“E-PARKING SYSTEM”

A Project Report

Submitted in the partial fulfilment of the requirements

for the award of the degree of

Bachelor of Technology

in

Department of Mechanical Engineering

By

190070119 – Vemula Mahesh

190070140 – B Ravi Prabhat

190079028 – K Rahul

KL

(DEEMED TO BE UNIVERSITY)

Koneru Lakshmaiah Education Foundation

(Deemed to be University estd., u/s 3 of UGC Act 1956)

Greenfields, Vaddeswaram, Guntur (Dist.), Andhra Pradesh - 522502

December, 2020

Declaration

The Project Report entitled "**E-PARKING SYSTEM**" is a record of bona fide work of submitted in partial fulfilment for the award of B.Tech in mechanical engineering to the K L University. The results embodied in this report have not been copied from any other department/university/institute

190070119 – Vemula Mahesh

190070140 – B Ravi Prabhat

190079028 – K Rahul

Certificate

This is to certify that the Project Report entitled "**E-PARKING SYSTEM**" is being submitted by 190070119 – Vemula Mahesh, 190070140 – B Ravi Prabhat and190079028 – K Rahul submitted in partial fulfilment for the award of B. Tech in Mechanical to the K L University is a record of Bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of the Co-Supervisor
Name and Designation

Signature of the Supervisor
Name and Designation

Signature of the HOD

Signature of the External Examiner

Acknowledgement

I also wish to extend my thanks to **Mr.K Dileep sir** and other facultiesfor guiding and providing the knowledge related to machinery and processes.

I express the sincere gratitude to our Mechanical Head of the Department **Dr. D. V. A. Rama Sastry** for his Admission towards our academic growth.

Finally, it is pleased to acknowledge the indebted to all those who devoted themselves directly or indirectly to make this project success.

S.no	Name	Id No
1.	Vemula Mahesh	190070119
2.	B Ravi Prabhat	190070140
3.	K Rahul	190079028

Abstract

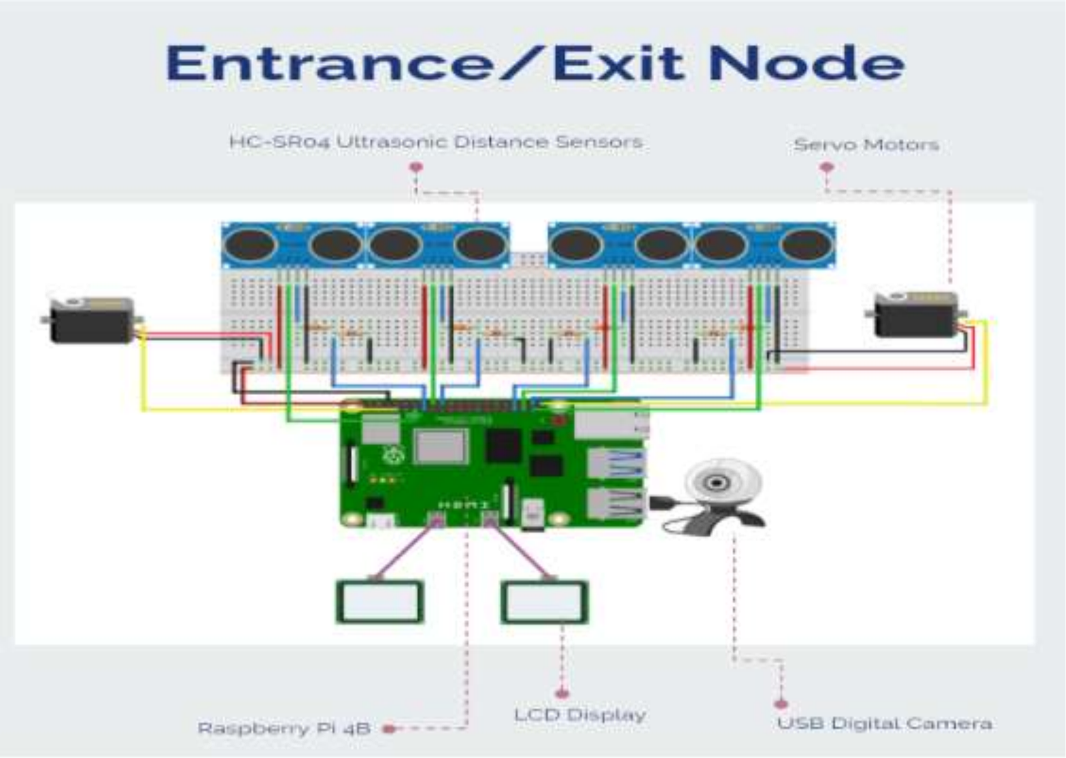
E-parking system is an effective tool for management of parking space through better access control and parking revenue method in busy areas in big cities such as shopping complexes, stadiums and other popular places, especially during their peak hour.

