IOT- SMART PARKING

NAME : N.Nibiya Rose

ID : aut962921104708

EMAIL : nibiyanibiya6@gmail.com

TABLES OF CONTENTS

DEVELOPMENT PART 1

start building the IOT enabled smart parking system in public.

INRODUCTION :

Smart parking is an efficient and cost-effective system to monitor the availability of parking spaces in real-time. Technologies such as sensors and cameras detect free parking spaces and direct drivers to the most convenient spot via digital signage such as LED-displays. By leveraging the power of artificial intelligence, edge computing, and interconnectivity, the complexity of driving in a complicated environment is simplified.

Smart parking solutions not only significantly minimize search traffic, but also help ease congestion in cities and reduce vehicle emissions. Finding a free parking space easily in an often-confusing commercial car park enhances customer experience, thus improving revenue. Simultaneously, parking operators and civic authorities can gather intelligent data on parking and customer habits that can be used to drive future infrastructure developments.

Data is generated via in-ground sensors that have to be installed in each parking space or via Overhead sensors as developed by Cleverciti. The latter can be easily mounted on existing infrastructure such as lampposts or buildings and provide a comprehensive real-time overview of the occupancy status of up to 100 parking spaces per sensor.

Overhead sensors scan the vicinity and identify the position of a parked car or an empty parking space while also measuring the length

of an available parking space and registering whether vehicles have been in an illegal space (e.g. in front of a fire hydrant or other no-parking zone).

The data gathered by these sensors is transmitted via cellular (LTE), Wi-Fi or wired Internet connection. The best smart parking solutions process data “at the edge”, which means the only data leaving the sensor itself is the GPS coordinates of a parking space and its availability. This approach ensures compliance with all the relevant data privacy regulations.

REAL-TIME PARKING INFORMATION :

Relevant insights can then be passed on to drivers via a mobile application, website or digital signage. LED displays such as Cleverciti’s Circ360™ and Circ180™.

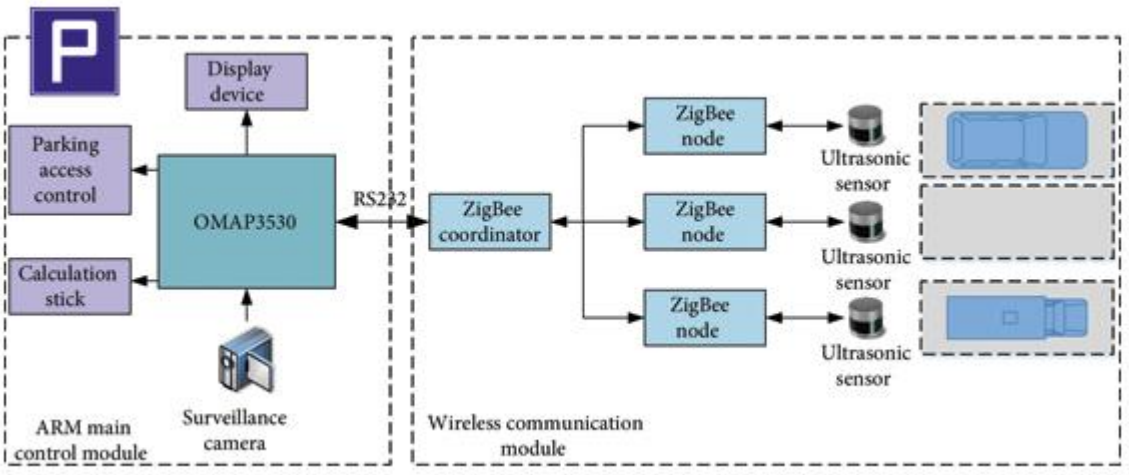
Smart Parking can be utilized in private parking lots, hospitals, hotels, shopping malls, public parking garages, offices, etc. to make the parking hassle free and time consuming. The intelligent parking system enables drivers to book the parking spots in advance and also get real-time accessibility of the parking spaces on their mobile devices.

Smart Parking includes the utilization of real-time data, low cost sensors and applications that empower users to analyze available parking slots. The aim is to automate and reduce time spent manually searching for the optimal parking floor, spot and even lot.

