INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Dharwad

ज्ञानेन विकासः

Mini Project 1

A Project Report on

Smart Vehicle Hardware for Finding Parking Spots

Submitted By-

Abdul Rahman (18BEC001)
Ajmal A (18BEC003)
Aryan Kumar (18BEC004)
Likhith Kumar (18BEC005)

Under Guidance of -
Dr. Mukesh Kumar Mishra
Dept. of E&C Engineering

Friday, 14 May 2021

INDEX

INTRODUCTION

Traffic congestion caused by vehicles is an alarming problem at a global scale and it has been growing exponentially. Car parking problem is a major contributor and has been, still a major problem with increasing vehicle size in the luxurious segment and confined parking spaces in urban cities. Searching for a parking space is a routine and often frustrating activity for many people in cities around the world. This search burns about one million barrels of the world's oil every day. As the global population continues to urbanize, without a well-planned, convenience-driven retreat from the car these problems will worsen. Smart Parking systems typically obtain information about available parking spaces in a particular geographic area and process it real-time to place vehicles at available positions. It involves using low-cost sensors, real-time data collection, and mobile-phone-enabled automated payment systems that allow people to reserve parking in advance or very accurately predict where they will likely find a spot. When deployed as a system, smart parking thus reduces car emissions in urban centers by reducing the need for people to needlessly circle city blocks searching for parking. It also permits cities to carefully manage their parking supply. Smart parking helps one of the biggest problems in driving in urban areas; finding empty parking spaces.

Parking facility managers will be able to optimize the use of space and resources within their parking lots, efficiently strategize and plan future development.

Need of Smart Parking

With the increase of the urban population, the need for living and infrastructure space is higher than ever before. On the other hand, the growing number of car owners it's estimated there will be over **two billion cars on the road by 2035** creates a high demand for parking spaces. Drivers searching for parking are estimated to be responsible for about 30% of traffic congestion in cities. Historically, cities, businesses, and property developers have tried to match parking supply

to growing demand for parking spaces. It has become clear that simply creating more parking spaces is not sufficient to address the problem of congestion. New approaches using smart parking systems look to provide a more balanced view of parking that better manages the relationship between supply and demand.

Benefits of Smart Parking

1. Optimized parking – Users find the best spot available, saving time, resources and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities.
2. Reduced traffic – Traffic flow increases as fewer cars are required to drive around in search of an open parking space.
3. Enhanced User Experience – A smart parking solution will integrate the entire user experience into a unified action. Driver's payment, spot identification, location search and time notifications all seamlessly become part of the destination arrival process.
4. New Revenue Streams – Many new revenue streams are possible with smart parking technology.Integrated Payments and POS – Returning users can replace daily, manual cash payments with account invoicing and application payments from their phone. This could also enable customer loyalty programs and valuable user feedback.
5. Increased Safety – Parking lot employees and security guards contain real-time lot data that can help prevent parking violations and suspicious activity. License plate recognition cameras can gather pertinent footage. Also, decreased spot searching traffic on the streets can reduce accidents caused by the distraction of searching for parking.
6. Real-Time Data and Trend Insight – Over time, a smart parking solution can produce data that uncovers correlations and trends of users and lots. These trends can prove to be invaluable to lot owners as to how to make adjustments and improvements to drivers.

LITERATURE SURVEY

Presently, many high traffic parking structures such as those found in airports, sports stadiums and shopping centres utilize vehicle entry and exit counters. While this simple approach indicates the number of parking sports that are vacant on a given level, this cannot provide drivers with any information regarding the specific location of vacant parking structures. A system that could keep track of available spots and push the data to driver smartphones or even the ticket received upon entry to the structure would improve driver experience and reduce congestion. One such system was implemented by the San Francisco Municipal Transportation Authority, which funded the SFpark project to cover 5,100 metered spaces throughout the city using embedded wireless optical sensors.

1. The sensors provide variable accuracy depending on the operational cost The highest accuracy reported is 92%.
2. The sensors are fine-tuned to detect vehicles within a defined area, and as such require clear demarcations on the street. The cost of installing and operating the sensor metering is approximately an annual cost of $500 per unit.
3. Another approach has been to embed ultrasonic sensors on-board vehicles coupled with GPS, which can provide drive-by sensing of parking spot vacancy.
4. However, the placement of the sensor on a mobile platform introduces additional complexity, and deployment of this technology presents its own challenges. Other approaches using image processing have relied predominantly on image segmentation based on color his-tograms. This has the disadvantage of not being robust towards cars that are of similar color to the background; but also requiring variable thresholds depending on lighting conditions which can be a difficult problem to quantify.
5. Another aspect of existing algorithms is the reliance on using hand labelled parking regions - while this is acceptable for small problems, it would be tedious and unrealistic to implement for large parking structures with a large number of spots.

Hardware Requirements

1. Raspberry Pi Model 3B+

Fig 1: Raspberry pi

Raspberry pi is the main gateway used for processing and sending data to the cloud. It acts as a microprocessor and a gateway.

- **RAM:** 1GB - 8GB
- **SoC:** Broadcom BCM2711 SoC
- **Inbuilt Modules:** Wifi/Bluetooth/HDMI
- **Working Voltage:** 5 Volts
- **Programming Capability:** Any Language
- **GPIOS:** 40 Digital / Analog Pins
- I2C/UART Supported
- Linux Based Operating system

2. Vehicle Detection Sensor

Fig 2: Ultrasonic Sensor

Ultrasonic sensors are immune to target color, reflectivity, and transparency, which simplifies product changeover and reduces downtime. Additionally, they are unaffected by area light conditions and perform well in dirty and wet environments. Hence it is best suitable for our project.

