School of Information Technology and Engineering (SITE)

M.Tech (Software Engineering)

Project Report

Traffic Control intimation system based on traffic density

Submitted for the Course

SWE4005 Internet of Things

Offered during winter 2017-2018

Prof. Kavitha BR

by

SRIRAM K	**15MIS0173**
SANJANA V	**15MIS0392**
VAISHNAVI CH	**15MIS0420**

April 2018

School of Information Technology & Engineering
M.Tech (Software Engineering) Winter 2017-18

SWE4005 Internet of Things

A Report on the Course Project

Smart Car Parking System

TEAM Number :

Team Member(s) with Reg # and Name :
SRIRAM K - 15MIS0173; 9444321961; sriram.k2015@vit.ac.in

SANJANA V - 15MIS0320; 9445910152; bala.nagasanjana2015@vit.ac.in

VAISHNAVI CH- 15MIS0420; 9989446522; vaishnavi.sandhyarani2015@vit.ac.in

Project Title : Smart Car Parking System

1. Introduction

1.1 Background (System Study Details in brief)

1.2 Problem Statement

- Difficulty in Finding Vacant Spaces Quickly finding a vacant space in a multilevel parking lot is difficult if not impossible, especially on weekends or public holidays.
- Finding spaces during weekends or public holidays can take more than 10 minutes for about 66% of visitors. Stadiums or shopping malls are crowded at peak periods, and difficulty in finding vacant slots at these places is a major problem for customers.
- If a car is parked in such a way that it occupies two parking slots rather than one, this is called improper parking .Improper parking can happen when a driver is not careful about another driver's rights.

1.3 Importance

The main importance of the system is to predict the availability of slots based on the previous data obtained from the parking slot. As we obtain the data set of parked slots of 1 or 2 months of data and on the basis of week, day and timing the slot availability is predicted and said to the customers. Also whoever has the app of our system can be able to reserve the slot even from their home or wherever they are. This also reduces the traffic problems in malls, hospitals, theatres.

2. Overview and Planning

2.1 Proposed System Overview

The smart parking system consists of two IR sensors, Node MCU Wi-Fi module will act as the intermediate between the sensors and the cloud. Also we have RFID sensors to check the vehicle that is being entered into the parking, which also alarms if any terrorist car or theft car detected.

Basically the IR sensors are placed in the parking slot. IR sensor sends a signal or wave, which hits any obstacle or object, and receiver receives the wave. If IR sensor senses any such obstacle, which indicates the presence of car. This data is captured by Node MCU and sent to the cloud server. The users will be provided with an application where he/she can check if any slot is available or not. If any of the slot is available then the user can reserve the slot earlier before he could actually reach the parking point.

2.2 Challenges:

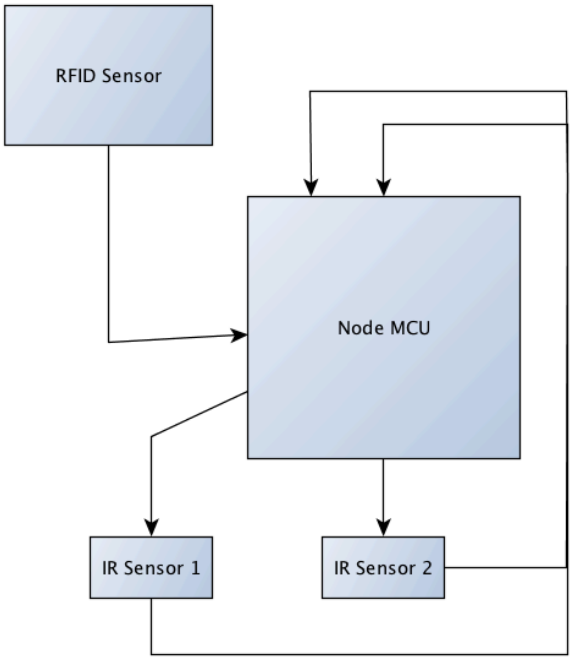
Every time the predictions that are made may not be true but the accuracy can be about 90%. The parking slot detection may fail rapidly in the case of trolley or any other obstacle unknowingly placed by any of the worker of mall or theatre or whatever, which may fail the systems predictions. The main challenge is to provide with the continuous detection of vehicle occupancy and updating status for each and every minute.

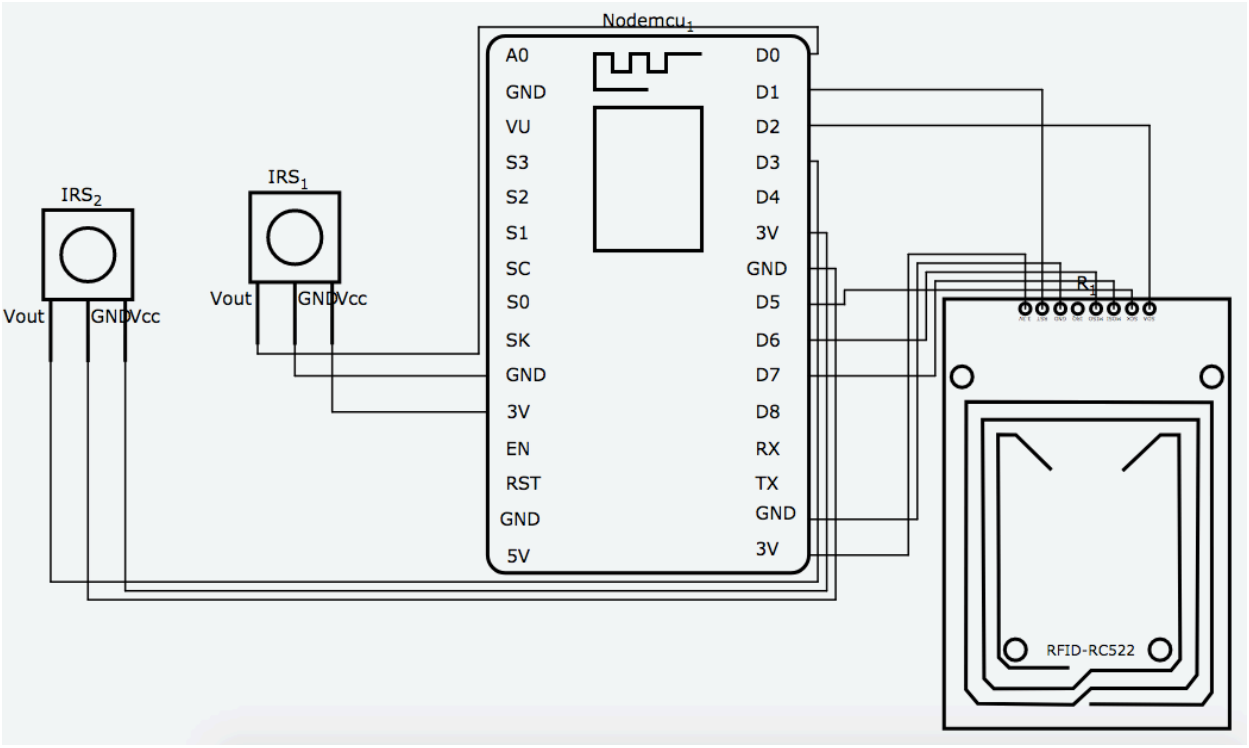
2.3 Assumptions:

No assumptions are assumed.

2.4 Architecture Specifications:

Overall View


RFID Sensor
Node MCU
IR Sensor 1
IR Sensor 2


Nodemcu1
A0
GND
VU
S3
S2
S1
SC
S0
SK
GND
3V
EN
RST
GND
5V
D0
D1
D2
D3
D4
3V
GND
D5
D6
D7
D8
RX
TX
GND
3V
IRS2
Vout
GND
Vcc
IRS1
Vout
GND
Vcc
R1
RFID-RC522

2.4 Hardware Requirements

- Node MCU
- RFID Module MRFC522
- Infrared Sensor(2)
- Connecting Wires

2.5 Software Requirements

Following steps should be performed for software implementation:

- 1. A MQTT dashboard application should be installed in the user's smart phone.
- 2. Once the app is installed and in runnable mode, a project is to be created using an email address.
- 3. After creating a project, an authorization code is sent to the email address through which the project can be linked to the hardware.
- 4. Create Labeled value widgets for displaying the Parking slots and Reservation of parking lot and also a listview to indicate if any suspecting vehicle enters.
- 5. To send the data between the hardware and the app we need internet connection like Wi-Fi or mobile hotspot.

3. IoT Design Methodologies

STEP 1: Purpose and Requirements

Purpose: Parking slot intimation system based on car occupancy to determine the occupancy of car and let the users to reserve for any of the available slot

Behavior: Auto Mode

In auto mode the availability of slot of determined which is then sent to cloud and retrieved by the user.

System Management Requirements: The system should provide remote monitoring and control functions.

Data analysis Requirements:

The system should provide remote monitoring and control functions.

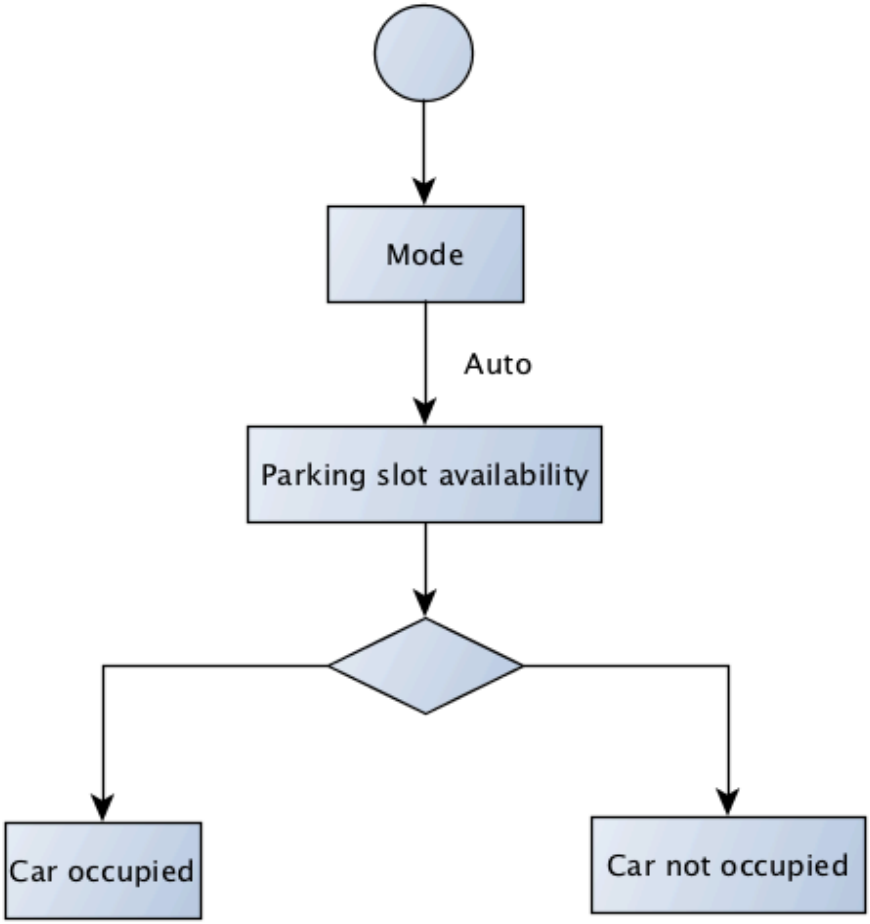
Application Development Requirements: The application should be deployed locally on the

device, but should be accessible remotely.

