

PARKING MANAGEMENT SYSTEM USING IOT TECHNOLOGY

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electrical and Electronics Engineering

by

GUNA SEKAR J (Register No. 40140012)

HARIPRASATH S (Register No. 40140014)



**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119, TAMILNADU

APRIL-2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

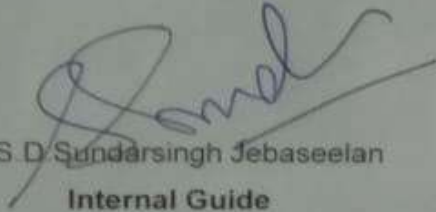
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

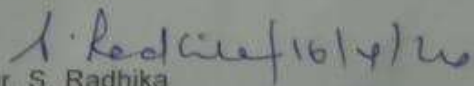
www.sathyabama.ac.in

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **GUNA SEKAR J** (Reg. No. 40140012) and **HARIPRASATH S** (Reg. No. 40140014) who carried out the project entitled "**PARKING MANAGEMENT SYSTEM USING IOT TECHNOLOGY**" under my supervision from November 2023 to April 2024.

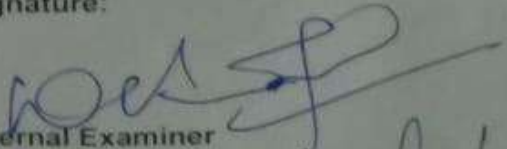

Dr. S. D. Sundarsingh Jebaseelan
Internal Guide


Dr. S. Radhika
Head of the Department

Submitted for Viva voce Examination held on 17/4/24

Name: H/H/2

Signature:


Internal Examiner


External Examiner

(Dr. J. Kalithuruman)

DECLARATION

We, GUNA SEKAR J (Reg.No.40140012) and HARIPRASATH S (Reg.No.40140014) hereby declare that the Project Report entitled "PARKING MANAGEMENT SYSTEM USING IOT TECHNOLOGY" done by us under the guidance of Dr.S.D.SUNDARSINGH JEBASEELAN is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electrical and Electronics Engineering.

1. J. Sekar

2. S. Hariprasath

SIGNATURE OF THE CANDIDATE

DATE: 17/4/24

PLACE: CHENNAI

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **Sathyabama Institute Of Science And Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey my thanks to **Dr. N.M. Nandhitha, M.E., Ph.D., Dean, School of Electrical and Electronics Engineering**, and **Dr. S. Radhika , Head of the Department, Dept. of Electrical and Electronics Engineering** for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to our Project Guide **Dr.S.D.SUNDARSINGH JEBASEELAN**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electrical and Electronics who were helpful in many ways for the completion of the project.

ABSTRACT

In recent trends, traffic has become a major problem in most cities. There are few parking spaces and few precautions for road safety. So, they are done in IoT field to solve traffic problems. This paper will present the basic concept of cloud-based parking system. The cloud will store and process data between device and mobile. Files stored in the cloud can be accessed from anywhere using a network connection. Using IoT, the latest technology for smart cities, we can book parking anywhere. The microcontroller sends the status of all IR sensors to the LCD modules which show the available slots and telnet the IOT server. The microcontroller reads the data display through LCD This will reduce the waiting time in the parking lot and this system will allow us to use the parking lot more efficiently. The main objective of this project was to design and develop intelligent parking systems using software solutions connected by sensors through IOT technology, which provide solutions to reduce traffic congestion and associated anxiety the causes of the. As a proof of this proposed approach, a prototype will be developed using Android and Python to demonstrate the proposed framework. In addition, a Python application will be developed using PYTHON IDLE to facilitate users to register, edit, check parking status, etc. The application runs on a smart slot recommendation device and connects smart features such as entry/exit to the system.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	TABLE OF CONTENT	vi
	LIST OF FIGURES	viii
	LIST OF TABLES	x
	LIST OF ABBREVIATION	xi
1	INTRODUCTION	
	1.1 General Introduction	1
	1.2 Literature Survey	3
	1.3 Scope Of The Project	6
	1.4 Advantages Of Proposed Technique	7
	1.5 Applications Of Proposed Technique	7
2	PROJECT DESCRIPTION	
	2.1 Block Diagram Of Parking System	8
	2.2 Concept Explanation	9
	2.3 Circuit Diagram	13
	2.4 Components	14
	2.4.1 Hardware Components	14
	2.4.2 Software Components	16
	2.5 Modules Name	17
	2.6 Flow Chart	17
	2.6.1 Flow Chart for Existing system	17
	2.6.2 Flow chart for Slot Reversation	18

	2.6.3 Flow chart for Vehicle Entry	19
	Flow	
	2.6.3 Flow chart for Vehicle Entry	19
	Flow	
	2.6.4 Flow chart for Vehicle Search	19
	Space Process	
	2.6.5 Flow chart for Vehicle Exit Flow	20
3	METHODOLOGY	
	3.1 Existing Methodology	22
	3.2 Proposed System	23
	3.2.1 Implementation	24
	3.2.2 Managing Locations	25
	3.2.3 Application Framework	25
	3.2.4 Libraries	26
	3.2.5 Android runtime	27
	3.2.6 System Requirements	28
4	SIMULATION THEORY	
	4.1 Proteus Description	46
	4.2 Simulation	47
5	SIMULATION CIRCUIT AND RESULT	
	5.1 Simulation Circuit	49
	5.2 Application Results	51
	5.3 Hardware Results	55
6	CONCLUSION & FUTURE SCOPE	
	6.1 Conclusion	58
	6.2 Future Scope	59
	REFERENCE	60

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Management of parking system	2
2.1	Block diagram of parking system	8
2.2	Hardware Circuit Diagram of Parking system	9
2.3	Circuit diagram for Proposed parking system	13
2.4	Arduino UNO	14
2.5	IR Sensor	14
2.6	LCD Display	15
2.7	Servo Motors	15
2.8	LED Lights	16
2.9	Arduino IDE	16
2.10	Flow Chart for Existing System	18
2.11	Flowchart for Slot Reservation	18
2.12	Flowchart for Vehicle Entry Flow	19
2.13	Flowchart for Vehicle Search space process	20
2.14	<i>Flowchart for Vehicle Exit Flow</i>	20
3.1	Board showing available parking parking slot	22

FIGURE NO	TITLE	PAGE NO
5.1	Simulation Circuit Diagram of parking system	50
5.2	Admin add parking slot page	51
5.3	Admin home page	51
5.4	Slot booking page	52
5.5	Login page	52
5.6	Welcome page	53
5.7	Manage your locations	53
5.8	Registration page for new user	54
5.9	Text view	55
5.10	Photocopy of Hardware circuit	56
5.11	Display output	57

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
3.1	Hardware Requirement For Android SDK	30
3.2	Python Environment	37

LIST OF ABBREVIATION

SDK	Software Development Kit
USB	Universal Serial Bus
IOT	Internet of things
API	Application Programming Interface
AVD	Android Virtual Device
OS	Operating system
HTML	Hypertext Markup Language
ADB	Android Debug Bridge
DDMS	Dalvik Debug Monitor Service
VGA	Video Graphics Array
GSM	Global System for Mobile Communications
EDGE	Enhanced Data rates for GSM Evolution
IDEN	Integrated Digital Enhanced Network
CDMA	Code Division Multiple Access
EVDO	Evolution-Data Optimized
UMTS	Universal Mobile Telecommunications System
WiMAX	Worldwide Interoperability for Microwave Access
C2DM	Android Cloud to Device Messaging
SMS	Short Message Service
MMS	Multimedia Messaging Service
CPU	Central Processing Unit
JPEG	Joint Photographic Experts Group
PNG	Portable Network GraphicsGIF

GIF	The Graphics Interchange Format
HE-AAC	High-Efficiency Advanced Audio Coding
AMR	Antimicrobial resistance
MIDI	Musical Instrument Digital Interface
ISMA	Inhibit Sense Multiple Access
RTMP	Real Time Messaging Protoco
HTTP	Hypertext Transfer Protocol
RTSP	Real Time Streaming Protocol
HID	Human Interface Devices
HDD	Hard Disk Drive
NTFS	New Technology File System

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

Smart parking sensing technology is leading the way in the delivery of proven, fully integrated, end-to-end solutions, resulting in a transformation of the overall parking experience. Smart park systems in general are designed with capable features change and fully integrate parking, navigation, payment analytics and other services along with a variety of options. This makes the overall smart park solution truly the sum of its parts, and leads the industry in terms of technology and parking. As the number of people in urban areas has increased, the transportation use is common. In most cities, especially throughout the push hours, finding a parking space is annoying for drivers. Undisciplined parking can lead to car injuries. Smart Cloud is a Smart Parking real-time, global, intelligent IoT service system that enables sophisticated and convenient services through an open web network. So cars must provide attractive parking spaces, and let alone many places to help the user position. Essentially, parking systems are the first acceptable and fastest growing response for any of the tricky city. Currently, most of the old parking lots do not have a science program. Most of those square measures are maintained manually and are somewhat inefficient | The requirements of each user should be I. user-friendly ii. Can be more expensive iii. They need to be given additional protection. The idea behind our humanoid application-“validspot” is to help the user for on-line parking booking. In this application, the user will read multiple parking options together to see if the parking space is accessible or not. If the parking is accessible at the parking lot, the user will co-book a specific time, this option provides an additional factor to cancel the booking.

