INTERNET OF THINGS GROUP-4

IOT- SMART WATER MANAGEMENT
PHASE-5

NAME	**: R.MUKESH**
REGISTER NO	**: 822621106303**
NM ID	**: aut2282260006**
DEPARTMENT	**: ELECTRONIC& COMMUNICATION ENGINEERING**
COLLEGE CODE	**: 8226**
COLLEGE NAME	**: ARIFA INSTITUTE OF TECHNOLOGY NAGAPPATTINAM**
EMAIL ID	**: mukeshttk9178@gmail.com**

ABSTRACT

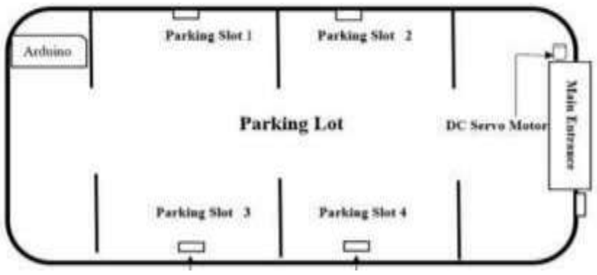
The project entitled SMART PARKING SYSTEM using Iot , the major motivation of this project is to reduce the traffic congestion in roads, multi-storeyed buildings and malls due to unavailability of parking spaces .The project displays the nearest empty slot if present with respect to user location. Our project aims to make efficient use of parking spaces. We track vacant slots in the parking space and assign that to the user. Smart parking system as described above can lead to an error-free ,reliable, secure and fast management system. In recent times the concept of smart cities have gained great popularity. Thanks to the evolution of the Internet of things the idea of smart city now seems to be achievable. Consistent efforts are being made in the field of IoT in order to maximize the productivity and reliability of urban infrastructure. Problems such as, traffic congestion, limited car parking facilities and road safety are being addressed by IoT. The proposed Smart Parking system consists of an on-site deployment of an IoT module that is used to monitor and signalize the state of availability of each single parking space. A mobile application is also provided that allows an end user to check the availability of parking space and book a parking slot accordingly. The paper also describes a highlevel view of the system architecture. Towards the

INTRODUCTION

The project entitled smart parking system is to manage all the parking facilities to an user. The recent growth in economy and due to the availability of low price cars in the market, an every average middle-class individual can afford a car, which is good thing, however the consequences of heavy traffic jams, pollution, less availability of roads and spot to drive the motor car. One of the important concerns, which is to be taken in accounting, is the problem of parking those vehicles .Though, if there is space for parking the vehicle but so much time is squandered in finding that exact parking slot resulting in more fuel intake and not also environment friendly. It will be a great deal if in some way we find out that the parking itself can provide the precise vacant position of a parking slot then it'll be helpful not limited to the drivers also for the environment . Initially when the user is about to enter the location the LCD displays the number of empty and filled spots and when the user is with its vehicle near to the parking detect sensor ,he/she would be thrown with a notification on their mobile app of the parking slot number ,where they should park there vehicle.

Relevance of the project

The main important benefit of a smart parking system is its advanced technology. It follows the latest technologies and concepts to assure profitable outcomes . The design and implementation of smart parking is very easy to supervise and manage. This system can be easily handled by the staff members because of its well organized structure.



block diagram of smart parking system

Problem Statement

In recent research in metropolitan cities the parking management problem can be viewed from various angles such as high vehicle density on roads. This results in annoying issues for the drivers to park their vehicles as it is very difficult to find a parking slot.

The drivers usually waste time and effort in finding parking space and end up parking their vehicles finding a space on the street which further leads to space congestion. In worst case, people fail to find any parking space especially during peak hours and festive season.

Objective

Smart Parking involves the use of low cost sensors, real-time data and applications that allow users to monitor available and unavailable parking spots. The goal is to automate and decrease time spent manually searching for the optimal parking floor, spot and even lot. Some solutions will encompass a complete suite of services such as online payments, parking time notifications and even car searching functionalities for very large lots. A parking solution can greatly benefit both the user and the lot owner.

Optimized parking – Users find the best spot available, saving time, resources and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities.

Reduced traffic – Traffic flow increases as fewer cars are required to drive around in search of an open parking space.

Reduced pollution – Searching for parking burns around one million barrels of oil a day. An optimal parking solution will significantly decrease driving time, thus lowering the amount of daily vehicle emissions and ultimately reducing the global environmental footprint.

Increased Safety – Parking lot employees and security guards contain real-time lot data that can help prevent parking violations and suspicious activity. License plate recognition cameras can gather pertinent footage. Also, decreased spot-searching traffic on the streets can reduce accidents caused by the distraction of searching for parking.

Decreased Management Costs – More automation and less manual activity saves on labor cost and resource exhaustion.

Enhanced User Experience – A smart parking solution will integrate the entire user experience into a unified action. Driver's payment, spot identification, location search and time notifications all seamlessly become part of the destination arrival process.

Scope of the project

At present some countries have portals which users can gain information about parking areas via the internet. This system can give users the information about parking space, but it won't be able to give which parking slot is vacant and occupied. Hence, such a system cannot smartly handle the issue. Car lifts along with an automated robotic system, which automatically takes the car to a particular parking spot as soon as the car enters on a platform. This system cannot be installed by medium scale shopping malls, movie theatres as it can cost them a huge amount. At many public places, the system only shows the availability but it cannot show the exact slot and path to the slot available. Hence, there is the need to smartly find the path to the vacant spot.

Methodology

In this project we are using NodeMCU, IR sensors, and servo motors. One IR sensor is used at entry and exit gate to detect the car while two IR sensors are used to detect the parking slot availability. Servo motors are used to open and close the gates according to the sensor value. NodeMCU is an open source IoT platform .It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware, which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language. The ESP8266 is a low-cost Wi-Fi enabled microchip with full TCP/IP stack and microcontroller capability. NodeMCU includes CPU core, faster Wi-Fi, more GPIOs, and supports Bluetooth 4.2, and low power Bluetooth. The ESP8266 is a low-cost WiFi enabled microchip with full TCP/IP stack and microcontroller capability. NodeMCU includes CPU core, faster Wi-Fi, more GPIOs, and supports Bluetooth 4.2, and low power Bluetooth. As soon as the IR sensors get the presence of a car in front of the entrance, it will send signal to the NodeMCU to check if there is an empty slot inside the parking lot. When NodeMCU acknowledges that there is an empty slot or more then it will send a signal to the dc servo motor which will open the main entrance. On the other hand if an NodeMCU encounters no empty slots at the time of a car trying to make an entrance, the gate will just not open. In addition, there will be a website linked with the NodeMCU board to show the number of parking.

The idea behind our methodology is very simple , usually users spend most of their time in looking for an empty slot where they can park their vehicle which increases fuel consumption and time wastage. We came-up with a new method where we provide the user an empty slot number where he can park his vehicle without wasting his time for finding one . Similarly we try to display the start time and end time so that the user can know for what amount of time he has parked his vehicle.

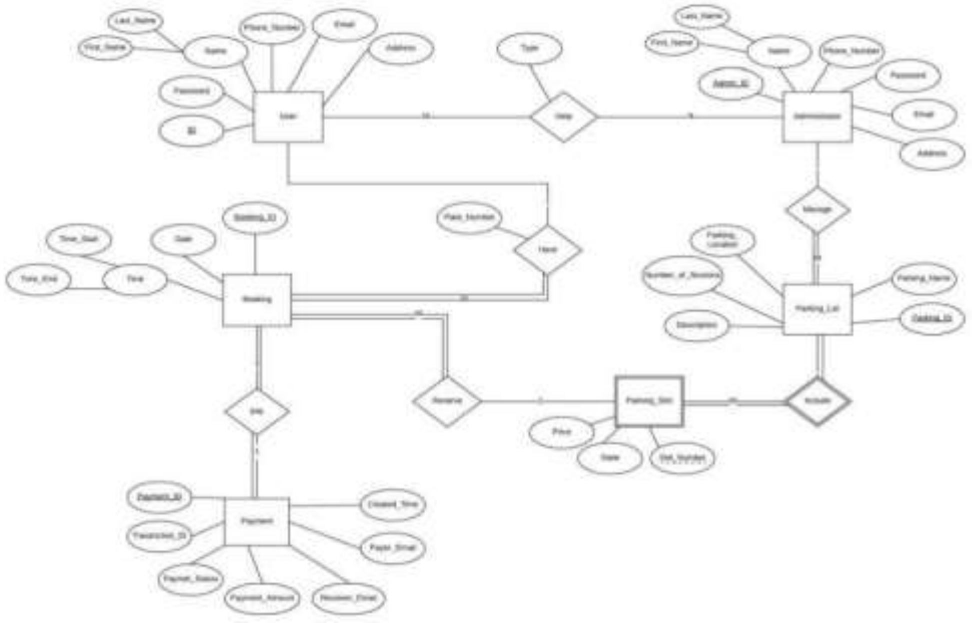
LITERATURE SURVEY

[1] Developing a Smart Parking Management System Using the Internet of Things

Searching for parking wastes significant amounts of time and effort and leads to substantial financial costs. This is particularly the case for people who are always pressured to be on time. Smart cities employ all kinds of modern technologies to manage and enhance resources effectively. Urban parking facilities are one of the essential assets that must be managed. We developed a smart parking management system (SPMS) as a modern solution to manage parking and save users time, effort and cost. In the context of today’s modern life, it has become necessary to improve search methods for available parking and minimize the congestion that occurs at the parking entrance. Searching or booking available parking online earlier is a better substitute than searching at a parking lot where there is a possibility of not being able to find parking. Our smart parking management system was developed to: • Manage parking and solve problems efficiently using technology

• Apply technical solutions to improve the smart cities concept

The proposed system uses a variety of technologies that help manage parking. It provides essential services for users, including searching for parking, reservations and payment. It is extended to cover more advanced services such as receiving notifications, statistics and monitoring the parking state. The system is connected to sensors to detect occupancy and an automatic number plate recognition (ANPR) camera to control access. The remainder of the paper is organized as follows.