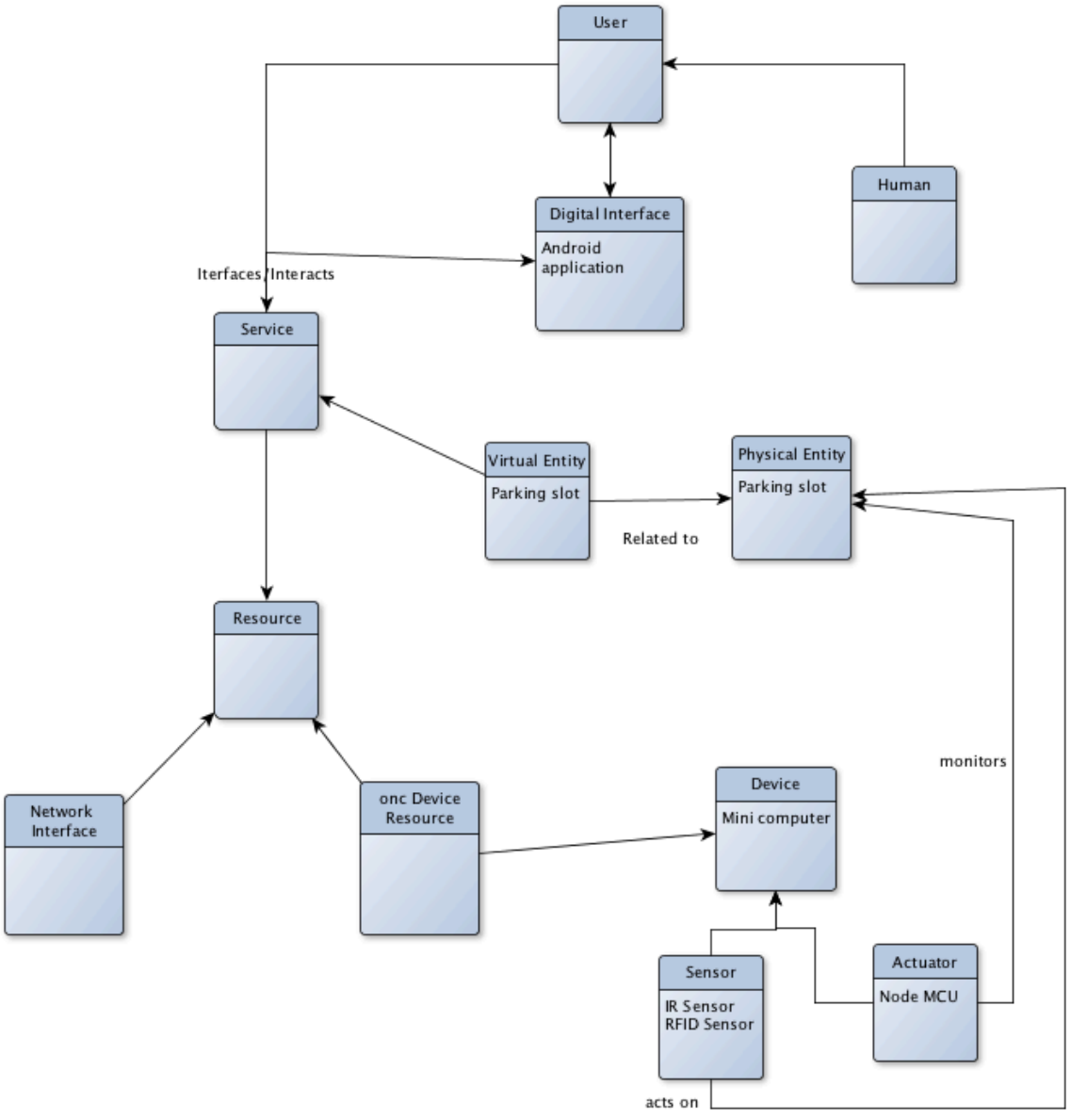
Security Requirement: The system should have basic user authentication capability.

STEP 2: Process Model Specification

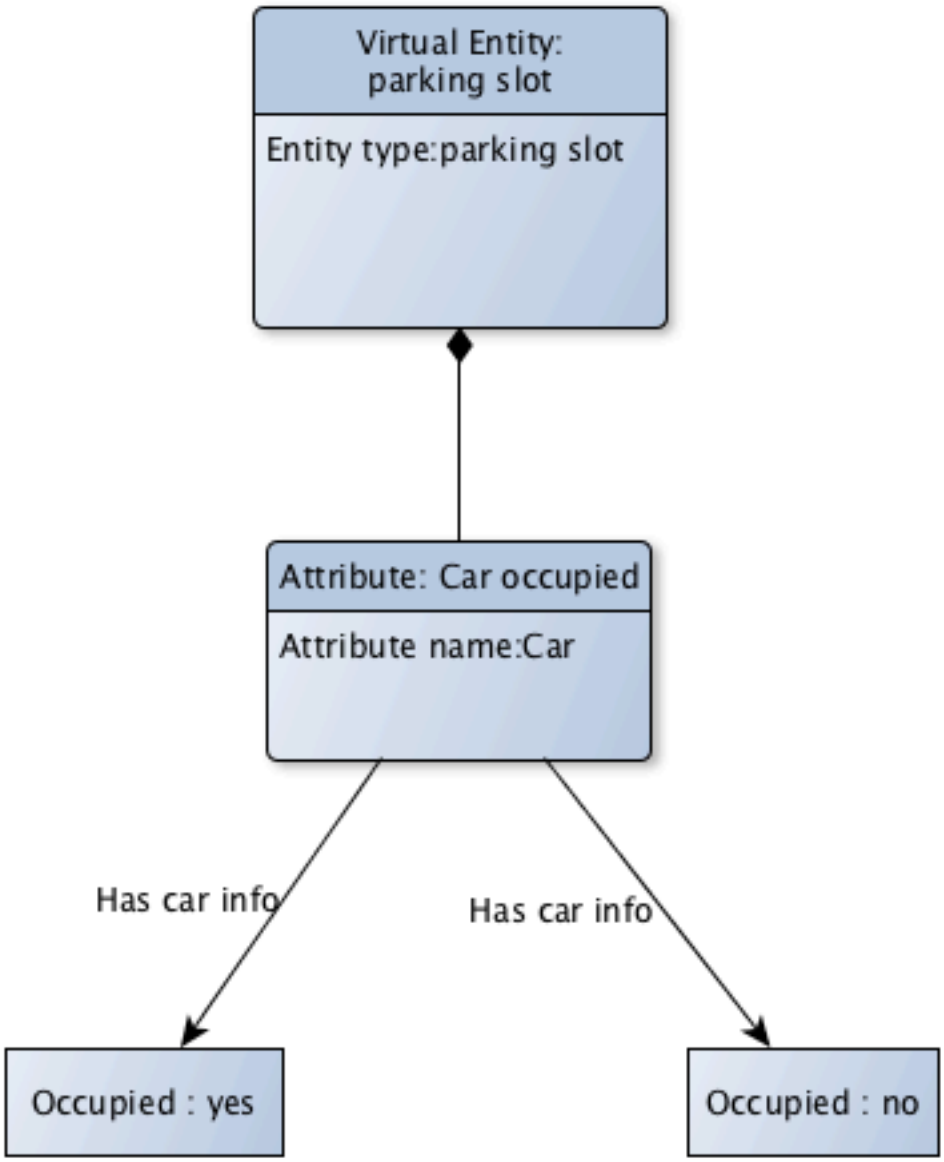
The second step in the IOT design methodology is to define the process specification. In this step, we design the use case of IOT system derived from the purpose and requirement specification.



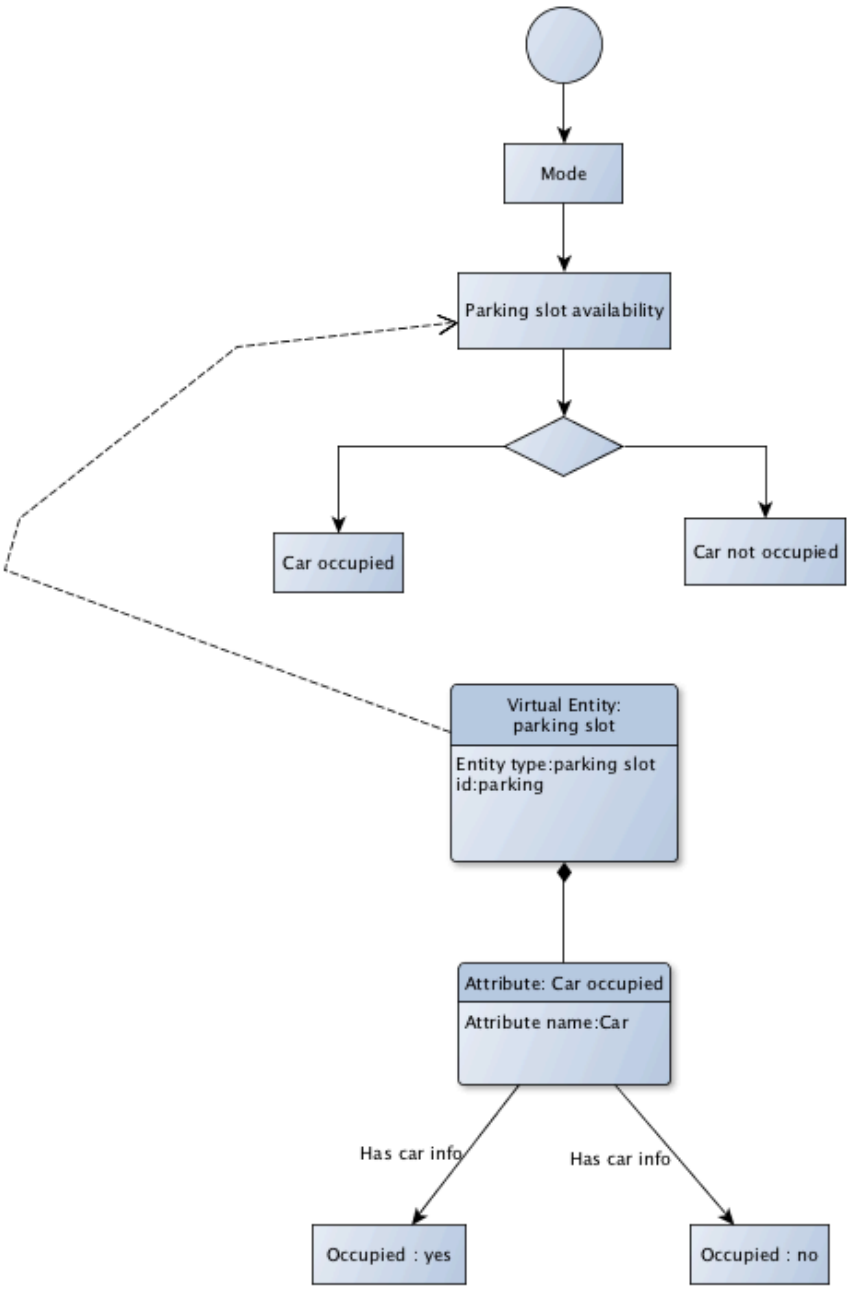
STEP 3: DOMAIN MODEL SPECIFICATION:



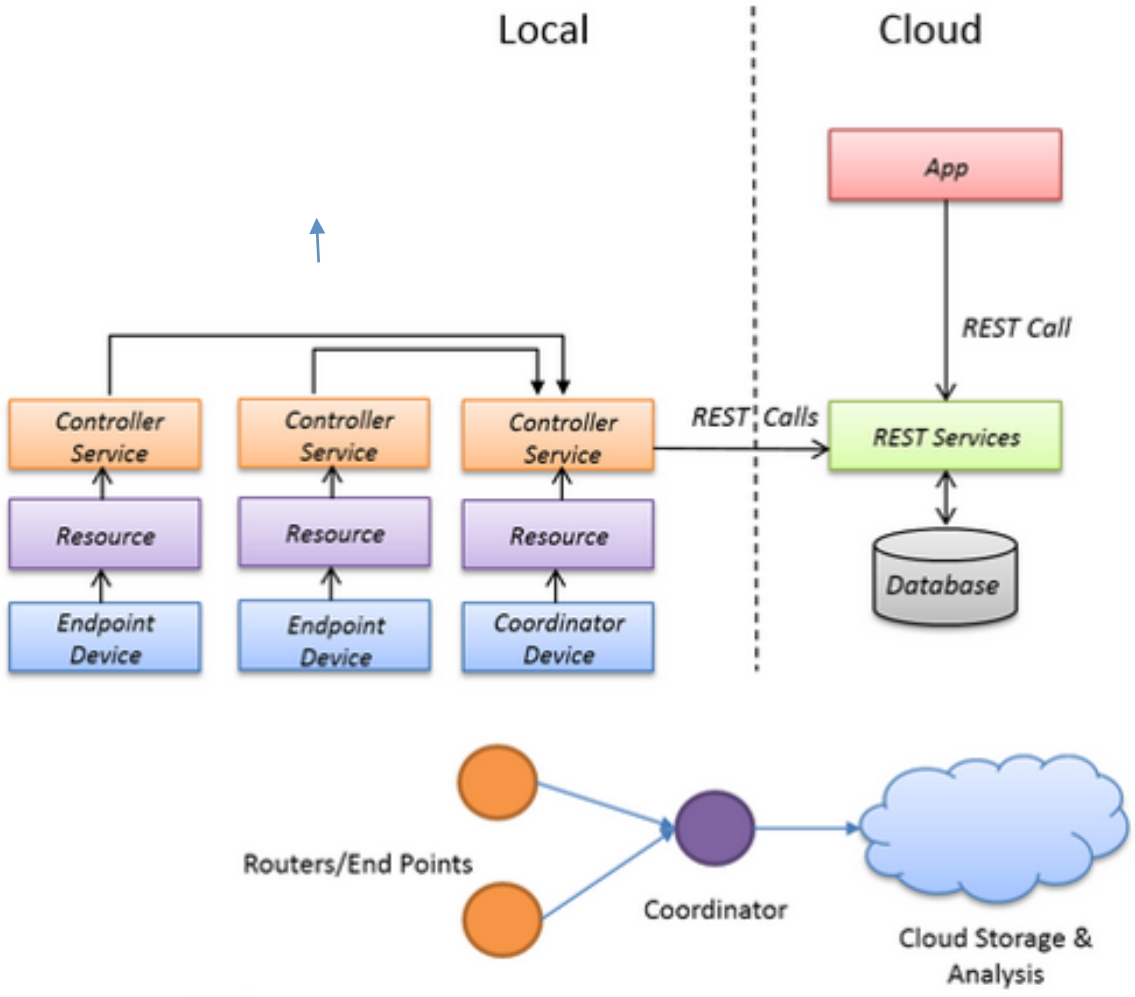
STEP 4: INFORMATION MODEL SPECIFICATION:



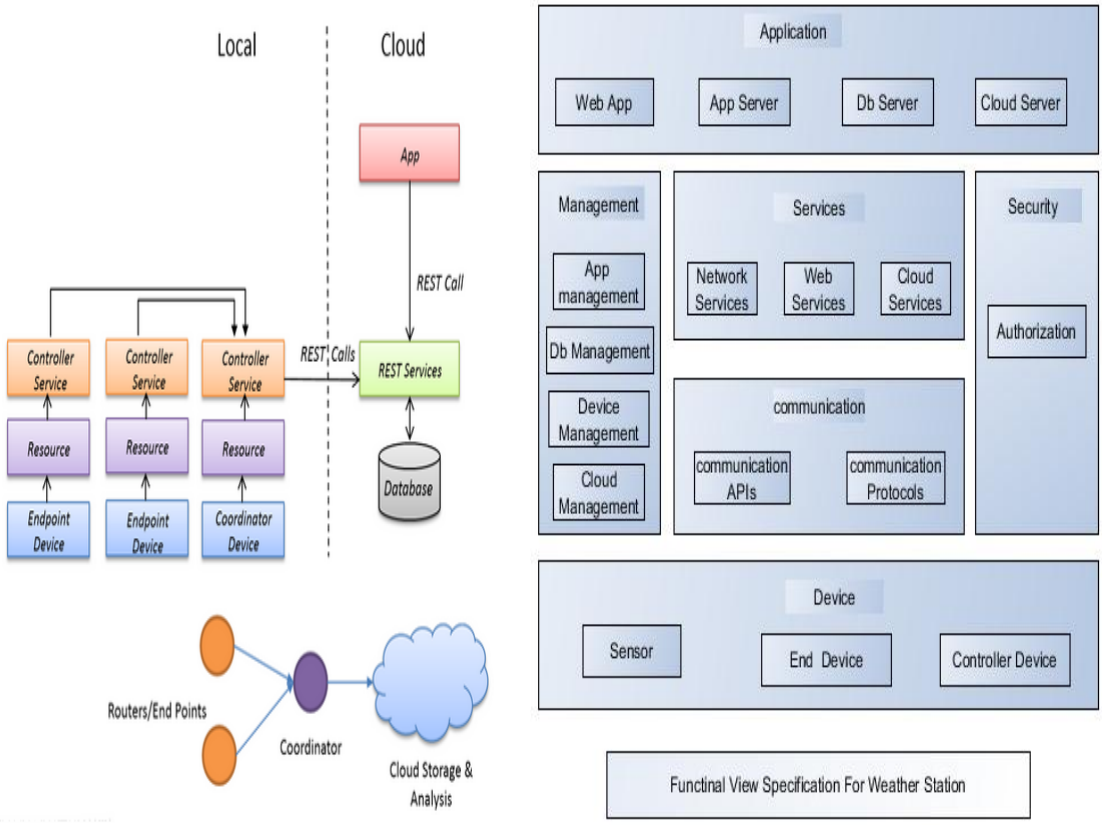
STEP 5: SERVICE SPECIFICATIONS:



STEP 6: IOT LEVEL SPECIFICATION:



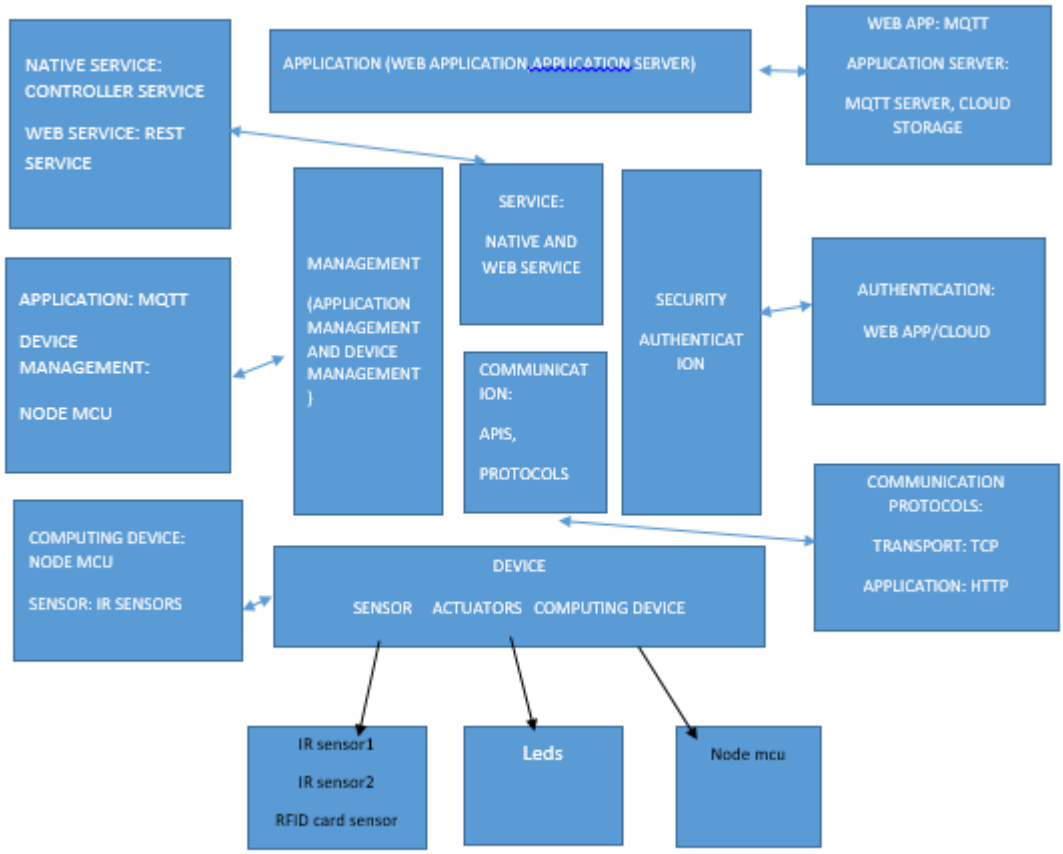
STEP 7: FUNCTIONAL VIEW SPECIFICATION:



Description:

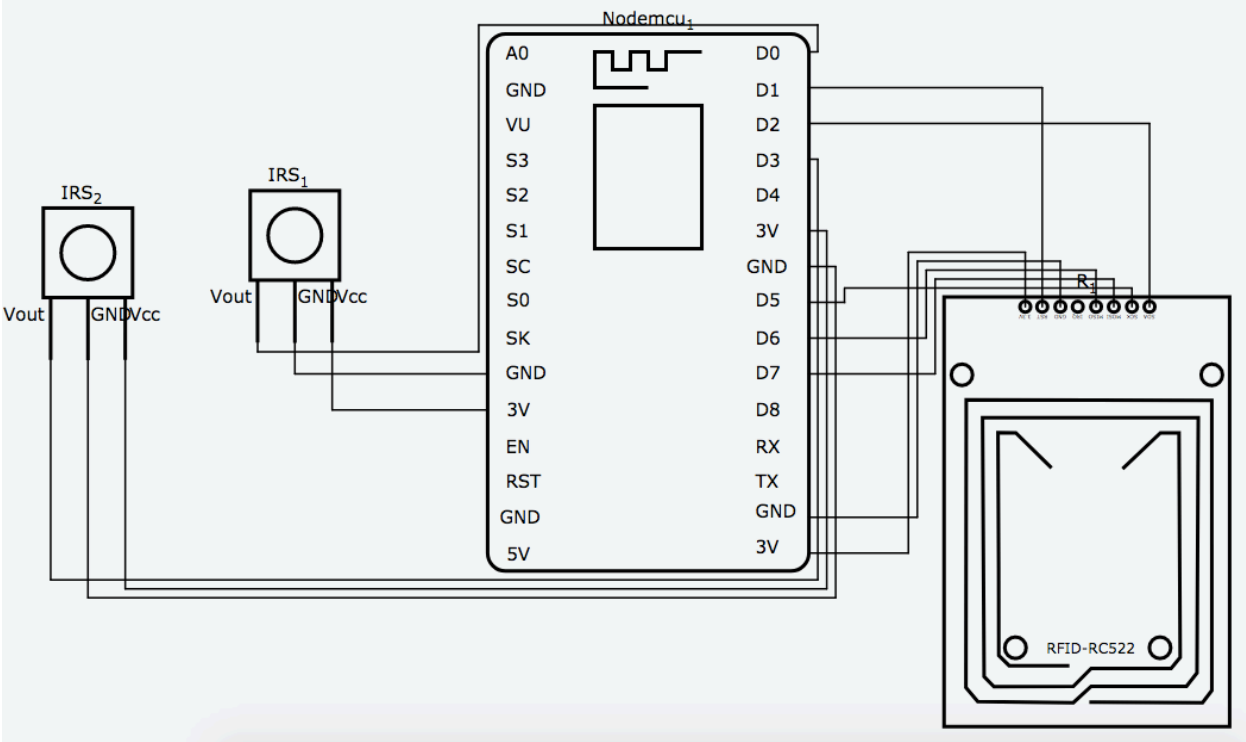
1. IOT devices (2 IR SENSORS, NODE MCU, NFC MODULE) are map with Device.
2. Resources are map to the device and communication API'S
3. Controller services map to services.
4. Rest services map to service (web services)
5. Cloud is map to security and services.
6. Application maps to web application and application server

STEP 8: OPERATIONAL VIEW SPECIFICATION:



STEP 9: DEVICE & COMPONENT INTEGRATION:

Overall View



STEP 10: APPLICATION DEVELOPMENT:

We are using 000webhost server provider to create our own server and an app is created to Monitor the availability of parking slots. Phpmysql is used as a database storage to store and retrieve the elements.