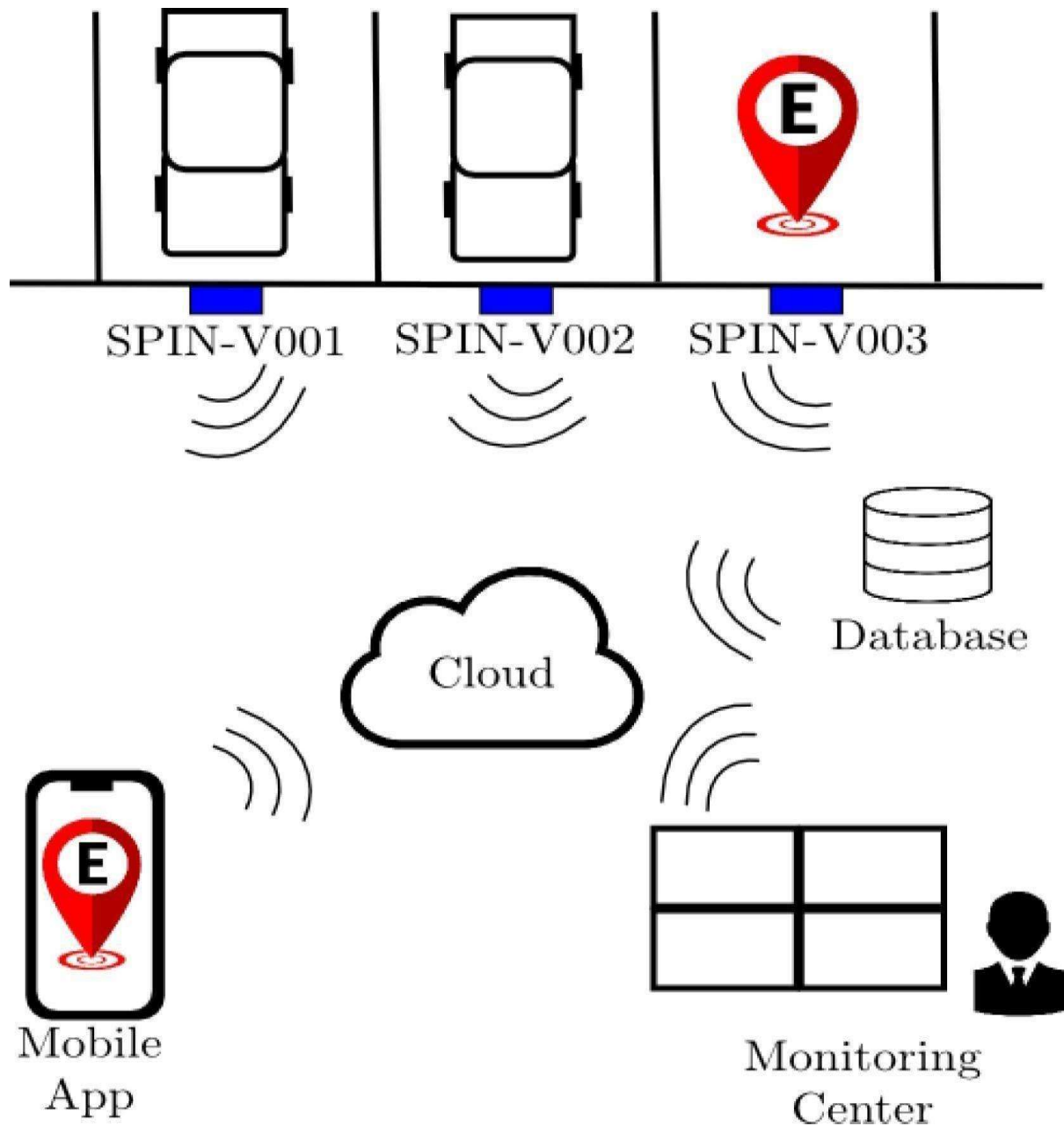


Fig 1.1. Management of Parking System

Together it uses open land for safe parking. Thus, it will solve the trailer traffic accident. Other expensive cameras and scanners cannot be used for verification at this time. In this system, the Registration Certificate (R.C) book is used for verification, thus collecting the excess value. Smart parking development means IoT- based System that

sends data about free parking spaces via web/mobile application. An IoT device with sensors and microcontrollers is located in every parking lot. This leads to a complete solution for integrating, managing and retrieving data from IoT devices, and means that we have been able to completely redefine and optimize how we look at how we use certain smart sensors and devices, we do work and we use it on. In this project... the main role cloud It will be because, the data of all the parking slots will be available in the cloud. Cloud computing is the provision of computer services such as servers, storage, databases, networking, software, analytics, intelligence, etc. Through the cloud, cloud computing provides an alternative to on-premises datacenters. With a live data center, we can manage everything, including buying and installing hardware, virtualization, installing operating systems, and other critical applications, configuring networks, configuring firewalls , and setting up data storage. We all know that one of the most common problems nowadays is overcrowded parking spaces. Traffic continues to exceed existing parking spaces, clogging the roads. Violence around residents, broken down vehicles due to lack of space, and charges for parking charges are some of the problems that arise. Smart parking development refers to policies that based on IoT that sends data about free parking spaces via web/mobile applications. An IoT device with sensors and microcontrollers is located in every parking lot.

1.2 LITERATURE SURVEY

Jung , H. G. , Choi , C. (1999). G. , Kim , D.K. S. , & Yoon , P. (1999). J. 2006.
 “System Configuration of Intelligent Parking Assistant Systems,” Proc. 13th World Congress. Intelligence. Transp. Syst. Services, London, UK, Oct). is hereby proposed a new light-band projection-based target positioning method for intelligent parking assist system, which economically free space for poorly light indoor -Provides a target -based positioning method Parking spaces without degrading rear view integrated into the body of the vehicle. The proposed method can be applied to existing background camera installations by adding a small, inexpensive display plane. Light band characteristics Light plane projectors are recognized by the difference between the on and off images. The three-dimensional information of the parking lot is reconstructed using the light-

band projection theory. Orientation normalization and depth mapping detect open areas and select an area close to your target. Experimental results show that the proposed method can correctly identify the target location, although in the absence of poor lighting and dark glare from nearby vehicles.

Kianpisheh, A., Mustafa, N., Limtrairut, P., and Keikhosrokiani, P. 2012. "Smart Parking System (SPS) Architecture Using Ultrasonic Detector," *International Journal of Software Engineering and Applications* (6:3), pp. 55-58. The increase in automobile production and world population requires more parking spaces and infrastructure. This paper proposes a new parking system called Smart Parking System (SPS) to help drivers identify empty spaces in parking lots in a short period of time. The new system uses ultrasonic (ultrasound) sensors to detect people sitting in a parking lot or inappropriate behavior in a parking lot. Different detection technologies are reviewed and compared to determine the best technologies for SPS development. Features of the SPS include identification of empty parking spaces, identification of poor parking spaces, display of available parking spaces, vacant parking spaces using specific features, pay zones and directions indicating various parking spaces (vacant, occupied, reserved and disabled) and LEDs. This paper also describes the use of SPS systems from entrance parking spaces to vacant parking space detection. The layout is designed for four car parks with 100 parking spaces per floor and five lanes. The system design describes the design features required such as the location of the sensors, the number of sensors and LEDs required for each layer, and the indoor and outdoor display panels.

Panayppan , R. , Trivedi , J.M. , Studer , A. , and Perrig , A. Age . "A VANET- based approach for parking space availability," *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, ACM 2007, pp. 1-13. 75-7 Parking information and parking location are two useful functions of VANET. In this paper, we propose a network model by which parking spaces and available parking spaces can be identified efficiently and without significant administrative support. Security vulnerabilities in the model are noted and appropriate security measures are proposed.

Pham , T. (1999). N. , Tsai , M.-F. , Nguyen , D.K. B. , Dow , C.-R. , & Deng , D.-J. 2015 is the 21st century focused on technology, communication and information processing. It offers high-quality papers in vibration, acoustic vibration, neuroscience, mathematical science, and communication. It includes engineering and construction engineering, computer science, biology, and biostatistics, among other related fields. It also encourages researchers to work in these areas by providing valuable insights. The papers included in this book were presented at the 1st International Conference on Engineering Vibration, Communication and Information Processing, India (ICoEVCI-2018).

