Department of Computer Science

CS5453 - Internet of Things

Smart Car Parking System

Author:
Reshu Verma
Nikhil Dinesh Prasad
Shubham Lahoti

01 May, 2021

CONTENTS

LIST OF FIGURES

1 PROBLEM STATEMENT

The basic idea of the project is to:

- Implement smart parking system capable of identifying free space in parking lot.
- Support OCR based car search(actual OCR implementation is not required).
- A simple command line or graphical user interface for control of the system.

What are the chief requirements of the car parking system?

- When asked about free space in the parking lot, the system should list all the free spaces in the parking lot.
- When asked about the location of a parked car, the system should ask for 4 digits car number and list all the cars with a matching car number.

Deliverables:

- Simulating parking space sensor
- Images and Message Logs captured during the course of the activity
- Project Report & Source Code

2 METHODOLOGY

Intelligent parking system requires data to be independent bidirectional flow - between sensor and vehicle owner. Sensor has to maintain a database comprising records of vehicle numbers and the parking slots allocated to them. The database is to be updated after a new vehicle is parked or a vehicle leaves the space. MQTT has a lightweight release and subscription mechanism, which supports bidirectional flow. So, we use a Python-based open source tool Paho to implement the MQTT client and sensor.

There are 3 types of queries possible in the smart car parking space :

- Finding and choosing a parking slot.
- Inquiring the parking slot of a particular vehicle number
- Clearing the parking slot

In order to accomplish the above tasks, the car owner needs to subscribe to the following topics:

- **free.space/pub -** when the vehicle owner wishes to park his vehicle in a desired slot.
- **carlocation/pub -** when the vehicle owner wishes to know the parking slot of his vehicle.
- **location/pub -** when the vehicle owner wants to vacate the parking slot.

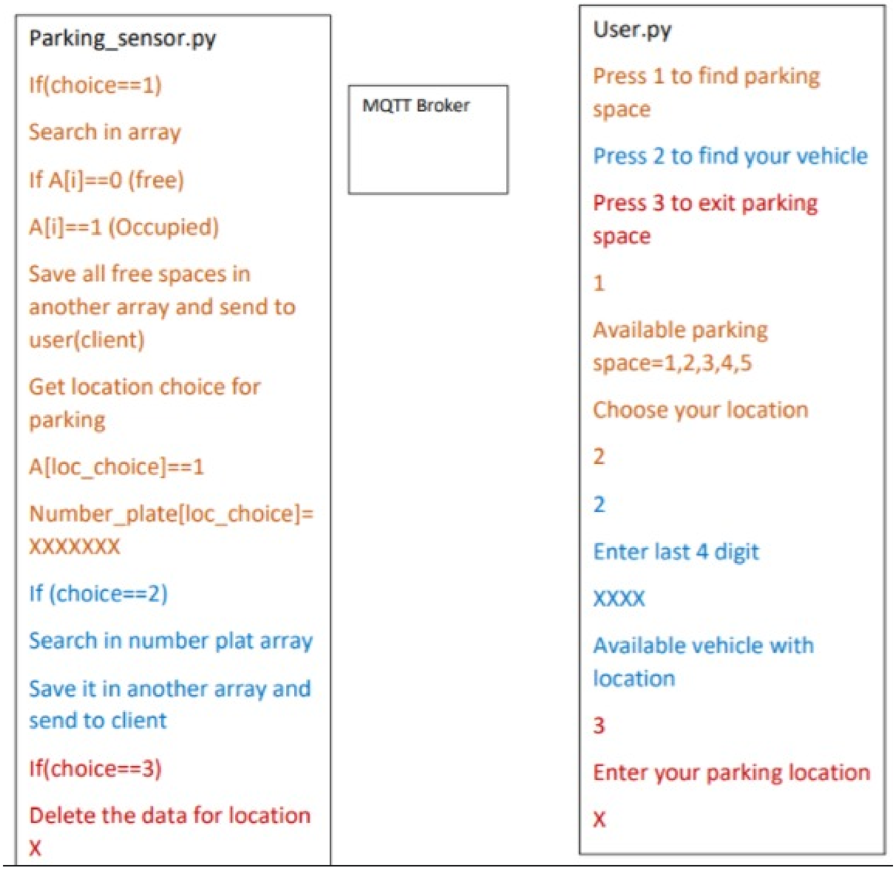
Correspondingly, the parking sensor needs to subscribe to the following topics:

- **location/sub -** sensor accepts
- **location/sub55 -** sensor collects the last 4 digits of vehicle whose location is to be found.
- **location/sub3 -** sensor collects vehicle number and updates database after vehicle vacates the space.

Dictionary is used in order to store the parking slot number of vehicle as :

$$\{\, Vehicle\ number\ :\ Parking\ space \,\}$$

The basic structure of the intelligent parking system is displayed below :



Parking_sensor.py

If(choice==1)

Search in array

If A[i]==0 (free)

A[i]==1 (Occupied)

Save all free spaces in another array and send to user(client)

Get location choice for parking

A[loc_choice]==1

Number_plate[loc_choice]=XXXXXXX

If (choice==2)

Search in number plat array

Save it in another array and send to client

If(choice==3)

Delete the data for location X

MQTT Broker

User.py

Press 1 to find parking space

Press 2 to find your vehicle

Press 3 to exit parking space

1

Available parking space=1,2,3,4,5

Choose your location

2

2

Enter last 4 digit

XXXX

Available vehicle with location

3

Enter your parking location

X

Figure 2.1: Structure of Intelligent parking system

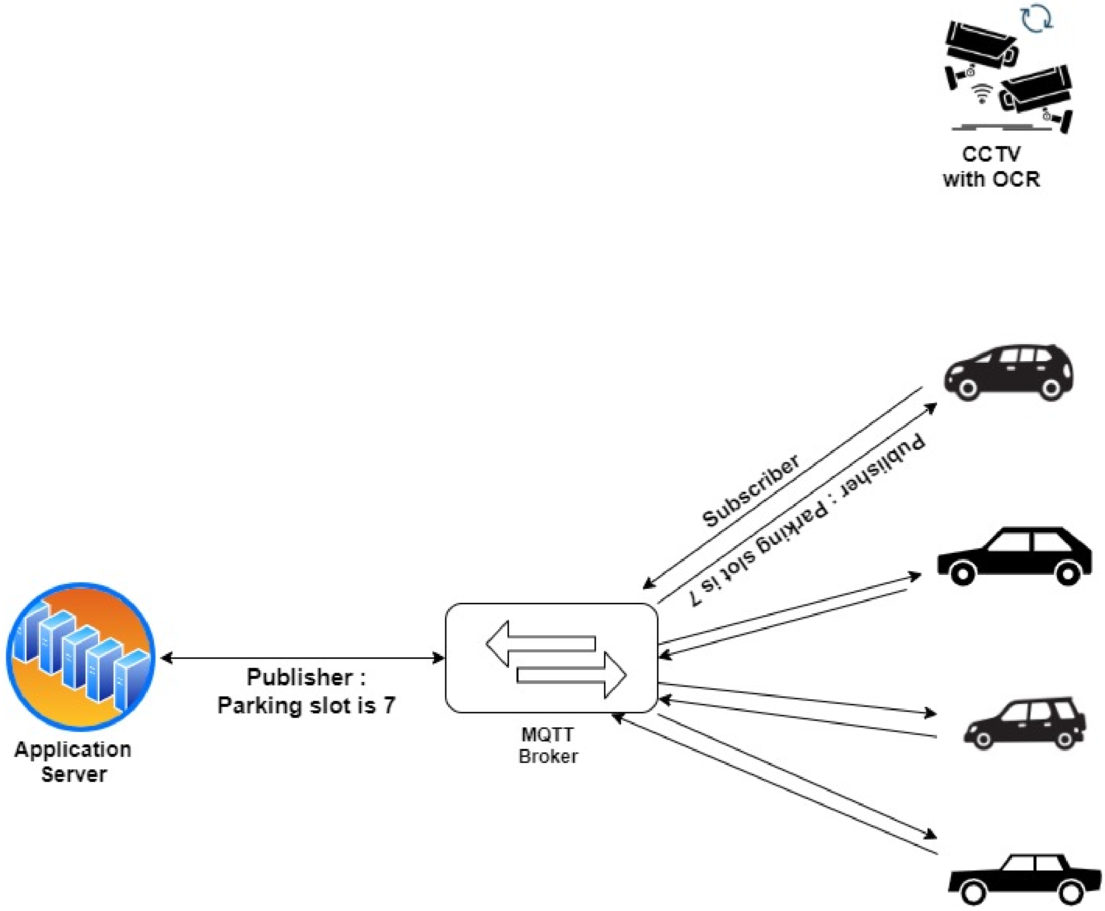
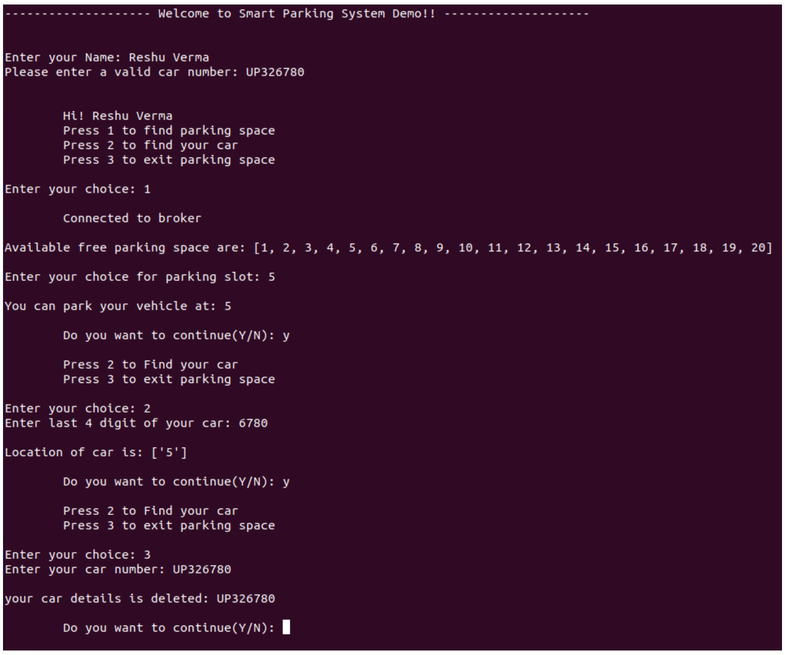