Entity Relationship Diagram of Smart Parking System

[2] An IoT-based E-Parking System for Smart Cities

The huge proliferation in the number of vehicles on the road along with mismanagement of the available parking space has created parking related problems as well as increased the traffic congestion in urban areas. Thus, it is required to develop an automated smart parking management system that would not only help a driver to locate a suitable parking space for his/her vehicle, but also it would reduce fuel consumption as well as air pollution. It has been found that a drivers search for a suitable parking facility takes almost 15 minutes which increases the fuel consumption by the vehicle, traffic congestion and air pollution. A significant amount of research works exist in the area of design and development of smart parking system. Various features of smart parking system are listed below.

• Inquiry on availability of parking space and reservation of parking lot

• Real-time parking navigation and route guidance

• Vehicle occupancy detection and management of parking lots .

Most of the smart parking systems (SPS) proposed in literature over the past few years provides solution to the design of parking availability information system, parking reservation system, occupancy detection and management of parking lot, real-time navigation within the parking facility etc. However, very few works have paid attention to the real time detection of improper parking and automatic collection of parking charges. Thus, this paper presents an internet-of thing (IoT) based E-parking system that employs an integrated component called parking meter (PM) to address the following issues.

• Real-time detection of improper parking

• Estimation of each vehicles duration of parking lot usage

• Automatic collection of parking charges

The E-parking system proposed in this paper also provides city-wide smart parking management solution via providing parking facility availability information and parking lot reservation system and it is named as parking meter (PM) based E-parking (PM-EP).

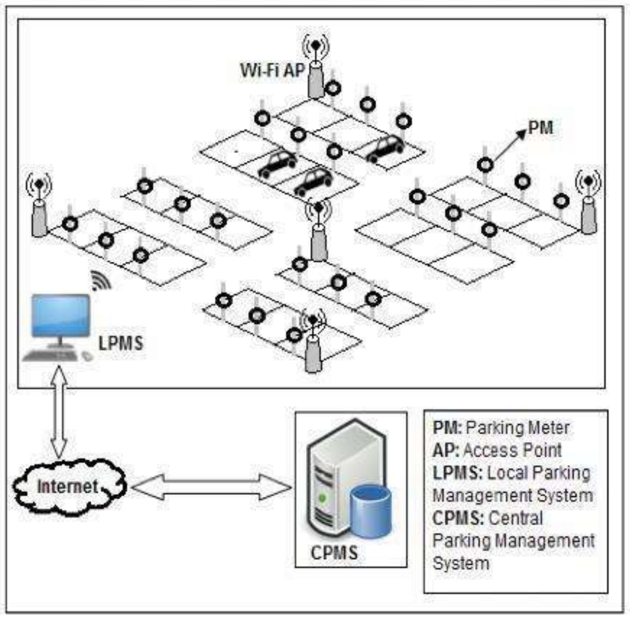

Wi-Fi AP
PM
LPMS
Internet
CPMS
PM: Parking Meter
AP: Access Point
LPMS: Local Parking Management System
CPMS: Central Parking Management System

Fig 2.2 Network Architecture of proposed System

[3] Smart Parking based System for smarter cities

India is getting motorized i.e. the rate of private vehicles is more as compared to public transports. As the rate of people owning their vehicles increases ,the need of parking slots to park vehicles also increases. But currently the scenario is that there are not sufficient parking slots available or there is also possibility that people are not now aware about the legal parking slots available in their locality. This situation leads to the unnecessary crowding of vehicles on the road and also results in inconveniency of people walking on the road. To overcome above problems, We are proposing the solution in the form of a multilingual android application which will be helpful for the people to find their parking slots digitally. By digitally we mean that this particular system will assign the parking slot based on the current location of the user and the parking slot which the user wants according to his/her ease. Ease in terms of finding the exact slot. The payments can be done digitally or through vending machines. The end user can register and login with his/her account which will help the system to find the location and displaying the nearest parking area and nearest parking slot ,whether it is available or not. If not then it will direct user to the next nearest slot and so on. The existing system comprises of both traditional and application based approach for parking. If we talk about the traditional approach it utilizes manual method of parking i.e user has to find the spot for parking by traveling to far distances and paying extra money. An application based approach consist of the applications which provides the parking slots for the particular locality for example .The application named ‘Parking Panda ’ provides the parking slots to the areas like stadium, sports leagues etc.

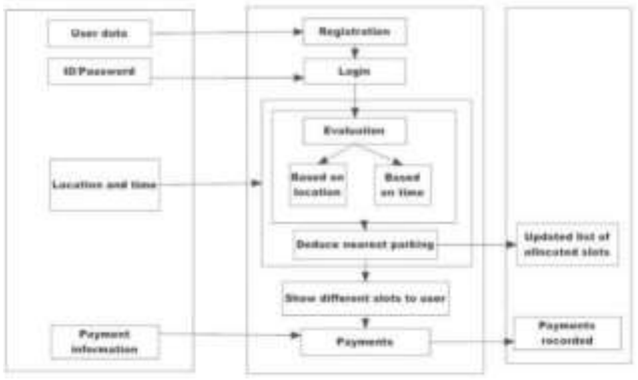


Fig 2.3 Block Diagram of parking system

[4] SMART PARKING SYSTEM TO REDUCE TRAFFIC CONGESTION

Transportation is the key-success for any of the country. Now a day, many people have options to use their own vehicle for travelling. This will surely increase the demand in trading but one of the problems created by road traffic is "parking". To park all these vehicles in the major metro cities is quite tedious and difficult task and it became problematic to park vehicles. Lot of research and development is being done all over the world to implement better and smarter parking management mechanisms. The current smart parking systems or Wireless Sensors Network Parking requires the combination of wireless sensor networks module, Embedded web-server, Central WebServer. Sensor networks make use of Infrared (IR) Sensor nodes to check the parking slot state and send this information to embedded web-server. It thereby displays the information on a LED screen with which the user can check for empty vehicle slots. These systems not guide the users to reach to the parking lot. If the slot is not available at that time than drivers will start searching for another parking zone so that this process is time consuming and will increase the traffic congestion.

This paper proposes a Reservation-based Smart Parking System for avoiding the traffic problems that provides the pre- booking of slots through the use of the mobile application. This application is expected to provide an efficient and cost- effective solution to the vehicle parking problems. Application must be installed in the user's mobile. Unlike the existing system, our idea is to use client-server architecture where client request for the reservation of slots and server responds with the slots which are available at that time. Our system is that the user has an option to go for the parking area according to his/her convenience. The advantage of this will greatly reduce the time taken by the vehicle to search for a parking area. Advanced payment modules are also included like e- wallet, debit card, credit card from which the user can pay. Penalty will be added on late exit as well as an over use of the slot after user specified entry and exit time .The refund will be given on cancelation of parking slot and early exit. The supervisor is required to monitor the area.

Many of the vehicles parking facilities are unable to cope with the influx of vehicles on roads and parking area. The current smart parking systems or Wireless Sensors Network Parking requires the combination of wireless sensor networks module, Embedded web-server, Central Web-Server. Sensor networks make use of Infrared (IR) Sensor nodes to check the parking slot state and send this information to embedded web-server. It thereby displays the information on a LED screen with which the user can check for empty vehicle slots . Also image capturing devices are used for continuously clicking pictures of parking area to ensure empty slots which results in high power consumption and also high maintenance cost is required . There are some systems in the market like the smart parking services which are based on the wireless sensor networks which uses wireless sensors to effectively find the available parking space. But to use this system, additional hardware needs to be installed in the car which is not feasible . Finding a parking slot in a congested city is very hard. In many cases people go to a parking station and they find it full and there is no space available for parking. Then in search of parking space they have to again roam with their vehicle to find available parking.

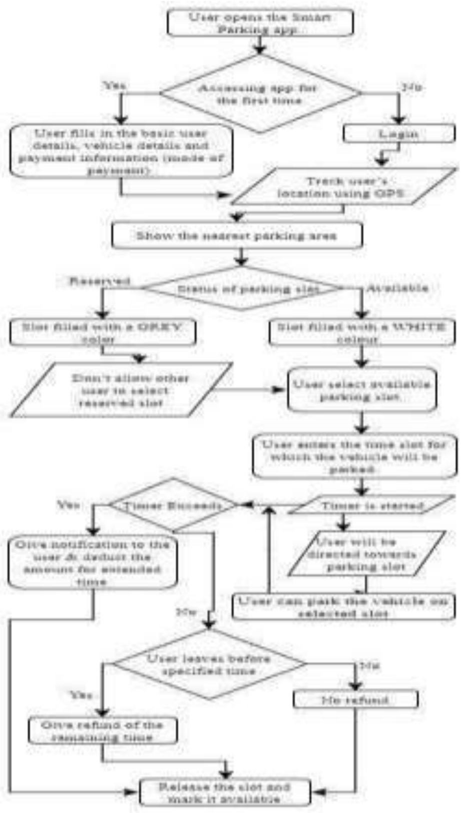
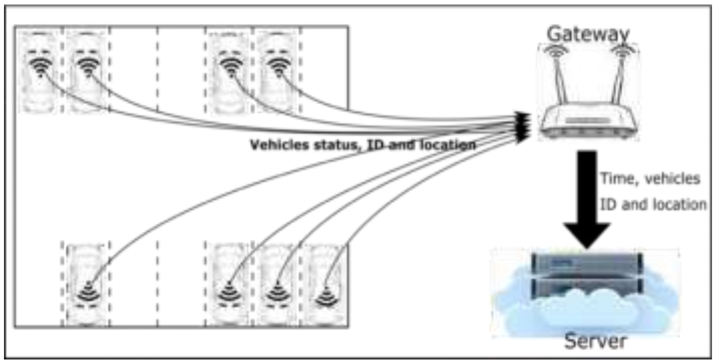