4. System Implementation

4.1 Module Development –Code

Parking Code:

```
#include<ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN D1
#define SS_PIN D2
#define DSEN1_PIN D3
#define DSEN2_PIN D0
MFRC522 mfrc522(SS_PIN, RST_PIN);



int RfidNo = 0;
int x = 0;
int y = 0;
int z = 0;
int amt = 130;
int rej;
String buf;
String slot1;
String slot2;

const char* ssid = "Virus";
const char* password = "123456789";
const char* host = "scps.000webhostapp.com";

long lastMsg = 0;
char msg[50];
int value = 0;
String dsen1;
String dsen2;
int dsenc1;
int dsenc2;


String slrf="0";



WiFiClient client;
```

```
void setup_wifi() {
  delay(100);
  // We start by connecting to a WiFi network
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}


void setup() {
  Serial.begin(115200);
  pinMode(DSEN1_PIN, INPUT);
  pinMode(DSEN2_PIN, INPUT);
  SPI.begin();
  setup_wifi();
  mfrc522.PCD_Init();

}

void loop() {


    host_connect();
    RfidScan();
    readPin();
    delay(1);
    upd();

}

void readPin()
{
  long now = millis();
  if (now - lastMsg > 100) {
    lastMsg = now;
```

```
    dsen1 = digitalRead(DSEN1_PIN);
    dsenc1 = digitalRead(DSEN1_PIN);
    dsen2 = digitalRead(DSEN2_PIN);
    dsenc2 = digitalRead(DSEN2_PIN);
    char message1[4];
    char message2[4];
    dsen1.toCharArray(message1, 4);
    dsen2.toCharArray(message2, 4);
    Serial.print("sen1=");
    Serial.println(dsen1);
    Serial.print("sen2=");
    Serial.println(dsen2);
    if(dsenc1==0)
    {
      Serial.println("HIGH");
      slot1="occupied";
    }
    if(dsenc1==1)
    {
      Serial.println("LOW");
      slot1="avail";
    }
    if(dsenc2==0)
    {
      Serial.println("HIGH");
      slot2="occupied";
    }
     if(dsenc2==1)
    {
      Serial.println("LOW");
      slot2="avail";
    }
  }

}

 void upd()
 {
  if(slrf=="0")
  {
  String url = "/smartcar/insert.php?slot1="+slot1+"&slot2="+slot2;
  Serial.print("Requesting URL: ");
  Serial.println(url);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +"Host: " + host + "\r\n" + "Connection:
close\r\n\r\n");
```

```
  delay(500);

  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
  }
  else if(slrf=="54")
  {
    String url = "/smartcar/insertrf.php?slrf="+slrf;
  Serial.print("Requesting URL: ");
  Serial.println(url);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +"Host: " + host + "\r\n" + "Connection:
close\r\n\r\n");
  delay(500);

  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);

  }
  slrf="0";
  }

  Serial.println();
  Serial.println("closing connection");
  delay(10000);
}

 void host_connect()
{
  Serial.print("connecting to ");
  Serial.println(host);

  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  Serial.println("Connected Successfully to host");
  Serial.println("");
}
```

```
void dump_byte_array(byte *buffer, byte bufferSize)
{

  if ((buffer[0] == 54 ) && (x == 0))
  {
    Serial.println("owner Name   : SRIRAM ");
    Serial.println("Age  : 25 years");
    Serial.println("ACCESS GRANTED");
    slrf="54";
    Serial.println();
    Serial.println();
    x = 2;
    y = 0;
  }
  else if ((buffer[0] == 06) && (y == 0))
  {
    Serial.println("owner Name   : ABCD ");
    Serial.println("Age  : 22 years");
    Serial.println("ACCESS DENIED");
    slrf="06";
    y = 2;
    x = 0;
  }
  else
  {
    slrf="0";
    Serial.println("Wrong ID");
  }
}

void RfidScan()
{
  if ( ! mfrc522.PICC_IsNewCardPresent())
    return;

  if ( ! mfrc522.PICC_ReadCardSerial())
    return;
  dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size);
}
```

Entrance code:

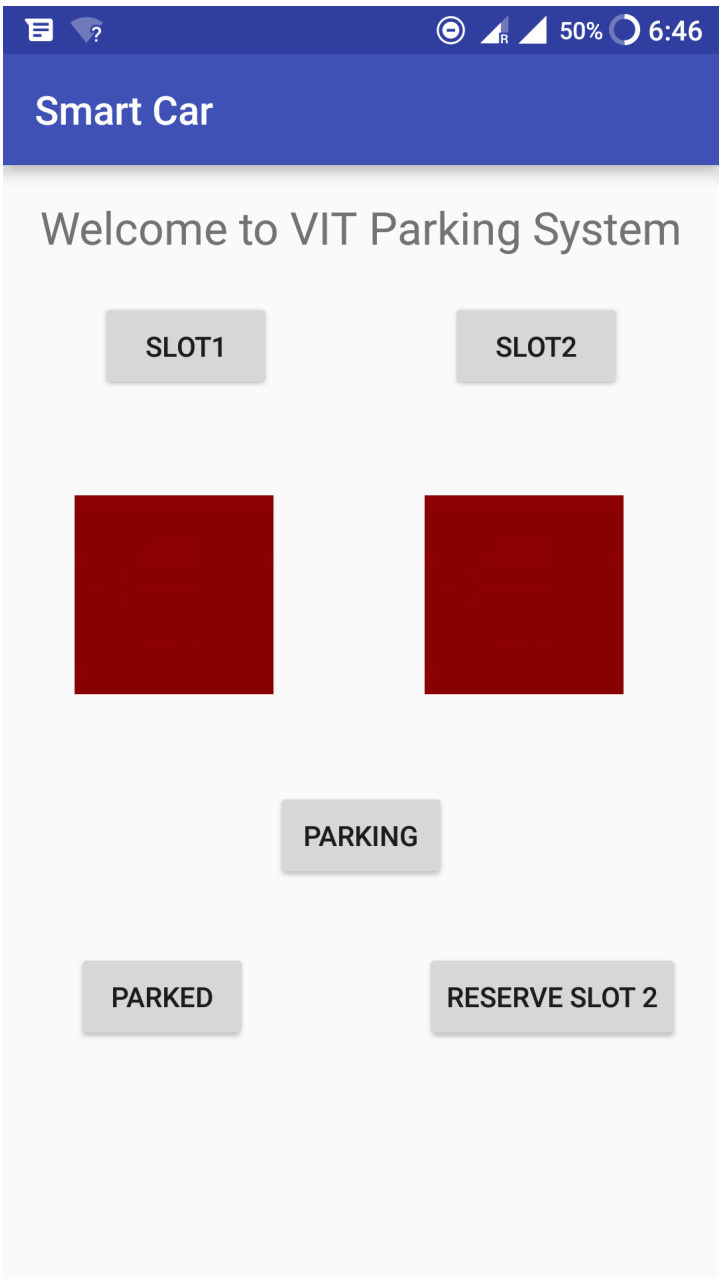
```
#include<ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN D1
#define SS_PIN D2
#define DSEN1_PIN D3
#define DSEN2_PIN D0
MFRC522 mfrc522(SS_PIN, RST_PIN);



int RfidNo = 0;
int x = 0;
int y = 0;
int z = 0;
int amt = 130;
int rej;
String buf;
String slot1;
String slot2;

const char* ssid = "Virus";
const char* password = "123456789";
const char* host = "scps.000webhostapp.com";

long lastMsg = 0;
char msg[50];
int value = 0;
String dsen1;
String dsen2;
int dsenc1;
int dsenc2;

String rcv="";


String slrf="0";



WiFiClient client;
```

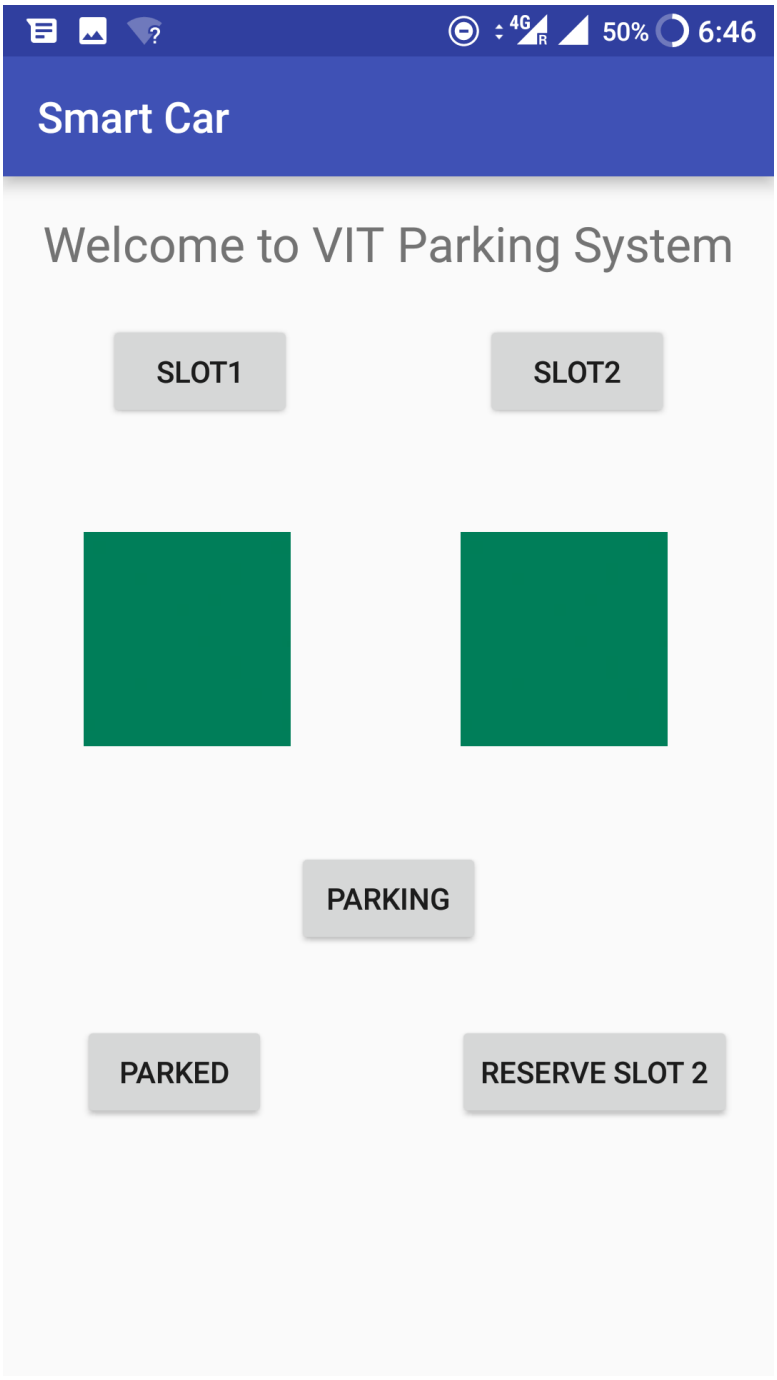
```
void setup_wifi() {
  delay(100);
  // We start by connecting to a WiFi network
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void setup() {
  Serial.begin(115200);
    pinMode(D3, OUTPUT);
    digitalWrite(D3, LOW);

  SPI.begin();


  setup_wifi();
  mfrc522.PCD_Init();

}

void loop() {

  host_connect();
  RfidScan();

    delay(1);
    //upd();
    upda();

}
void upd()
{
 client.print(String("GET ") + "/smartcar/parf.php/" + " HTTP/1.1\r\n" + "Host: " + host + "\r\n"
+ "Connection: close\r\n\r\n"); //GET request for server response.
  delay(500);
```