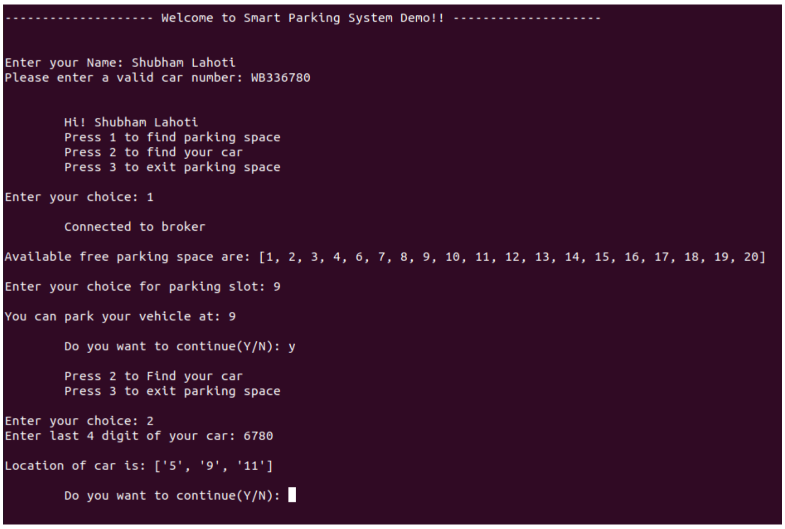
Figure 2.2: Working of MQTT Broker

3 Illustrations



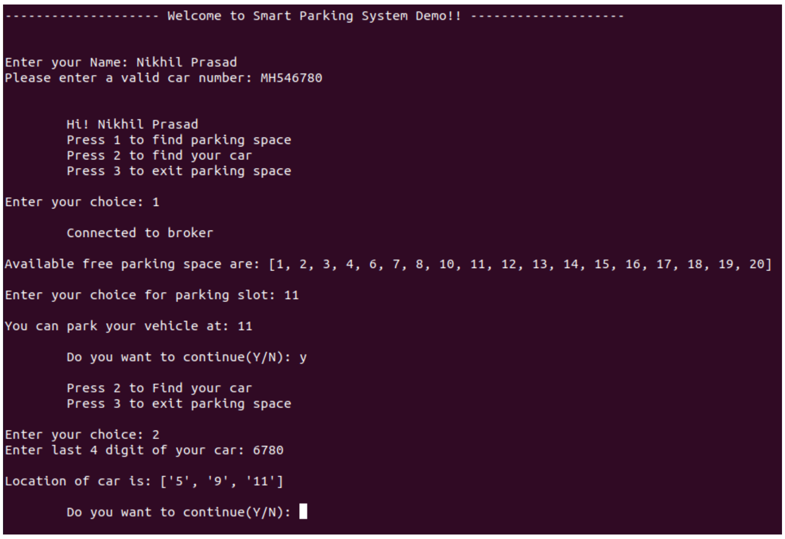
```
------------------- Welcome to Smart Parking System Demo!! -------------------


Enter your Name: Reshu Verma
Please enter a valid car number: UP326780


        Hi! Reshu Verma
        Press 1 to find parking space
        Press 2 to find your car
        Press 3 to exit parking space

Enter your choice: 1

        Connected to broker

Available free parking space are: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

Enter your choice for parking slot: 5

You can park your vehicle at: 5

        Do you want to continue(Y/N): y

        Press 2 to Find your car
        Press 3 to exit parking space

Enter your choice: 2
Enter last 4 digit of your car: 6780

Location of car is: ['5']

        Do you want to continue(Y/N): y

        Press 2 to Find your car
        Press 3 to exit parking space

Enter your choice: 3
Enter your car number: UP326780

your car details is deleted: UP326780

        Do you want to continue(Y/N):
```

