

# Smart Parking System Using Blockchain and OpenALPR

Sanjeev S Sasalu<sup>1</sup>, Srikanth J Pai<sup>2</sup>, Srivallabh Mudumba<sup>3</sup>, Surya Shanmukh A<sup>4</sup>, Kumar P<sup>5</sup>

<sup>1,2,3,4</sup> Dept. of Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bengaluru

<sup>5</sup>Assistant Professor, Dept. of Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bengaluru

\*\*\*

**Abstract** - The standard of living in India, as a result of globalization, has significantly improved. Most of the population now own automobiles. So much so that traffic and parking in the major cities are a huge problem. We propose a solution to the public and private parking systems by automating and monetizing them. The system consists of a parking module which can detect if a vehicle has parked in a slot and can read the license plate number of the vehicle. This module is integrated into a blockchain network with all other similar modules. All the transactions take place in the form of a cryptocurrency on the blockchain network. The user can register and reserve parking lots in advance by using the mobile application, which also functions as a wallet. The system is capable of unsupervised operation once it is set up and at no additional cost to the users. If the license plate read by the module belongs to a registered user, then the amount is automatically debited from the user's mobile wallet until the vehicle is taken out.

**Key Words:** Smart Parking, Internet of Things, Blockchain, OpenALPR.

## 1. INTRODUCTION

Bangalore, being the IT capital of the country, currently has a significantly large population of 11.4 million [1] and the vehicle count in the city has crossed 72.58 lakh, out of which over 50 lakh are two-wheelers and 14 lakh are cars [2] as of March 2018. This has led to traffic congestion in the city and this growth isn't likely to reduce. This problem is also prevalent in other largely populated cities like Mumbai, Delhi, Hyderabad, Chennai, etc. Due to this rapid increase in the vehicle count in the city, parking is a major problem for citizens. Drivers waste a significant amount of time in just trying to find a parking spot. Parking lots are also a business in the cities, where you pay to park your vehicle. This is prominent in malls, industrial complexes and public parking lots.

We propose a smart parking solution for these issues. The parking system consists of a Parking Module, the Blockchain and the Mobile Application. The module consists of a Raspberry Pi 3B+, a camera, and an Ultrasonic Sensor for range detection. The module can detect if a vehicle has parked within the confines of the parking space and reads the number plate of the vehicle. This is done using the OpenALPR (ALPR: Automatic License Plate Recognition) cloud API [3]. These modules are integrated into a

blockchain network. The user must install an application and register his/her vehicle license plate, to make use of these facilities. All transactions are performed using the native cryptocurrency in the system and are recorded on the blockchain. The application is a wallet similar to Paytm or any Bitcoin wallet which allows you to load balance from your bank account. The application also allows the user to reserve a parking spot in advance.

This system focuses on transparency and ease of use. It is capable of unsupervised operation once set up and imposes no additional cost to the user, other than the fee for parking the vehicle. Once the vehicle has been parked correctly, the module captures an image and sends it to the OpenALPR cloud API for processing. Once the license plate characters are identified, the module checks if it is a registered user. Once this is done, it notifies the user of the parking rates in the application and automatically debits the amount from the user's wallet every hour. When the user takes his vehicle out, the transaction is stopped. In case of any issues, the user can also reject the charge and take his vehicle out within a time frame. This entire operation is unaided, and the user doesn't have to do anything. He can just park his vehicle and walk away.

Since each module operates independently, the system is suitable for small scale applications like mall parking lots as well as large scale applications like city-wide implementation, similar to the parking meter system in the United States. This system can also be a tool for enforcing parking policies and to aid in traffic and mobility management in congested cities.

### 1.1 Why Blockchain?

This system is made possible due to its integration into a blockchain network where all the transactions are recorded on a ledger. A blockchain is a data structure that consists of a chain of blocks. All the nodes in the network have their own local copy of the ledger of transactions and any change in the local ledger of one node is reflected in all the nodes in the network.

The ledger is a hash chain where each subsequent block has a cryptographic hash value that is linked to the hash value of the previous block. So, any change in one block will require changing the hash values of all the previous blocks in the chain. This is almost impossible as these hash values are

calculated using the SHA256 algorithm which generates a 256-bit hash value which takes some time and processing power. This is what ensures the security of the system and discourages any attempts at tampering with the transactions. The blockchain cannot be corrupted, as changing any unit of information on the blockchain would require a huge amount of computing power to override the entire network.