OVERALL DESIGN :

1.General Structure of the System :

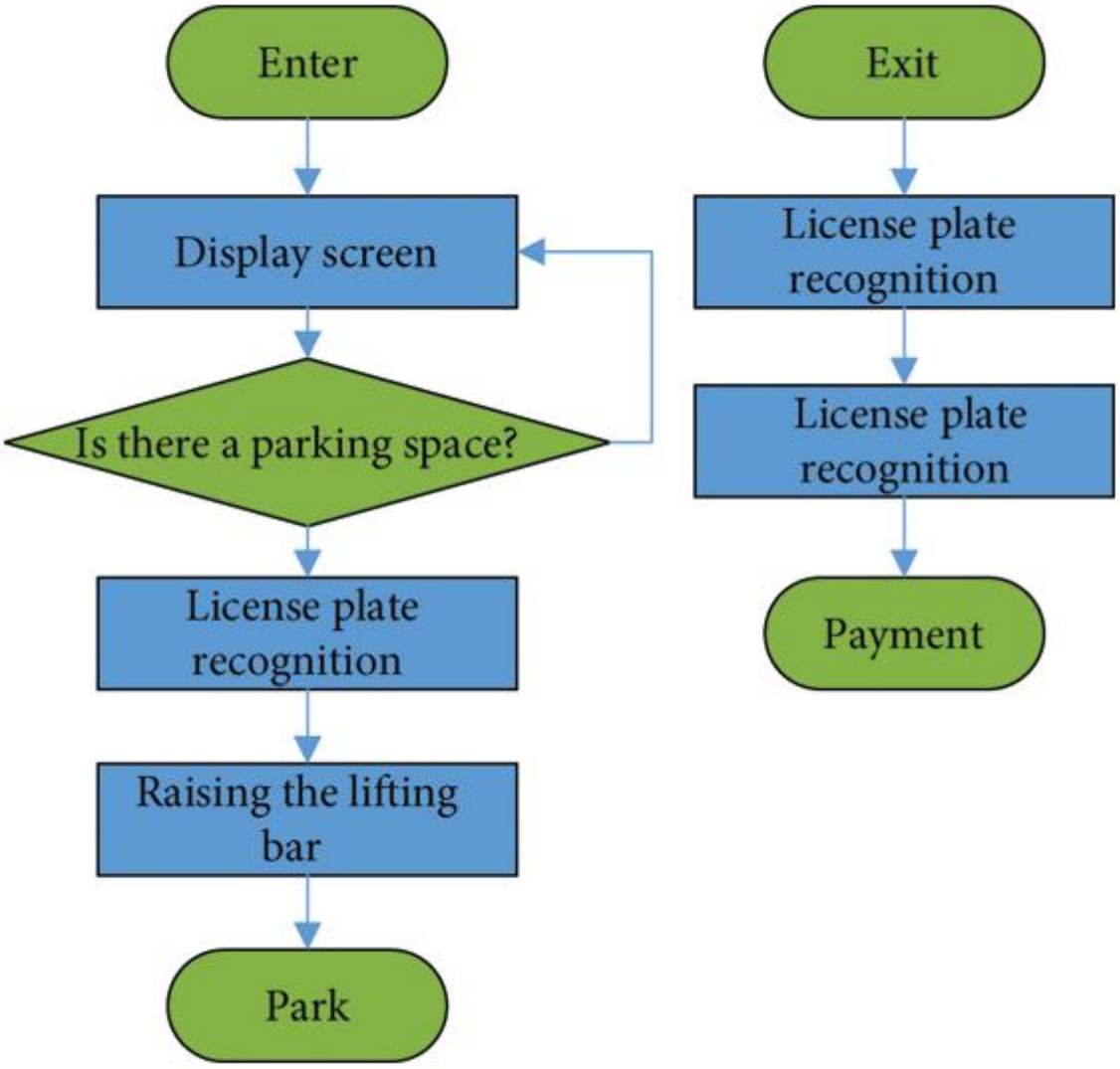
According to the function of the smart car park and the characteristics of ZigBee technology, the smart car park management system designed in this paper uses the ARM processor OMAP3530 as the main controller. The network transmission part of the system consists of a coordinator node, multiple router nodes, and multiple sensor data collection nodes. The overall structure of the system is shown in Figure.



At the entrances and exits of the car park, stop bars and surveillance cameras are placed to control the entry and exit of vehicles. When the vehicle arrives at the entrance, the system identifies the vehicle number and records the corresponding time and plate number. At the same time, the ARM controls the stop bar and releases the vehicle. License plate recognition can improve the efficiency of vehicles passing through entrances and exits. License plate recognition technology is currently used in many car parks. Each parking space detector is connected to a ZigBee node. The data detected by the car space detector is sent from the end node to the ARM main control module via a router. The coordinator does not need to add a parking space detection module.

