospitals, city centers, and residential societies. However, organizing the parking environment has become inconvenient and time-consuming.

Currently, manual car parking systems [2] rely on paper records, making it difficult to filter through a large number of records. Employees at these parking facilities waste significant time searching for specific information about a vehicle and preparing reports for revenue or other documents. The process of entering data on paper is also time-consuming. Additionally, relying on manual parking systems incurs redundant expenses.

On the other hand, smart car parking systems utilize RFID cards. Radio-frequency identification uses electromagnetic fields to automatically identify and track tags attached to objects. All the data is collected and stored in a database. This eliminates the need for paperwork and reduces unnecessary capital expenditure.

In this study, we have developed a QR-based smart parking system that consists of various hardware and software components. We have created a web-based user-friendly interface for users to access the system. A database server has also been deployed to store usage records and keep track of the number of available parking slots. Users will be able to see the number of free parking spaces displayed on a screen, eliminating the need to waste time checking for availability when the parking lot is full.

II. RELATED WORK

There are numerous options available for automated car parking management, including both commercial solutions and research articles. Researchers have utilized various technologies such as RFID, ultrasonic detectors, and IoT devices to create parking management systems. One example is a face recognition-based automated car parking system proposed in [7], which eliminates the need for traditional payment methods by detecting the driver's face at entry and exit points. iParker, described in [3], is a new smart car-parking system that utilizes dynamic resource allocation and pricing. The Smart parking system (SPS) in [6] suggests an architecture that incorporates ultrasonic detectors. In [5], a cloud-based car parking middleware is designed and implemented for IoT-enabled smart cities. Additionally, [4] introduces a Vehicle-2-Vehicle (V2V) technology that enables wireless communication to share information about available street-parking spots and distances.The format presented in this document is adapted from [8].

After analyzing these solutions, we have identified the essential features that we need to develop in our own system. These requirements have been gathered from various sources, tailored to suit our country's specific scenario.

III. DESIGNING A SMART CAR PARKINGSYSTEM

Our developed smart car parking system consists of mainly four operations: entry, slot allocation, slot updation andexit.

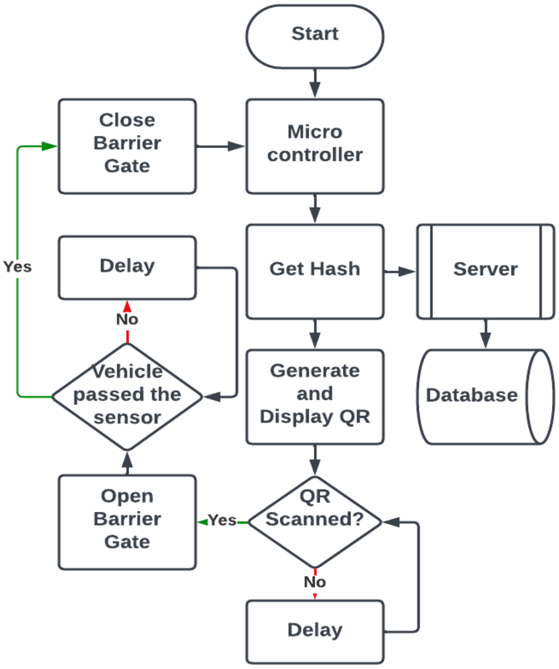

Fig. 1: Flowchart of entry process of the system.

A. Entry Process

You can see the detailed process of entry in Figure 1, where we've illustrated the steps inflowchart.When someone wants to enter our parking area, they use our special app to scan a code displayed on a screen. This code, called a QR code, is unique and comes from our main server. To make this QR code, we use a specific 16-digit code from our server. If it's the first time someone is using the system, they need to make an account. If not, they just log in using their details.

To make sure everything is correct, the app asks users to show pictures of their car's license and a registration paper. The app checks these pictures using special reading software. It makes sure the name on the papers matches with the car's number.

Once the user scans the code, the app saves this information in our database with the time. This helps us keep track of who enters and when. After saving, a gate opens to let the car in. As the car moves forward and crosses a certain point, the gate closes shown in figure 2. Then, our system creates a new unique code for the next user to use, and the process starts again.