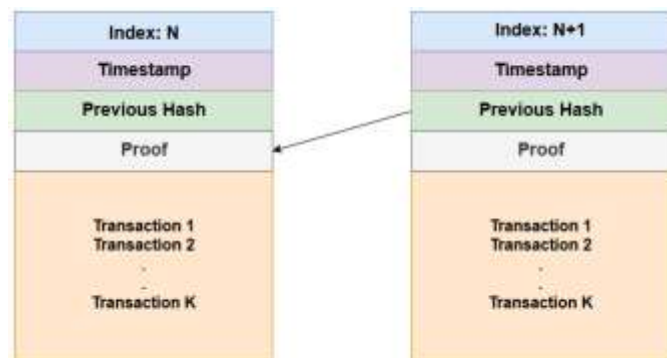


Fig -1: Structure of a Block

The 'Proof' value in the above figure is the value which gives us the required hash when we apply the SHA256 algorithm. So, every time a new block must be added, all the blocks are validated i.e. all the hash values are re-checked. An important feature of the blockchain is non-repudiation. This feature ensures that the parties involved in the transaction cannot deny having either paid or received the money. This means that no illegal activities are possible within the system.

## 2. RELATED WORK

A system called iParking [4] was proposed where users could monitor parking states, search for vacant parking spots and reserve it, make a payment, etc. Usually, in large parking lots, it is difficult for drivers to know ahead of time if they will be able to secure a parking spot in a particular area. The authors also noted that it is time-consuming to wait in line for paying the parking fees when leaving the parking lot. In this system, the authors have proposed to set up magnetic field sensors in each parking lot, which can detect vehicles as they contain significant amounts of ferrous metals and affect the magnetic field nearby. This change in the magnetic field can be used to detect if a parking spot is vacant. This sensor node then communicates the state of the slot to the server. The users can then check the states of the parking slots and reserve them. However, the authors don't mention how the payments are processed, and they have used a web-based interface for reserving parking slots.

A parking system called CampusSense [5] used Automatic Number Plate Recognition (ANPR) cameras and an android based mobile application to monitor and manage the parking facilities at university campuses. The authors note that

vehicle parking and management has become a big challenge for educational institutions, resulting in blockage of vehicles, congestion and sometimes damage to vehicles. The proposed system consists of surveillance cameras which can detect the number plates of the vehicles and then registers the time and location when the particular vehicle has parked. This system is a parking surveillance system and the object here is to keep track of the vehicles that parked in the slots. This allows security personnel to monitor the system and find out the users responsible for any damages. It doesn't allow any sort of reservation or payment system.

The authors in [6] proposed a smart reservation-based parking system which allows you to book and make payments for your parking slot. The authors note that most commuters spend more time finding spaces for parking. They developed an application which allows you to search for a parking slot and book it. However, it requires the user to enter the entry time and exit time, it isn't dynamic. It also requires the user to pay beforehand. The payments aren't automated. The hardware consists of a NodeMCU and a HC-SR04 Ultrasonic Sensor to detect vehicles.

The authors in [7] have proposed a reservation-based parking system where the user has to register into their system, and the system shows the available parking spaces based on the GPS location of the user. The user here has to mention the time for which he will need the parking lot and make the cashless payment. The system also allows for refunds in case the user leaves before the stipulated time and charges extra if the user exceeds it. However, the system is trust-based which assumes that only the user's car will park in the slot and not someone else. There is no way to detect which vehicle has parked in the slot.

The authors in [8] have proposed a brilliant method for vacant parking space detection for smart cities in real time by using a crowd of taxis to sense on-street parking space availability. They suggest that this is a better alternative than installing static parking sensors in each slot. This is still under development and is not suitable for certain countries like India where there aren't proper demarcated parking slots on the streets.

The authors in [9] propose a secure and intelligent parking system based on a secured wireless network and sensor communication. The system is reservation based and involves attaching a short-range wireless transceiver and a simple processor. This is an inconvenience when we consider the number of vehicles in the city alone. The system proposed here accepts payments first and then each parking slot is equipped with wireless transceivers, parking belts, an Infrared Device (IFD) and a control computer. This is a lot of equipment to set up for each slot in the parking lot. The system involves lots of setup and requires planning with the layout of the parking slot under consideration.