Solik , P. , Marasovich , I. , Stefanizzi , M. (2010). L. , Patrono , L. , & Manetti , L. (1999). "RFID-Based Efficient Method for Parking Slot Car Detection," Software, Telecommunications and Computer Communications (SoftCOM),2015 23rd International Conference on, IEEE2015, pp. 1-12. 108. The literature review for this paper provides an in-depth analysis of Recent developments in sensing and communication technologies related to parking systems. In addition, this paper presents an academic, qualitative review and analysis of the literature. It includes in-depth study of selected topics and provides a step-by-step process for implementation. It reviews intelligent parking systems used for parking guidance and parking management and examines technical aspects and such systems published in academia in the past 15 years specifically.

Xu, J., Chen, G., and Xie, M. Year. "Vision-guided computerized parking for smart vehicle," Proceedings of the IEEE Intelligent Vehicles Symposium2000, pp. 725-730. This describes about how to efficaciously and successfully finding parking-slots using a imaginative and prescient- based device remains an unresolved difficulty. Even more severely, there's no publicly to be had categorised benchmark dataset for tuning and testing parking-slot detection algorithms. In this paper, we strive to fill the above-noted research gaps to a degree and our contributions are twofold. Firstly, to facilitate the observe of vision-based parking-slot detection, a big-scale parking-slot picture database is installed. This database comprises 8600 surround-view images accrued from regular indoor and outside parking sites. For every photograph on this database,

the marking-points and parking- slots are cautiously categorized. Such a database can function a benchmark to design and validate parking-slot detection algorithms. Secondly, a learning-based totally parking-slot detection method, particularly PSDL , is proposed. Using PSDL , given a surround-view image, the marking-factors will be detected first and then the valid parking-slots may be inferred. The efficacy and performance of PSDL had been corroborated on our database. It is anticipated that PSDL can serve as a baseline while the alternative researchers increase more state-of-the-art strategies.

Ye , H.-T. , Chen , B.-C. , and Wang , B.-X. 2016. "Urban parking integration system integrated with cloudcomputing technology and smart mobile devices," Eurasia Journal of Mathematics, Science and Technology Education (12:5), pp. 1-13. 1231-1 The current study used cloud computing technology and smart mobile devices together with a streaming server for parking lots to create a city parking integration system that includes a parking search system, a parking navigation system, a parking reservation service and a car retrieval service. With this configuration, users can easily locate parking spaces near their destination, and combine the connection with cloud server using various parking servers and planning centers with Cloud server computing power five closest parking locations with available parking space to the user smart mobile device s in which information is scanned parking space using RFID indoor positioning system can be possible such as the current location of users, so that an empty parking space in the parking lot is the location of moving vehicles on their... smart mobile device. Additionally, the traffic records stored can provide local governments with information for future inner-city parking.

1.3 SCOPE OF THE PROJECT

The parking arrangement uses IOT technologies, enabling the user to make better use of parking spaces, reducing the time spent searching for parking a free of charge. The IoT-based system is a centralized system that allows drivers to search and reserve parking remotely using their smart phones. It provides a convenient parking system for

motorists when they are looking to avoid potential traffic congestion. In a parking garage system, there is an automatic parking system every step of the experience is installed and digitally enhanced. Businesses no longer rely on servants to check in and out of vehicles, check locations, check occupants, and check in to withdraw money or process credit cards.

1.4 ADVANTAGES OF PROPOSED TECHNIQUE

- Real-time monitoring.
- Advanced practical experience.
- Effective use of space.
- reduced traffic congestion and carbon emissions.
- Cost savings.
- Enhanced security.
- predictive maintenance.
- Data-Driven Insights.

1.5 APPLICATIONS OF PROPOSED TECHNIQUE

- Urban parking policy.
- supermarkets and retail outlets.
- Airport parking lot.
- hospitals and health care facilities.
- Smart buildings and smart cities.
- Parking events.
- Transportation Center.

CHAPTER 2

PROJECT DESCRIPTION

2.1 BLOCK DIAGRAM OF PARKING SYSTEM

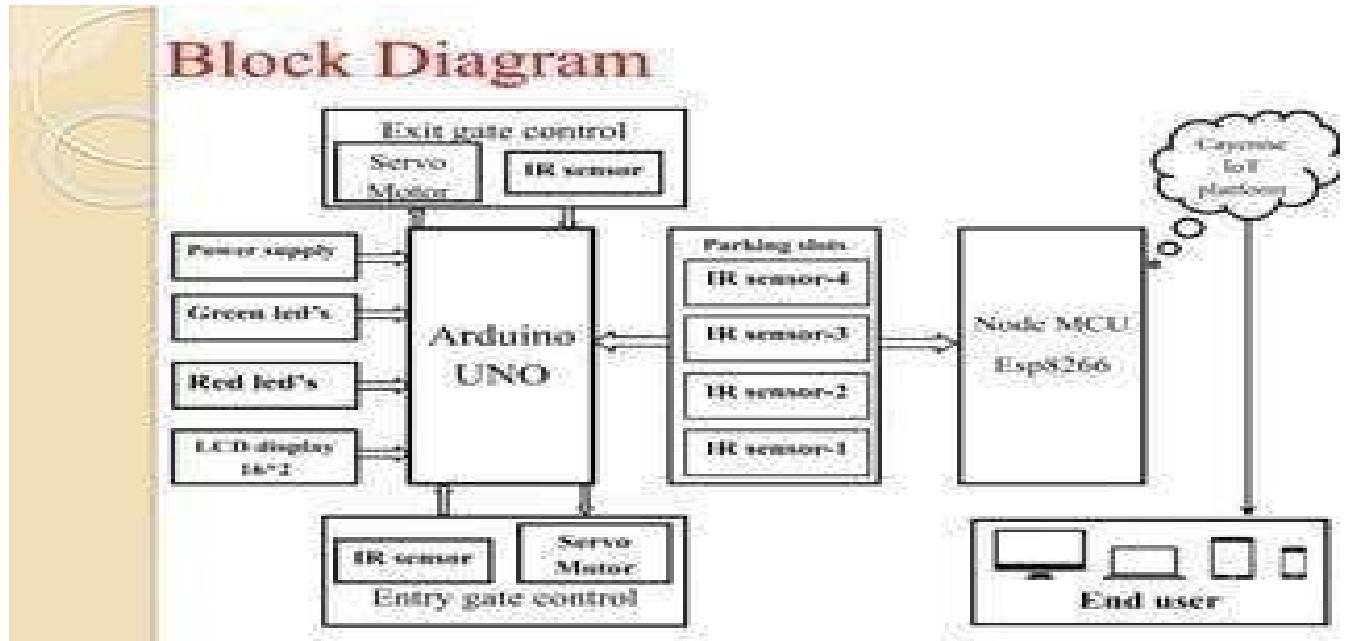


Fig 2.1 Block diagram of parking system

The above figure 2.1 shows the general block diagram of *parking system* respectively.

Entrance: This may be a device that controls the entrance to the parking lot. This can be a servo motor or other motor that opens and closes a door depending on certain conditions.

Servo motor: This is a type of motor that is capable of rotating at a certain angle and holding that position. This parking system may use servo motors to control the entry and exit doors.

Parking: This section represents the parking lot.

IR sensor-4: This is an infrared (IR) sensor that detects the presence of a vehicle. It will likely be used to detect the presence of a vehicle at an exit in this parking system.

Green led's: These are light-emitting diodes (LEDs) that emit green light. In this installation process, it may be used to confirm that the parking system is available for use.

Red led's: These are LEDs that emit red light. It is used in this installation process to indicate that the installation process may not be available for use.

Arduino UNO: It is a microcontroller board that can be programmed to perform various tasks. This parking system may be used to control the entrance and exit gates, and to process sensor data and display the information on the LCD display

IR sensor-3 and IR sensor-2: These are IR sensors that detect the presence of a vehicle. In this parking system, it is used to detect the presence of a vehicle at the entrance and in the parking lot.

LCD Display: This is a liquid glass that displays information to the user. It will likely be used to display information such as the number of parking spaces available in this parking system.

IR sensor-1612: This is an IR sensor that detects the presence of a vehicle. It will likely be used to detect the presence of a vehicle at the entrance of this parking system.

Node MCU Esp8266: This is a microcontroller board that can be programmed to perform various tasks. This installation system can be used to connect the parking system to the Internet and allow remote access to the system.

Exit: This section represents the end user of the parking system.

The block diagram shows how these elements combine and interact with each other to form a complete parking system. The entrance gate control and servomotor are connected to an Arduino UNO which processes sensor data and controls the gate based on certain conditions. An IR sensor and LCD display are also available around the Arduino UNO which processes the sensor data and displays the information on the

LCD display. The node MCU interfaces with the Esp8266 Arduino UNO and provides remote access to the parking system. The end user communicates with the parking system through Node MCU Esp8266.