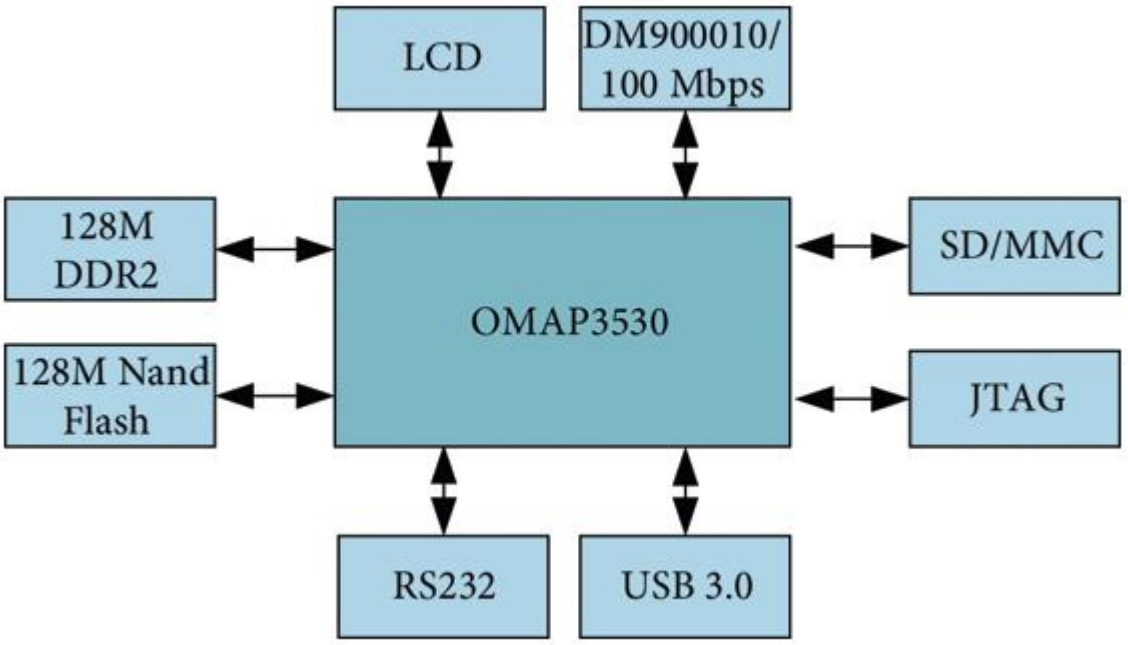
2. Workflow of the Parking Management System:

The user can observe the display at the entrance to the car park. If a space is available in the car park, the system displays the serial number and location of the available space. The user then drives the vehicle into the car park. At this point, the camera recognises the number plate and records the relevant data. If there is no space available in the car park, no space is displayed. When the user drives the vehicle to the exit of the car park, the camera recognises the number plate. The system charges the user a fee based on the rate and parking time. After the user has paid the fee, the system raises the stop bar and releases the vehicle. The workflow is shown in Figure.