### 3. HARDWARE/SOFTWARE TOOLS USED

#### 3.1 Hardware Components

i) Raspberry Pi 3B+



Fig -2: Raspberry Pi 3 Version B+

Specifications

- ARM Cortex A53 1.4GHZ
- RAM: 1GB
- 40 GPIO Pins
- Operating System: Raspbian

ii) HC-SR04 Ultrasonic Sensor



Fig -3: HC-SR04 Ultrasonic Sensor

Specifications

- Working Voltage: DC 5V
- Working Current: 15mA
- Max Range: 4m
- Min Range: 2cm

iii) PiCamera V2

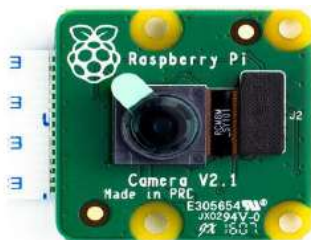


Fig -4: PiCamera V2 Module

Specifications

- Resolution: 8MP
- Supports 1080p Recording

#### 3.2 Software Tools

- Blockchain Class built using Python 3.7.
- Flask 1.0 for building the REST API.
- OpenALPR REST API for license plate recognition.
- Postman for API testing.
- Google Firebase Database.
- Android Application built using Java in Android Studio.

### 4. WORKING PRINCIPLE

The system has 3 parts to it:

- The parking module which consists of the Raspberry Pi 3B+, a camera and the HC-SR04 Ultrasonic Sensor.
- The Blockchain REST API which serves as the backend.
- The Android application and the Database.

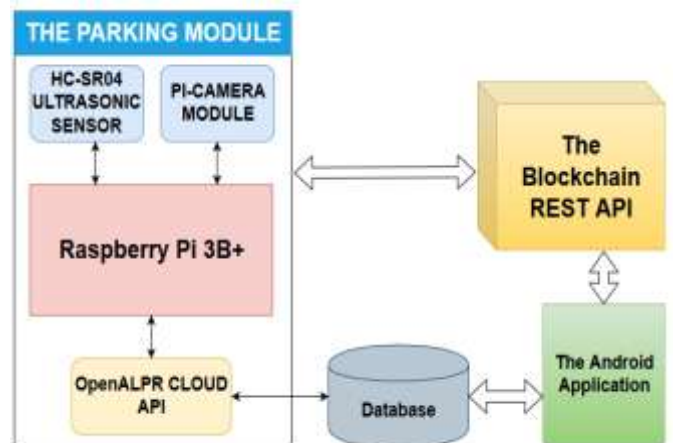


Fig -5: Block Diagram of the Parking System

#### 4.1 The Parking Module

The parking module comprises of a Raspberry Pi 3B+ running the Raspbian Operating System. It is connected to a HC-SR04 Ultrasonic Sensor for distance measurement and a PiCamera V2 module to capture images. The module runs a program written in Python 3.7. The ultrasonic sensor is used to check if an object is in the defined range, and if it is, the camera captures an image. When an image is captured, the module sends the image along with a POST request to the OpenALPR Cloud API, which then returns the license plate number. The module then checks if this plate is registered in the database, and if not, it activates a buzzer. If it does exist in the database, the module automatically debits the amount from the User's wallet. All the transactions are in the form of



a cryptocurrency and are recorded in a blockchain. All these actions are performed using the 'Requests' Python library.

### 4.2 The Blockchain REST API

The Blockchain class is built using Python 3.7 and the REST API is built using the Flask framework for Python. The Blockchain network records all the transactions and validates them. This provides transparency. The blocks are encrypted using the SHA 256 algorithm for security.

### 4.3 The Android Application and the Database

The Mobile Application is built using Java in Android Studio. It acts as a wallet for storing the User's cryptocurrency. It also allows the user to register into the network and reserve parking spaces. The Database can be any SQL or NoSQL database.