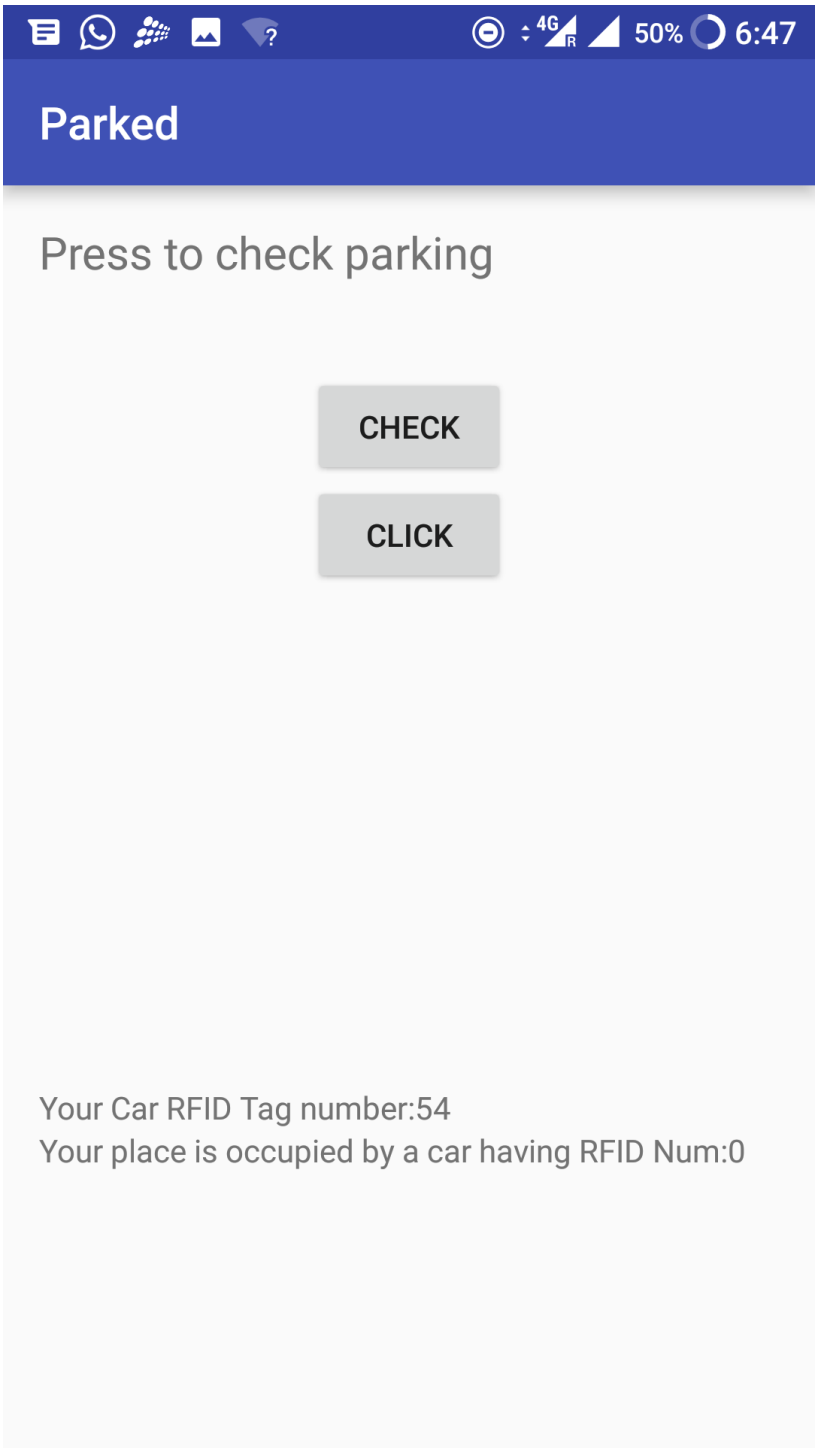
```
  while(client.available()){

    String line = client.readStringUntil('\r'); //Read the server response line by line..
      rcv+=line; //And store it in rcv.
  }

  Serial.println("Received string: ");
  Serial.println(rcv);

  Serial.println();
  Serial.println("closing connection");
  delay(5000);

}

void upda()
{
  String url = "/smartcar/insertprf.php?slrf="+slrf;
  Serial.print("Requesting URL: ");
  Serial.println(url);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +"Host: " + host + "\r\n" + "Connection:
close\r\n\r\n");
  delay(500);

  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }
  Serial.println();
  Serial.println("closing connection");
  delay(5000);
}
  void host_connect()
{
  Serial.print("connecting to ");
  Serial.println(host);

  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }
  Serial.println("Connected Successfully to host");
  Serial.println("");
```

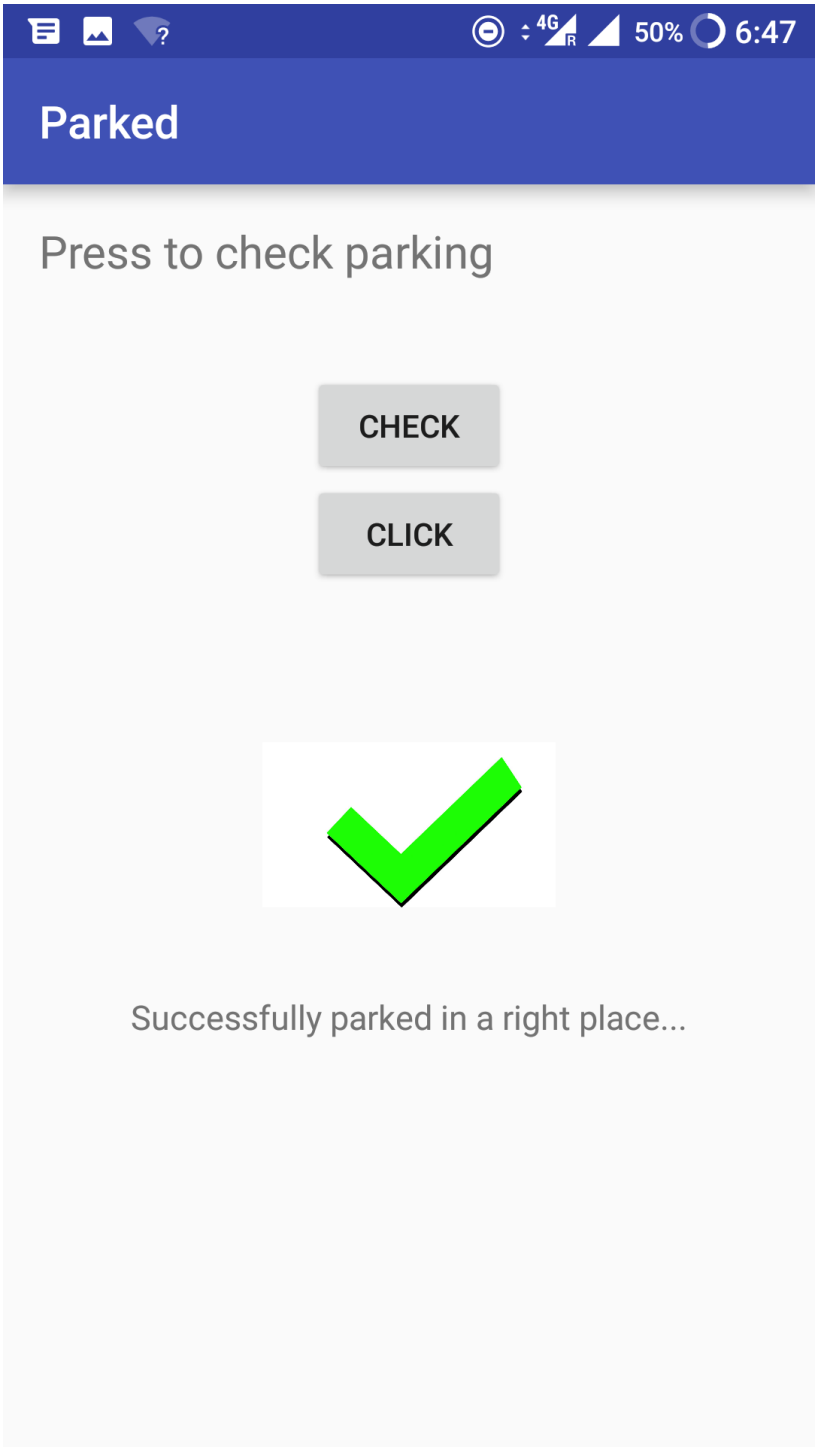
```
}
void dump_byte_array(byte *buffer, byte bufferSize)
{

  if ((buffer[0] == 54 ) && (x == 0))
  {
    Serial.println("owner Name   : SRIRAM ");
    Serial.println("Age  : 25 years");
    Serial.println("ACCESS GRANTED");
    slrf="54";
    Serial.println();
    Serial.println();
    x = 2;
    y = 0;
    digitalWrite(D3, LOW);
  }
  else if ((buffer[0] == 06) && (y == 0))
  {
    Serial.println("owner Name   : NIVETHA ");
    Serial.println("Age  : 22 years");
    Serial.println("ACCESS DENIED");
    slrf="06";
    y = 2;
    x = 0;
  }
  else
  {
    digitalWrite(D3, HIGH);
    slrf="0";
    Serial.println("Wrong ID");
  }
}
void RfidScan()
{
  if ( ! mfrc522.PICC_IsNewCardPresent())
    return;

  if ( ! mfrc522.PICC_ReadCardSerial())
    return;
  dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size);
}
```