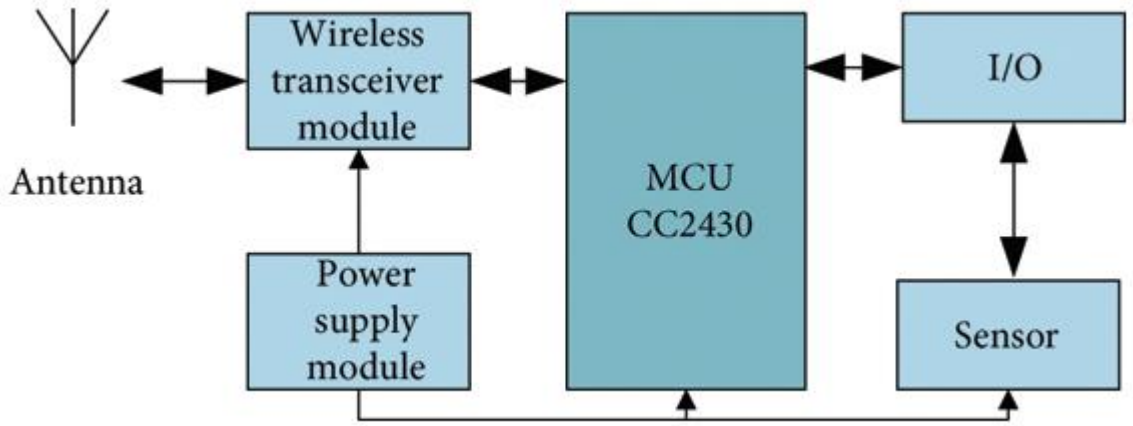


3. Design of the Hardware System:

The OMAP3530-Mini was chosen as the main controller platform for this system. The OMAP3530 is the latest generation of high-performance ARM processors from TI. The OMAP3530 uses the advanced Cortex A8 architecture and is capable of reaching a main frequency of over 1 GHz. The ARM main controller module uses the OMAP3530 processor as the core and contains modules such as LCD, DDR2, Nand Flash, and ZigBee wireless communication module. The structure of the ARM main controller module is shown in Figure.



The wireless communication module consists of a ZigBee wireless sensor network. The ZigBee coordinator node is connected to the ARM host controller module via the RS 232 serial port. The ZigBee coordinator node is responsible for data forwarding between the two modules. The ZigBee node consists of the CC2430 core processor, wireless transceiver module, sensors, I/O interface circuit, and power supply module. The structure is shown in Figure.

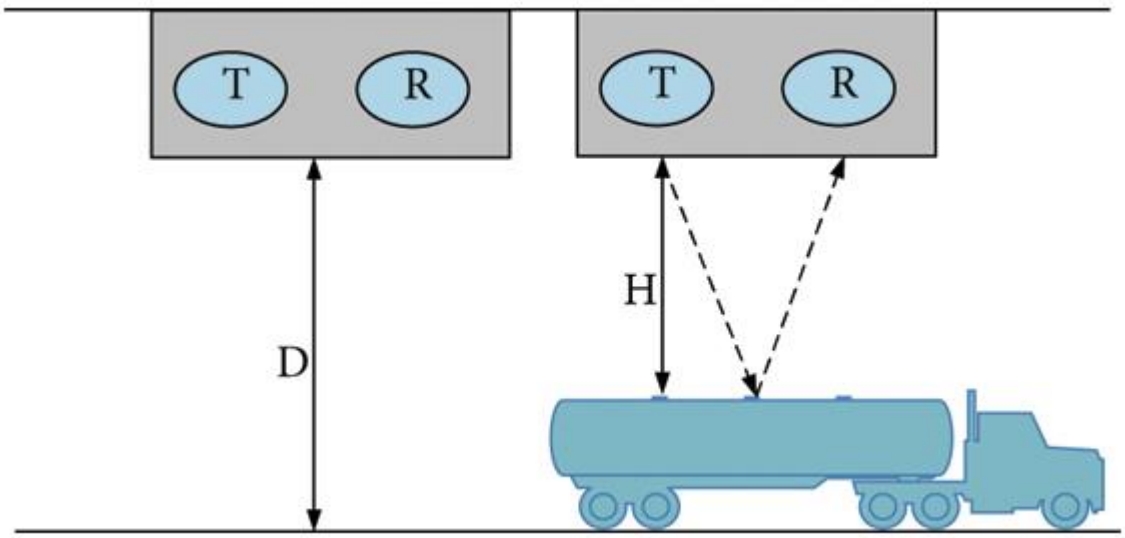


The core CC2430 processor is responsible for controlling the logic, routing protocols, synchronous positioning, power management, and task management of the entire node. The sensors are responsible for collecting the temperature in the designated monitoring area and completing the data conversion.