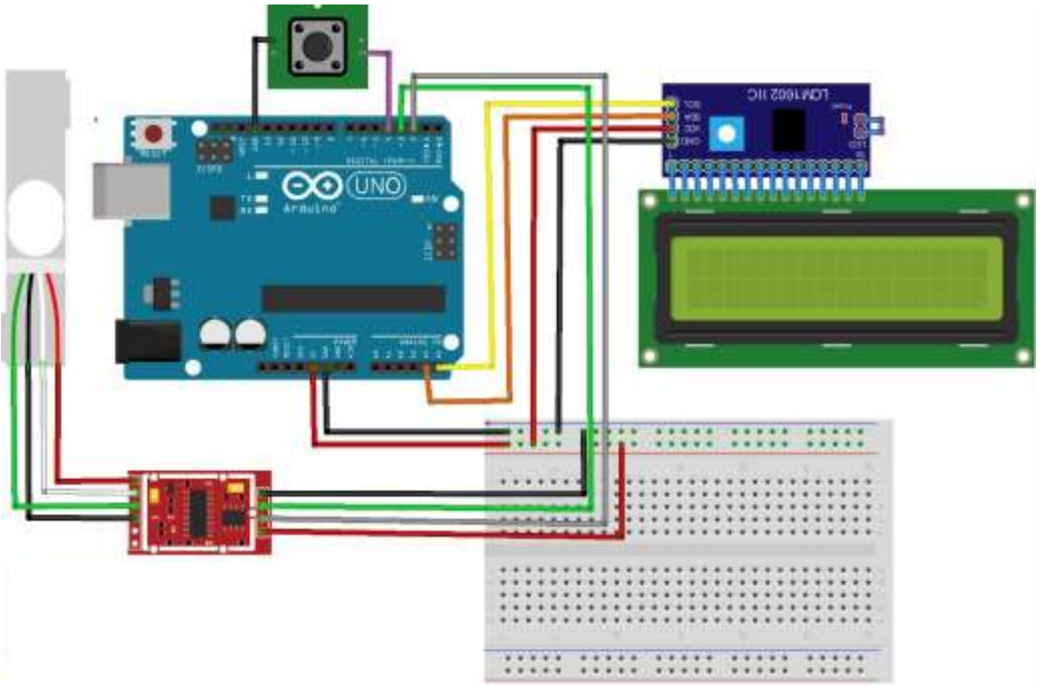
E-Parking is smart parking system which helps in solving the common problem of allocating parking space.E-Parking System improves safety, efficiency and mobility, optimizes the parking economic performance.

Introduction

Today India is facing a new problem which is lack of sufficient and secure parking space. Due to rapid growth in population the total number of motor vehicles increasing, the parking scenario is falling short of the current requirements in the country. A centralized and single database system avoids the critical error of assigning vacant spot to several users who come at the same time.Advanced parking guidance system uses sensors and CCTV to guide users towards available spaces.RFID technology for the automatic vehicle identification and Advanced payment system available.RFID enabled smart card ensures rechargeable balance mechanism.Highly advanced system as total integration with mechanical parking equipment.