2.2 CONCEPT EXPLANATION

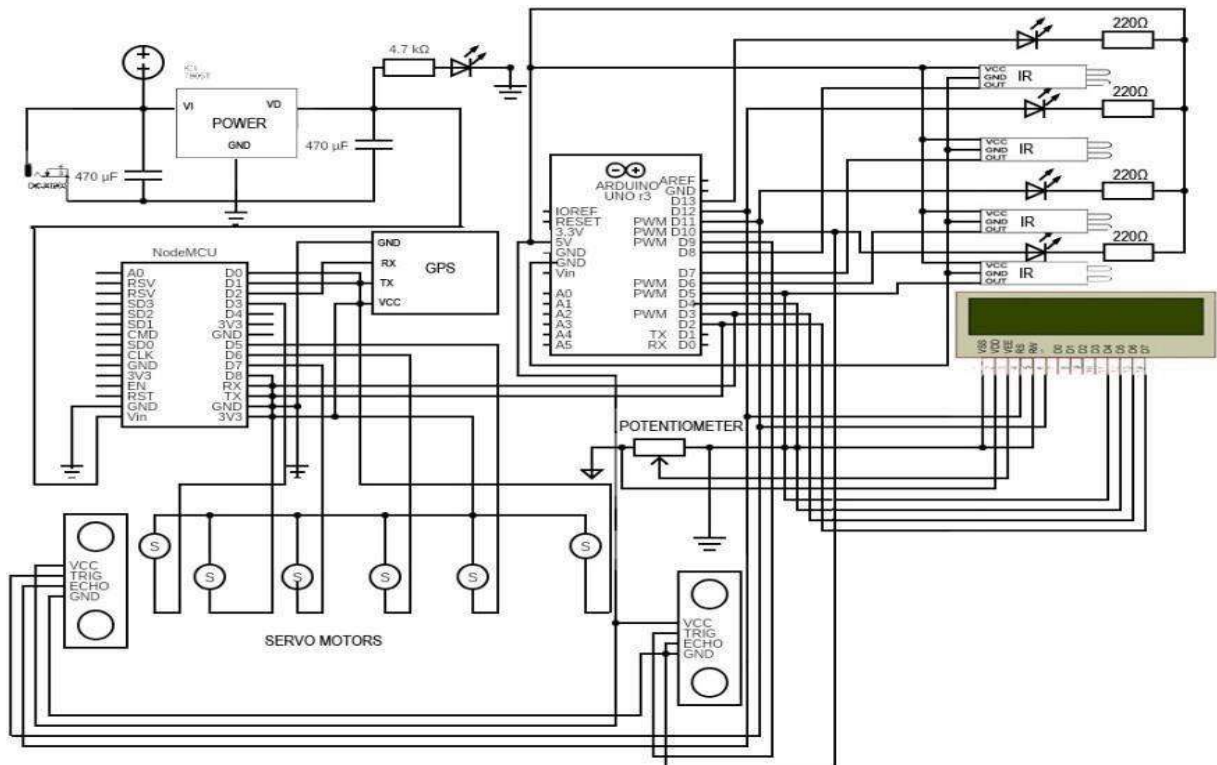


Fig 2.2 Hardware Circuit Diagram of parking system

The above figure 2.2 shows the general Hardware Circuit diagram of *parking system* respectively.

The hardware circuit diagram provided in the text describes the electrical connections and components of the parking system. Here is a breakdown of its features and functions.

470 μF: This is a capacitor used to smooth voltage changes in the power supply.

DCX: It is a voltage regulator which converts high voltage to low voltage.

7806T: This is a voltage regulator that converts the high voltage to a stable 6V output.

VI: This is the voltage input of the circuit.

Power: This is the primary power source for the circuit.

VD: This is the voltage output from the voltage regulator.

4.7 K Ω : This is the resistor that limits the current in the circuit.

2200: This is a capacitor to remove the power supply.

VCC: This is the output voltage of the voltage regulator.

GND: This is the ground connection of the circuit.

IR: This is an infrared sensor that detects the presence of a vehicle.

OUT: This is the output signal from the infrared sensor.

Arduino - This is a microcontroller board used to control the parking system.

UNO: This is a typical example of an Arduino microcontroller board.

D13: This is the digital input/output pin on the Arduino board.

IOREF: This is a pin on the Arduino board that provides a reference voltage.

RESET: This is a pin on the Arduino board that resets the microcontroller.

PWM: This is a pulse width modulation signal used to control a servo motor.

A1-A5: These are the analog input pins on the Arduino board.

RX: This is a pin on the Arduino board that receives serial data.

NodeMCU: This is a microcontroller board used to connect the parking system to the internet.

GPS: This is a module for location information.

SD1-SD3: These are the serial peripheral interface (SPI) pins on the Arduino board.

CMD: This is the command signal used to control the GPS module.

SDO: This is the sequential data output from the GPS module.

CLK: This is the same as the clock signal used to synchronize SPI connections.

POTENTIOMETER: It is a variable resistor used to adjust the sensitivity of the infrared sensor.

SERVO MOTORS: These are motors for controlling the entry and exit gates in the parking system.

TRIG: This is the pin on the Arduino board used to trigger the ultrasonic sensor.

ECHO: This is a pin on the Arduino board that receives an echo signal from the ultrasonic sensor.

The hardware circuit diagram shows how the various components of the parking system communicate and interact with each other. The power supplies are connected to voltage regulators, which provide a stable voltage output to components such as the Arduino board. The infrared and ultrasonic sensors are connected to the Arduino board, which processes sensor data and controls the servomotors, etc. The GPS module is connected to the NodeMCU board, which provides an internet connection to the parking system. A potentiometer is used to adjust the sensitivity of the infrared sensor, and an LCD display is used to display the information to the user.

2.3 CIRCUIT DIAGRAM

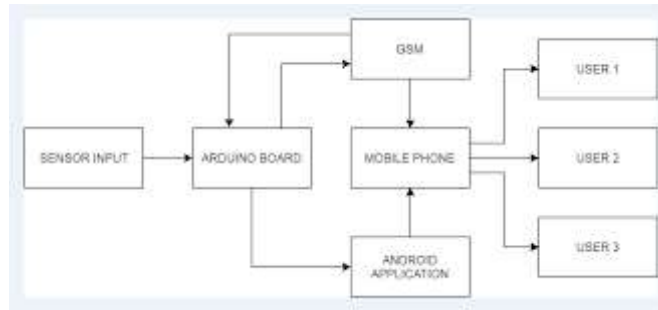


Fig.2.3 Circuit diagram for Proposed parking system

IR Sensor: This system uses an ultrasonic sensor to detect the presence of a vehicle in a parking space.

Servo Motor: The system uses a servo motor to control the opening and closing of the parking door.

Arduino Uno: The system uses an Arduino Uno board to communicate with sensors and actuators.

GSM Module: The system uses a GSM module to send SMS alerts to users about available parking spaces.

LCD Display: The system uses an LCD display to display the status of parking spaces.

Android Application: The system includes an Android application that allows users to reserve a parking space and receive SMS alerts.

2.4 COMPONENTS

2.4.1 HARDWARE COMPONENTS

1. Arduino UNO



Fig 2.4 Arduino UNO

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- Flash Memory: 32 KB
- Clock Speed: 16 MHz

2. IR Sensors

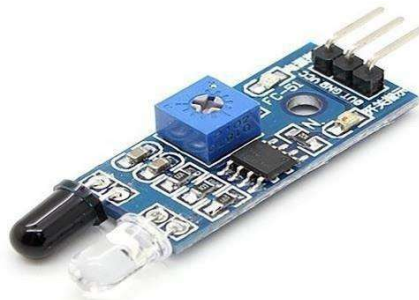


Fig 2.5 IR Sensor

- Active Output Level: Outputs Low Logic Level When Obstacle is Detected.
- On board Obstacles Detection LED Indicator.

3. LCD Display



Fig 2.6 LCD Display

- A 16x2 LCD Display is very basic module and is very commonly used in various devices and circuits.
- A 16x2 LCD means it can display 16 characters per line and there 2 such lines.

4. Servo Motors



Fig 2.7 Servo Motors

- A servomotor is a rotary actuator that allows for precise control of angular position, velocity and acceleration.
- It consists of a suitable motor coupled to a sensor for position feedback.
- Servo motors are used to lift the gates at the entry and exit of the parking area.

5. LED Lights



Fig 2.8 LED Lights

- A light-emitting diode (LED) is a semiconductor light source.
- LED's are used as indicator lamps in many devices, and are increasingly used for lighting.

2.4.2 SOFTWARE COMPONENTS

1. Arduino IDE



Fig 2.9 Arduino IDE

- The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java.

- It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one click mechanisms to compile and upload programs to an Arduino board.