User opens the Smart Parking app
Accessing app for the first time
Yes
No
User fills in the basic user details, vehicle details and payment information (mode of payment)
Login
Track user's location using GPS
Show the nearest parking area
Status of parking slot
Reserved
Available
Slot filled with a GREY color
Slot filled with a WHITE colour
Don't allow other user to select reserved slot
User select available parking slot
User enters the time slot for which the vehicle will be parked
Timer is started
Timer Exceeds
Yes
Give notification to the user & deduct the amount for extended time
User will be directed towards parking slot
User can park the vehicle on selected slot
No
User leaves before specified time
Yes
No
Give refund of the remaining time
No refund
Release the slot and mark it available

Fig 2.4 Flow chart of Smart Parking System

[5] An IoT-Based Intelligent System for Real-Time Parking Monitoring and Automatic Billing

Today, the parking industry is being transformed by new technologies that are allowing cities to reduce rates of congestion significantly. Sensor networks that sense vehicle occupancy are providing the basic intelligence behind smart parking systems. Thanks to the Smart Parking technology, it is now possible to know in real-time the location of free parking spaces and to help drivers to get to their ultimate destination. A variety type of vehicle detectors has been used in parking information acquisition. These vehicle detectors mainly include the inductive loop , acoustic sensor , infrared sensor , or ultrasonic sensor . System using video camera sensor technologies have been proposed to collect the information in vehicle parking field. However, a video camera sensor is vulnerable to bad weather and night time operation. Furthermore, it is expensive, and can generate a large amount of data that can be difficult to transmit in a wireless network. The magneto-resistive based detection systems combined with a wireless area network are the most popular technique due to their high accuracy. Yet, this type of sensor is facing different issues, i.e. it can be bedeviled by electromagnetic interference, which affects the accuracy , the reading from sensor needs to be collected constantly which will result in wearing out the battery . To extend the battery lifetime and increase the vehicle detection accuracy, a parking sensor system has been proposed. While power management technique has been implemented to optimize energy consumption, high occupancy monitoring accuracy is achieved using two-fold sensing approach. It is a sequence of darkness and Signal Strength Indicator (RSSI) measurement based techniques. The wireless sensors are still intrusive, they are embedded in the pavement, or taped to the surface of each individual parking lot. Existing sensors, such as ground based parking sensors costs up to $200 per parking lot . As consequence, smart-parking technology using wireless sensors for outdoor parking is costly due to the large number of sensors units required to cover the entire parking lot . Although, parking occupancy monitoring systems have made a significant progress, smart parking payment is rarely studied in smart parking research . Yet, there are companies working on the patents of parking systems for payments. A first approach consists in using a camera or an RFID transceiver for vehicle detection and identification . A limitation of this solution lies in that the system is complex and its implementation is expensive when a detection device is installed on each parking lot. Furthermore, when only RFID transceiver is used for vehicle detection and identification, the system can be bedeviled by electromagnetic interference, which affects the accuracy. Moreover, this system is designed to detect a vehicle when entering a parking and seek payment, whereas information on vacant parking lots is not provided. A technique for monitoring vehicle parking using one camera to record the entrance of a vehicle and a second camera to record the vehicle leaving the parking has been proposed . Moreover, in a system and method for obtaining and displaying information on vacant parking space is described. When a user occupies a parking space designated with an individual ID, he enters this ID into a parking meter or via a smart phone mobile app., and pays the parking fees. The database processes the received data and changes the status of the parking space with its ID from unpaid to paid. These data are used as information on the occupation of a parking space. In this paper, we propose a smart sensor system allowing outdoor parking monitoring and payment without requiring any user/driver interaction. It will be deployed without having to install new components on each parking lot. The proposed sensor has benefits in terms of detection and payment reliability, and reduced expense by reducing the system complexity and installation, and extending batteries lifetime through the reduction of the system power consumption.


Gateway
Vehicles status, ID and location
Time, vehicles ID and location
Server

Proposed system architecture; wireless occupancy sensor; wireless gateway; data storage and processing unit.

Summary of the Approaches

Approaches	
It deals with saving financial cost by developing the system in the most efficient manner .	Keeps a count on number of vacant spots and prior booking.
Real time monitoring vehicle occupancy	Displays appropriate message
Android app for providing all parking	Specifying every minute
details to the user.	information to the user using android application .

SYSTEM REQUIREMENTS SPECIFICATION

Functional Requirement

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs/or conditions. These may include calculations, data manipulation and processing and other specific functionality. In these systems following are the functional requirements

- The application should not display in-appropriate message for valid conditions.
- The application must not stop working when kept running for even a long time.
- The application should process information for any kind of input case.
- The application should generate the output for a given input test case .

Non-Functional Requirement

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours.

Given below are the non-functional requirements:

- Product requirements
- Organizational requirements
- Basic operational requirements

Hardware Specifications

- ENODE MCU (ESP8266)
- JUMPER WIRES
- INFRARED SENSORS
- 16*2 LED DISPLAY
- DC MOTOR

Software Specification

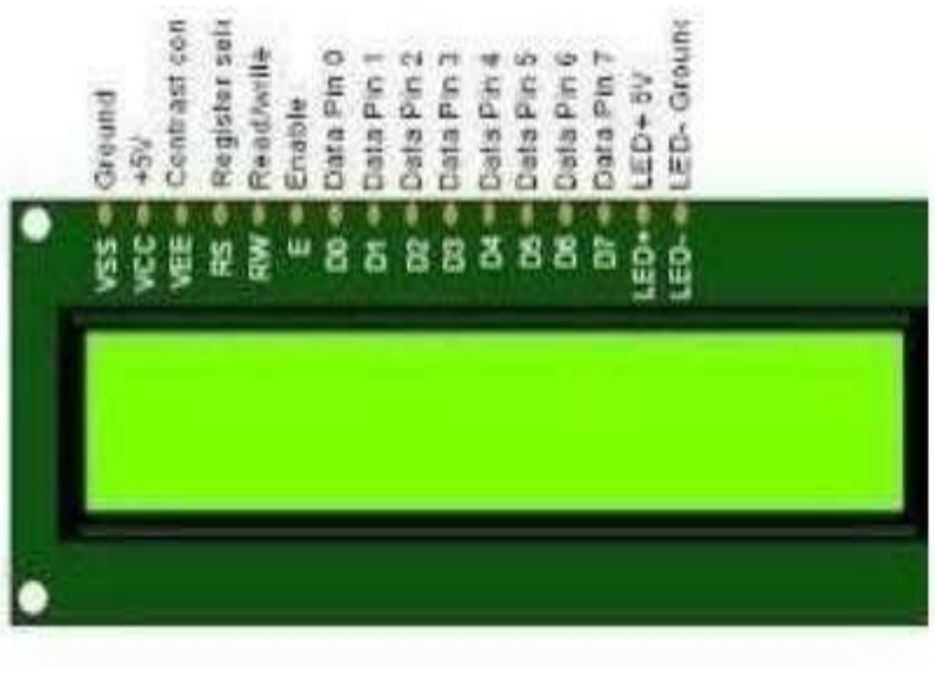
- ARDUINO IDE

Node MCU

The NodeMCU as shown in Fig has assimilated TCP/IP protocol that can give any microcontroller entrance to the Wi-Fi network that supports 2.4 GHz Wi-Fi (802.11 Wi-Fi standards). NodeMCU is capable of either connecting to an existing wireless connection or hosting an application over http protocol. Each Node MCU module comes pre-programmed with an AT command set firmware which means one can simply link this up to your Raspberry Pi device and get about like Wi-Fi shield. The reason why we use node mcu is that it is more cost-efficient with respect to Arduino uno , in Arduino we have to use ethernet shield which provides us secure ethernet connectivity whereas all these features are provided by node mcu and it also comes with a updated feature of wi-fi , where you can power or connect your system by WiFi.

16*2 LCD Display

An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIY's and circuits. The 16×2 translates o a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix. The 16*2 display is used to display the number of vacant and spilled spot . It also gets updated on the display LCD when a vehicle parks or unparks the vehicle .



IR sensor:

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum). Anything that emits heat (everything that has a temperature above around five degrees Kelvin) gives off infrared radiation. We are using three IR detect sensor in our project , one IR detect sensor is used to sense the vehicle near the parking sensor and other two IR detect sensor is used to send data to the node mcu which is the brain of our system whether a vehicle is parked in that slot or is unparked .

SYSTEM REQUIREMENTS SPECIFICATION

Functional Requirement

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs/or conditions. These may include calculations, data manipulation and processing and other specific functionality. In these systems following are the functional requirements

- The application should not display in-appropriate message for valid conditions.
- The application must not stop working when kept running for even a long time.
- The application should process information for any kind of input case.
- The application should generate the output for a given input test case .

Non-Functional Requirement

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. Given below are the non-functional requirements:

- Product requirements
- Organizational requirements
- Basic operational requirements