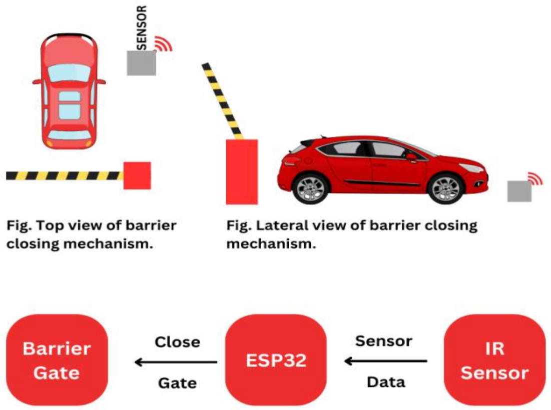
SENSOR
Fig. Top view of barrier closing mechanism.
Fig. Lateral view of barrier closing mechanism.
Barrier Gate
Close Gate
ESP32
Sensor Data
IR Sensor

Fig. 2: Barrier Gate Closing Mechanism.

B. Parking Slot Allocation

The process for updating the parking slots is illustrated in the flowchart provided in Figure 2 below.

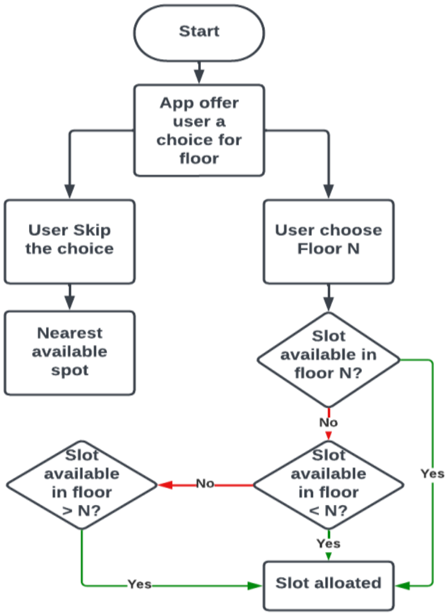
Start
App offer user a choice for floor
User Skip the choice
User choose Floor N
Nearest available spot
Slot available in floor N?
No
Yes
Slot available in floor > N?
No
Slot available in floor < N?
Yes
Yes
Slot alloated

Fig. 3: Flowchart of parking slot allocation process.

After successfully entering, our app offers users a choice: they can pick a specific floor they want to go to, especially in places like a mall. If they know where they want to go, they can choose that floor. But if they're not sure or don't want to pick, no worries! They can skip it andthe app willautomatically find the nearest available spot for them. Once it does, the app shows which spot it is and gives easy directions right on the phone. This saves time and frustration because users don't have to drive around looking for an open space.

Now, if someone does pick a floor, like saying they want to go to the second floor, our smart system gets to work. First, it checks if there's an open spot on that floor. If there is, great! It assigns them a spot there. But if all spots on the second floor are taken, the system is smart. It starts looking on the floors just below that one. It keeps searching in those lower floors until it finds an open spot. Only if there are no spots in the lower floorsthen look on the higher floors. The pictorial representation of the process is shown in figure 4.

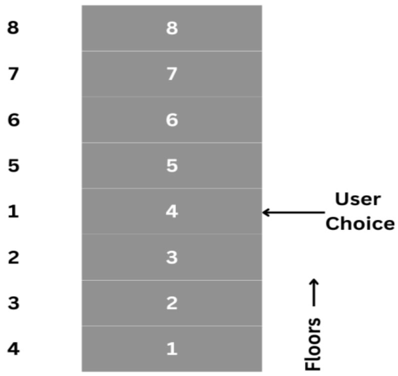

Fig. 4: Parking slot allocation preference order as per user choice

This way of working has many benefits. First, it makes sure everyone gets a spot as efficiently as possible. It also reduces traffic inside the parking area because people aren't driving around aimlessly.

C. Updating Parking Slot Status

In our system, we have a smart way to check if a parking spot is free or taken. We used IR module combined with a small computer chip called ESP8266 a micro controller for this task. If you look at Figure 5, you'll see that we connect four sensors to one of these micro controllers.

These sensors do the job of checking if a spot is empty or if a car is parked there. Once they gather this information, they send it to the micro controller. Now, the micro controller doesn't keep this info to itself. It talks to our main server by making an API call. This is like the micro controller telling the server, "Hey, this spot is now free," or "Someone just parked here."