Figure 3.1: Smart parking demo for Reshu Verma



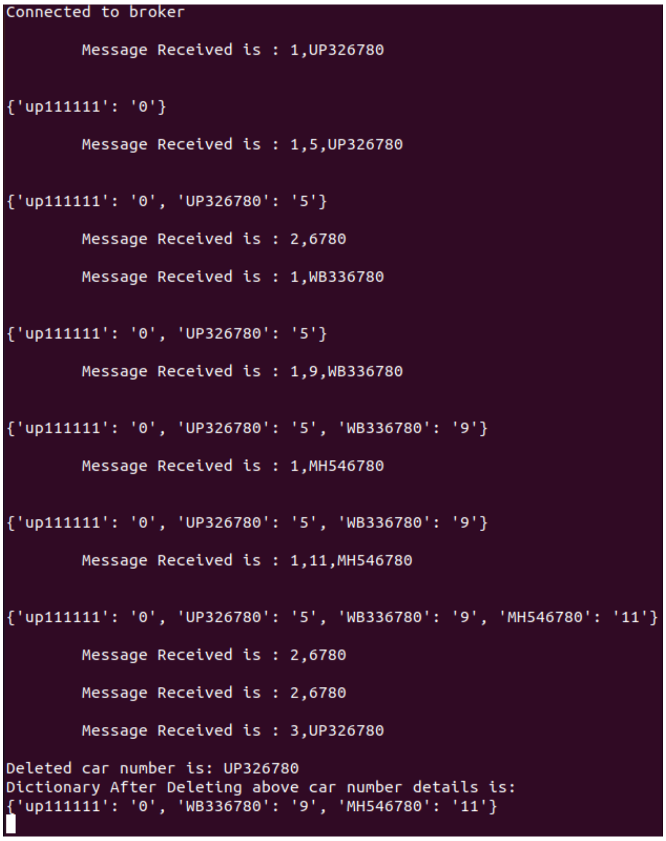
```
------------------- Welcome to Smart Parking System Demo!! -------------------


Enter your Name: Shubham Lahoti
Please enter a valid car number: WB336780


        Hi! Shubham Lahoti
        Press 1 to find parking space
        Press 2 to find your car
        Press 3 to exit parking space

Enter your choice: 1

        Connected to broker

Available free parking space are: [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

Enter your choice for parking slot: 9

You can park your vehicle at: 9

        Do you want to continue(Y/N): y

        Press 2 to Find your car
        Press 3 to exit parking space

Enter your choice: 2
Enter last 4 digit of your car: 6780

Location of car is: ['5', '9', '11']

        Do you want to continue(Y/N):
```

Figure 3.2: Smart parking demo for Shubham Lahoti



```
------------------- Welcome to Smart Parking System Demo!! -------------------


Enter your Name: Nikhil Prasad
Please enter a valid car number: MH546780


        Hi! Nikhil Prasad
        Press 1 to find parking space
        Press 2 to find your car
        Press 3 to exit parking space

Enter your choice: 1

        Connected to broker

Available free parking space are: [1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

Enter your choice for parking slot: 11

You can park your vehicle at: 11

        Do you want to continue(Y/N): y

        Press 2 to Find your car
        Press 3 to exit parking space

Enter your choice: 2
Enter last 4 digit of your car: 6780

Location of car is: ['5', '9', '11']

        Do you want to continue(Y/N): █
```

Figure 3.3: Smart parking demo for Nikhil Prasad

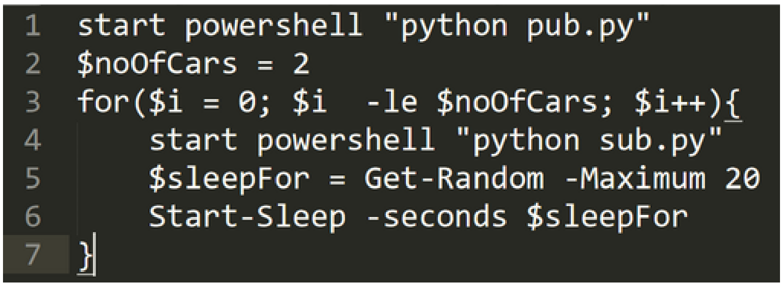


```
Connected to broker

        Message Received is : 1,UP326780


{'up111111': '0'}

        Message Received is : 1,5,UP326780


{'up111111': '0', 'UP326780': '5'}

        Message Received is : 2,6780

        Message Received is : 1,WB336780


{'up111111': '0', 'UP326780': '5'}

        Message Received is : 1,9,WB336780


{'up111111': '0', 'UP326780': '5', 'WB336780': '9'}

        Message Received is : 1,MH546780


{'up111111': '0', 'UP326780': '5', 'WB336780': '9'}

        Message Received is : 1,11,MH546780


{'up111111': '0', 'UP326780': '5', 'WB336780': '9', 'MH546780': '11'}

        Message Received is : 2,6780

        Message Received is : 2,6780

        Message Received is : 3,UP326780

Deleted car number is: UP326780
Dictionary After Deleting above car number details is:
{'up111111': '0', 'WB336780': '9', 'MH546780': '11'}
█
```