Hardware Specifications

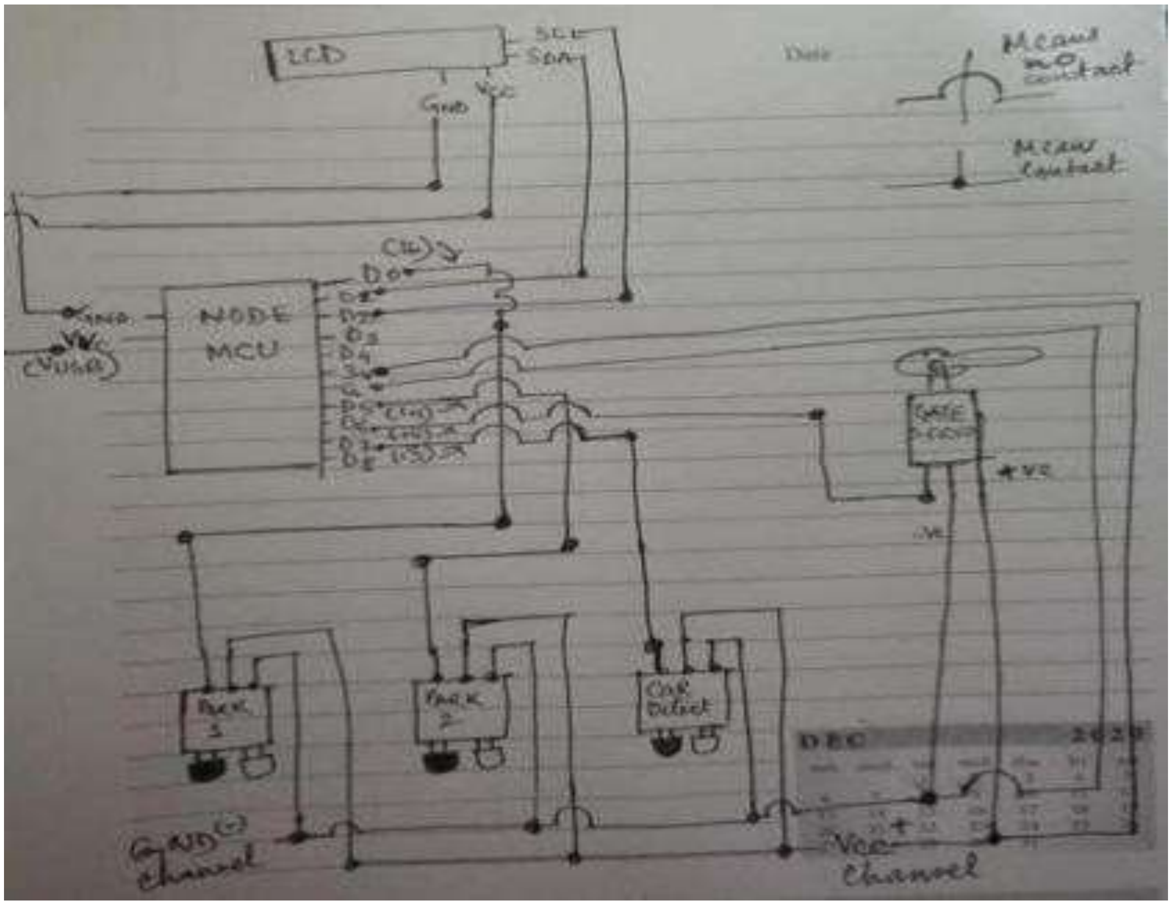
- ENODE MCU (ESP8266)
- JUMPER WIRES
- INFRARED SENSORS
- 16*2 LED DISPLAY
- DC MOTOR

Software Specification

- ARDUINO IDE

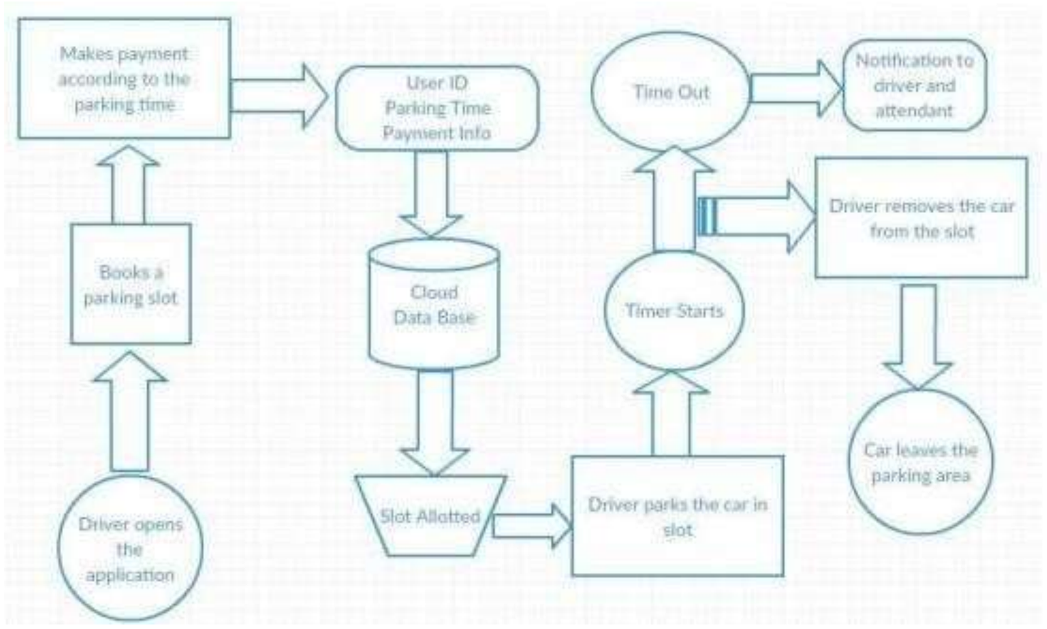
System Architecture

The below diagram shows the pin diagram of our model. It consists of one node mcu , one dc motor , one 16*2 LCD display and three IR sensors .The node mcu is the brain of our system which powers all the other devices .The 16*2 LCD display is powered by node mcu by connecting jumper wires from the display to node mcu . The DC motor is also powered by node mcu with connecting its pins to node mcu. The IR sensor consists of three pins, where two pins refer to the power supply and ground and the other pins refer to the pin which is going to be connected in the Node mcu. On successfully connecting all the components in the given figure now we have to connect the blynk app. While using the blynk app we have to specify the widgets used in our android app and the pin number to which they are connected to node mcu in the actual model so that the mobile app will react exactly to the inputs provided in the model .



IMPLEMENTATION

Flowchart of the System:


Makes payment according to the parking time
User ID Parking Time Payment Info
Time Out
Notification to driver and attendant
Driver removes the car from the slot
Books a parking slot
Cloud Data Base
Timer Starts
Car leaves the parking area
Driver opens the application
Slot Allotted
Driver parks the car in slot

Below are the steps that a driver needs to follow in order to park its car using our parking system.

Step 1: Install the smart parking application on your mobile device. Step 2: On the 16*2 display the number of vacant and filled spots are displayed so that the user can see the status of parking zone. Step 3: Once the user logs into the app he would see the parking architecture with the cars filled at which position and positions which are empty . Step 4: When the user is near to the parking IR detect sensor , he would receive a notification on his app on which slot he can park his vehicle if there is a empty slot. Step 5: If there is no empty slot the user will be displayed with an appropriate message on the mobile application . Step 6: On availability of parking area and user parking into the respective slot he/she would receive a message which states the start time of the parking and the slot in which he/she has parked. Step 7: On successfully un-parking your vehicle from the parking slot the user will receive a message which states the start time and end time of his parking time and an amount which he needs to pay for the parking duration.

Design of the System

This model has the capacity of containing two cars. There are two sensors at the entrance to detect the presence of a car before going inside or outside of the parking lot. The other two sensors are plotted inside the parking lot to detect the car individually for each parking slot. A DC Servo motor has been used at the entrance to open and close the gate according to the signals sent by the sensors through Arduino. The projection on the screen corresponds to the system model parking slots. This is a real time display regarding the status of the parking lot. As this is a web-based representation, anyone will be able to get the status of the parking lot by visiting the website on the URL through their cell phones, laptops, desktops and other internet supporting devices. The model of the parking lot has two parking slots. Thus, we can park a maximum number of two cars through the system. We have used two IR sensors which when vehicle parked will show appropriate message to the user and when all the parking slots the dc motor would not open gate for the vehicle to be parked. Displaying of appropriate message for any action which takes place in the parking zone is done effectively and efficiently .

Network Time protocol :-

The Network Time Protocol is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. We have used NTP for fetching time from the NTP server so that we can show the start time and end time for the user when he parks or unparks his vehicle making information real-time.

Blynk app:-

Blynk app is a hardware-agnostic IoT platform with white label mobile apps, private cloud ,device management, data analytics and machine learning .On using the blynk app we tried to pop notification to every possible event that is occurring in the parking zone . Used a serial algorithm to display the slot number to the user who is going to park his vehicle .For example we display the empty slot number in a serial manner which gets filled , if the slot 1 is filled and when an another vehicle turns up we display slot 2 and further like these for all other vehicles , and if any vehicle leaves the slot number then we display the earlies slot number , not making the user to travel long if an initial spot is vacant .

RELATED WORKS :

- The sensors used in IoT based smart parking systemstores and accesses data from remote locations with the help of the cloud these factors give raise to cloud of things (COT). The nodes could be monitored and controlled from any location the system that we propose provides information regarding the availability of the parking slots with the help of the mobile application the users fromthe remote location can book the parking slots.
- An algorithm is used to increase efficiency of cloud-based parkingsystem and network architecture technology is used. This algorithm is used to find the lowest cost parking space.
  Considering the number of parking space available and also considering the distance of the parking space from the user. The user can directly access the cloud-based server and find the information on the parking space. The user can also install an application in their mobile phones to access this information. With the help of this algorithm, waiting time of the user to find a parkingspace can be minimised. Security aspects are not included in this paper.
- A wireless sensor node along with smart phone application is beingused to find the parking space.Since, wireless technology is used here the system has high accuracy and efficiency. In this system, onboard units are used to communicate with other vehicles. The user parks his vehicle in any one of the several bays available a mechanical lift lifts the vehicle out.ticket key and id are given to the user and it is only known to the user which is used to retrieve the vehicle. The user need not carry any paper ticket since anRfid card is given to the user. The technology used here is economical.Security features must be improved to protect the user’s privacy.