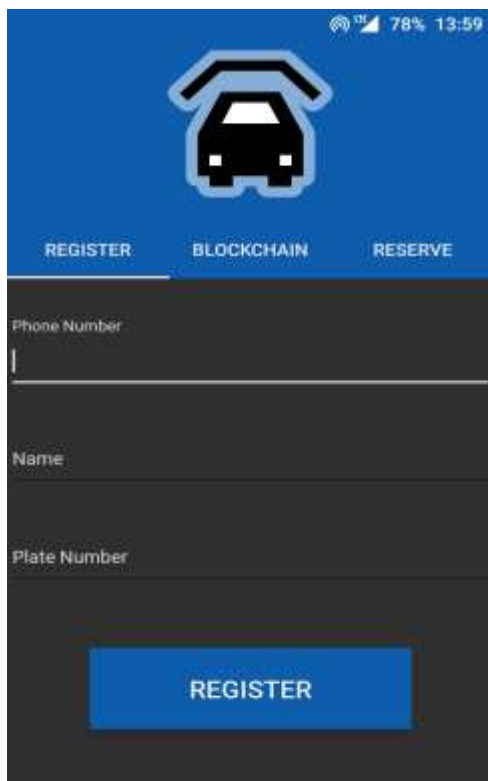


Fig -6: Android App for the Parking System

The Android App allows the user to register his vehicle into the system, view the blockchain as well as reserve a parking slot in an area. The fee for parking can be set by the owner of the parking lot.

The Database in this project was implemented using Google Firebase, but any such database can be used for the system. The Database will store the users' phone numbers, name and plate numbers.

### 4. FLOWCHART

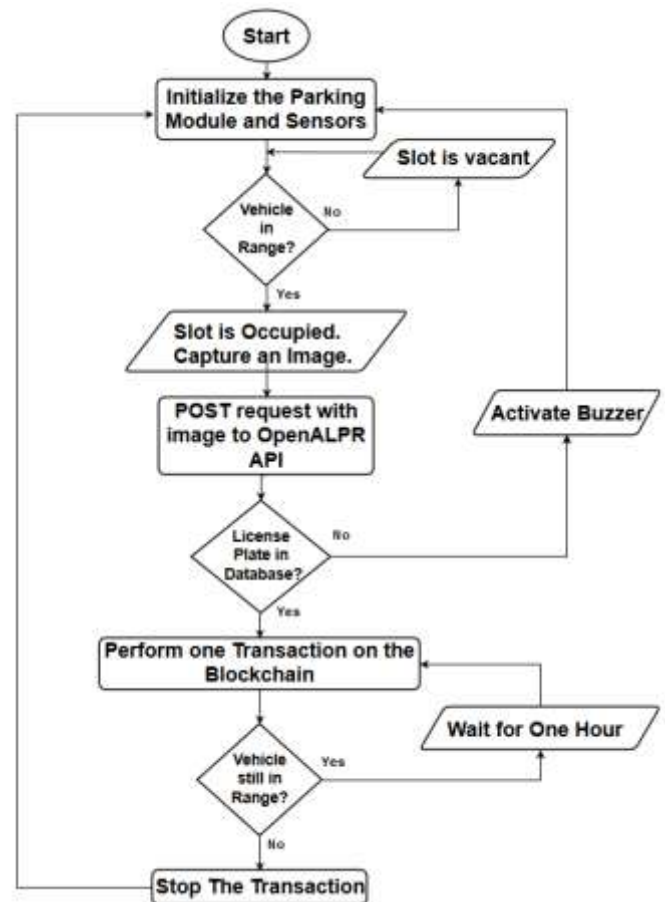


Fig -7: Flowchart of the Parking System

#### Steps:

1. Start and Initialize the Module.
2. Check if the object is within the defined range.
3. If there is no object in the range, the parking slot is still vacant, keep checking.
4. If there is an object in the defined range, the parking slot is now occupied, capture an image with the camera.
5. Send a POST request to OpenALPR with the image file to retrieve the license plate characters.
6. Check if the license plate returned exists in our database.
7. If the plate is not registered, activate the buzzer, reset the module.
8. If the plate exists in the database, then start performing the transaction.
9. Wait for the defined time, check if the vehicle is still in the range, if so perform another transaction.
10. If the vehicle isn't in the range anymore, stop the transactions and reset the module.

## 5. RESULTS

The final project hardware implementation:

- The Raspberry Pi 3B+ is connected to the PiCamera V2 and the HC-SR04 Ultrasonic Sensor.
- The Yellow LED is on when there is no object in the defined range, and indicates that the parking slot is vacant.
- The Green LED turns on when an object is detected and the camera takes a picture.