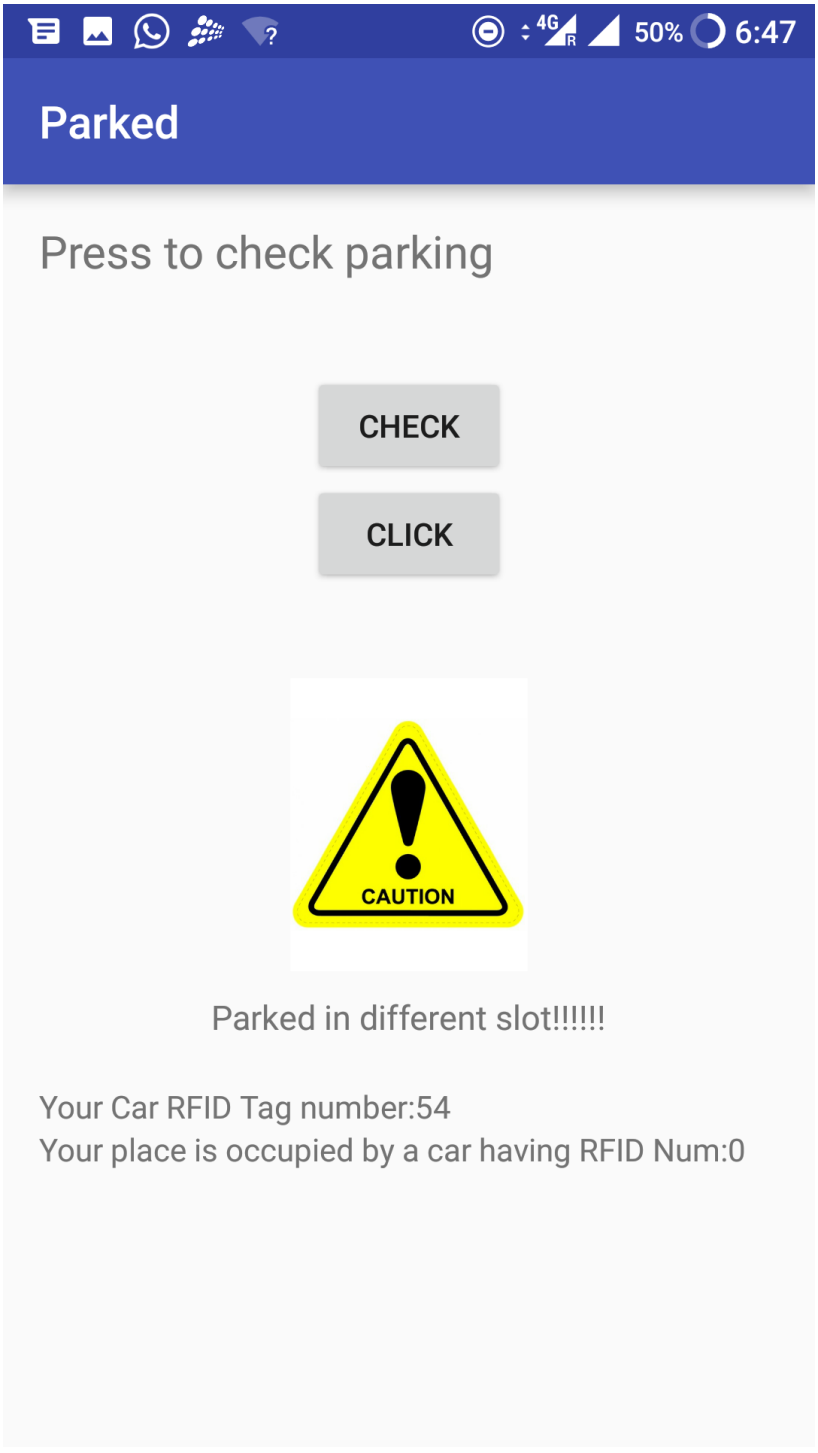
4.2 Output/Results

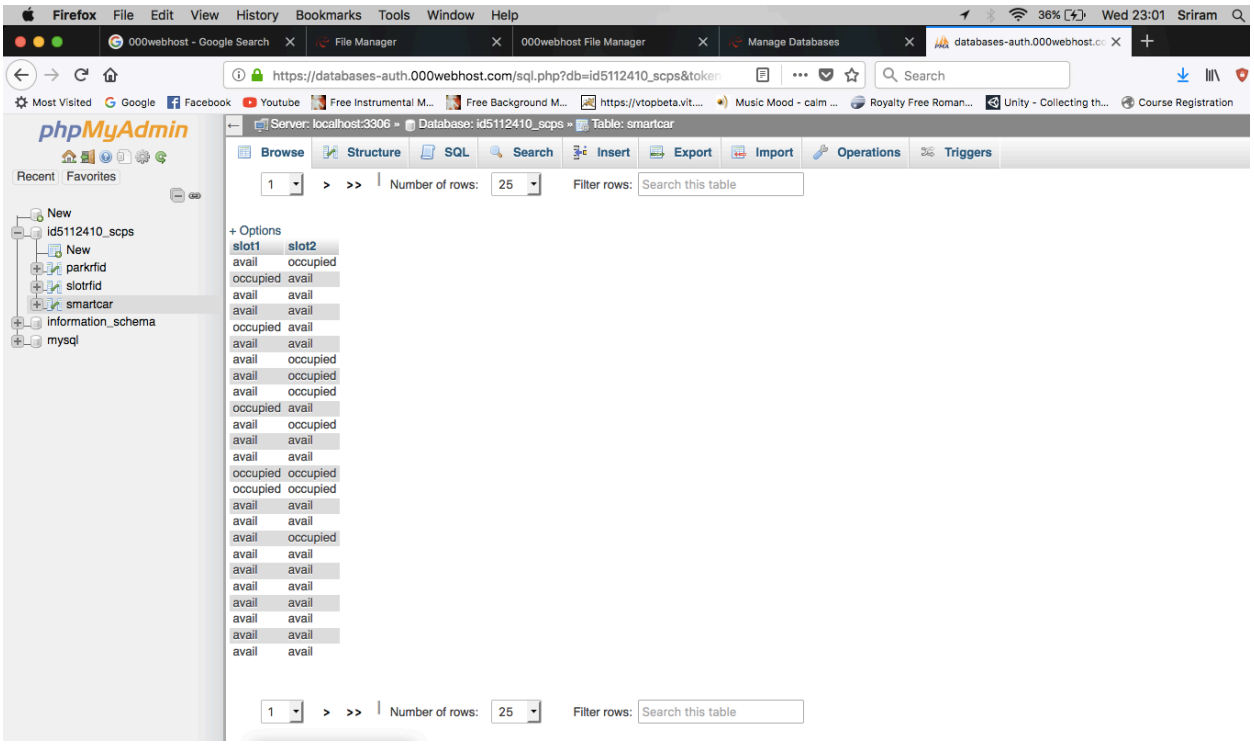


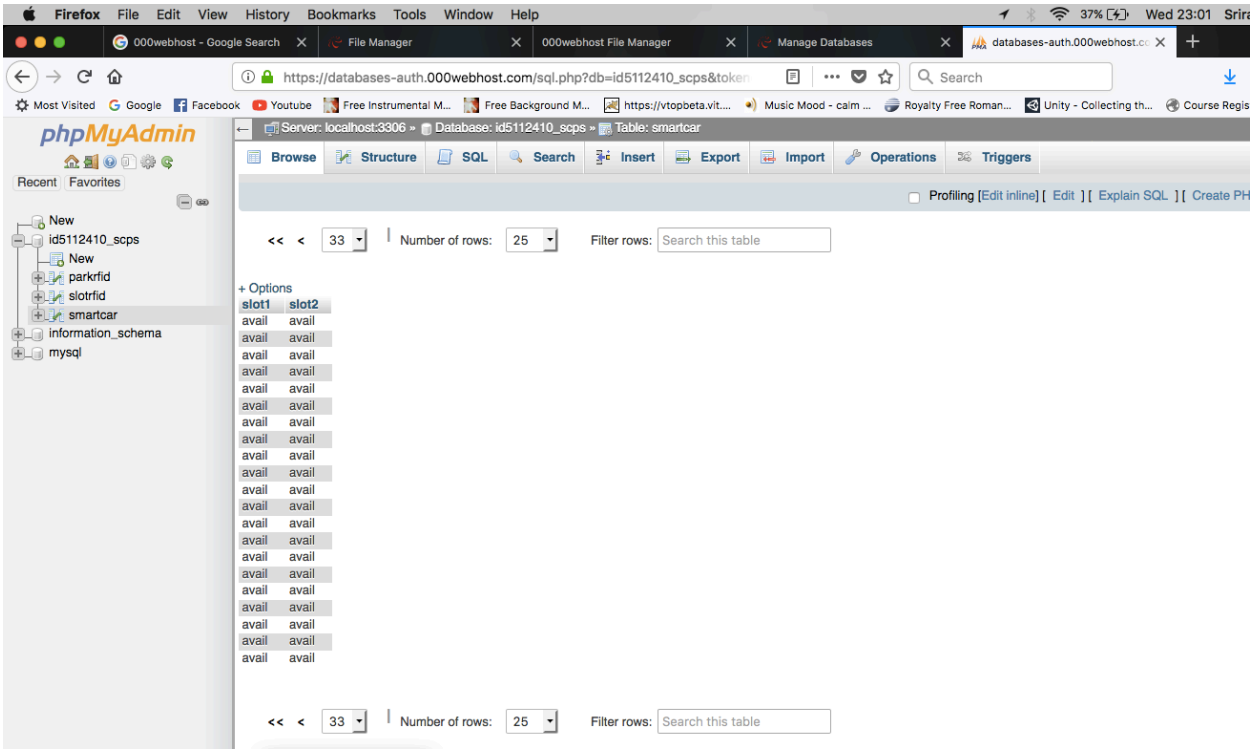

50% 6:46
Smart Car
Welcome to VIT Parking System
SLOT1
SLOT2
PARKING
PARKED
RESERVE SLOT 2

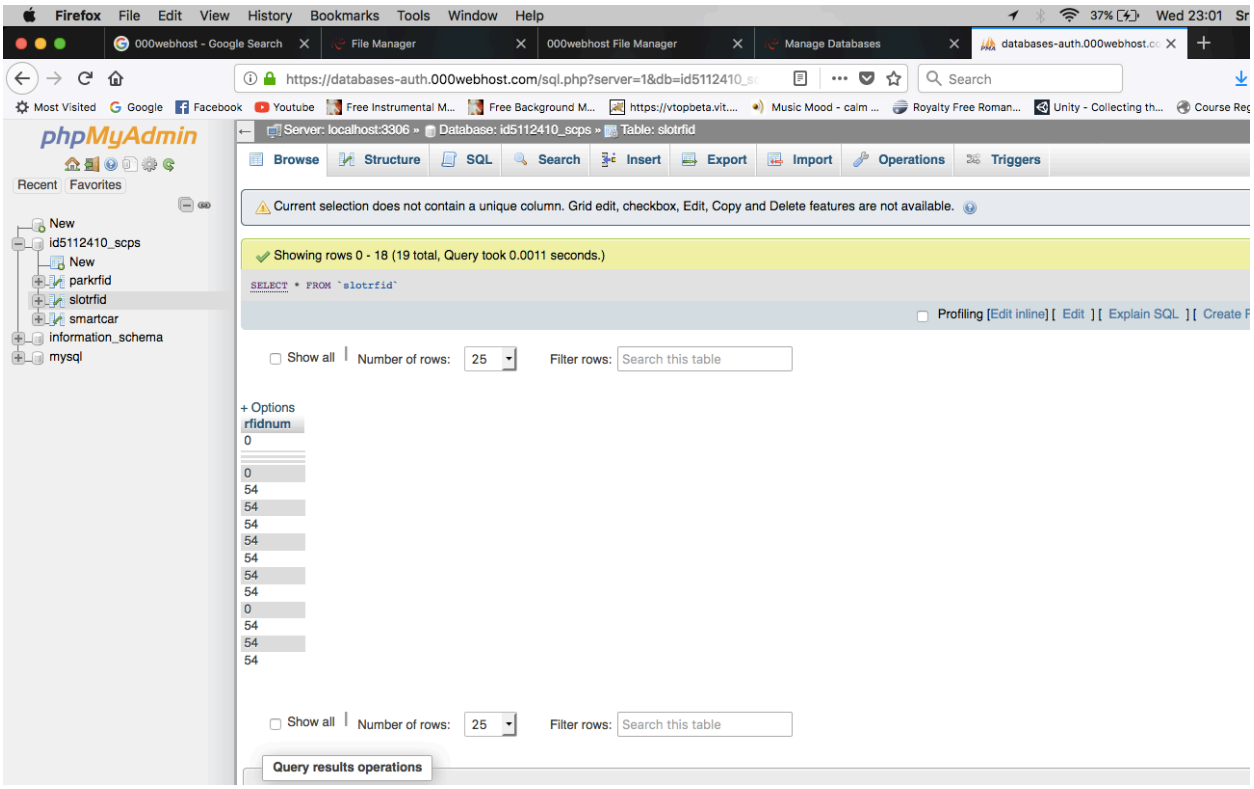

50% 6:47
Parked
Press to check parking
CHECK
CLICK
Your Car RFID Tag number:54
Your place is occupied by a car having RFID Num:0