2.5 MODULES NAME

- Exit gate control.
- Servo (motor) for exit gate.
- Parking area.
- EI sensor-4 (perhaps a type of sensor for parking management).
- Green and Red LEDs (possibly for indicating parking status).
- Arduino UNO (microcontroller board).
- IR sensors (3 and 2, might be used for sensing objects or vehicles).
- LCD display (for showing information).
- IR sensor-1612 (possibly a different type of IR sensor).
- Servo (motor) for entry gate.
- Entry gate control.
- Node MCU (microcontroller unit).
- Esp8266 (a Wi-Fi module).
- End user (the person using the parking system).

2.6 FLOW CHART

2.6.1 Flow Chart for Existing System

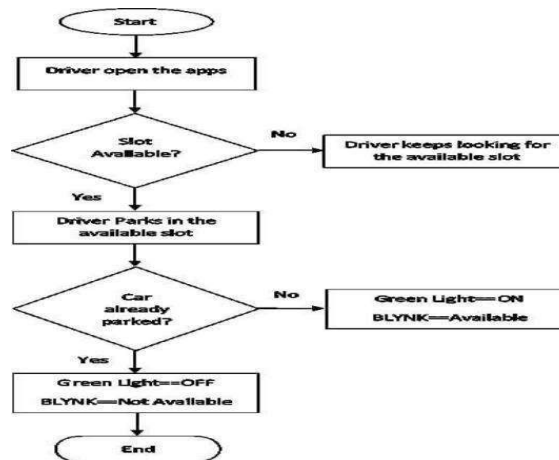


Fig 2.10 Flow Chart for Existing System

The existing parking system involves drivers arriving at the parking lot, searching for an available spot, parking their vehicles, retrieving parking tickets if applicable, and leaving their vehicles until they return, at which point they retrieve their vehicles and present their parking tickets before exiting the lot.

2.6.2 Flowchart for Slot Reservation

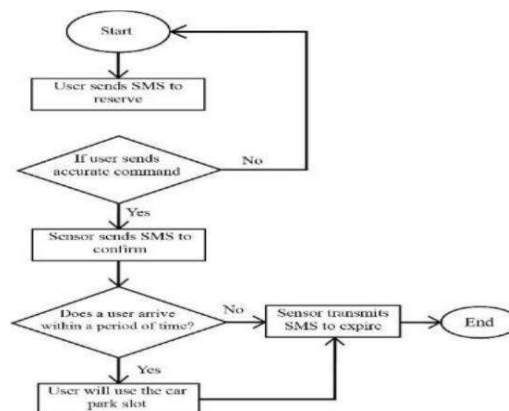


Fig 2.11 Flowchart for Slot Reservation

The process of slot reservation in the parking system begins with the user initiating a reservation request, followed by the system displaying available parking slots, the user selecting a desired slot, and the system checking availability before either reserving the slot and generating confirmation or notifying the user of unavailability.

2.6.3 Flowchart for Vehicle Entry Flow

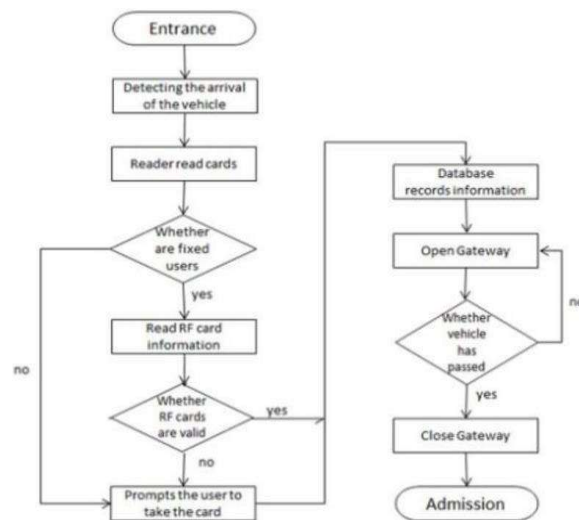


Fig 2.12 Flowchart for Vehicle Entry Flow

The vehicle entry flow in the parking system begins with the vehicle arriving at the entry gate, where the gate recognition system may identify the vehicle, followed by automatic gate opening or interaction with the driver for collecting an entry ticket or pass before logging the entry in the system, if attended by a gate attendant.

2.6.4 Flowchart for Vehicle Search Space Process

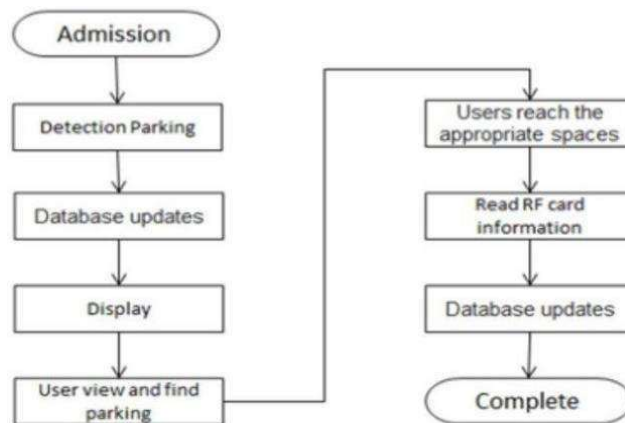


Fig 2.13 Flowchart for Vehicle Search Space Process

The vehicle search space process begins with the vehicle entering the parking lot, followed by the parking system identifying available spaces, displaying them to the driver, who then selects a preferred space, with the system verifying its availability, allowing the driver to proceed if the space is available, or updating the display if not, until the driver parks the vehicle.

2.6.5 Flowchart for Vehicle Exit Flow

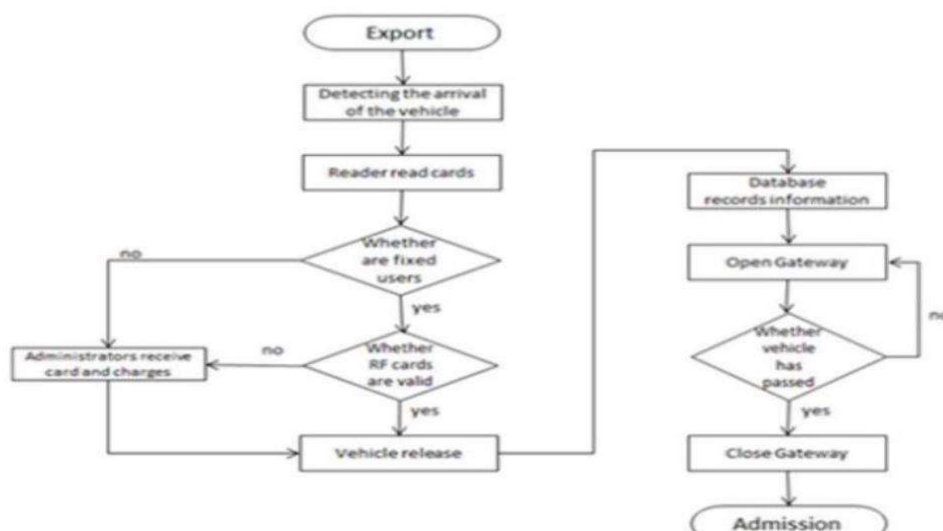


Fig 2.14 Flowchart for Vehicle Exit Flow

The process of vehicle leaving in the parking system entails the user returning to the parking lot, locating the parked vehicle, retrieving it, presenting any necessary documentation, such as a reservation confirmation, to the attendant if applicable, and then exiting the parking lot. The start of the car approaching the exit. Make sure the vehicle has a valid exit ticket.

If the ticket is active, proceed to the next step; If not, have the driver pay the required amount. If payment is required, direct the driver to the payment counter. Wait for the driver to pay. Check the payment status. If payment is OK, issue a certified exit ticket. If the ticket has already been validated, cancel the payment. Show the certified ticket at the exit. Reconfirm the validity of the ticket.

If the ticket is valid, open the exit door. Get the car out of the parking lot. If the ticket is invalid or expired, refuse to exit and ask the driver to rectify the issue. Offer options for resolving the issue, such as paying any outstanding bills or seeking professional help. Once the issue is resolved, verify the ticket again. Once repaired properly, continue to open the exit door.

Allow the vehicle to exit. Log exit connections in the system. Update the number of available parking spaces. Check the exits for obstructions or safety hazards. Make sure traffic flows smoothly. If there is no restriction, execute the next exit transaction. If there is an obstruction, alert the appropriate personnel for assistance. Provide alternate transportation routes if necessary. Keep watching the exit. Promptly address any mechanical problems with the exit door. Establish communication with drivers to facilitate a smooth exit. Make sure all cars exit in order.

CHAPTER 3

METHODOLOGY

3.1 EXISTING METHODOLOGY

currently in use In general, the parking area is managed by human resources. As a consequence, there are traffic queues. It is impossible to know the number of open slots ahead of time. In emergency cases, the user cannot book their slot using the current technique. When the city becomes congested, everyone will be unable to find a parking spot. Proposed Methodology The proposed framework is an Android application that is made up of three key components.