3. Ticket Vending/Payment Machine

Fig 3: Payment Machine

There will be a ticket vending machine at the entrance that will generate tickets and a payment machine will be there at the exit point in which the user will have to either scan the QR on the ticket or enter the ticket number to fetch the details. After successful payment the exit servo barrier will open and the user can exit.

Softwares & Programming Languages/ Modules

1. **Python 3.x and PIL Image Library**
   Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

2. **MQTT API By Cayenne**
   The Message Queuing Telemetry Transport is a lightweight, publish-subscribe network protocol that transports messages between devices.

3. [Cayenne IoT](#) Cayenne Iot online platform takes most of the complications out of creating hardware oriented programming. This online platform contains remote monitoring and control mqtt API , lorawan, task scheduling and many more.
   This platform stores all the previous data with its check in and check out times so that it can be used later for security or for other data analysis.

Device Name	Device RPI
Device Icon	Cayenne
MQTT Username	cc19b960-9ded-11eb-b767-3f1a8f1211ba
MQTT Password	b6f7930368322aef505a03697ab38c52ff9bba1a
Client ID	80839f80-9e64-11eb-a2e4-b32ea624e442
Remove Device	Remove Device This action cannot be undone

Fig 4: MQTT User Access Control Key

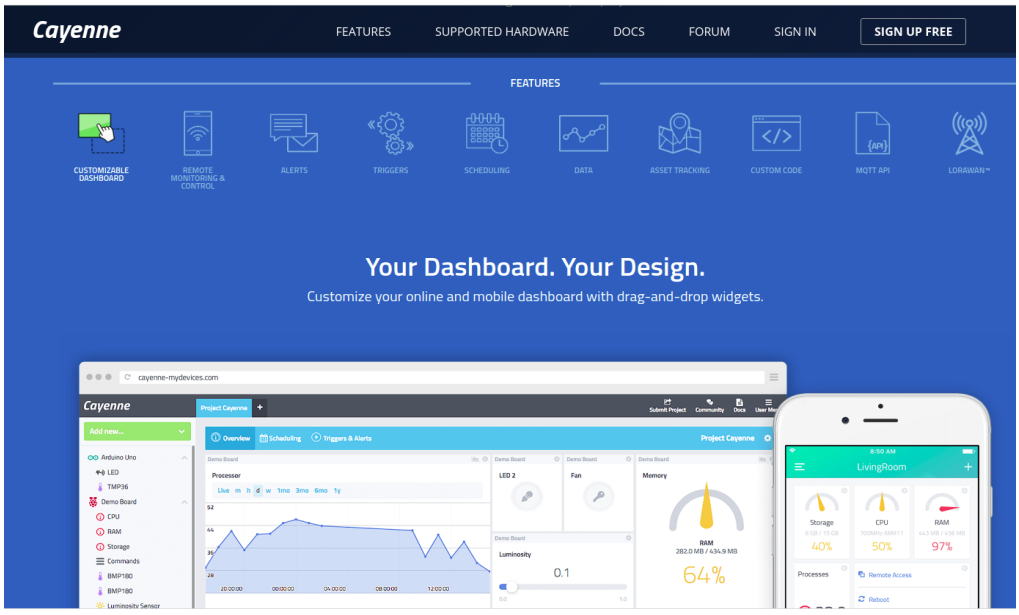

Fig 5: Cayenne IoT Dashboard

Basic Architecture

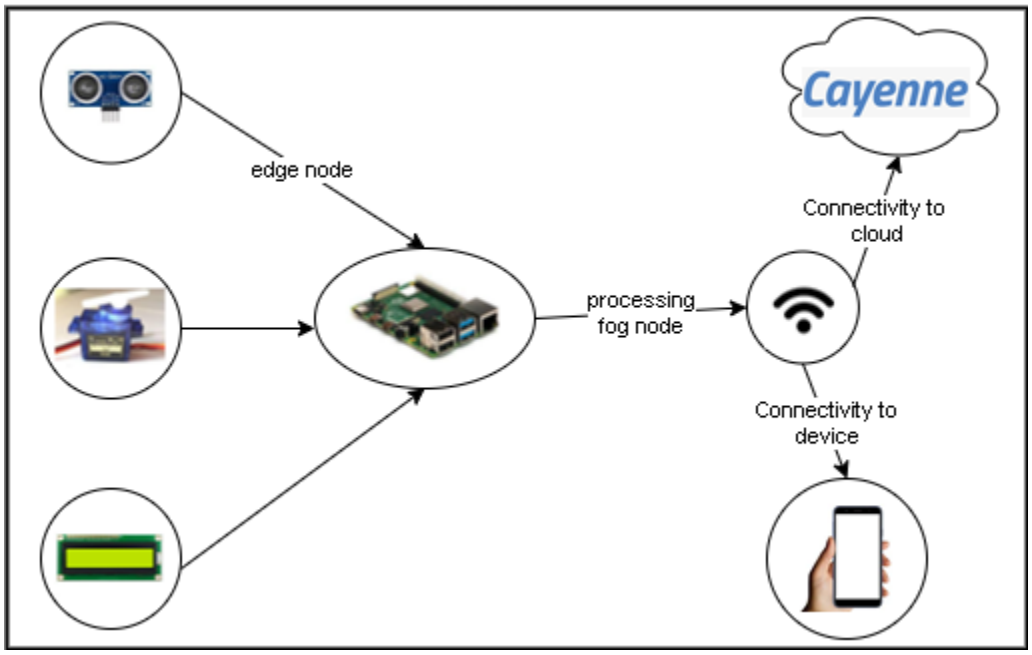

Fig 6: Basic Architecture