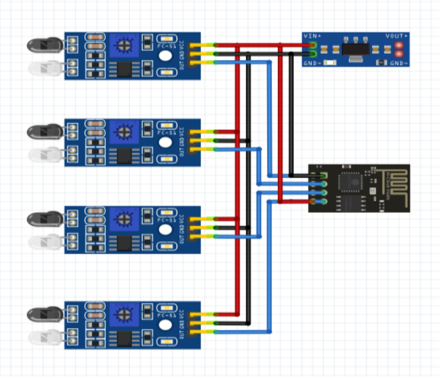

Fig. 5: Sensor Node Setup

But we don't update the database every single time the sensor notices a change. We have a smart rule: unless the sensor detects a change for 5 seconds, we don't update the status (Refer: Figure 6). This helps make sure our system is accurate and reliable.

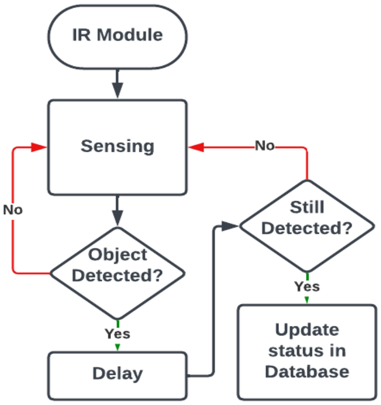

Fig. 6: Flowchart of Slot Status updating mechanism

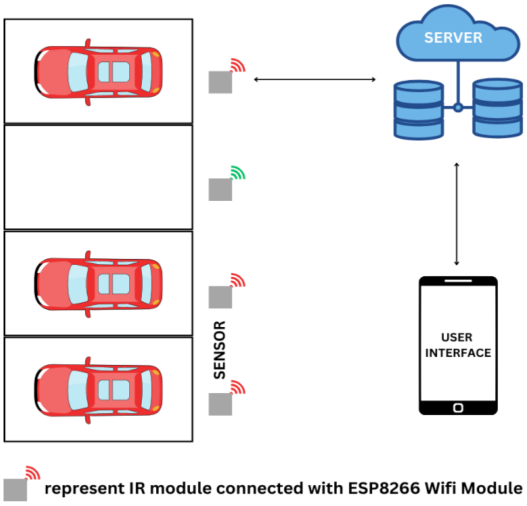

Fig 7: System design

The sensor is strategically positioned on the wall so that it can effectively detect the presence of a vehicle as it approaches. For a clearer visual understanding of this setup, please refer to Figure 8.

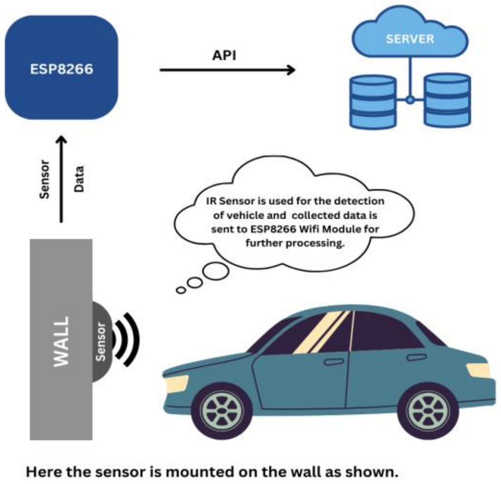
ESP8266
API
SERVER
Sensor
Data
IR Sensor is used for the detection of vehicle and collected data is sent to ESP8266 Wifi Module for further processing.
WALL
Sensor
Here the sensor is mounted on the wall as shown.

Fig. 8: Sensor Setup

D. *Exit Process*

When someone is ready to leave, they just need to use our app to scan an QR code, that's right at the exit. Imagine if the parking lot has a few different ways out; each of these exits has its own unique QR code. So, when someone scans the code, it tells the system, "This is the gate I'm leaving from!" and then the gate opens up.

After scanning, our app shows a bill or receipt for the parking fee. Now, our app is pretty cool because it has a digital wallet feature. Think of it like a virtual pocket in the app where you can keep some money. You can add money to this digital wallet whenever you want and take it out too.

So, once you see the bill on the app, you just tap once to pay using the money from this digital wallet. And just like that, you're good to go and can exit the parking lot hassle-free!

III. DEVELOPMENT AND DEPLOYMENT

The system consists of various parts, both in hardwareand software components. Let's break down and explaineach of these components individually.