Fig 3.1 Board showing available Parking Slots

One of the most common problems today is a saturation of parking spaces. Vehicles continue to outnumber existing parking spaces, thus clogging roads. Incidences of violence over occupancy, deformed cars due to a space crunch, and overcharging for parking are some problems that result. Admist this people are still not able to perfectly find the available slots for parking their vehicles.

3.2 PROPOSED SYSTEM

After logging into the app, the user can choose a parking space that is closest to his destination. When a consumer books a parking space, the administrator changes the status of that parking space to "RESERVED." If the user does not arrive at the parking slot within 20 minutes of making the reservation, the reservation is cancelled and the status is changed to "EMPTY." The client-server architecture underpins the Intelligent Parking System (IPS). It's economically beneficial since it doesn't require any heavy infrastructure. It is neither sensitive to temperature change nor affected by any extreme air turbulence.

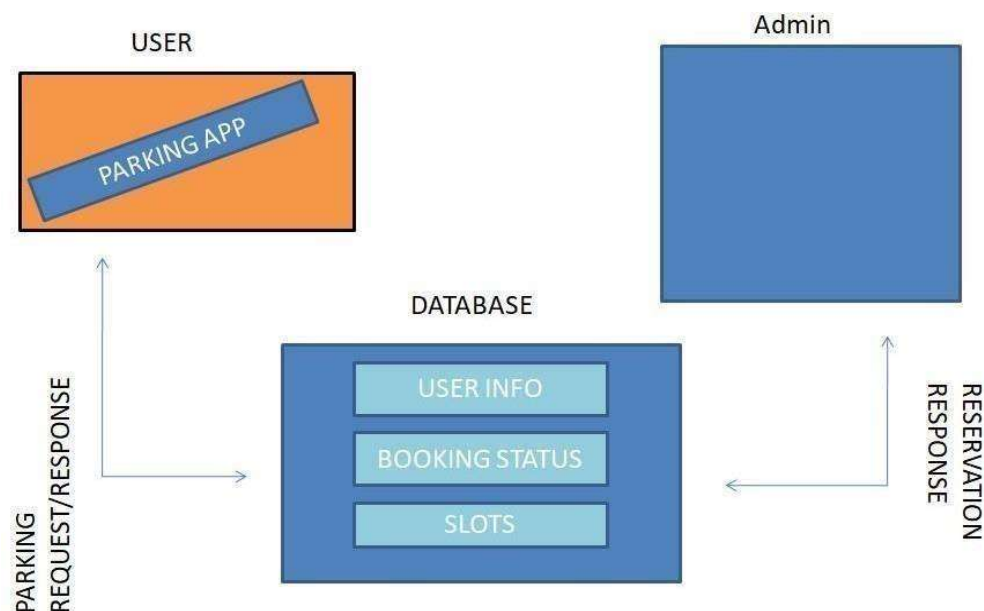


Fig 3.2 Block Diagram for proposed system

The following are the key goals of the Intelligent Parking System(IPS)application:

3.2.1 An intelligent, ubiquitous, and user-friendly automated parking system application that saves users time and helps them escape traffic congestion in major cities.

3.2.2 To ensure that parking spaces are safe and protected within a restricted area.

3.2.1 IMPLEMENTATION

The following is the sequence of the slot allocation method:

Step 1: Initially, the user selects the slot from his mobile phone. He looks for a parking spot that is close to his location and sees if one is available. If it is open, he advances to the next stage; otherwise, he returns to the previous state.

Step 2: Using an Android application, sends a request for a parking spot from the phone.

Step3: The Parking Control Unit (PCU) receives the user's requested slot number

Step 4: If the payment is completed successfully, the requested parking space will be reserved.

Step 5: After the user has reserved a specific slot, the status of that slot is labeled.

Step 6: As soon as the vehicle is parked in the designated space

Step7: As soon as the vehicle leaves the parking spot, the total cost will be shown. A module is a component of a larger device.

The Intelligent Parking System is made up of three modules. They are the following three:

1: Administrator Module

2: User Module

3: Booking Module Server Side

The administrator first enters his username and password into the application.

Server Side Initially the administrator logs in the application by using his username and password. The administrator has authority to add new users and stores their details in the database which are used for further purpose.

Login: The administrator can login to the application by giving email and password. If the administrator gets successfully login then the administrator is said to be authorized.

After getting login to the application the administrator can carry out many tasks such as:

- Adding Parking Locations
- View Parking Locations
- View All Users
- View All Bookings

3.2.2 MANAGING LOCATIONS

Add and view parking locations:

The administrator may check the status of different parking slots as well as display different locations where parking slots are open. View All Users and Their Scheduled Slots: The administrator will see all of the users who are using the app and review their booking information, such as the time and date the user needs a slot, the number of hours a person is using the assigned slot, the position at which he needs a slot, and so on. The reserved slots of all authorised users are visible to the administrator. The administrator takes this as a reference for further allocation.

3.2.3 Application Framework

Framework for Applications By making an open development platform available,Android allows developers to create apps that are incredibly rich and creative. Developers can use the device's hardware to access location data, run background services, set alarms, display alerts in the status bar, and much more.

The same platform APIs that the core applications use are available to developers. The application architecture is designed to facilitate component reuse; any application can publish its capabilities, which can then be used by other applications (subject to security constraints enforced by the framework). The consumer can substitute components using the same mechanism.

All applications are supported by a range of facilities and systems, which include Lists, grids, text boxes, icons, and even an embeddable web interface are among the Views that can be used to build an application.

Content Providers allow apps to access data from other apps (like Contacts), as well as share their own data. A Resource Manager is someone who manages resources. A Notification Manager that allows all apps to display custom notifications in the status bar. An activity manager that handles application lifecycles and offers a unified navigation backend.

3.2.4 Libraries

Android contains a collection of C/C++ libraries that are used by various Android components. The Android application platform makes these features available to developers.

The following are some of the most important libraries:

System C library library (libc) is a BSD-derived implementation of the regular C system library (libc) that is optimised for embedded Linux devices.

Media Libraries - The libraries are based on Packet Video's Open CORE and allow playback and recording of a variety of audio and video formats, as well as static image

Surface Manager - - handles access to the display subsystem and seamlessly composites 2D and 3D graphic layers from various applications.

Lib Web Core - a modern web browser engine which powers both the Android browser and an embeddable web view.

SGL - the underlying 2D graphics engine.

3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer.

FreeType - bitmap and vector font rendering.

SQLite - a powerful and lightweight relational database engine available to all applications.

3.2.5 Android Runtime

Android includes a collection of core libraries that has most of the practicality offered within the core libraries of the Java programming language. Every mechanical man application runs in its own method, with its own instance of the virtual machine referred to as Dalvik virtual machine. Dalvik has been written in order that a tool will run multiple VMs with efficiency. The Dalvik VM executes files within the Dalvik feasible (.dex) format that is optimized for lowest memory footprint. The VM is register-based, and runs categories compiled by a Java language compiler that are reworked into the .dex format by the enclosed "dx" tool. The Dalvik VM depends on the Linux kernel for underlying practicality like threading and low-level memory management.

Linux Kernel

Android's core system services, such as security, memory management, process management, network stack, and driver model, are all based on Linux version 2.6. Between the hardware and the rest of the software stack.

ANDROID SDK

Rather than installing a new SDK starter kit, you can use the Android SDK and AVD Manager to upgrade to the current tools or framework
http://dl.google.com/android/android-sdk_r16-macosx.zip

Here's a rundown of the steps you'll need to take to install the Android SDK:

Prepare your development computer and ensure it meets the system requirements.

Install the SDK starter package from the table above. (If you're on Windows, download the installer for help with the initial setup.)

you'll need to install the ADT Plugin for Eclipse.

To your SDK, Add Android platforms and other components.

Examine the Android SDK's contents.

3.2.6 System Requirements

Supported Operating Systems

Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)

Mac OS X 10.5.8 or later (x86 only)

Linux (tested on Ubuntu Linux, Lucid Lynx)

GNU C Library (glibc) 2.7 or later is required.

On Ubuntu Linux, version 8.04 or later is required.

64-bit distributions must be capable of running 32-bit applications.

For information about how to add support for 32-bit applications

Supported Development Environments for Android SDK

Eclipse IDE

Eclipse 3.6 (Helios) or greater

Note: Eclipse 3.5 (Galileo) is no longer supported with the latest version of ADT.

Eclipse JDT plugin (included in most Eclipse IDE packages)

If you need to install or update Eclipse, you can download it from

<http://www.eclipse.org/downloads/>.