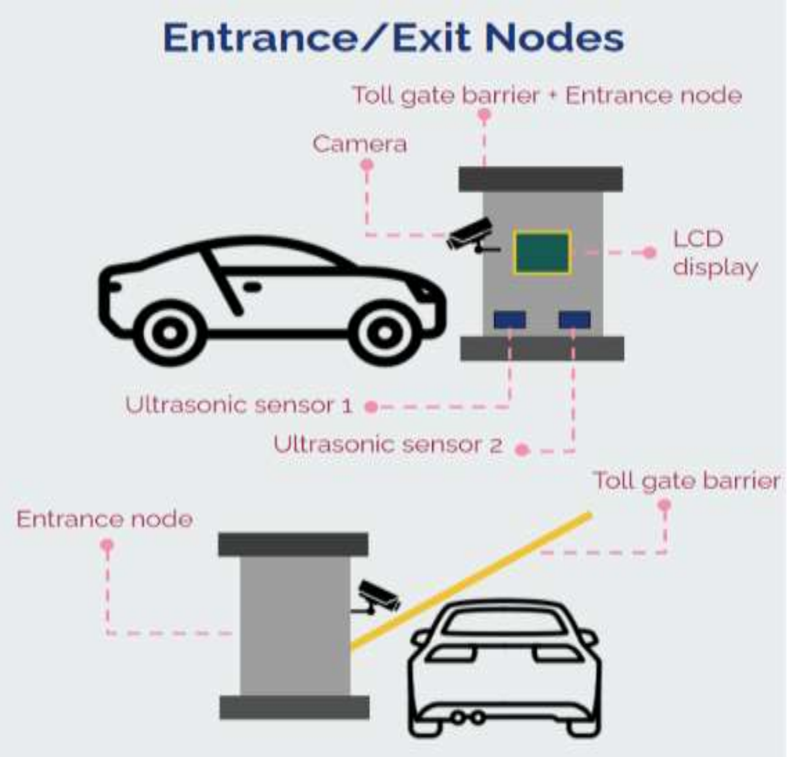
CIRCUIT DIAGRAM





Entrance/Exit Node:

HC-SR04 ultrasonic distance sensors will sense the incoming vehicle at the entrance node and the digital camera will get the license plate image. This image will be processed at the Raspberry Pi 4 (Python and Tessaract OCR will be used for the Automatic Number Plate Recognition in the Raspberry Pi) and the necessary data will be send to the server. Once a spot number is received , the Raspberry Pi 4 will signal the servo motor to open the gate while displaying the spot number on the LCD display. Each hardware node has an AC-DC power supply attached and the components operate on 5V DC. The time of entry will be saved and at the time of exit time being in parking will be calculated. The amount directly collected through RFID technology.



Software's used

Arduino IDE:

- Arduino Integrated Development Environment
- The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux.
- Python

Hardware requirements

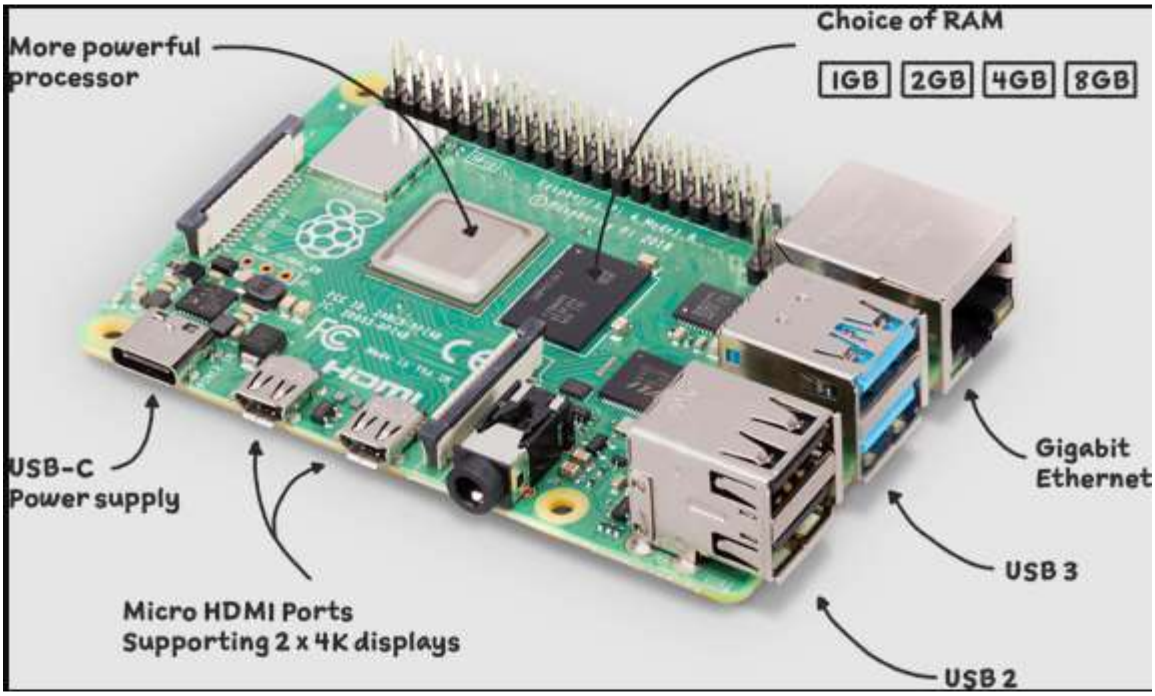
Arduino UNO:

- Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.