A. *Hardware Components:*The hardware components we use in developing the complete system are: ESP32 Micro controller, Display Unit, ESP8266 ESP-01, IR Module, Barrier Gate and Power Supply.

1. ESP32: The ESP32 is like the brain of our parking system. It helps make sure everything works smoothly, like keeping track of cars coming in and going out. When a car needs to enter or leave, the ESP32 helps manage that process, making sure everything happens right.

2. *Barrier gate*:A barrier refers to a bar or pole that can be pivoted to block the passage of vehicles or pedestrians at a specific point. Once all the necessary entry procedures are completed, the barrier gate will automatically open, allowing the user's vehicle to enter the parking area.

*3. Display Unit:*The display unit shows the QR code made by the ESP32. The ESP32 gets this special code by asking the server using an API call.

*4. ESP8266 ESP-01:*The ESP8266 ESP-01 is like a small computer with Wi-Fi abilities. It's connected to our sensor node. This means it gathers information from the sensors and then updates the database with that data.

5. *IR Module:*The IR module is used in two main places:

1. At the entry and exit points, it senses when a vehicle is passing by. This helps in closing the barrier gate.
2. Inside the parking slots, it checks if a spot is free or occupied, helping people find available spaces.

B. Software Components

1. *User Interface:*Our system features a web-based application for user interaction. This application is crafted using Angular and Bootstrap, ensuring a modern and responsive design for optimal user experience.

.

*2. Server:*The server is created using Python and Flask. After development, it's tested on both a local server and a render server to ensure its reliability and efficiency..

3. *Database:*We selected MongoDB as our database solution for several reasons. Its flexible schema allows easy modifications as our system evolves. Additionally, MongoDB's scalability ensures we can handle growing data demands efficiently. Its efficient performance and indexing capabilities enhance data retrieval. Moreover, its adaptability fits seamlessly with our application's dynamic need.

C. User Interface Features

1. User Sign Up/ Sign In..
2. User Verification using OCR.
3. Smart Parking Slot Allocation.
4. Responsive User friendly interface.

5. Live Slot availability details.
6. Invoice Generation.
7. Digital wallet for easy checkout.

IV. CONCLUSION

In this paper, we have created a cost-effective smart car parking system in order to improve the current parking situation in our country. All the necessary components for the system are easily obtainable and relatively inexpensive. Consequently, our developed system can be produced at a significantly lower cost.

Moving forward, our future plans involve incorporating image processing, computer vision, and Artificial Intelligence into the system.

REFERENCES

[1] Ministry of Road Transport and Highways https://morth.nic.in/annual-report

[2] Rentcubo - Problems in manual parking system https://www.linkedin.com/pulse/problem-manual-parking-system-rentcubo/

[3] A. O. Kotb, Y.-C. Shen, X. Zhu, and Y. Huang, "iparkera new smartcar-parking system based on dynamic resource allocation and pricing,"IEEE transactions on intelligent transportation systems, vol. 17, no. 9,pp. 2637–2647, 2016.

[4] W. Balzano and F. Vitale, "Pam-sad: Ubiquitous car parking availabilitymodel based on v2v and smartphone activity detection," in Intelligent Interactive Multimedia Systems and Services 2017, G. De Pietro, L. Gallo,R. J. Howlett, and L. C. Jain, Eds. Cham: Springer InternationalPublishing, 2018, pp. 232–240.

[5] Z. Ji, I. Ganchev, M. O'Droma, L. Zhao, and X. Zhang, "A cloudbased car parking middleware for iot-based smart cities: Design andimplementation," Sensors, vol. 14, no. 12, pp. 22 372–22 393, 2014.

[6] A. Kianpisheh, N. Mustaffa, P. Limtrairut, and P. Keikhosrokiani, "Smartparking system (sps) architecture using ultrasonic detector," International Journal of Software Engineering and Its Applications, vol. 6,no. 3, pp. 55–58, 2012.

[7] Z. Mahmood, "Towards a fully automated car parking system," IETIntelligent Transport Systems, vol. 13, pp. 293–302(9), February 2019.

[8] L. H. Chowdhury, Z. N. M. Z. Mahmud, I. -U. Islam, I. Jahan and S. Islam, "Smart Car Parking Management System," 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON), Dhaka, Bangladesh, 2019, pp. 122-126.