Several types of Eclipse packages are available for each platform. For developing Android applications, we recommend that you install one of these packages:

Eclipse IDE for Java Developers

Eclipse Classic

Eclipse IDE for Java EE Developers

JDK 5 or JDK 6 (JRE alone is not sufficient)

Android Development Tools plugin (recommended)

Not compatible with Gnu Compiler for Java (gcj)

Other development environments or IDEs

JDK 5 or JDK 6 (JRE alone is not sufficient)

Apache Ant 1.8 or later

Not compatible with Gnu Compiler for Java (gcj)

Hardware requirements for Android SDK

The Android SDK requires disk storage for all of the components that you choose to install. The table below provides a rough idea of the disk-space requirements to expect, based on the components that you plan to use.

Table 3.1 Hardware requirements for Android SDK

Component type	Approximate size	Comments
SDK Tools	35 MB	Required.
SDK Platform-tools	6 MB	Required.
Android platform (each)	150 MB	At least one platform is required.
SDK Add-on (each)	100 MB	Optional.
USB Driver for Windows	10 MB	Optional. For Windows only.
Samples (per platform)	10M	Optional.
Offline documentation	250 MB	Optional.

Note that the disk-space requirements above are in addition to those of the Eclipse IDE, JDK, or other prerequisite tools that you may need to install on your development computer.

INSTALLING THE SDK

Step 1. Preparing Your Development Computer

Before getting started with the Android SDK, take a moment to confirm that your development computer satisfies the system's requirements. If you don't already have it, you'll probably need to install the JDK. If you're going to build in Eclipse with the Android Development Tools (ADT) Plugin—which is the recommended route if you're new to Android—make sure you have the right version of Eclipse installed on your device, as mentioned in the System Requirements. document.

If you need to install Eclipse, you can download it from this location:
<http://www.eclipse.org/downloads/>

The "Eclipse Classic" version is recommended. Otherwise, a Java or RCP version of Eclipse is recommended.

Step 2. Downloading the SDK Starter Package

The SDK starter package is not a full development environment—it includes only the core SDK Tools, which you can use to download the rest of the SDK components (such as the latest Android platform).

If you haven't already, get the latest version of the SDK starter package from the [SDK download page](#).

If you downloaded a .zip or .tgz package (instead of the SDK installer), unpack it to a safe location on your machine. By default, the SDK files are unpacked into a directory named android-sdk-<machine-platform>.

If you downloaded the Windows installer (.exe file), run it now and it will check whether the proper Java SE Development Kit (JDK) is installed (installing it, if necessary), then install the SDK Tools into a default location (which you can modify).

Make a note of the name and location of the SDK directory on your system—you will need to refer to the SDK directory later, when setting up the ADT plugin and when using the SDK tools from the command line.

Step 3. Installing the ADT Plugin for Eclipse

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT), that is designed to give you a powerful, integrated environment in which to build Android applications. It extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, debug your applications using the Android SDK tools, and even export signed (or unsigned) APKs in order to distribute your application. In general, developing in Eclipse with ADT is a highly recommended approach and is the fastest way to get started with Android.

If you'd like to use ADT for developing Android applications, install it now. Read [Installing the ADT Plugin](#) for step-by-step installation instructions, then return here to continue the last step in setting up your Android SDK.

If you prefer to work in a different IDE, you do not need to install Eclipse or ADT. Instead, you can directly use the SDK tools to build and debug your application. The [Introduction](#) to Android application development outlines the major steps that you need to complete when developing in Eclipse or other IDEs.

Step 4. Adding Platforms and Other Components

The last step in setting up your SDK is using the Android SDK and AVD Manager (a tool included in the SDK starter package) to download essential SDK components into your development environment.

The SDK uses a modular structure that separates the major parts of the SDK—Android platform versions, add-ons, tools, samples, and documentation—into a set of separately installable components. The SDK starter package, which you've already downloaded, includes only a single component: the latest version of the SDK Tools. To develop an Android application, you also need to download at least one Android platform and the associated platform tools. You can add other components and platforms as well, which is highly recommended.

If you used the Windows installer, when you complete the installation wizard, it will launch the Android SDK and AVD Manager with a default set of platforms and other components selected for you to install. Simply click Install to accept the recommended set of components and install them. You can then skip to Step 5, but we recommend you first read the section about the Available Components to better understand the components available from the Android SDK and AVD Manager.

You can launch the Android SDK and AVD Manager in one of the following ways:

From within Eclipse, select **Window > Android SDK and AVD Manager**.

On Windows, double-click the SDK Manager.exe file at the root of the Android SDK directory.

On Mac or Linux, open a terminal and navigate to the tools/ directory in the AndroidSDK, then execute:

Android

To download components, use the graphical UI of the Android SDK and AVD Manager to browse the SDK repository and select new or updated components (see figure 1). The Android SDK and AVD Manager installs the selected components in your SDK environment. For information about which components you should download, see Recommended Components.

VERSION HISTORY

mechanism has seen variety of updates since its original unleash, every fixing bugs and adding new options. every version is called, in alphabetical order, when a sweet.

2.3 cake refined the computer programme, improved the soft keyboard and copy/paste options, improved recreation performance, additional SIP support (VoIP calls), and additional support for close to Field Communication.

3.0 Honeycomb was a tablet-oriented unleash that supports larger screen devices and introduces several new computer programme options, support for multi-core processors, hardware acceleration for graphics and full system encoding. the primary device that includes this version, the Motorola Xoom pill, went on sale in Feb 2011.

3.1 Honeycomb, discharged in could 2011, additional support for further input devices, USB host mode for transferring info directly from cameras and alternative devices, and therefore the Google Movies and Books apps.

Honeycomb, discharged in July 2011, additional improvement for a broader vary of screen sizes, new "zoom-to-fill" screen compatibility mode, loading media files directly from Coyote State card, associated an extended screen support API. Huawei MediaPad is that the initial seven in. pill to use this version.

4.0 frozen dessert Sandwich, declared on Gregorian calendar month nineteen, 2011, brought Honeycomb options to smartphones and additional new options together with identity verification unlock, network information usage watching and management, unified social networking contacts, photography enhancements, offline email looking, app folders, and knowledge sharing exploitation NFC. Android 4.0.3 frozen dessert Sandwich is that the latest automaton version that's out there to phones. The ASCII text file of automaton four.0.1 was discharged on Gregorian calendar month one4, 2011.



[Prefer a video? We have one for you.](#)

Python Program

Python programs are composed of modules

Modules contain statements

Statements contain expressions

Expressions create and process objects

Features of Python

Open source: Python is publicly available open source software,any one can use source code that doesn't cost anything.

Easy-to-learn: Popular (scripting/extension) language, clear and easy syntax, no type declarations, automatic memory management, high-level data types and operations, design to read (more English like syntax) and write (shorter code compared to C, C++, and Java) fast.

High-levelLanguage:

High-level language (closer to human) refers to the higher level of concept from machine language (for example assembly languages). Python is an example of a high-level language like C, C++, Perl, and Java with low-level optimization.

Portable:

High level languages are portable, which means they are able to run across all major hardware and software platforms with few or no change in source code. Python is portable and can be used on Linux, Windows, Macintosh, Solaris, FreeBSD, OS/2, Amiga, AROS, AS/400 and many more.

Object-Oriented: Python is a full-featured object-oriented programming language, with features such as classes, inheritance, objects, and overloading.

Python has an interactive console where you get a Python prompt (command line) and interact with the interpreter directly to write and test your programs. This is useful for mathematical programming.

Interpreted : Python programs are interpreted, takes source code as input, and then compiles (to portable byte-code) each statement and executes it immediately. No need to compiling or linking.

Extendable : Python is often referred to as a "glue" language, meaning that it is capable to work in mixed-language environment. The Python interpreter is easily extended and can add a new built-in function or modules written in C/C++/Java code.

Libraries : Databases, web services, networking, numerical packages, graphical user interfaces, 3D graphics, others.

Supports :Support from online Python community

Python Interpreter

In interactive mode, type Python programs and the interpreter displays the result:

Type python into your terminal's command line

After a short message, the >>> symbol will appear

The above symbol signals the start of a Python interpreter's command line.

Python interpreter evaluates inputs (For example >>> 4*(6-2) return 16)

How stable is Python?

Very stable. New, stable releases are popping out roughly each half dozen to eighteen months since 1991, and this appears seemingly to continue. presently there ar sometimes around eighteen months between major releases. The latest stable releases will invariably be found on the Python transfer page. There ar 2 counseled production-ready versions at this time in time, as a result of at the instant there ar 2 branches of stable releases: a pair of.x and 3.x. Python 3.x could also be less helpful than a pair of.x, since presently there's additional third party code accessible for Python a pair of

than for Python three. Python a pair of code can typically not run unchanged in Python three.

History

The name Python was elect from "Monty Pythons Flying Circus" that was a British sketch comedy series created by the comedy cluster Monty Python and broadcast by the BBC from 1969 to 1974. Python was created within the early Nineties by Guido van Rossum at the National analysis Institute for arithmetic and technology in Holland. Python was created as a successor of a language known as rudiment (All Basic Code) and discharged publically in1991. Guido remains Python's principal author, though it includes several contributions from active user community. Between 1991 and 2001 there area unit many versions discharged, current stable unharness is three.2. In 2001 the Python software system Foundation (PSF) was shaped, a non-profit organization created specifically to possess Python-related holding. Zope Corporation could be a sponsoring member of the terrorist organization. All most all Python releases area unit Open supply. to check the main points of unharness versions and licence agreement of Python check here.