USB DIGITAL CAMERA:

The USB cable supplies power to the webcam from the computer and takes the digital information captured by the webcam's image sensor back to the computer

RASPBERRY PI 4B:

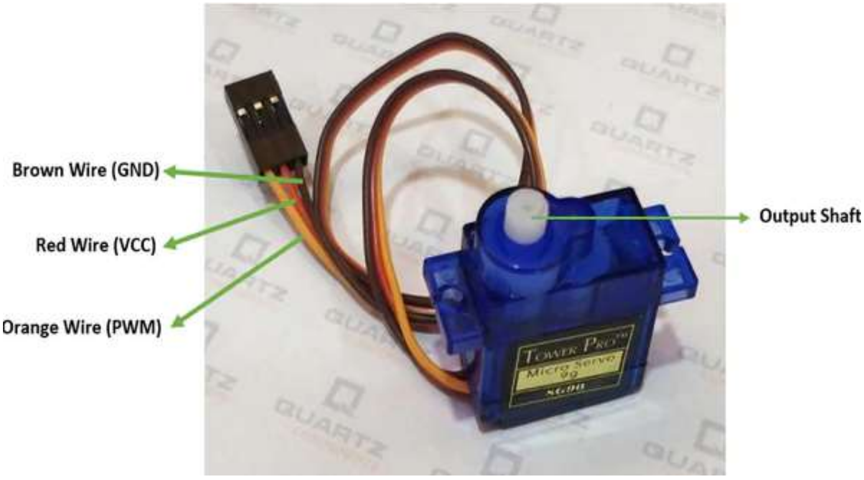


HR-SR04 ULTRASONIC DISTANCE SENSOR:

This is the HC-SR04 ultrasonic distance sensor. This economical sensor **provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm**. Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit.

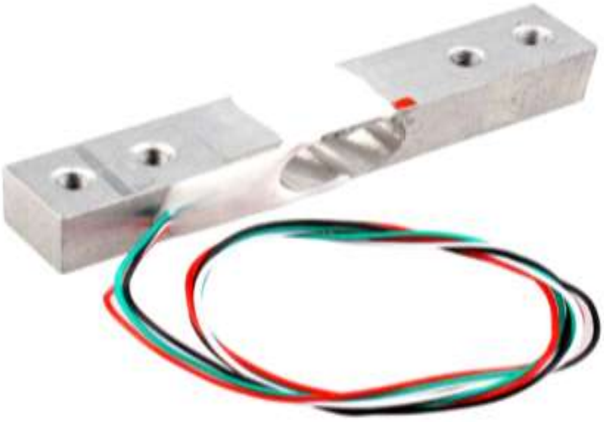
SERVO MOTOR:

A servo motor is **a rotary actuator that allows for precise control of angular position**. It consists of a motor coupled to a sensor for position feedback. It also requires a servo drive to complete the system. The drive uses the feedback sensor to precisely control the rotary position of the motor.



Load Cell:

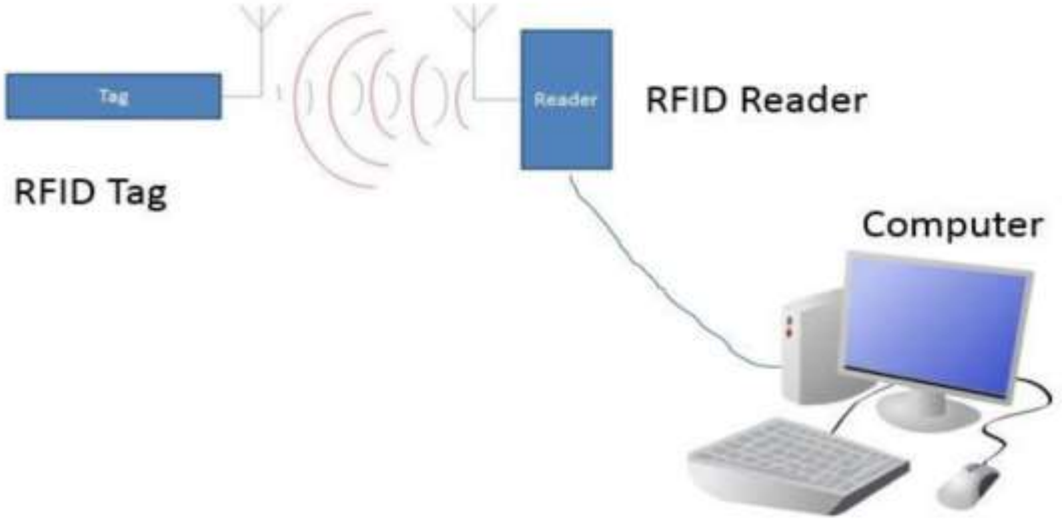
A load cell is an electrical device that transforms force or pressure into electrical output. The magnitude of this electrical output is directly proportional to the force being applied. Load cells have a gage, which deforms when pressure is applied to it. And then gage generates an electrical signal on deformation as its effective resistance changes on deformation. A load cell sometimes consists of 4 strain gauges during a Wheatstone bridge configuration. Load cell comes in varied ranges like 5kg, 10kg, 100kg.