- The author of smart parking system the survey has divided detectorsystem and vehicle sensors into two math categories as intrusive sensors and non - intrusive sensors. Intrusive sensors are installedin holes on the road surface by tunnelling under the road.
  Nonintrusive sensors do not affect the surface of the road and it canbe easily installed and maintained. Smart parking system helps us to resolve the grounding problems of the traffic congestion and it also reduces the emission from a car.
- A paper proposes efficient way to unfold the issue of parking availability in the real time scenario and to reduce the time consumption. In this, the data is sent locally with devices which filters the data. This signal is transmitted over the cloud for the process as well as for evaluation which uses machine learning algorithms. This paper uses mobile phone application that connectsthe user with the real time traffic status via Google API. Thus
  ,avoiding traffic congestion. This paper does not provide thereservation facility for the car parking.
- Smart parking using IoT technology helps to designs and developsa real smart parking system which provides information for vacant spaces and also helps the user to locate the nearest availability.
  This paper uses a computer vision to detect vehicle number plate inorder to enhance the security. The user can pay for the parking space prior to the entry of the car through mobile payment. Thus, insuring the reservation of the parking. The user is notified about the parking location, number of slots available and all other relevant information. The paper uses efficient algorithms and techniques for extracting license plate text. An algorithm operates on the ultrasonic sensor detection of the vehicle entering into the parking slot and calculates the minimum cost for the user.
- Smart parking system based on reservation allows the reservation of a vacant space which involves smart parking system based onreservation (SPSR).This consists of host parking database

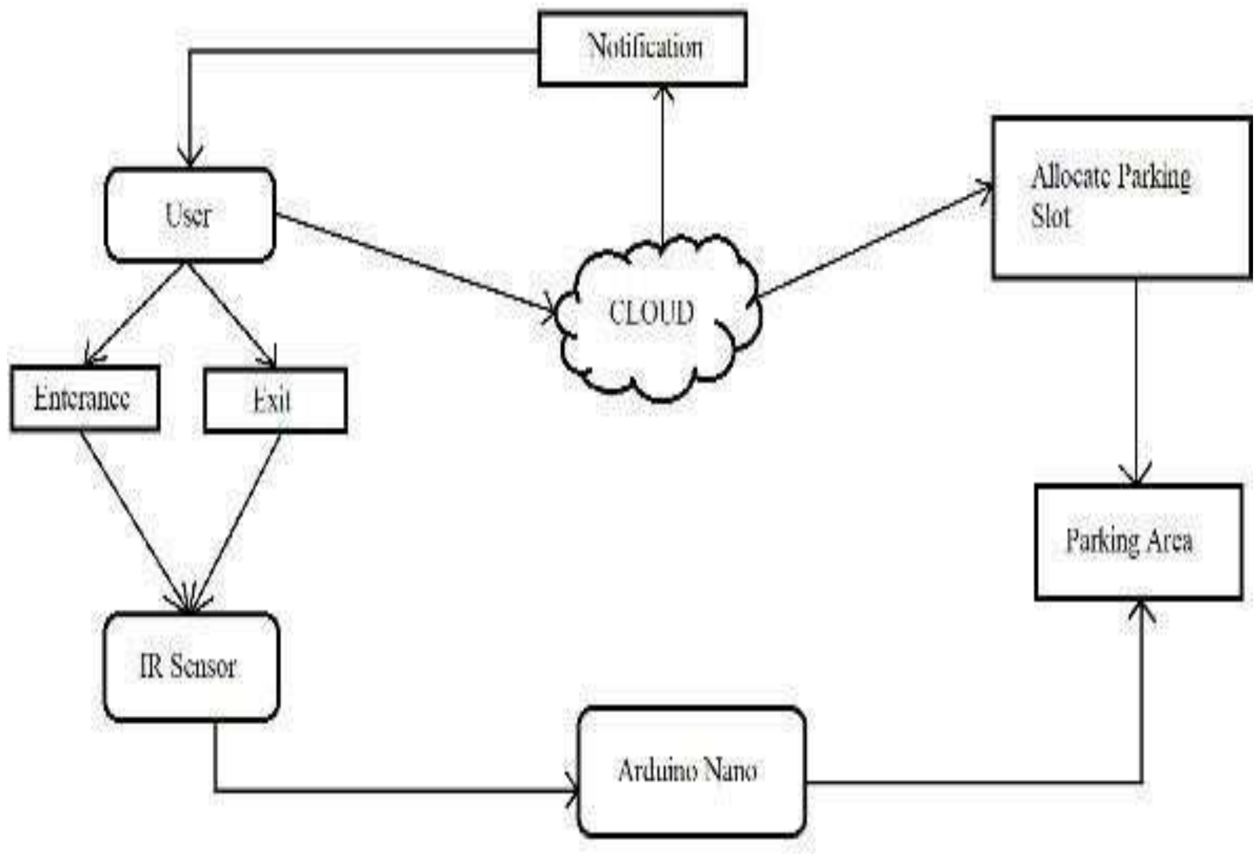
management which collects and stores data about the driver’s identity and parking location. When the parking reservation time isabout to expire a notification will be sent to the user through the web service that has been provided to the user by the admin. The main drawback is that some other user can occupy a reserved parking space to avoid this QR scanners are used to identify the user.

- It helps us to propose a way in which the user can reserve his parking space by mentioning the destination and the vehicle type with the help of mobile applications .The booking details will be stored in the cloud which finds the shortest path from the user to the parking space , the location of the user is updated regularly in the cloud with the help of GPS . When the user reaches the car parking the Rfid is scanned and the user is allowed into the parkingspace.The billing is done by the cloud server. The main disadvantage is that the car parking space must be registered in the smart parking system for the user to use it.
- This paper describes the implementation of wireless sensor networks (WSN) used in a car parking system with the help of a server which is using xbeezigbee. The car parking system can detect the car which is parked in the parking slot. The aim of thisproject is to make it cost effective and user friendly. Car parking system helps the user to sustain the data with 90% of accuracy.
- Smart car parking system provides a comprehensive parking solution for the user as well as admin of the parking area. It provides the feature for a reserved parking slot and identify reserved user. In this, user can navigate to the nearest parking areadepending upon the size of the vehicle. The user can reserve parking slot based on hourly, daily, weekly or monthly basis. An algorithm is designed to identify the nearest parking according to the size. The mobile application provided to the user is used to reserve and pay-as-you go service.

SYSTEM ARCHITECTURE

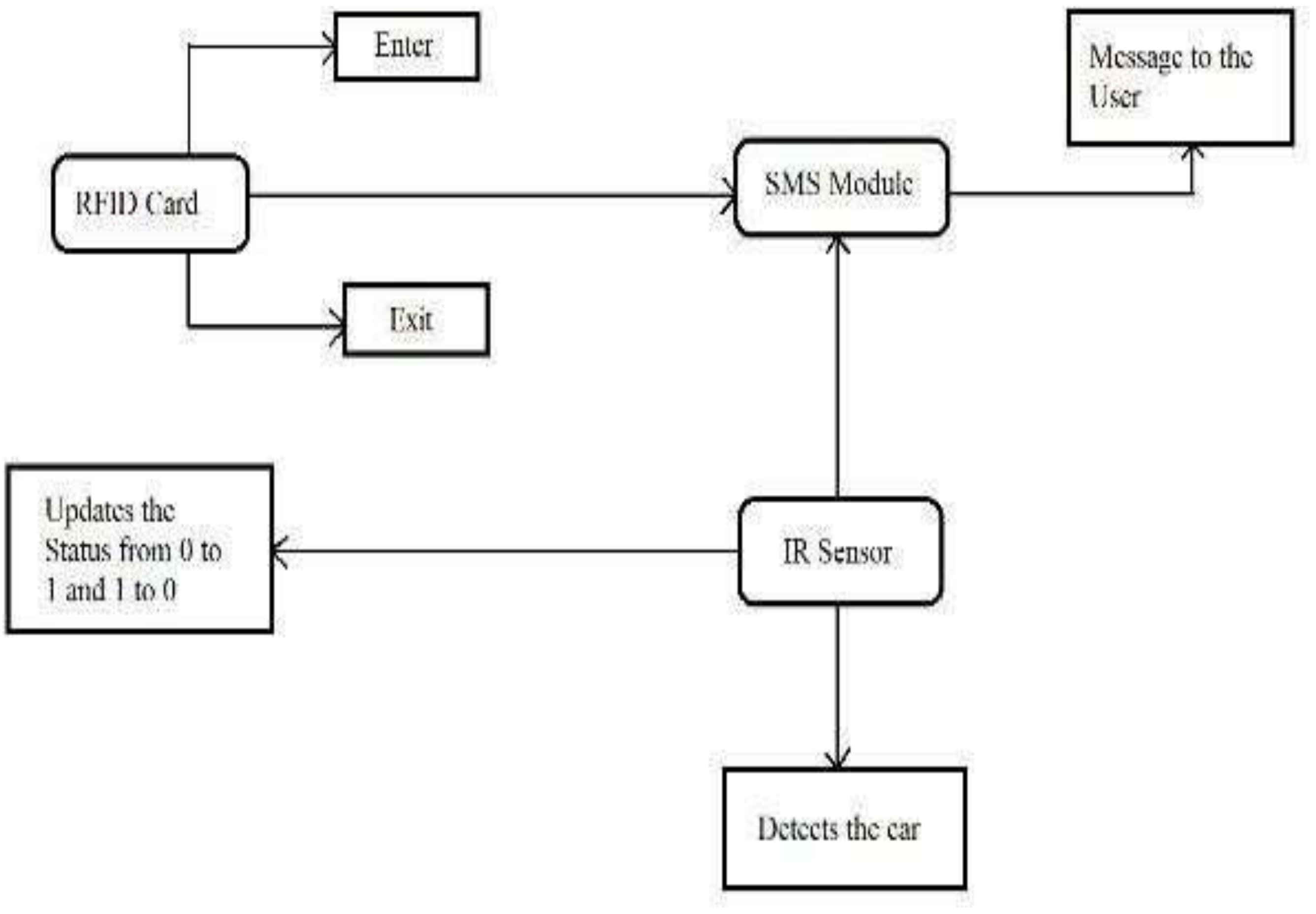
A. Proposed System

It consists of three sections: first section is the parking area which includes Arduino devices along with IR sensor. The user interacts with the parking area with the help of these devices. The user cannot enter the parking area without the help of RFID card. The second section contains the cloud-based web services which acts a mediator between the user and parking area. The cloud is updated depending upon the availability of the parking area. The admin administers the cloud services and it can also be viewed by the user for checking the availability. The third section is the user side. The user gets notificationon the basis of the availability via SMS through GSM module.