★ EDGE-NODE: Starting from the front end, the edge node is the first node of any IoT system, without this node the Things part of IoT is not achieved, these edge nodes are sometimes also called as objects and they mostly work as sensing nodes. These nodes usually have dual nature. Examples of end-users are all types of sensors but normally these sensors are basic and it can be converted into an active device by a designer. In our project we have used an Ultrasonic sensor, Servo motor and LCD display as edge devices.

★ PROCESSING- FOG NODE: It is the central important block because it provides an artificial intelligence to the whole circuitry. It processes the data and information received from edge nodes and transfers it further for next action which may be software application or cloud based service and data received from application to the previous edge nodes. Usually this node involves one or more microcontrollers, microprocessors, etc. and may be relatively bigger in size as compared to edge nodes. In our project we have used a Raspberry pi microcontroller as the processing node.

★ CONNECTIVITY: Connectivity is must in any system to establish a connection, which may be wired or wireless. The main task of this node is to transfer the data gathered after

processing of data sent by the edge node to the application software or to the cloud. Connectivity allows data transfer in duplex form. In our project we have used WiFi to send data over the cloud.

★ APPLICATION: Cloud based service is an endpoint in any IoT system which is essential to see the output of the project or the output of the data sent forward by the sensor nodes or the edge devices. Users can use that information and apply various techniques to make its representation effective. In this smart parking system we have used Cayenne. Cayenne allows us to quickly design, prototype and visualize IoT solutions.

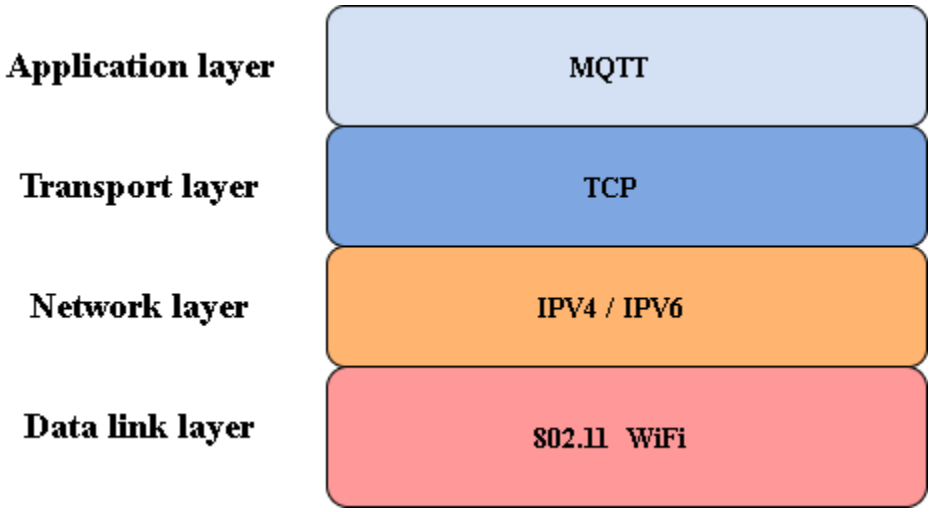

Fig 7: IoT's Generalized Architecture

Methodology

- The basic block diagram of project is as follows:

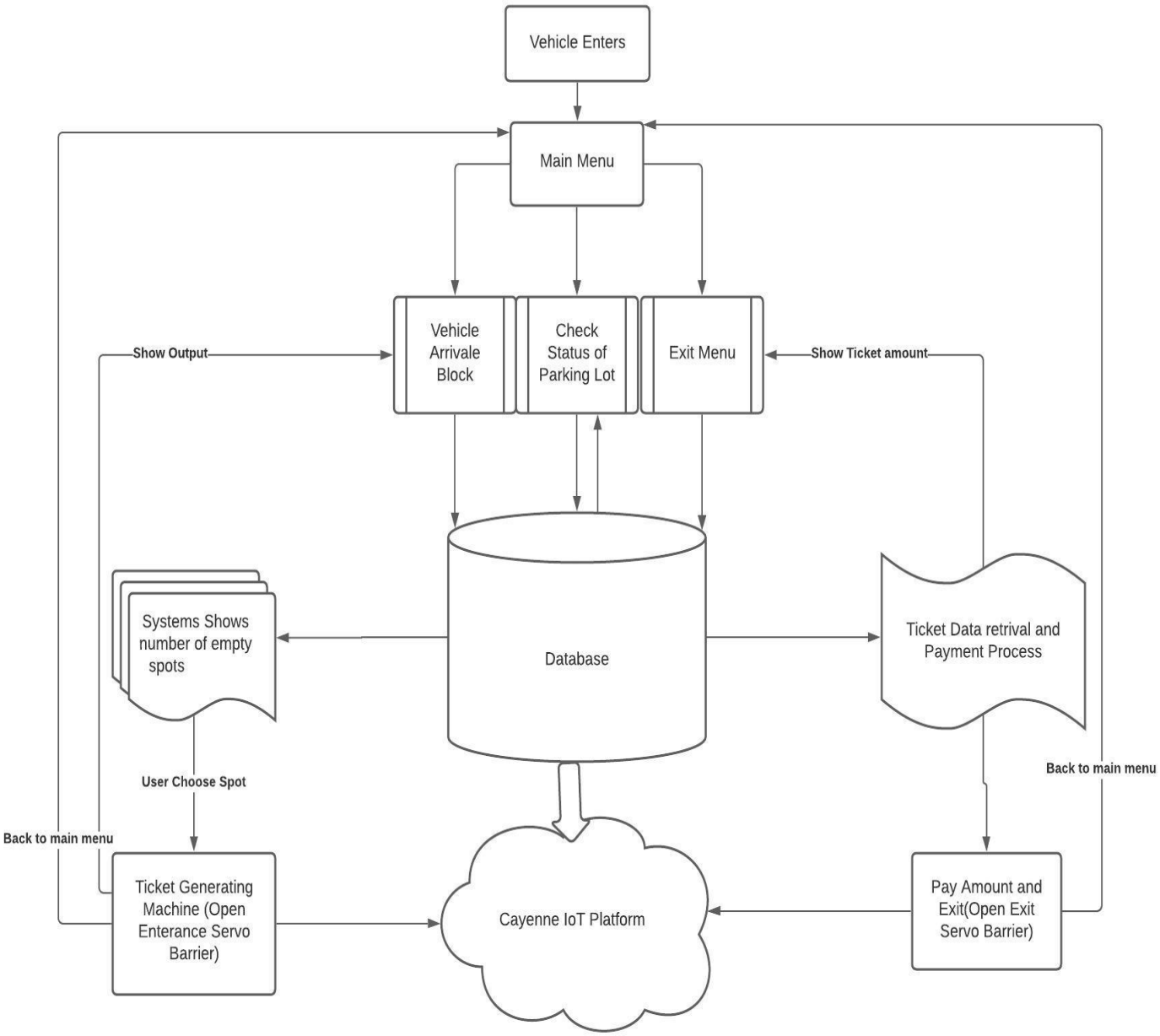

Fig 8: Block Diagram

1. Vehicle Arrival

When a Vehicle arrives the Screen will show the number of empty spots remaining. Driver will then select the prefered spot as per convenience and the vehicle type(4-Wheeler/2-Wheeler) after which the system will generate a ticket and the vehicle is all set to park!

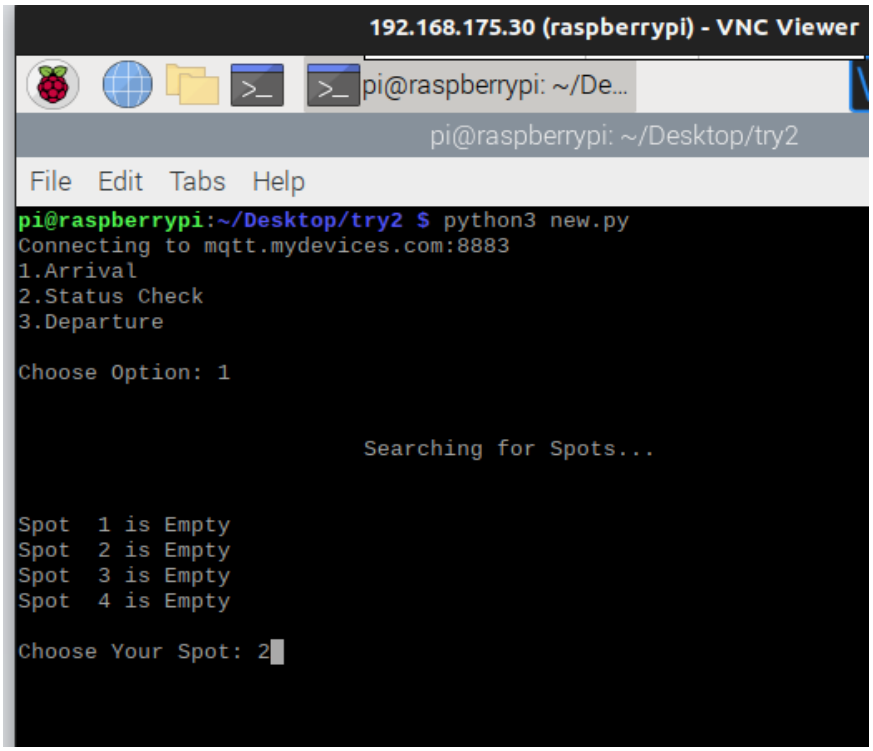

Fig 9: System Showing number of empty spots

Fig 10: Cloud Application Dashboard

- On the Cayenne dashboard, all the spots are visible as empty. All the spots have a key I'D that is connected to the Physical sensor. When a particular spot is reserved physically, the same will be updated on the server.