Bread Board:

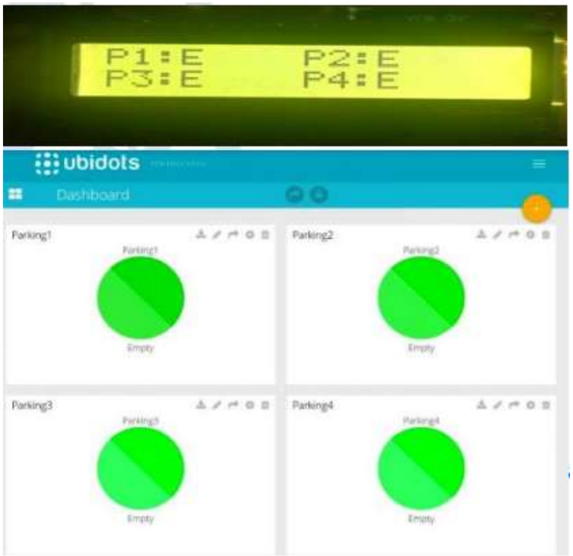
RFID :

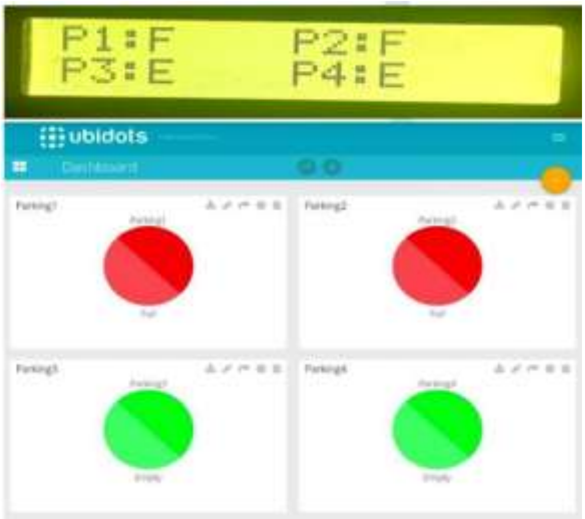
Radio Frequency Identification (RFID) refers to a wireless system comprised of two components: tags and readers. The reader is a device that has one or more antennas that emit radio waves and receive signals back from the RFID tag.Tags, which use radio waves to communicate their identity and other information to nearby readers, can be passive or active. Passive RFID tags are powered by the reader and do not have a battery. Active RFID tags are powered by batteries.



Working Principle

HC-SR04 ultrasonic distance sensors will sense the incoming vehicle at the entrance node and the digital camera will get the license plate image. This image will be processed at the Raspberry Pi 4 (Python and Tessaract OCR will be used for the Automatic Number Plate Recognition in the Raspberry Pi) and the necessary data will be send to the server. Once a spot number is received , the Raspberry Pi 4 will signal the servo motor to open the gate while displaying the spot number on the LCD display. Each hardware node has an AC-DC power supply attached and the components operate on 5V DC. The time of entry will be saved and at the time of exit time being in parking will be calculated. The amount directly collected through RFID technology. The person can see the vacant places at the entry where it is displayed at the entry gate.





Parking spaces which are vacant.

Parking spaces which are filled and vacant.