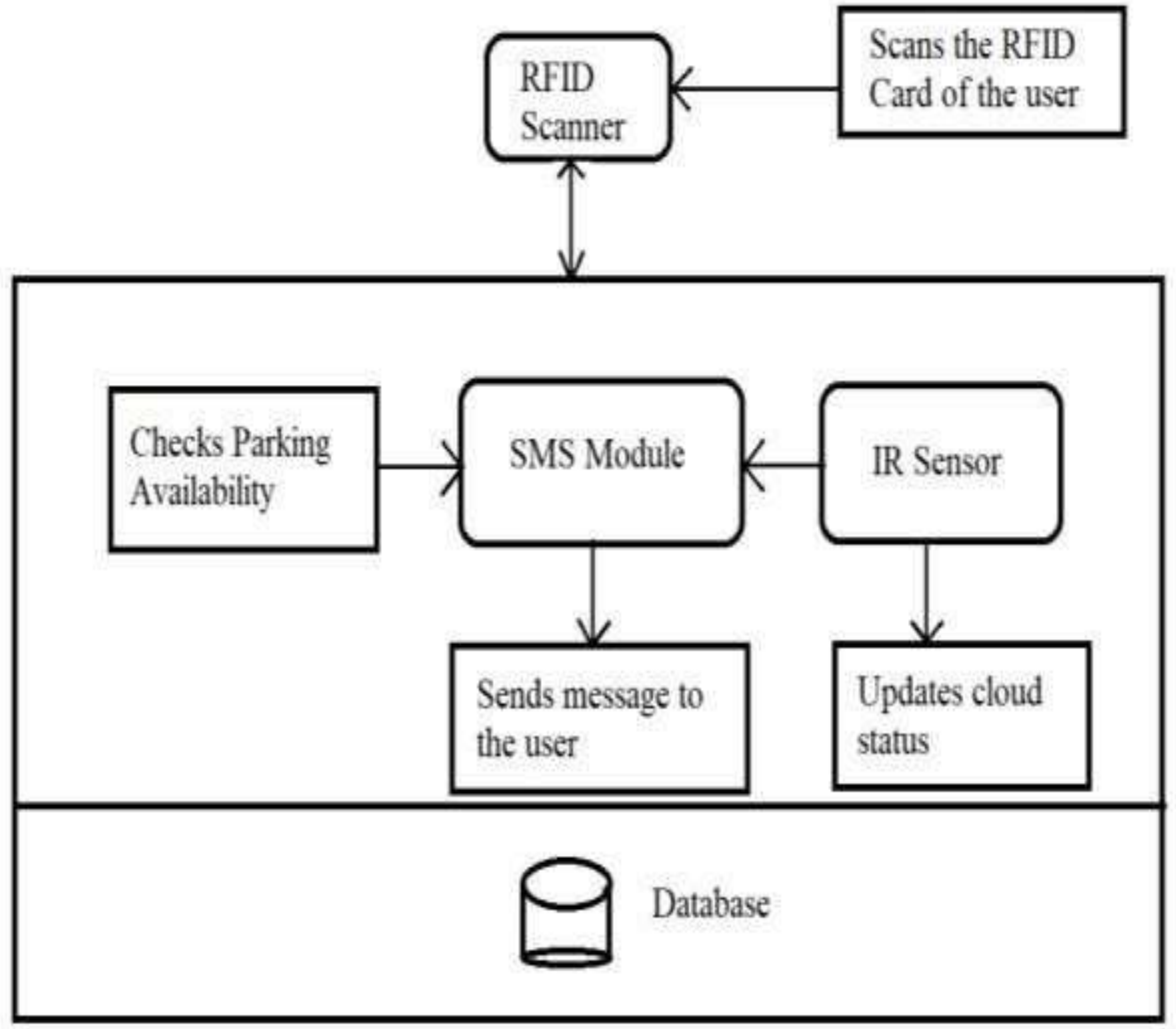
B. Hardware

The three main hardware components used are GMS module, RFID card, IR sensors. A user is allowed inside a parking space only if the user has a RFID card. RFID card contains the information of the registered user. As the car enters the parking slot, reader module scans the registered user's RFID tag. The data is sent to the ardunio for checking the availability of the car parking and simultaneously, the useris notified through SMS about the status of the parking area. The GSM module sends the message according to the availability. IR sensor sendsthe signals according to the presence of the vehicle.



C. Software

The cloud server acts as a mediator between the modules. The cloud server is connected to the Wi-Fi module. The user receives messagesthrough the SMS module while the car enters and exits the parking area using RFID card. The messages sent by the SMS moduleare managed by the cloud. As soon as the IR sensor detects the car, the status of the cloud will be updated from 0 to 1 and when the car leavesthe parking area the status of the car will be updated from 0 to 1.


RFID Scanner
Scans the RFID Card of the user
Checks Parking Availability
SMS Module
IR Sensor
Sends message to the user
Updates cloud status
Database

DETAILS OF THE MODULE A. GSM Module

The GSM module is a circuit which is used to setup communication between mobile phones and microcontroller. It is used to send SMS, MMS and voice messages through mobile network. GPRS extension inGSM allows high data transmission. GSM uses time division multiple access approach for transmission.

C. RFID Card

RFID tags are made up of integrated circuit (IC), an antenna, and a substrate. It is an identification badge or credit card that transfers its contents about an object to the reader module. RFID tag transfers dataabout an object through radio waves.When RFID tags are attached to devices they can also be used for tracking.

D. READER Module

This module is a device which scans and gathers the information from the RFID Card. This card can be used to track objects. As the car entersthe parking area, the user scans the RFID card and all the information stored in card is transferred to the admin through this module.

E. Servo Motor

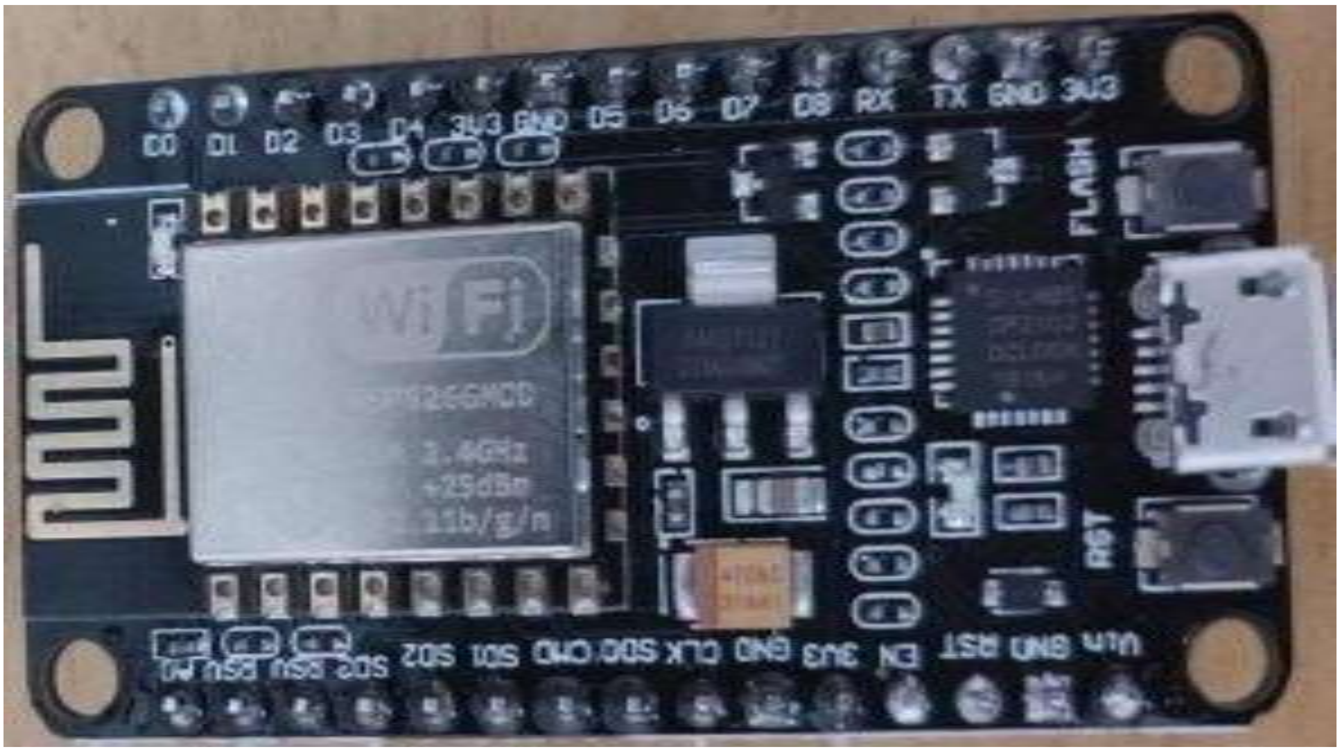
It is a rotator device that allows the control of angular as well as linear motion. A servo motor is used for the opening and closing of thegate.Servo drivetransmits electrical signals to the servo motor for producing motion.

F. Arduino Nano

It is a compact board which can be used in various devices and variousfield. It has overall 22 input/output pins out of which 14 pins are digitalpins. It has a flash memory of about 32 kb. These pins can control the operations of digital pins as well as analogy pins. This module is a breadboard-friendly board which can be easily used anywhere.

G. WIFI Module

It is used to send data from embedded system to the internet using URLby HTTP POST method using TCP/IP protocol. It is developed by espressif systems. It is a 32 bit microcontroller with 80kb user data. It contains 16 gpio pins.