4. Ultrasonic Sensor-Based Parking Space Detection :

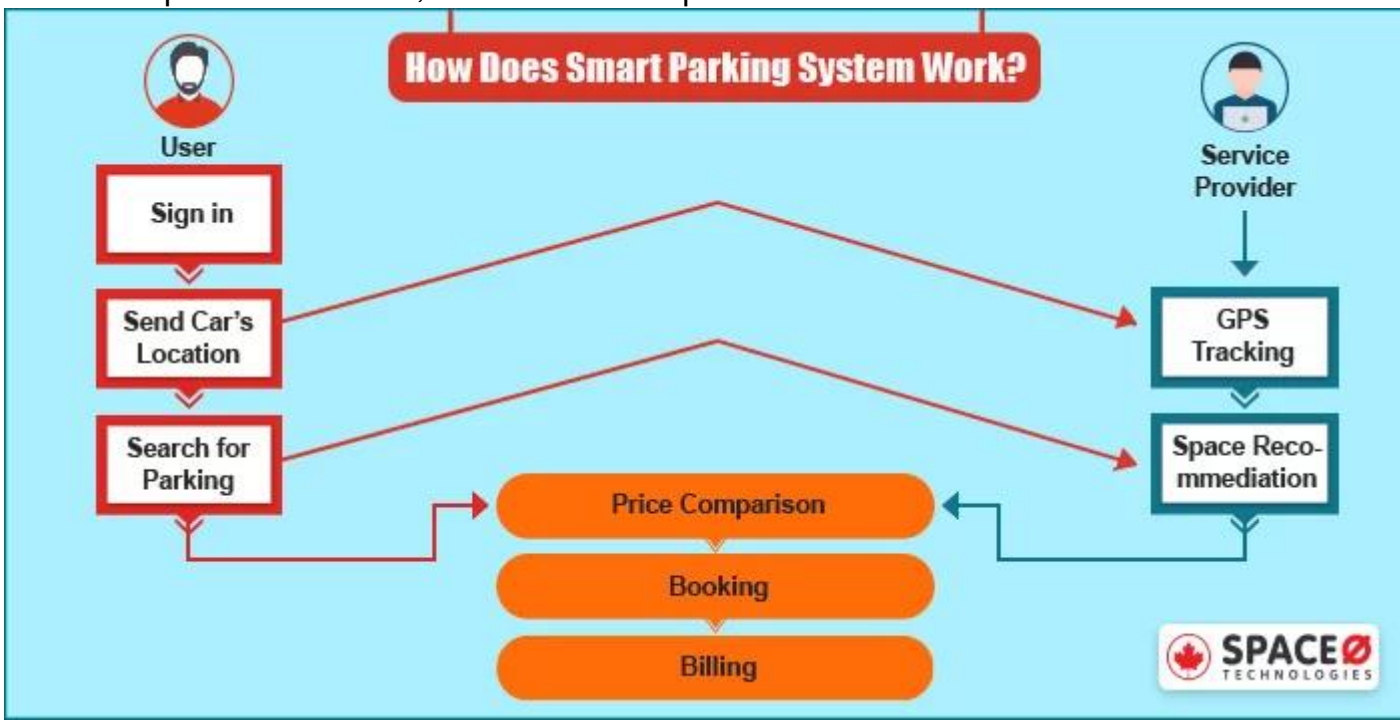
In an intelligent parking system, it is necessary to detect parking space information and determine whether there are any vehicles in the parking space. This function is the key part of the whole system. At present, the actual common detection technologies are geomagnetic detection technology, infrared detection technology, video image processing detection technology, ultrasonic detection technology, and so on. Ultrasonic sensors have the advantages of low cost, simple maintenance, strong anti-interference ability, and high measurement accuracy; therefore, this paper uses ultrasonic detection technology for real-time detection of car parking status in the car park. As the ultrasonic sensor is affected by the temperature, temperature compensation measures can be taken.

The ultrasonic distance measuring principle used in the intelligent parking system is shown in Figure . *T* indicates the ultrasonic transmitting device, *R* indicates the ultrasonic receiving device, *D* indicates the distance of the ultrasonic device from the ground, and *H* indicates the distance of the ultrasonic device from the roof of the car. The model number of the ultrasonic sensor is HC-SR04. The ultrasonic timing is shown in Figure .



SOFTWARE DESIGN :

Before you directly jump into the car parking application development process, you must know how the web application works from the user's point of view. So, here are the steps to consider.



1. User Send Car's Location :

1. Smartphone navigator enables GPS location in the app once downloaded.

2. Users send the car's location to the parking provider in the smart city.

2. Look for Space Availability :

1. The app provides a variety of parking slots for a better experience.

2. The user checks the occupied and available parking spaces of the area.

3. Parking Garage Allotment :

1. The app, if available, recommend the parking spot to the navigator.

2. Drivers also compare recommended slots by price or distance to find the perfect match.

SYSTEM TESTING :

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces (UI/UX) of those components so as to meet the end-user requirements and expectations. Major tasks performed during the system design process are initialising design definition, establishing design characteristics, assessing all possible alternatives and managing the design.

Requirement Specification:

For any project, narrowing down to a specific goal and targets is very important so that things do not get complicated later on. Requirements are basically of two types:

- **Functional**
- **Non-Functional requirements**

Estimation of Requirements:

Now that the system requirements are planned, the next step is to estimate the quantity or the amount of resources needed. This can include estimating the amount of storage needed, computing resources required, etc.