Cloud Connection via MQTT Protocol Python Script:

```
MQTT_USERNAME  = "cc19bxxxxxxxxxxxxxxxxxxxxxf1a8f1211ba"
MQTT_PASSWORD  = "b6f7xxxxxxxxxxxxxxxxxxxxxxxxxx2ff9bba1a"
MQTT_CLIENT_ID = "80839f8xxxxxxxxxxxxxxxxxxxxa624e442"
client = cayenne.client.Cayenne MQTT Client()
client.begin(MQTT_USERNAME, MQTT_PASSWORD, MQTT_CLIENT_ID,
port=8883)
def reserve(i):
    client.loop()
    client.virtualWrite(i, 1, "Spot Filled", "d")
def empty(j):
    client.loop()
    client.virtualWrite(j, 0, "Spot Empty", "d")
```

Full Python Project Code at: https://github.com/aryanboss/Smart-Parking-System

- Spot 2 is selected:

  If the connection is successful, the system will then send a signal to the reserved spot channel so that it can be updated on the server.

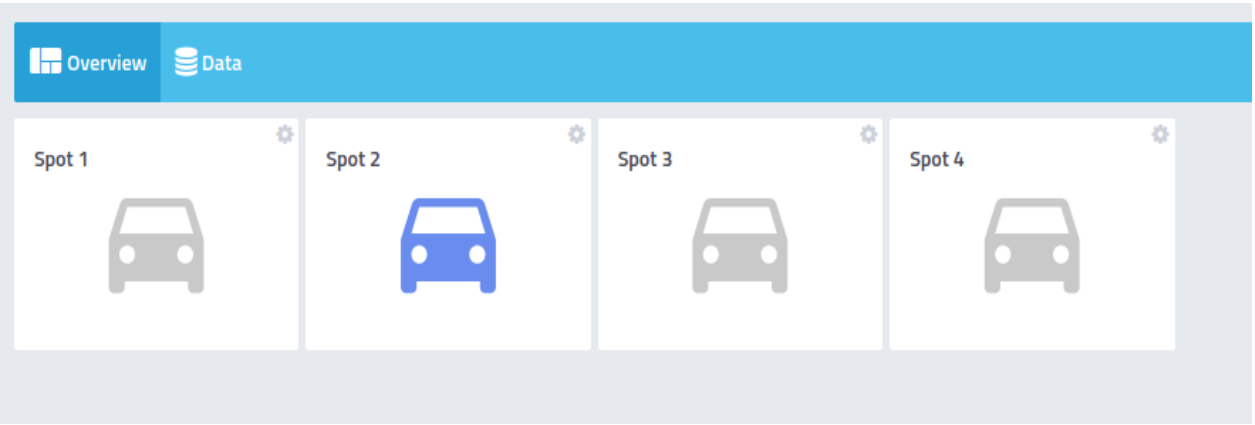

Fig 11: Cloud Application Dashboard

```
Choose Your Vehicle Type Car/Bike: IoT Bike

Please Wait! Your Ticket is being Generated!
Ticket generated successfully!

Opening Gate, Please Collect Your Ticket and Enter
Closing Gate
1.Arrival
2.Status Check
3.Departure
```

Fig 12: Confirmation Message

After Successful reservation, the system will provide the user with a Token

The Ticket generated will have entrance time, Vehicle Type, ticket number, date and QR Code on to it.

Fig 13: Generated Ticket

2. Status Menu

This menu is provided so that one can check the current status of the parking lot and how many spots are vacant/filled.

Fig 14: Status Check

3. Exit Menu

While Checking-out, the driver needs to go to a payment machine where he/she will either have to scan the QR code that is on the ticket or enter the ticket number. The system will retrieve back the entrance time of that particular vehicle associated with that ticket and calculate the charges as per the vehicle size, as different types of vehicles will have different rates. The Driver will then have to pay the full amount in Cash or through an online payment method. After successful payment the driver will be provided with a payment slip and will then later be used at the last exit barrier for payment confirmation.