Source Code:

```
import colorama

from termcolor import colored

options_message = """
Choose:
1. To park a vehicle
2. To remove a vehicle from parking
```

```
3. Show parking layout
4. Exit
"""



class Vehicle:


    def _init_(self, v_type, v_number):
        self.v_type = v_type
        self.v_number = v_number
        self.vehicle_types = {1: 'c', 2: 'b', 3: 't'}


    def _str_(self):
        return self.vehicle_types[self.v_type]



class Slot:


    def _init_(self):
        self.vehicle = None


    @property
    def is_empty(self):
        return self.vehicle is None
```

```
class Parking:


    def _init_(self, rows, columns):
        self.rows = rows
        self.columns = columns
        self.slots = self._get_slots(rows, columns)


    def start(self):
        while True:
            try:
                print(options_message)


                option = input("Enter your choice: ")


                if option == '1':
                    self._park_vehicle()


                if option == '2':
                    self._remove_vehicle()


                if option == '3':
                    self.show_layout()
                if option == '4':
                    break

            except ValueError as e:
                print(colored(f"An error occurred: {e}. Try again.", "red"))

        print(colored("Thanks for using our parking assistance system", "green"))

    def _park_vehicle(self):
        vehicle_type = self._get_safe_int("Available vehicle types: 1. Car\t2. Bike\t3.
Truck.\nEnter your choice: ")

        if vehicle_type not in [1, 2, 3]:
            raise ValueError("Invalid vehicle type specified")

        vehicle_number = input("Enter vehicle name plate: ")
        if not vehicle_number:
            raise ValueError("Vehicle name plate cannot be empty.")
        vehicle = Vehicle(vehicle_type, vehicle_number)

        print('\n')
        print(colored(f"Slots available: {self._get_slot_count()}\n", "yellow"))
        self.show_layout()
        print('\n')
        col = self._get_safe_int("Enter the column where you want to park the vehicle: ")
        if col <= 0 or col > self.columns:
            raise ValueError("Invalid row or column number specified")

        row = self._get_safe_int("Enter the row where you want to park the vehicle: ")
        if row <= 0 or row > self.rows:
            raise ValueError("Invalid row number specified")

        slot = self.slots[row-1][col-1]
        if not slot.is_empty:
            raise ValueError("Slot is not empty. Please choose an empty slot.")

        slot.vehicle = vehicle

    def _remove_vehicle(self):
        vehicle_number = input("Enter the vehicle number that needs to be removed from
parking slot: ")
        if not vehicle_number:
            raise ValueError("Vehicle number is required.")

        for row in self.slots:
            for slot in row:
                if slot.vehicle and slot.vehicle.v_number.lower() == vehicle_number.lower():
                    vehicle: Vehicle = slot.vehicle
                    slot.vehicle = None
                    print(colored(f"Vehicle with number '{vehicle.v_number}' removed from
parking", "green"))
                    return
        else:
            raise ValueError("Vehicle not found.")

    def show_layout(self):
        col_info = [f'<{col}>' for col in range(1, self.columns + 1)]
        print(colored(f"|{''.join(col_info)}|columns", "yellow"))

        self._print_border(text="rows")

        for i, row in enumerate(self.slots, 1):
            string_to_printed = "|"
            for j, col in enumerate(row, 1):
                string_to_printed += colored(f"[{col.vehicle if col.vehicle else ' '}]",
                                             "red" if col.vehicle else "green")
            string_to_printed += colored(f"|<{i}>", "cyan")
            print(string_to_printed)

        self._print_border()

    def _print_border(self, text=""):
        print(colored(f"|{'-' * self.columns * 3}|{colored(text, 'cyan')}", "blue"))
```

```
def _get_slot_count(self):
    count = 0
    for row in self.slots:
        for slot in row:
            if slot.is_empty:
                count += 1
    return count

@staticmethod
def _get_slots(rows, columns):
    slots = []
    for row in range(0, rows):
        col_slot = []
        for col in range(0, columns):
            col_slot.append(Slot())
        slots.append(col_slot)
    return slots

@staticmethod
def _get_safe_int(message):
    try:
        val = int(input(message))
        return val
    except ValueError:
            raise ValueError("Value should be an integer only")



def main():
    try:
        print(colored("Welcome to the parking assistance system.", "green"))
        print(colored("First let's setup the parking system", "yellow"))
        rows = int(input("Enter the number of rows: "))
        columns = int(input("Enter the number of columns: "))

        print("Initializing parking")
        parking = Parking(rows, columns)
        parking.start()

    except ValueError:
        print("Rows and columns should be integers only.")

    except Exception as e:
print(colored(f"An error occurred: {e}", "red"))



if _name_ == '_main_':
    colorama.init() # To enable color visible in command prompt
    main()
```

ArduinoProperties, Schematic Diagram, and Power

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, anICSP header, and a reset button.

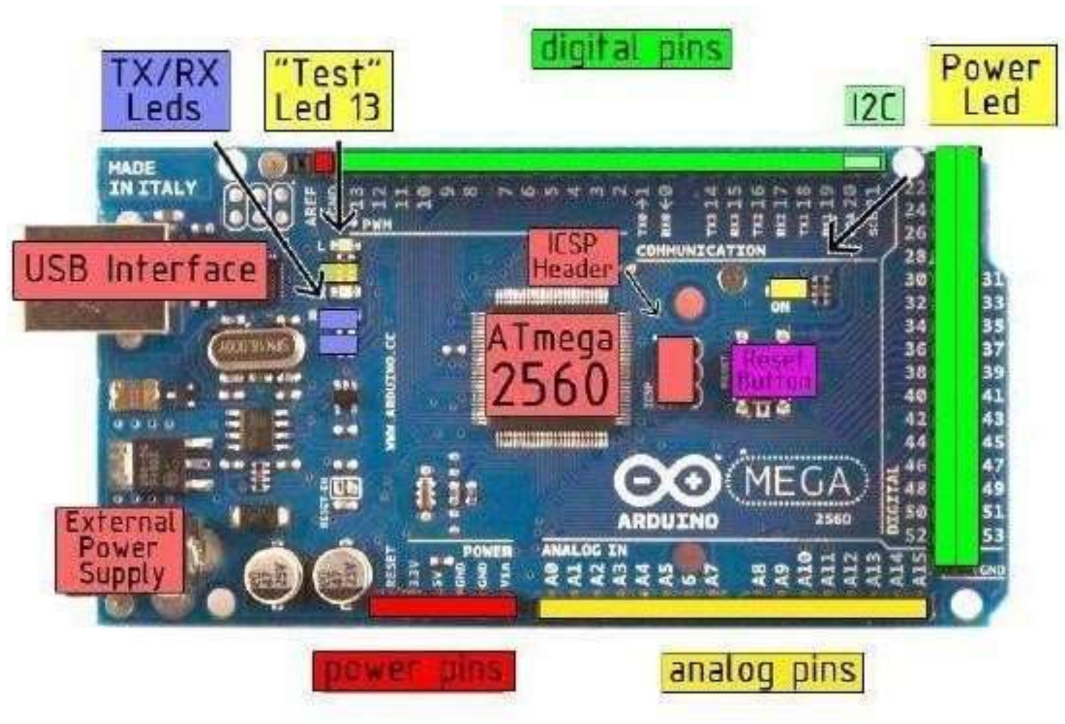
Specifications and Features of Arduino Mega 2560

Some of the detailed technical specification is mentioned below:- •
Operating voltage of 5 V.

- 16 MHz clock speed.
- The number of digital I/O's pins is 54.
- The number of analog input pins is 16.
- 4 hardware serial ports.
- Flash Memory: 32 KB
- SRAM: 2 KB
- EPROM: 1 KB
- Clock Speed: 16 MHZ

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. There is an important feature of Arduino which is Software Serial library that allows for serial communication on any of the Mega‘s digital pins. TheArduino 2560 has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. Thesefeatures as illustrated in figure 2.1 next pageare:

- Reset button
- Digital I/O
- Power pins
- Analog inputs
- Voltage regulator
- In-circuit serial programming
- FTDI USB chip and USB jack
- Power jack
- ICSP header



The Arduino can be used to develop stand-alone interactive objects or itcan be connected to a computer to retrieve or send data to the Arduino and then act on that data. The Arduino is an amazing device. Users can us to make many things from interactive works of art to robots. With a little enthusiasm to learn how to program the Arduino and make it interact with other components a well as a bit of imagination.Users can build many things.TheArduino can be connected to LED Matrix displays, RFID readers, buttons, switches,motors, temperature sensors, pressure sensors, distance sensors, webcams, printers, GPS receivers, and Ethernet modules. The Arduino board is made of an Atmel AVR Microprocessor, a crystal or oscillator (basically a crude clock that sends time pulses to the microcontroller to enable it to operate at the correct speed) and a 5-volt linear regulator. USB connector is used to connect to a PC or Mac to upload or retrieve data. The board exposes the microcontroller I/O (Input/Output) pins to enable you to connect those pins to other circuits or to sensors, etc.

DESIGN OF THE SYSTEM:

Network Time protocol :-

The Network Time Protocol is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. We have used NTP for fetching time from the NTP server so that we can show the start time and end time for the user when he parks or unparks his vehicle making information real-time.

Blynk app:-

Blynk app is a hardware-agnostic IoT platform with white label mobile apps, private cloud ,device management, data analytics and machine learning .On using the blynk app we tried to pop notification to every possible event that is occurring in the parking zone .

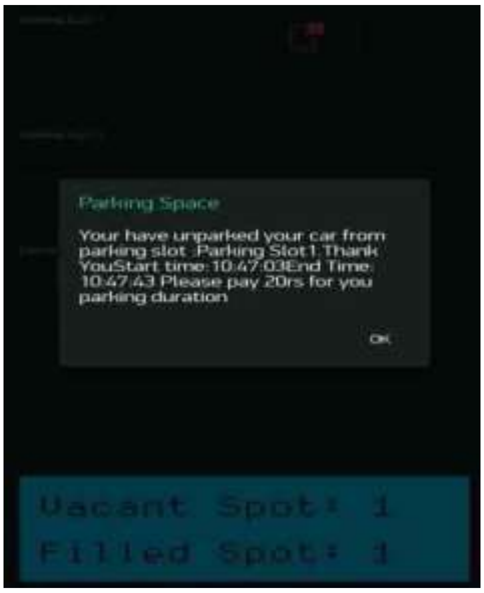
Used a serial algorithm to display the slot number to the user who is going to park his vehicle .For example we display the empty slot number in a serial manner which gets filled , if the slot 1 is filled and when an another vehicle turns up we display slot 2 and further like these for all other vehicles , and if any vehicle leaves the slot number then we display the earlies slot number , not making the user to travel long if an initial spot is vacant .

