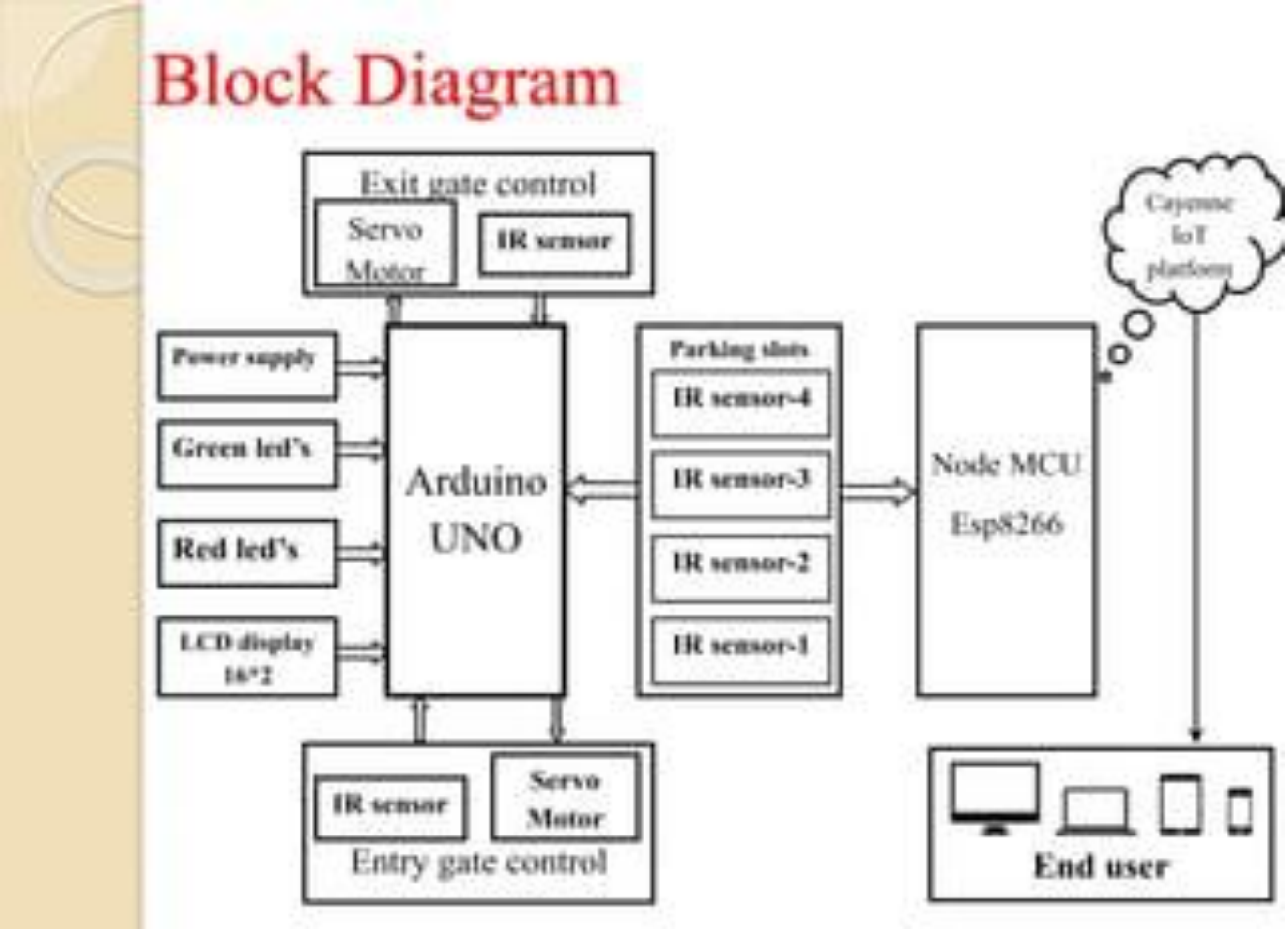

Smart parking system is a **software solution that incorporates IoT technologies such as sensing devices, cameras, or counting sensors** to identify which parking area is occupied or available and often uses this sensor data to design a real-time parking map. Such sensors communicate with the gateway and transfer the real-time data to the cloud.