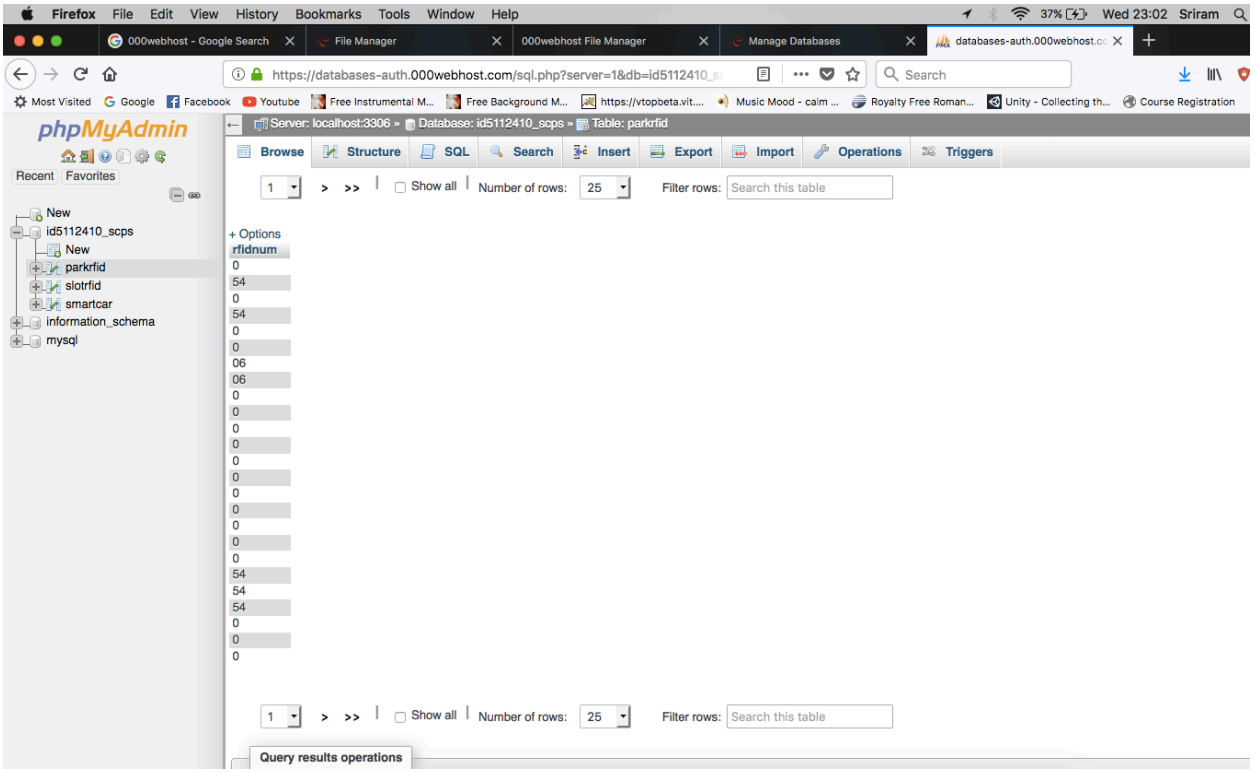

50% 6:47
Parked
Press to check parking
CHECK
CLICK
Successfully parked in a right place...

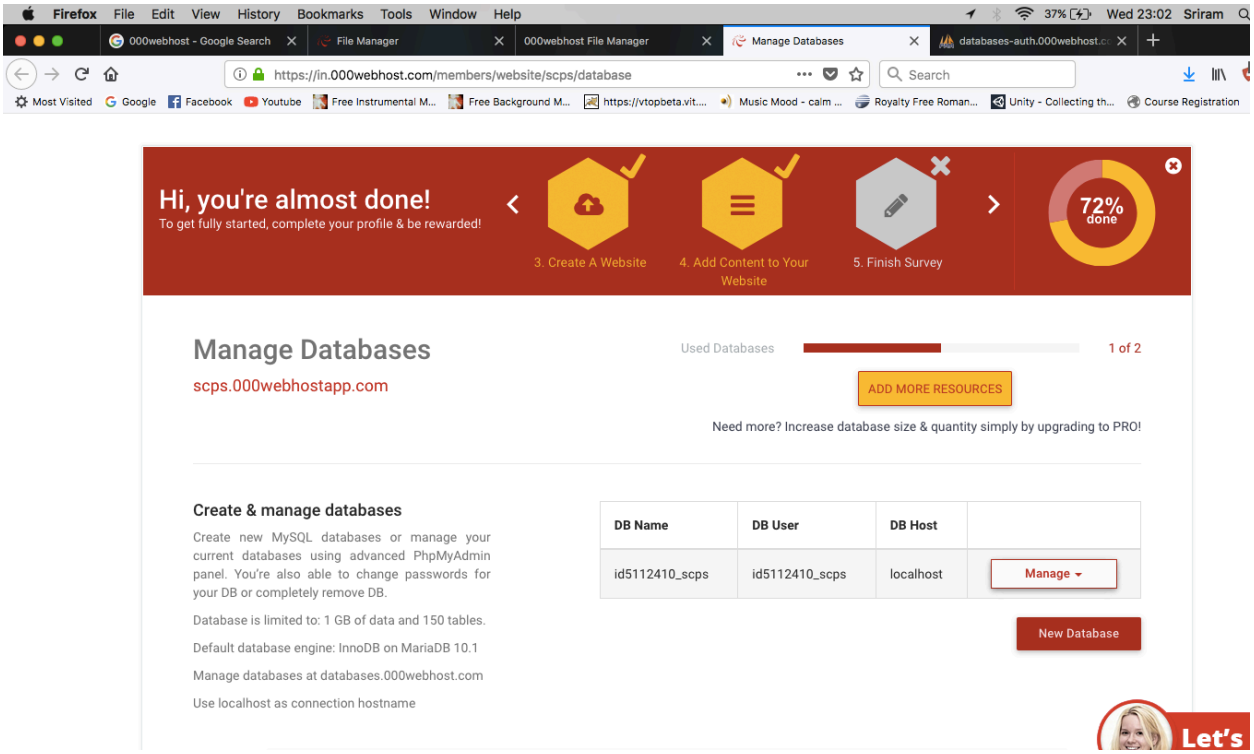

50% 6:47
Parked
Press to check parking
CHECK
CLICK
CAUTION
Parked in different slot!!!!!!
Your Car RFID Tag number:54
Your place is occupied by a car having RFID Num:0

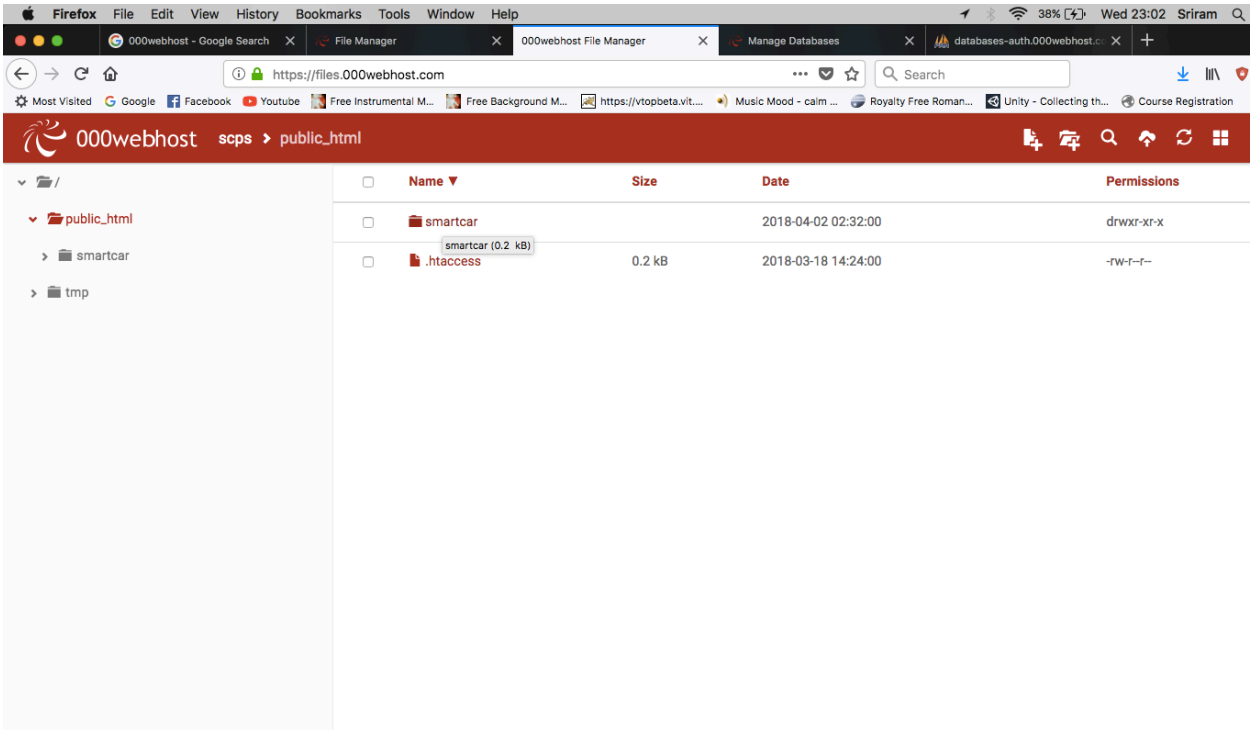

phpMyAdmin
Server: localhost:3306 » Database: id5112410_scps » Table: smartcar
Browse Structure SQL Search Insert Export Import Operations Triggers
Number of rows: 25
Filter rows: Search this table
+ Options
slot1 slot2
avail occupied
occupied avail
avail avail
avail avail
occupied avail
avail avail
avail occupied
avail occupied
avail occupied
occupied avail
avail occupied
avail avail
avail avail
occupied occupied
occupied occupied
avail avail
avail avail
avail occupied
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
New
id5112410_scps
New
parkrfid
slotrfid
smartcar
information_schema
mysql

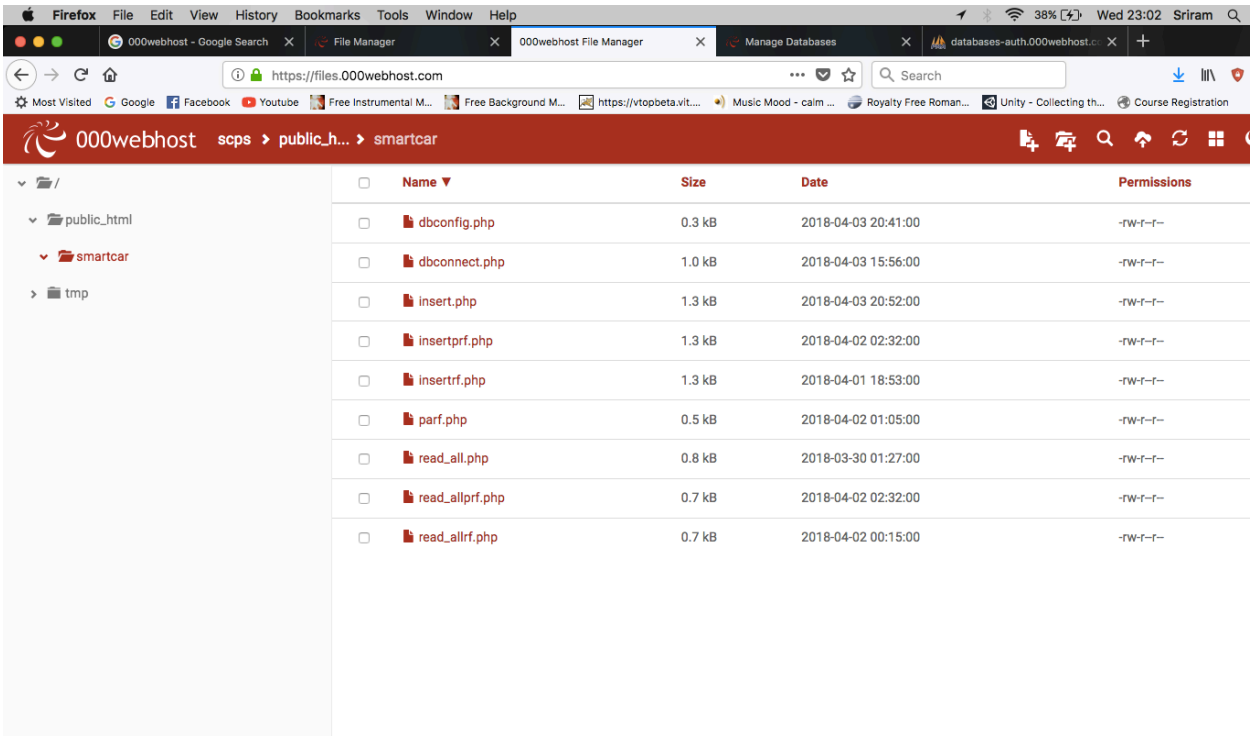

phpMyAdmin
Server: localhost:3306 » Database: id5112410_scps » Table: smartcar
Browse Structure SQL Search Insert Export Import Operations Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PH
<< < 33 Number of rows: 25
Filter rows: Search this table
+ Options
slot1 slot2
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
avail avail
New
id5112410_scps
New
parkrfid
slotrfid
smartcar
information_schema
mysql