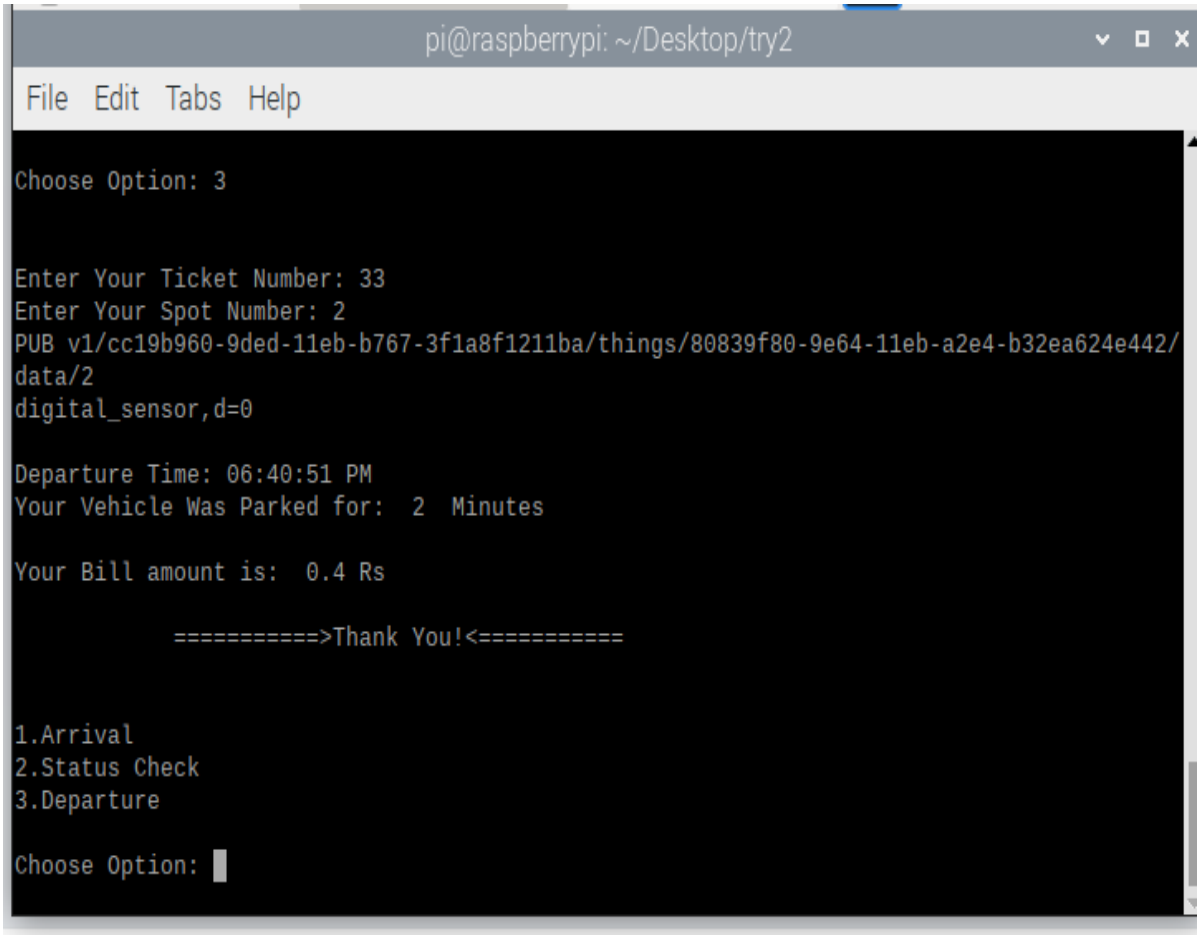

Fig 15: Exit Confirmation Message

The Workflow of Project:

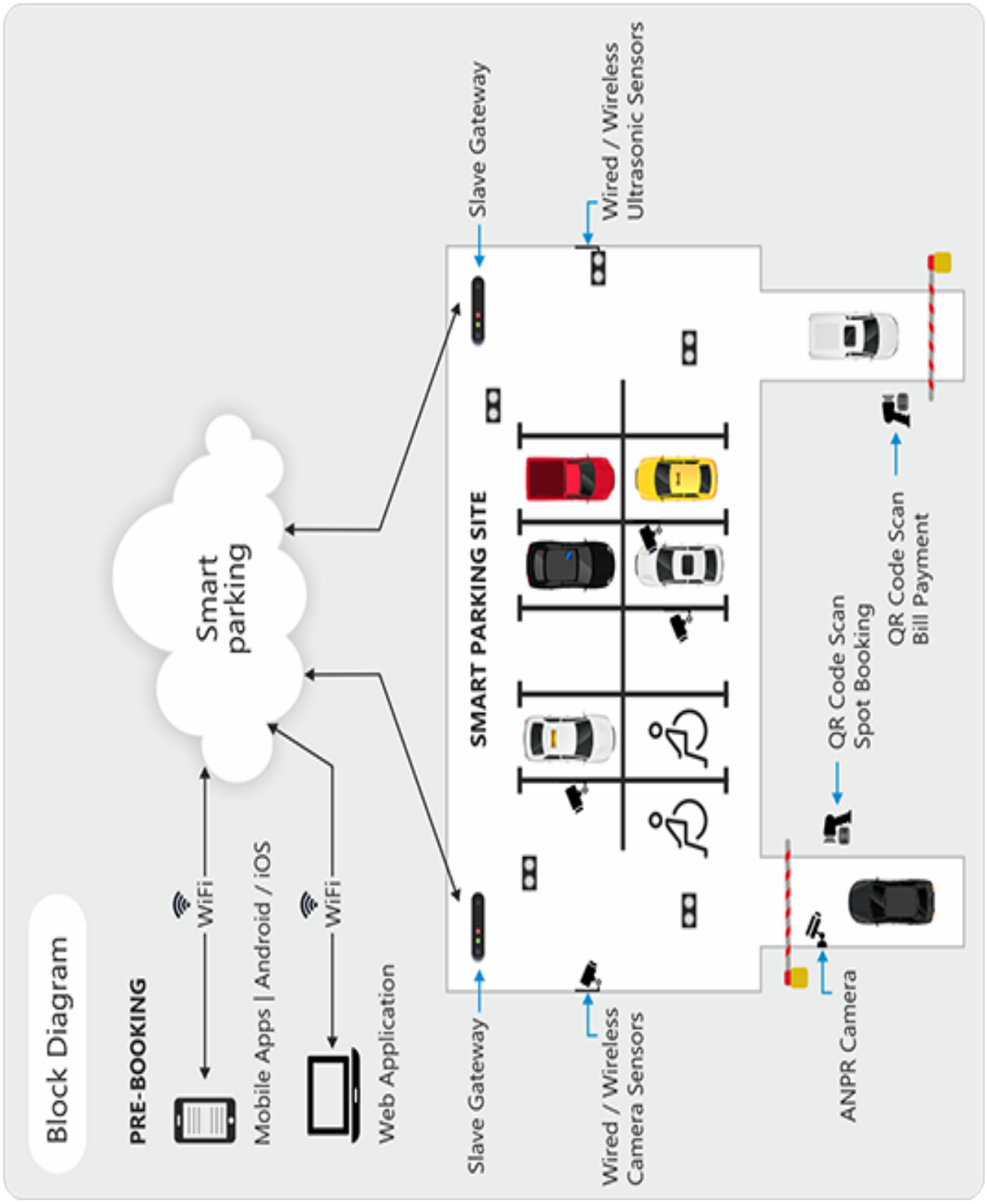

Fig 16: Workflow

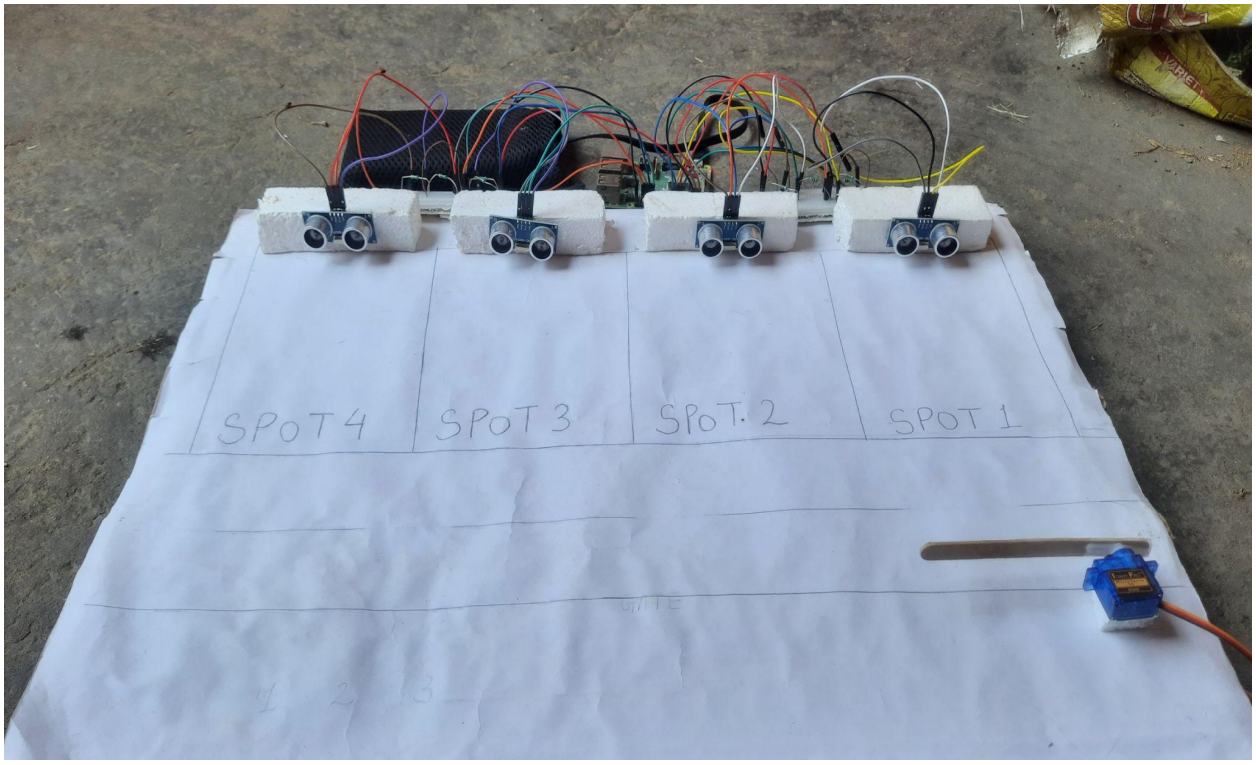

Fig 17(a): Hardware Prototype

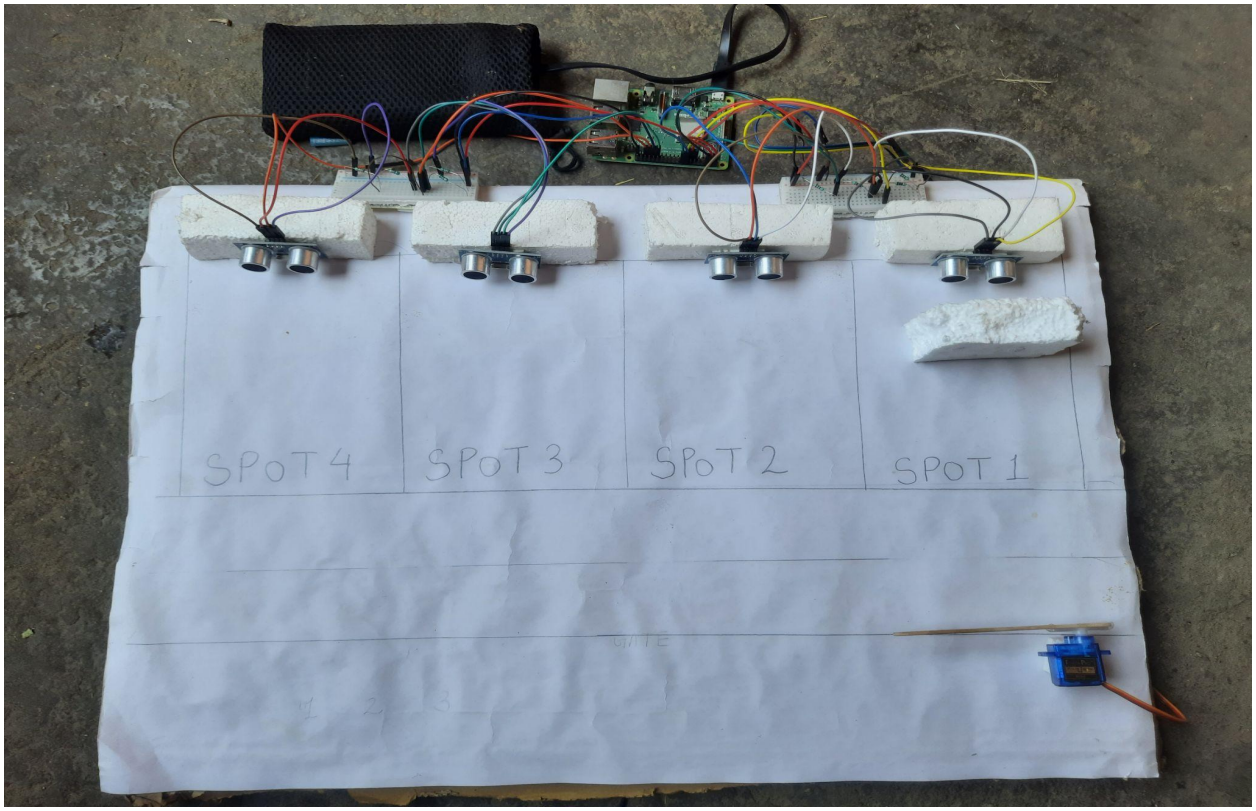

Fig 17(b): Hardware Prototype

RESULTS

- Accurately predict and sense spot/vehicle occupancy in real-time.
- Guides residents and visitors to available parking.
- Provides optimize Parking Space Usage, tools to optimize workforce management.
- Simplifies the parking experience and adds value for parking stakeholders, such as drivers and merchants.
- Help traffic in the city flow more freely leveraging IoT technology.
- Enables intelligent decisions using data, including real–time status applications and historical analytics reports.
- Smart Parking enables better and real time monitoring and managing of available parking space , resulting in significant revenue generation.

Conclusion

Smart Parking and its Smart Parking Sensors can be seen as a part of smart cities. These smart cities are cities that are driven by an IT infrastructure and by using this infrastructure, cities can enhance the quality of life and improve economic development for its inhabitants. By collecting this data, cities can analyse how processes, like parking, can be optimized. In this project, IoT enabled sensors and Online IoT platform is used to realise this real time project The smart parking system based on IoT concept has been implemented using sensors, raspberry pi and cloud platform. This is an efficient system for smart parking which reduces traffic congestion. This work is further extended by adding Temperature and LDR sensors for monitoring the temperature and lightning g conditions of the parking lot.

ACKNOWLEDGEMENT

First and foremost, we concede the surviving presence and the flourishing refinement of the Almighty God for his concealed hand yet substantial supervision all through the design project.