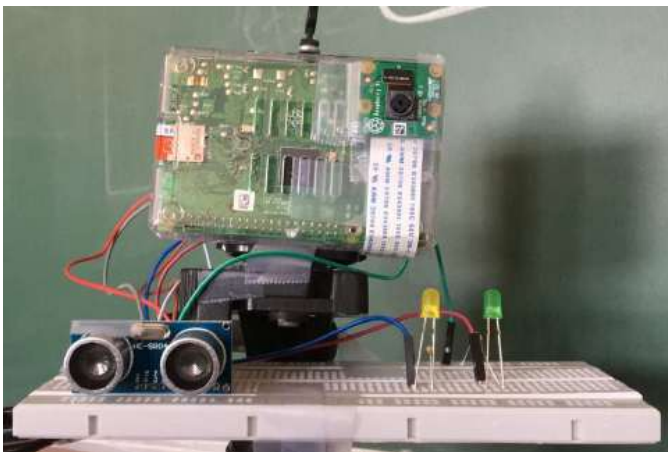


Fig -8: Hardware Implementation

The transactions on the Blockchain:

- Each block has an index, a proof and the previous Hash value.
- The transactions are recorded in each block.
- The timestamps are in UTC (Coordinated Universal Time).
- Once the vehicle leaves the spot, the transactions are stopped.

```

JSON Raw Data Headers
Save Copy Collapse All Expand All
current_chain:
  0:
    index: 1
    prev_hash: "0"
    proof: 1
    timestamp: "2019-04-24 03:37:37.255991"
    transactions: []
  1:
    index: 2
    prev_hash: "18b423fcaca87dd69b98a71b942df9ca943210c59bc5d6bb7bb8e2c941e2463c"
    proof: 533
    timestamp: "2019-04-24 03:37:45.807416"
    transactions:
      0:
        amount: "10"
        receiver: "owner"
        sender: "KA05JG90"
  
```

Fig -9: Transactions on the Blockchain

## 6. ADVANTAGES AND LIMITATIONS

### 6.1 Advantages

- **Round the clock operation:** The system works unsupervised once it is set up.
- **Transparency:** The transactions are stored in a blockchain, this means that there is very less scope for malicious users intending to cheat the system and makes it easy for auditing purposes.
- **User-friendly:** The user doesn't have to make any modifications to his vehicle or install any type of device, everything is managed by the app.

### 6.2 Limitations

- **Physical Security:** The module can be damaged due to rain and is vulnerable to theft. So physical security must be provided to the module.
- **Power Requirements:** The module requires a power supply and battery backup during working hours.

## 7. CONCLUSION

The parking system consists of a parking module, built using the Raspberry Pi 3B+, a HC-SR04 Ultrasonic Sensor and a PiCamera. The module detects if a vehicle is in range and captures the image. This is then sent to the OpenALPR cloud API, which then returns the License Plate number. Once this number is received, if the license plate is registered, the module starts to perform the transaction, based on the predefined fee, until the vehicle leaves the spot.

This can be improved by adding Vehicle object detection so that the camera captures an image only if there is a vehicle in front of it and not any other object. Another feature that can be added is to ensure that the vehicle is parking within the confines of the spot, to prevent any damage to neighbouring vehicles.

## REFERENCES

- [1] <http://worldpopulationreview.com/world-cities/bangalore-population/>
- [2] <https://www.thehindu.com/news/cities/bangalore/vehicle-count-breaches-72-lakh-mark/article22936333.ece>
- [3] <https://www.openalpr.com/>
- [4] D. Zheng, X. Zhang, Y. Shu, C. Fang, P. Cheng and J. Chen, "iParking: An intelligent parking system for large parking lots," 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, 2015, pp. 37-38.
- [5] M. Y. Aalsalem and W. Z. Khan, "CampusSense — A smart vehicle parking monitoring and management system using ANPR cameras and android phones," 2017 19th

*International Conference on Advanced Communication Technology (ICACT)*, Bongpyeong, 2017, pp. 809-815.

- [6] A. Somani, S. Periwal, K. Patel and P. Gaikwad, "Cross Platform Smart Reservation Based Parking System," *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, Mumbai, 2018, pp. 1-5.
- [7] S. Kazi, S. Nuzhat, A. Nashrah and Q. Rameeza, "Smart Parking System to Reduce Traffic Congestion," *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, Mumbai, 2018, pp. 1-4.
- [8] F. Bock, S. Di Martino and A. Origlia, "Smart Parking: Using a Crowd of Taxis to Sense On-Street Parking Space Availability," in *IEEE Transactions on Intelligent Transportation Systems*.
- [9] Gongjun Yan, Weiming Yang, Danda B. Rawat, Stephan Olariu SmartParking, "A Secure and Intelligent Parking System" in *IEEE Intelligent Transportation Systems Magazine* ( Volume: 3, Issue: 1, Spring 2011).