Code:

For Automatic License/Number Plate Recognition:

```
#Automatic License/Number Plate Recognition (ANPR)

import pytesseract
import cv2
import numpy as np
import imutils

pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

#License plate detection

img = cv2.imread('lp3.jpg',cv2.IMREAD_COLOR)
scale_percent =  90
width = int(img.shape[1] * scale_percent / 100)
height = int(img.shape[0] * scale_percent / 100)
dim = (width, height)
img = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = cv2.bilateralFilter(gray, 13, 15, 15)
edged = cv2.Canny(gray, 30, 200)

contours = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(contours)
contours = sorted(contours, key = cv2.contourArea, reverse = True)[:20]
screenCnt = None
peri = cv2.arcLength(c, True)
approx = cv2.approxPolyDP(c, 0.018 * peri, True)

if len(approx) == 4:
        screenCnt = approx
        break

if screenCnt is None:
    detected = 0
    print ("No contour detected")
else:
     detected = 1

image (img)
if detected == 1:

    cv2.drawContours(img, [screenCnt], -1, (0, 0, 255), 3)

    mask = np.zeros(gray.shape,np.uint8)
    new_image = cv2.drawContours(mask,[screenCnt],0,255,-1)
    new_image = cv2.bitwise_and(img,img,mask=mask)


    (x, y) = np.where(mask == 255)
    (topx, topy) = (np.min(x), np.min(y))
    (bottomx, bottomy) = (np.max(x), np.max(y))
    Cropped = gray[topx:bottomx+1, topy:bottomy+1]
    Cropped = cv2.medianBlur(Cropped,3)

    text = pytesseract.image_to_string(Cropped, config='--psm 11', lang="eng" )
    print("Text:",text)


    text = text[:-1]
    num = ".join(i.upper() for i in text if i.isalnum())
    print("License Plate Number:",num)


    img = cv2.resize(img,(500,300))
    Cropped = cv2.resize(Cropped,(400,200))
    cv2.imshow('car',img)
    cv2.imshow('Cropped',Cropped)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

To show the spot to park:(Code)

```
// defines pins numbers
const int trigPin = 2;  //D4
const int echoPin = 0;  //D3
const int ledPin = 4;  //D2 to indicate the state of the gate
bool gateSignal;
bool state;

// defines variables
long duration;
int distance;

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
pinMode(ledPin, OUTPUT); // Sets the ledPin as a
Serial.begin(9600); // Starts the serial communication

}

void loop() {

//If the server gives the signal--> open the gate



  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(1000);
  if(distance < 10)
{
    state = true;
    digitalWrite(ledPin, HIGH);
    delay(2000);
}
 else
{
    state = false;
    digitalWrite(ledPin, LOW);
    delay(2000);
}
  Serial.print("State: ");
  Serial.println(state);
}
```

RFID reader:

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup()
{
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println("Approximate your card to the reader...");
  Serial.println();
}
void loop()
{
 if ( ! mfrc522.PICC_IsNewCardPresent())
 {
   return;
  }
 if ( ! mfrc522.PICC_ReadCardSerial())
 {
return;
 }
Serial.print("UID tag :");
String content= "";  byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));  }  Serial.println();
Serial.print("Message : ");
content.toUpperCase();
if (content.substring(1) == "BD 31 15 2B"){
Serial.println("Authorized access");
 Serial.println();
 delay(3000);
}
 else
{    Serial.println(" Access denied");
  delay(3000);
}
}
```

Total Approximate value for this project:

Component	Qty	PRICE
Raspberry pi 4	1	Rs.9000
RFID READER	1	Rs.285
RFID TAG	1	Rs.10
Camera Modules	2	Rs.1000
LCD Display	2	Rs.5000
HC-SR04 Ultrasonic Sensor	5	Rs.200
AURDINO	1	Rs.780
Servo Motors	3	Rs.320
Power Supply (5V 12A)	2	Rs.1500
Load Cell	2	Rs.1000
TOTAL COST		**RS.18,195**

References:

https://iopscience.iop.org/article/10.1088/1755-1315/717/1/012031/meta

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.446.4724&rep=rep1&type=pdf

https://projects.ce.pdn.ac.lk/3yp/e17/E-Parking-System/

https://ieeexplore.ieee.org/abstract/document/7503445

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.474.3951&rep=rep1&type=pdf

Conclusion

The user interface will be having bill based where services such as parking supervision, online payment and vehicle guidance will be common features of e-parking. The selection of sensors in e-parking will vary depending on several indoor and outdoor conditions. Security in data communication protocols will also remain a major source of concern in future e-parking systems. As a result, more emphasize will be provided in wireless communication protocols to ensure data security.