Figure 3.4: Terminal view of sensor

4 IMPLEMENTATION

OCR : We have written a powershell script which will detect a car which comes at parking space in random sequence. In case of hardware implementation, the logic would be similar and performed by hardware camera module.



```
1 start powershell "python pub.py"
2 $noOfCars = 2
3 for($i = 0; $i  -le $noOfCars; $i++){
4     start powershell "python sub.py"
5     $sleepFor = Get-Random -Maximum 20
6     Start-Sleep -seconds $sleepFor
7 }
```

Figure 4.1: Code snippet for OCR

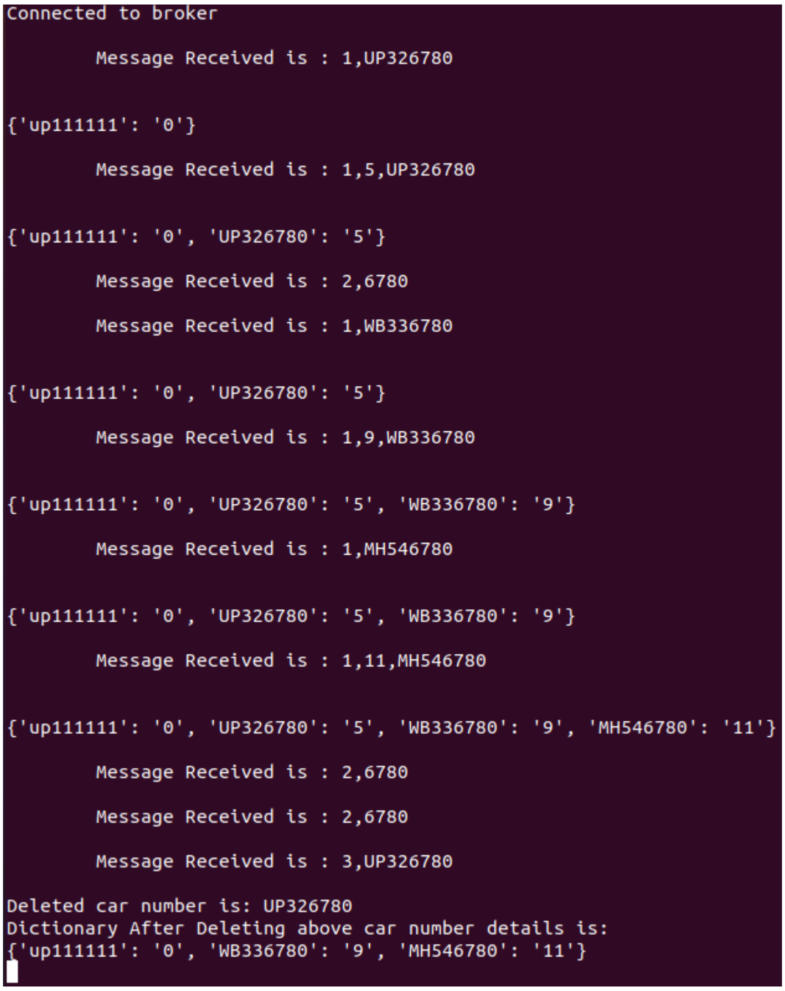
In the above code snippet we could see that initially we are executing publisher code named as pub.py in each line 1. In line 2, we initialized the number of cars which will come at the parking gate randomly. To get a car at the parking gate, the loop which will run sub.py at random interval between 0-20 seconds.

Pub.py
The main responsibility of pub.py is to:

- To distribute current available parking space when requested by the user
- Allocate new cars to a free parking space selected by the user.
- Free the parking space, when a car leaves the parking.