How Does Smart Parking Work?

Smart parking systems rely on a combination of hardware, software, and real-time data to provide efficient parking solutions. The process involves various steps, from data collection to information dissemination. Let’s take a closer look at how smart parking works:

1. Data Collection: Smart parking systems use IoT sensors and cameras to collect real-time data about parking occupancy. These sensors are deployed in parking lots, garages, or on-street parking spaces and can detect the presence or absence of vehicles.
2. Data Processing and Analysis: The collected data is then processed and analyzed by the smart parking system. Advanced algorithms and machine learning techniques are used to analyze the data and determine the availability of parking spaces in real-time.
3. Information Display: Once the data is processed, the information about available parking spaces is presented to drivers. This can be done through mobile applications, online platforms, or digital signs placed at strategic locations. Drivers can access this information and make informed decisions about where to park.
4. Reservation and Payment: In some cases, smart parking systems offer reservation and payment features. Drivers can reserve a parking spot in advance through a mobile app and pay for their parking using digital payment methods. This not only ensures a guaranteed parking space but also allows for a seamless and cashless payment process.
5. Navigation Assistance: Many smart parking solutions also provide navigation assistance to drivers. This feature helps drivers locate the nearest available parking spot and guides them with turn-

by-turn directions to reach it. This reduces the time spent searching for parking and enhances the overall parking experience.

6. Monitoring and Operations: Smart parking systems also have a backend monitoring and operations component. This allows parking operators to monitor the occupancy rates, collect data for future analysis, and manage any maintenance or issues with the sensors or the system itself.

By leveraging these technologies and processes, smart parking systems provide drivers with real-time parking information, optimize space utilization, reduce congestion, and enhance the overall parking experience

Coding

```
import RPi.GPIO as GPIO

from time import sleep

import requests

from firebase import firebase

fbase = firebase.FirebaseApplication('https://iot-smart-parking-bdf95.firebaseio.com/',None)
```

```
GPIO.setmode(GPIO.BCM)


GPIO.setup(14,GPIO.IN)

GPIO.setup(4,GPIO.IN)

GPIO.setup(26,GPIO.IN)

GPIO.setup(27,GPIO.IN)




while True:

    sensedata = GPIO.input(14)

    sense2 = GPIO.input(4)

    sense3 = GPIO.input(27)

    sense4 = GPIO.input(26)


    getdata = fbase.get('nodes/node1/reading','')

    getdata2 = fbase.get('nodes/node2/reading','')
```

```
getdata3 = fbase.get('nodes/node3/reading','')

getdata4 = fbase.get('nodes/node4/reading','')



if sensedata==int(getdata):

    print("No state change in node 1")

else:

    print("State changed in A1")

    res = fbase.put('nodes/node1','reading',str(sensedata))

    if sensedata == 0:

        print("car parked in A1")

        sleep(2)

    elif sensedata == 1:

        print("Car Left from A1")

        sleep(2)

if sense2==int(getdata2):

    print("No state change in node 2")
```

```
else:

    print("State changed in 2")

    res = fbase.put('nodes/node2','reading',str(sense2))

    if sense2 == 0:

        print("car parked in A2")

        sleep(2)

    elif sense2 == 1:

        print("Car Left from A2")

        sleep(2)

if sense3==int(getdata3):

    print("No state change in node 3")

else:

    print("State changed in A3")

    res = fbase.put('nodes/node3','reading',str(sense3))

    if sense3 == 0:

        print("car parked in A3")
        sleep(2)

    elif sense3 == 1:

        print("Car Left from A3")

        sleep(2)

if sense4==int(getdata4):

    print("No state change in node 4")

else:

    print("State changed in 4")

    res = fbase.put('nodes/node4','reading',str(sense4))

    if sense4 == 0:

        print("car parked in A4")

        sleep(2)

    elif sense4 == 1:

        print("Car Left from A4")

        sleep(2)
```