Data Flow:

This step involves identifying the type of data being handled and the type of database to use(Relational DB, NoSQL DB,etc.).

High level Component Design:

In this step, the components of the system are planned. In this step, the main components of the system and their relations are sketched.

JAVA SCRIPT:

IT IS VERY SIMPLE TO CREATE A SMART PARKING SYSTEM USING JAVASCRIPT.

```
<script>

    let btn=document.getElementById('btn')
    btn.addEventListener('click',fun)

    function fun(){

  var day = new Date();

  var entrance = day.getHours() + " AM:" + day.getMinutes() + ":" +
day.getSeconds();

  let parkingTable = document.getElementById("myTable");
```

```
         let name=document.getElementById("name").value
        let plate = document.getElementById("pl-num").value;
        let color = document.getElementById("carC").value;
        let phoneNum=document.getElementById("telNum").value;

var row = parkingTable.insertRow();
var numRow = document.getElementById('myTable').rows.length;

row.innerHTML = `
<td>${name}</td>
<td>${plate}</td>
<td>${color}</td>
<td>${phoneNum}</td>
<td>${entrance}</td>
`;

row.id = numRow;
var col= row.insertCell();

var rem = document.createElement('input');
    rem.type = "button";
    rem.value = "x";
    rem.style = "background-color:red;color:black;border-
radius:7px;width:50px;height:30px"
    col.appendChild(rem);

rem.onclick =
    function del() {
       document.getElementById(numRow).remove(numRow);
    };

};
```

HTML CODE :

```
<div class="row " style="margin-top: 70px;padding: 40px;">
        <div class="col-lg-6 col-md-6 ">
          <div class="cover">
            <img src="image/park.jpg" alt="">
            <div class="input">
            <div class="mb-3 " >
                <h2 style="padding: 20px;">Car Parking </h2>
                <form id="parking-form">
                <label for="inputPassword" class="col-sm-2 col-form-label"
style="font-size: 20px;">FullName</label>
                <div class="col-sm-10">
```

```
                    <input type="text" class="fullName" id="name">
                </div>

                </div>
      <div class="mb-3">
        <label for="inputPassword" class="col-sm-2 col-form-label"
style="font-size: 20px;">PlateNumber</label>
        <div class="col-sm-10">
          <input type="number" class="plate-number" id="pl-num">
        </div>
      </div>
```

```
<div class="mb-3">
        <label for="inputPassword" class="col-sm-2 col-form-label"
style="font-size: 20px;">Color</label>
        <div class="col-sm-10">
          <input type="text" class="car-color" id="carC">
        </div>
      </div>

      <div class="mb-3">
        <label for="inputPassword" class="col-sm-2 col-form-label"
style="font-size: 20px;">PhoneNumber</label>
        <div class="col-sm-10">
          <input type="tel" class="tel-num" id="telNum">
        </div>
      </div>
      <button type="button" class="btn btn-primary" id="btn"> Park
</button>
      </div>
    </form>
```

```
<div class="col-lg-6 col-md-6  card shadow-lg" style="background-color:
white;">
        <table id="myTable" class="table table-striped   table-
responsive">
            <thead>
                <tr>
                  <th scope="col">Full Name</th>
                  <th scope="col">Plate NUmber</th>
                  <th scope="col">Color</th>
                  <th scope="col">Phone Number</th>
                  <th scope="col">Check in time</th>
                  <th scope="col">Remove</th>
                </tr>
              </thead>
        </table>

      </div>
      </div>
```

CSS CODE:

```
body{
    margin: 0px;
    padding: 0px;
    box-sizing: border-box;

}
div.cover{
    position: relative;
}
div .input{
  font-size: 10px;
  color:white;
  position: absolute;
  top: 10%;
  padding: 10px;
  left: 5%;
}
<nav class="navbar navbar-expand-lg bg-dark border-bottom border-bottom-
dark">
        <div class="container-fluid">
          <h1 style="color:white;">ParkHere</h1>
          </div>
        </div>
      </nav>
```

CONCLUSION:

The more smart parking systems will be installed, the lower the unit cost will be in the end. Conclusion **Smart parking is a good investment for cities that have lots of traffic problems and want to increase their efficiency**. Also, for commercial real estate, this can be quite interesting.

Smart parking is a good investment for cities that have lots of traffic problems and want to increase their efficiency. Also, for commercial real estate, this can be quite interesting. Especially for large parking lots in

front of shopping malls, offices and industrial buildings, the technology can help to reduce emissions, time spent on searching for parking spots and cost.