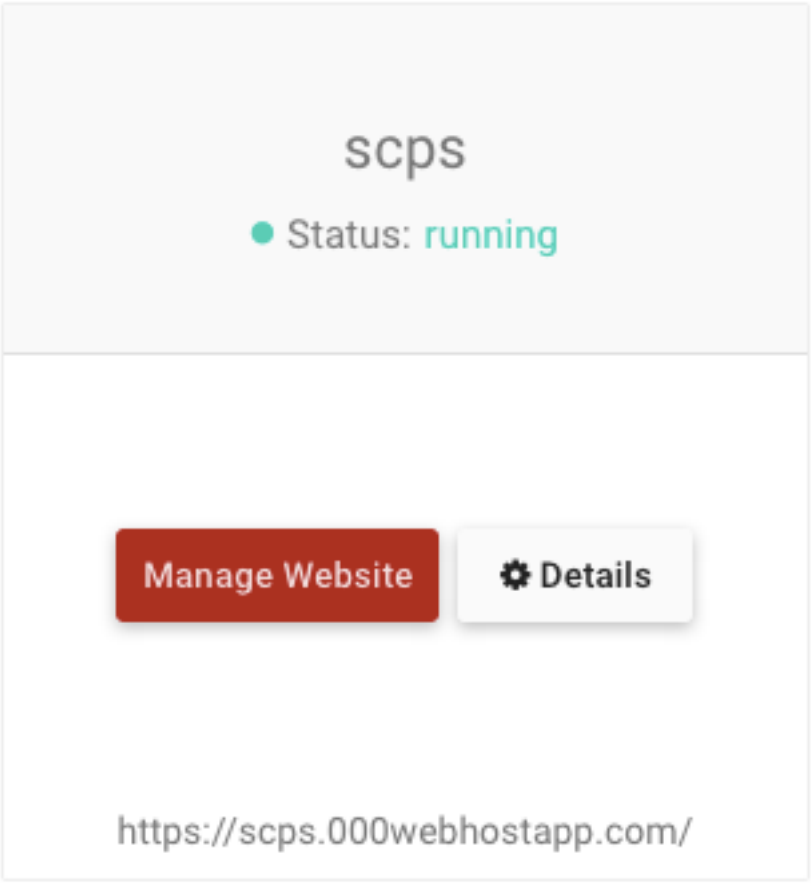

phpMyAdmin
Recent Favorites
New
id5112410_scps
New
parkrfid
slotrfid
smartcar
information_schema
mysql
Server: localhost:3306 » Database: id5112410_scps » Table: slotrfid
Browse Structure SQL Search Insert Export Import Operations Triggers
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.
Showing rows 0 - 18 (19 total, Query took 0.0011 seconds.)
SELECT * FROM `slotrfid`
Profiling [Edit inline] [Edit] [Explain SQL] [Create
Show all | Number of rows: 25 Filter rows: Search this table
+ Options
rfidnum
0
0
54
54
54
54
54
54
0
54
54
54
Show all | Number of rows: 25 Filter rows: Search this table
Query results operations


phpMyAdmin
Recent Favorites
New
id5112410_scps
New
parkrfid
slotrfid
smartcar
information_schema
mysql
Server: localhost:3306 » Database: id5112410_scps » Table: parkrfid
Browse Structure SQL Search Insert Export Import Operations Triggers
1 > >> Show all | Number of rows: 25 Filter rows: Search this table
+ Options
rfidnum
0
54
0
54
0
0
06
06
0
0
0
0
0
0
0
0
0
0
0
54
54
54
0
0
0
1 > >> Show all | Number of rows: 25 Filter rows: Search this table
Query results operations


Firefox File Edit View History Bookmarks Tools Window Help
Wed 23:02 Sriram
000webhost - Google Search
File Manager
000webhost File Manager
Manage Databases
databases-auth.000webhost.com
https://in.000webhost.com/members/website/scps/database
Search
Most Visited Google Facebook Youtube Free Instrumental M... Free Background M... https://vtopbeta.vit.... Music Mood - calm ... Royalty Free Roman... Unity - Collecting th... Course Registration
Hi, you're almost done!
To get fully started, complete your profile & be rewarded!
3. Create A Website
4. Add Content to Your Website
5. Finish Survey
72% done
Manage Databases
scps.000webhostapp.com
Used Databases
1 of 2
ADD MORE RESOURCES
Need more? Increase database size & quantity simply by upgrading to PRO!
Create & manage databases
Create new MySQL databases or manage your current databases using advanced PhpMyAdmin panel. You're also able to change passwords for your DB or completely remove DB.
Database is limited to: 1 GB of data and 150 tables.
Default database engine: InnoDB on MariaDB 10.1
Manage databases at databases.000webhost.com
Use localhost as connection hostname
DB Name
DB User
DB Host
id5112410_scps
id5112410_scps
localhost
Manage
New Database
Let's


Firefox File Edit View History Bookmarks Tools Window Help
Wed 23:02 Sriram
000webhost - Google Search
File Manager
000webhost File Manager
Manage Databases
databases-auth.000webhost.com
https://files.000webhost.com
Search
Most Visited Google Facebook Youtube Free Instrumental M... Free Background M... https://vtopbeta.vit.... Music Mood - calm ... Royalty Free Roman... Unity - Collecting th... Course Registration
000webhost
scps › public_html
/
public_html
smartcar
tmp
Name
Size
Date
Permissions
smartcar
2018-04-02 02:32:00
drwxr-xr-x
smartcar (0.2 kB)
.htaccess
0.2 kB
2018-03-18 14:24:00
-rw-r--r--


Firefox File Edit View History Bookmarks Tools Window Help
Wed 23:02 Sriram
000webhost - Google Search
File Manager
000webhost File Manager
Manage Databases
https://files.000webhost.com
Search
000webhost
scps › public_h... › smartcar
/
public_html
smartcar
tmp
Name
Size
Date
Permissions
dbconfig.php 0.3 kB 2018-04-03 20:41:00 -rw-r--r--
dbconnect.php 1.0 kB 2018-04-03 15:56:00 -rw-r--r--
insert.php 1.3 kB 2018-04-03 20:52:00 -rw-r--r--
insertprf.php 1.3 kB 2018-04-02 02:32:00 -rw-r--r--
insertrf.php 1.3 kB 2018-04-01 18:53:00 -rw-r--r--
parf.php 0.5 kB 2018-04-02 01:05:00 -rw-r--r--
read_all.php 0.8 kB 2018-03-30 01:27:00 -rw-r--r--
read_allprf.php 0.7 kB 2018-04-02 02:32:00 -rw-r--r--
read_allrf.php 0.7 kB 2018-04-02 00:15:00 -rw-r--r--


scps
Status: running
Manage Website
Details
https://scps.000webhostapp.com/

5. Conclusion and Future Developments

We can see that this reduces human effort and is fully automated. This system reduces the frustration or difficulty in finding the parking spaces in bigger buildings. Also people saves their time by reserving the parking spaces wherever they are. Also when it comes to accuracy this system in very accurate to tell the availability of parking slots to the users. Earlier people has to go to the respective place and wait for at least half an hour or one hour to park their vehicles. Which in turn causes a lot of pollution in the city. So this system will let the people know if the parking slot is free and if not they get the idea of staying back wherever they are instead of making such a long crowd and pollution.

In addition to this system we can retrieve the location of people who ever is reserving for a particular place and this can only be done if he is in 1km radius from the parking space or else there might occur a problem of people reserving it when they are in home and this in turn could make the availability of slot for a very longer time.

A product should not be so ease that one will become lazy. A product should be automated only at a particular level, or else it can become poison to them. Make sure that everything should be in normal and safe range and does not affect the user physically and mentally. That is called as a perfect automated product.

6. References

[1] J. Dongjiu Geng, Yue Suo, Yu Chen, Jun Wen, Yongqing Lu, Remote Access and Control System Based on Android Mobil Phone, vol.2. Journal of Computer Applications, 2011, pp. 560-562

[2] M. Fengsheng Yang, Android Application Development Revelation, China Machine Press, 2010

[3] Karthikeyan, T., Bhuvaneswari, N. S., & Sujatha, S. (2012). Traffic Handling Approach with Intelligent Speed Control and Prioritization of Emergency Vehicles using PCM Agent. International Journal of Computer Technology and Application (IJCTA), 3(4).

[4] Kumaar, M. A., Kumar, G. A., & Shyni, S. M. (2016, April). Advanced traffic light control system using barrier gate and GSM. In Computation of Power, Energy Information and Commuincation (ICCPEIC), 2016 International Conference on(pp. 291-294). IEEE.

[5] Singha, M. R., & Kalita, B. (2013). Using Mobile Phone Network for Urban Traffic Management. International Journal of Computer Applications, 65(2).

[6] Tahilyani, S., Darbari, M., & Shukla, P. K. (2012). A new genetic algorithm based lane-by-pass approach for smooth traffic flow on road networks. International Journal of Advanced Research in Artificial Intelligence, 1(3), 32-36.

[7] Latha, J. R., & Suman, U. (2015). Intelligent Traffic Light Controller. International Journal, 38.

[8] Viswanath, D. K., & Krishna, K. M. (2009). Towards load-balanced de-congested multi-robotic agent traffic control by coordinated control at intersections. Intelligent Service Robotics, 2(2), 81-93.

[9] Aavani, P., Sawant, M. K., Sawant, S., & Deshmukh, R. S. (2017). A review on adaptive traffic controls systems. International Journal of Latest Engineering and Management Research, 2(1), 52-57.

[10] Priya, M. P. R., Jose, A., & Sumathy, G. (2015). Traffic Light Pre-emption control System for Emergency Vehicles. SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE), 2(2).

OTHER INFORMATION:

All the data that is collected from the sensor is given to the cloud by MQTT protocol and the analysis is done in the cloud based on the algorithm used. The output of the live traffic updates and emergency vehicle approach can be viewed.

VIT® Vellore Institute of Technology (Deemed to be University under section 3 of UGC Act, 1956)	**School of Information Technology & Engineering** **M.Tech (Software Engineering) Winter 2017-18** **SWE4005 Internet of Things** **Course Project- Implementation Review(Final)** **Evaluation Sheet** **(Review Date 28.03.2018 – 30.03.2018 ,** **See Individual Schedule)**

Title: Smart Car Parking System

Team Name

Project Team

S.No	Register Number	Student Name	Signature	Guided By
1	15MIS0173	Sriram K		Dr. Kavitha Br
2	15MIS0392	Sanjana V		
3	15MIS0420	Vaishnavi CH		

Team Member(s) Contribution and Performance Assessment

Components	Student 1	Student 2	Student 3
Analysis Component (15)			
Deploying in Cloud(15)			
Completion of project(10)			
Upload and formatting of the report(10)			

Student Feedback(Student Experience in this Course Project)	Evaluator Comments

Name & Signature of the Evaluator(s)