PARKING A VEHICLE:

Once when the user enters the parking detect sensor he would receive a parking slot number on his mobile application which he is supposed to park his vehicle. Upon parking the vehicle in the respective slot and IR sensor successfully detecting the vehicle it would show a notification on the app the start time of the vehicle and the slot number in which the vehicle is parked and it would be similarly updated on the 16*2 display.

UN PARKING AVEHICLE:

Unparking your vehicle from the parking slot would pop a notification on the application app stating the start time and the end time the user has parked his vehicle in the parking slot and an small amount which the user needs to pay when he leaves the parking zone which is fixed for any duration .

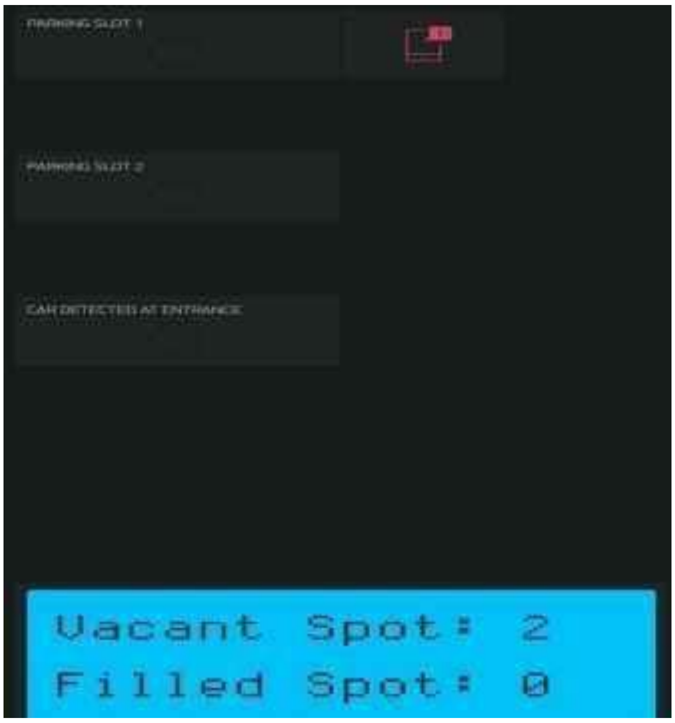


Parking Space

Your have unparked your car from parking slot :Parking Slot1 Thank YouStart time:10:47:03End Time: 10:47:43 Please pay 20rs for you parking duration

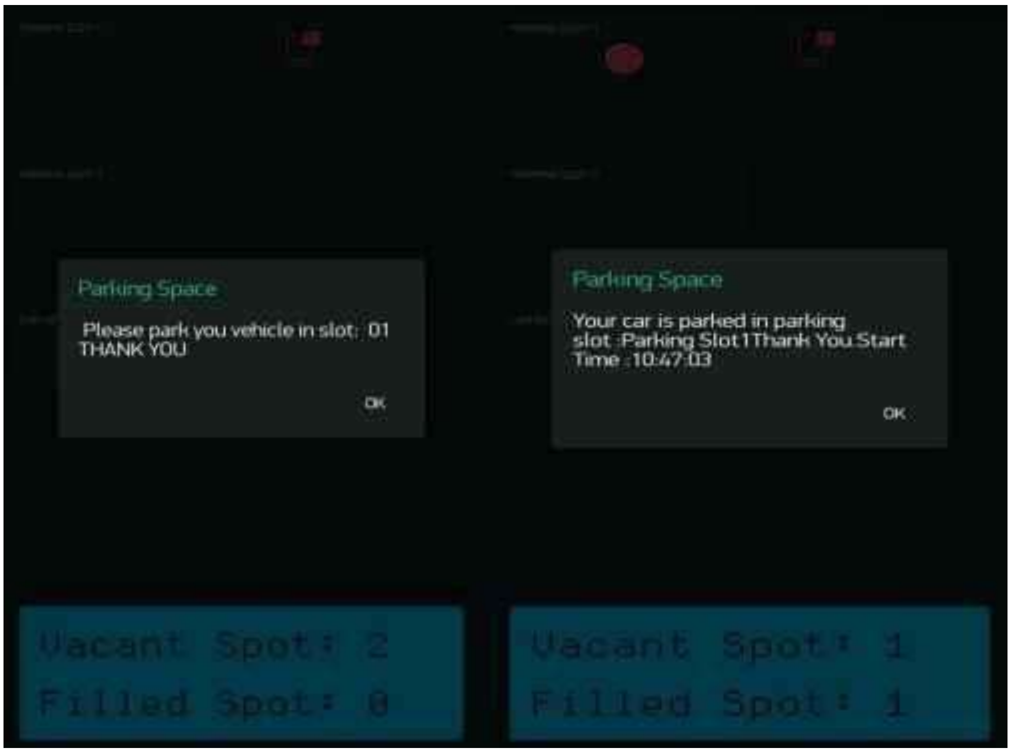
TESTING:

This case shows that all the parking slots are empty and therefore, the system will allow a car to enter into the parking zone . The 16*2 LCD will display the number of vacant spot and filled spot and similarly it would be displayed on the application.



```
Vacant  Spot:  2
Filled  Spot:  0
```

This following case focuses on showing a slot number when the user is near to the parking detect sensor. It shows a parking slot number where the user should park his vehicle and upon parking it shows the start time of his parking.



Please park you vehicle in slot: 01
THANK YOU

Your car is parked in parking slot :Parking Slot1Thank You.Start Time :10:47:03

```
Vacant  Spot:  2
Filled  Spot:  0
```

```
Vacant  Spot:  1
Filled  Spot:  1
```

PROGRAM:

Source Code:

```
import colorama


from termcolor import colored


options_message = """
Choose:
1. To park a vehicle
2. To remove a vehicle from parking
3. Show parking layout
4. Exit """



class Vehicle:


    def _init_(self, v_type, v_number): self.v_type =
v_type                    self.v_number =v_number
                          self.vehicle_types =  {1: 'c', 2: 'b',
3: 't'}


    def _str_(self):
        return self.vehicle_types[self.v_type]



class Slot:
    def _init_(self):
self.vehicle = None


    @property           def
is_empty(self):
        return self.vehicle is None


class Parking:


    def _init_(self, rows, columns): self.rows =
rows                    self.columns =columns
                        self.slots =
self._get_slots(rows, columns)


    def start(self):
while True:
try:
                print(options_message)


                option = input("Enter your choice: ")if option
                 ==

'1':
                    self._park_vehicle()if
                 option ==


'2':
                    self._remove_vehicle()if
                 option ==



'3':
```

```
                    self.show_layout()if
                option == '4':
                    break


            except ValueError as e:
                print(colored(f"An error occurred: {e}. Try again.", "red"))


        print(colored("Thanks for using our parking assistance system", "green"))


        def _park_vehicle(self):
        vehicle_type = self._get_safe_int("Available vehicle types: 1. Car\t2. Bike\t3.
      Truck.\nEnter your choice: ")


        if vehicle_type not in [1, 2, 3]:
            raise ValueError("Invalid vehicle type specified")


        vehicle_number = input("Enter vehicle name plate: ")if not
vehicle_number:
            raise ValueError("Vehicle name plate cannot be empty.")vehicle =
Vehicle(vehicle_type, vehicle_number)


        print('\n')
        print(colored(f"Slots available: {self._get_slot_count()}\n","yellow"))
                        self.show_layout()                    print('\n')

        col = self._get_safe_int("Enter the column where you want to park the vehicle: ")if col <= 0 or col >
self.columns:
            raise ValueError("Invalid row or column number specified")
        row = self._get_safe_int("Enter the row where you want to park the vehicle: ")if row <= 0 or row >
self.rows:
            raise ValueError("Invalid row number specified")


        slot = self.slots[row-1][col-1]if not
slot.is_empty:
            raise ValueError("Slot is not empty. Please choose an empty slot.")


            slot.vehicle = vehicle


         def _remove_vehicle(self):
        vehicle_number = input("Enter the vehicle number that needs to be removed fromparking slot: ") if not
vehicle_number:
            raise ValueError("Vehicle number is required.")


        for row in self.slots:for
slot in row:
                if slot.vehicle and slot.vehicle.v_number.lower() ==
vehicle_number.lower():
                    vehicle: Vehicle = slot.vehicle
                    slot.vehicle = None
                    print(colored(f"Vehicle with number '{vehicle.v_number}' removed fromparking", "green"))
                    return
else:
            raise ValueError("Vehicle not found.")


    def show_layout(self):
        col_info = [f'<{col}>' for col in range(1, self.columns + 1)]
print(colored(f"|{''.join(col_info)}|columns", "yellow"))


        self._print_border(text="rows")


        for i, row in enumerate(self.slots, 1):
            string_to_printed = "|"
for j, col in enumerate(row, 1):
                string_to_printed += colored(f"[{col.vehicle if col.vehicle else ' '}]","red" if col.vehicle else
                                            "green")
string_to_printed += colored(f"|<{i}>", "cyan")
print(string_to_printed)


        self._print_border()


    def _print_border(self, text=""):
        print(colored(f"|{'-' * self.columns * 3}|{colored(text, 'cyan')}", "blue"))


    def _get_slot_count(self):count
        = 0                     for
row in self.slots:
for slot in row:
if slot.is_empty:
count += 1          return
count


    @staticmethod         def
_get_slots(rows, columns):
        slots = []              for row in
range(0, rows):                        col_slot =[]
                for col in range(0, columns):
col_slot.append(Slot())
slots.append(col_slot)           returnslots


    @staticmethod            def
_get_safe_int(message):try:
            val =
int(input(message)) return
val                         except
ValueError:

            raise ValueError("Value should be an integer only")




def main():
try:
        print(colored("Welcome to the parking assistance system.", "green"))
                        print(colored("First let's setup the parking system","yellow"))
                          rows = int(input("Enter the number of rows: ")) columns =
int(input("Enter the number of columns: "))


        print("Initializing parking") parking
= Parking(rows, columns)parking.start()


    except ValueError:
        print("Rows and columns should be integers only.")


    except Exception as e:
print(colored(f"An error occurred: {e}", "red"))




if _name_ == '_main_':
    colorama.init() # To enable color visible in command promptmain()
```