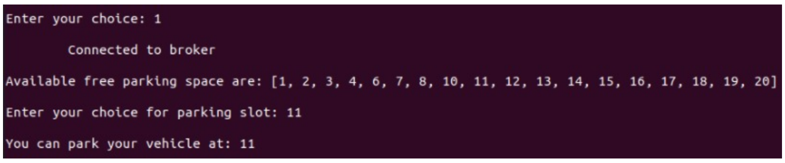
Sub.py :
The main responsibility of sub.py is to :

- Request sub for currently available parking space.
- Inquire a parking slot corresponding to a given vehicle number.
- Free the parking space, when a car leaves the parking.



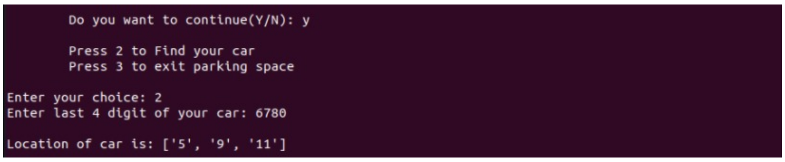
```
Connected to broker

        Message Received is : 1,UP326780


{'up111111': '0'}

        Message Received is : 1,5,UP326780


{'up111111': '0', 'UP326780': '5'}

        Message Received is : 2,6780

        Message Received is : 1,WB336780


{'up111111': '0', 'UP326780': '5'}

        Message Received is : 1,9,WB336780


{'up111111': '0', 'UP326780': '5', 'WB336780': '9'}

        Message Received is : 1,MH546780


{'up111111': '0', 'UP326780': '5', 'WB336780': '9'}

        Message Received is : 1,11,MH546780


{'up111111': '0', 'UP326780': '5', 'WB336780': '9', 'MH546780': '11'}

        Message Received is : 2,6780

        Message Received is : 2,6780

        Message Received is : 3,UP326780

Deleted car number is: UP326780
Dictionary After Deleting above car number details is:
{'up111111': '0', 'WB336780': '9', 'MH546780': '11'}
```

Figure 4.2: Terminal view of sensor



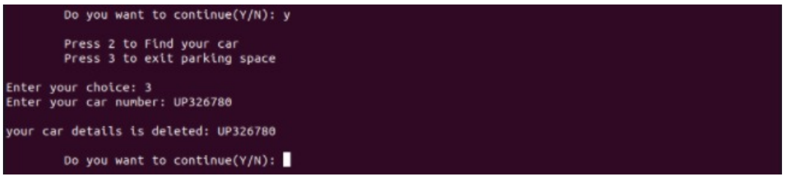
```
Enter your choice: 1

        Connected to broker

Available free parking space are: [1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

Enter your choice for parking slot: 11

You can park your vehicle at: 11
```

Figure 4.3: Request to find free parking space



```
        Do you want to continue(Y/N): y

        Press 2 to Find your car
        Press 3 to exit parking space

Enter your choice: 2
Enter last 4 digit of your car: 6780

Location of car is: ['5', '9', '11']
```

Figure 4.4: Inquire a parking slot



```
        Do you want to continue(Y/N): y

        Press 2 to Find your car
        Press 3 to exit parking space

Enter your choice: 3
Enter your car number: UP326780

your car details is deleted: UP326780

        Do you want to continue(Y/N): 
```

Figure 4.5: Exit the parking space

5 CONCLUSION

Based on the team's analysis, the smart parking system can provide free parking spaces, and helps in data sharing between the parking lot and build a IOT platform. In order to reduce human intervention in this system, we could employ the following features :

- Mobile self-help payment,
- Use computer vision for swift identification of the vehicle numbers,
- Use an automatic navigation system to help users get to their parking spaces.

6 References

1. Designing of Intelligent Parking Lot Based On MQTT - (ICCNEA 2017)

2. MQTT Assignment name: UAV Flight Control Assignment