We express our sincere thanks and deep sense of gratitude to the Director of this institution **PROF KAVI MAHESH** for providing all the necessary facilities.

We thank **DR DEEPAK KT ,** Head of the Department and our design project guide, for all the words of inspiration.

Next, we would like to thank our design project coordinator **DR SIBASANKAR PADHY** and our supervisor **DR MUKESH KUMAR MISHRA** and **DR CHINMAYANANDA A** for their valuable advice.

We have great pleasure to express our deep sense of gratitude and obligation to our design project guide **DR DEEPAK KT** for his valuable guidance and suggestions throughout the preparation of the design project.

Last but certainly not the least; we would also like to thank all faculties of **ELECTRONICS AND COMMUNICATIONS** Department and our friends for their help and cooperation.

ABDUL RAHMAN M

AJMAL A

ARYAN KUMAR

LIKHITH KUMAR

Appendix I

- To find the amount of ticket:
    - Rate : (0.5 Per Minute)
    - Variable : Hour
    - Variable : Minute
    - Variable : TotalTime
    - Variable: TicketFare

Number of Hours Car Parked is:

- DepartureHour - ArrivalHour

- Number of Minutes Car Parked is:
    - If ArrivalMinute > DepartureMinute
        - Minute = ArrivalMinute - DepartureMinute
        - TotalTime = 60 X Hour - Minute
    - If ArrivalMinute < DepartureMinute:
        - Minute = DepartureMinute - ArrivalMinute
        - TotalTime = 60 X Hour + Minute

- Total Fare :
    - TicketFare = TotalTime X Rate

References

- ResearchGate Project
    - https://www.researchgate.net/publication/313667380_Smart_Parking_System_Student_Activity_Project
- Traffic InfraTech Magazine
    - https://www.trafficinfratech.com/interview-providing-smart-vehicle-parking-management-system/
- Online Article
    - https://mobidev.biz/blog/iot-based-smart-parking-system
- SmartParking System
    - https://www.smartparking.com/smartpark-system
- Stanford Research Paper
    - https://stacks.stanford.edu/file/druid:yj296hj2790/Li_Jones_Vacant_Parking_Space_Detection_Algorithm.pdf