Python introduction video

Python Environment

Table 3.2 Python Environment

AIX	AROS	AS/400 (OS/400)	BeOS
Morph OS	MS-DOS	OS/2	OS/390 and z/OS
Palm OS	PlayStation and PSP	Psion	QNX
RISC OS	Series 60	Solaris	VMS
Windows CE or Pocket PC	HP-UX	Linux	

Major uses of Python

System utilities (system admin tools, command line programs).

Web Development.

Graphical User Interfaces (Tkinter, gtk, Qt).

Internet scripting.

Embedded scripting.

Database access and programming.

Game programming.

Rapid prototyping and development.

Distributed programming

Organizations Using Python (sector wise)

Web Development : Yahoo Maps, Yahoo teams, Google, Zope Corporation, Ultraseek,

Linux Weekly News, ElasticHosts Cloud Servers, Mojam.com, hunch, Shopzilla, Movieplayer.it, Multiplayer.it. Games: field of battle a pair of, Crystal area, Star Trek Bridge Commander, The Temple of Elemental Evil, Vampire: The Masquerade: Bloodlines, Civilization four, QuArK (Quake Army Knife) Graphics : Industrial light-weight & Magic, filmmaker Feature Animation, HKS, Inc. (ABAQUS/CAE), RoboFog, Caligari Corporation, Blender 3D, Jasc code, Paint look professional. Financial :Altis Investment Management, ABN AMRO Bank, Treasury Systems, Bellco bank, Journyx Timesheet and Resource Management code. Science : National Weather Service, measuring device Remote Sensing cluster, Applied Maths, Biosoft, The National analysis Council of Canada, town National Laboratory (LANL) Theoretical Physics Division, AlphaGene, Inc., LLNL, NASA, Swedish earth science and Hydrological Institute (SMHI), Environmental Systems analysis Institute (ESRI), Objexx Engineering,

Nmag procedure Micromagnetics Electronic style Automation: Ciranova, Productivity style Tools, Object Domain, Pardus, Red Hat, SGI, Inc., MCI Worldcom, Nokia, Education : University of Calif., Irvine, Smeal school of Business, The Pennsylvania State University, New island Digital Library, IT Certification examination preparation, SchoolTool, Business code : Raven Bear Systems Corporation, Thawte Consulting, Advanced Management Solutions opposition., IBM, Arakn, RealNetworks, dSPACE, Escom, the little Company, Nexedi, Piensa Technologies - Bufete Consultor First State United Mexican States, Nektra, WuBook.

Is Python a good language for beginning programmers?

Yes. it's still common to start out students with a procedural and statically written language like Pascal, C, or a set of C++ or Java. Students could also be higher served by learning Python as their natural language. Python contains a terribly straightforward and consistent syntax and an outsized normal library and, most significantly, victimization Python in a very starting programming course lets students focus on vital programming skills like drawback decomposition and knowledge sort style. With Python, students may be quickly introduced to basic ideas like loops and procedures. they will in all probability even work with user-defined objects in their terribly 1st course. For a student World Health Organization has ne'er programmed before, employing a statically written language appears unnatural. It presents extra complexness that the scholar should master and slows the pace of the course. the scholars are attempting to be told to assume sort of a pc, decompose issues, style consistent interfaces, and encapsulate knowledge. whereas learning to use a statically written language is very important within the future, it's not essentially the most effective topic to deal with within the students' 1st programming course. Many different aspects of Python build it an honest natural language. Like Java, Python contains a massive normal library so students may be assigned programming comes terribly early within the course that do one thing.

Assignments aren't restricted to the quality four-function calculator and check equalization programs. By victimization the quality library, students will gain the satisfaction of acting on realistic applications as they learn the basics of programming. victimization the quality library additionally teaches students regarding code apply. Third-party modules like PyGame are useful in extending the students' reach.

SYSTEM TESTING

The purpose of testing is to seek out errors. Testing is that the tactic of Associate to seek out every conceivable fault or weakness in an passing work product. It provides how to ascertain the practicality of elements, sub assemblies, assemblies and/or a finished product it's the method of exercise code with the intent of guaranteeing that the Software system meets its necessities associated user expectations and doesn't fail in an unacceptable manner. There are varied forms of take a look at every take a look at kind addresses a particular testing demand.

TYPES OF TESTS

1. Unit testing

Unit checking involves the design of check cases that validate that the within program logic is functioning properly, that program inputs prove valid outputs. All call branches and internal code flow need to be valid. it is the testing of individual software units of the appliance .it is done once the completion of a private unit before integration. this may be a structural testing, that depends on info of its construction and is invasive. Unit checks perform basic checks at half level and test a selected business process, application, and/or system configuration. Unit tests confirm that each unique path of a business methodology performs accurately to the documented specifications and contains clearly printed inputs and expected results.

2. Integration testing

Integration tests are designed to examine integrated package components to work out if they terribly lie program. Testing is event driven and is extra involved the essential outcome of screens or fields. Integration tests demonstrate that tho' the components were one by one satisfaction, as shown by successfully unit testing, the mixture of components is correct and consistent. Integration testing is specifically aimed toward exposing the issues that arise from the mixture of components.

3. Functional check

Functional tests supply systematic demonstrations that functions tested are gettable as nominal by the business and technical requirements, system documentation, and user manuals. Functional testing is targeted on the next items:

Valid Input : noted classes of valid input ought to be accepted.

Invalid Input : noted classes of invalid input ought to be rejected. Functions : noted functions ought to be exercised.

Output : noted classes of application outputs ought to be exercised. Systems/Procedures: interfacing systems or procedures ought to be invoked.

Organization and preparation of helpful tests is targeted on requirements, key functions, or special check cases. In addition, systematic coverage relating to verify Business methodology flows; data fields, predefined processes, and serial processes must be thought of for testing. Before helpful testing is complete, more tests are identified and so the effective price of current tests is set .

SYSTEM CHECK

System testing ensures that the total integrated computer code meets necessities.

It tests a configuration to form certain famed and inevitable results. AN example of system checking is that the configuration minded system integration test. System testing is predicated on method descriptions and flows, action pre-driven method links and integration points.

White Box Testing

White Box Testing is also a testing throughout that throughout that the computer code tester has knowledge of the inner workings, structure and language of the computer code, or a minimum of its purpose. it's purpose. it's accustomed check areas that cannot be reached from a recorder level.

Black Box Testing

Black Box Testing is testing the computer code with none data of the inner workings, structure or language of the module being tested. recording machine tests, as most alternative forms of tests, should be written from a definitive supply document, like specification or requirements document, like specification or necessities document. it's a testing during that the computer code underneath check is treated, as a recorder .you cannot “see” into it. The check provides inputs and responds to outputs while not considering however the computer code works.

Unit Testing

Unit checking is often conducted as a vicinity of a combined code and unit test section of the computer code lifecycle, though it isn't uncommon for writing and unit testing to be conducted as 2 distinct phases.

Test strategy and approach

Field testing is reaching to be performed manually and purposeful tests are reaching to be written intimately .

Test objectives

All field entries should work properly.

Pages should be activated from the known link.

The entry screen, messages and responses should not be delayed.

Features to be tested

Verify that the entries are of the proper format

No duplicate entries ought to be allowed

All links ought to take the user to the proper page.

INTEGRATION TESTING

Software integration testing is that the incremental integration testing of two or more integrated software components on one platform to supply failures caused by interface defects.

The task of the mixing test is to see that components or software applications, e.g. components during a software or - one intensify - software applications at the corporate level - interact without error.

Test Results: All the test cases mentioned above are passed successfully. No defects are encountered.

Acceptance Testing

User Acceptance Testing may be a critical phase of any project and requires significant participation by the top user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above are passed successfully. No defects are encountered.

CHAPTER 4

SIMULATION THEORY

4.1 PROTEUS DESCRIPTION

Proteus is a famous electronic layout automation (EDA) software program that offers a platform for designing, simulating, and analyzing electronic circuits. It is utilized by engineers, hobbyists, and students to create schematic diagrams, PCB layouts, and three-D fashions of digital circuits.

Proteus gives a wide range of capabilities and tools for electronic layout, such as:

Schematic Capture: Proteus lets in customers to create and edit schematic diagrams of digital circuits the usage of a drag-and-drop interface. It helps numerous additives, along with resistors, capacitors, transistors, ICs, and microcontrollers.

Simulation: Proteus provides a simulation environment that permits users to test and analyze the behavior of digital circuits before constructing them. It supports diverse simulation models, which includes SPICE, VHDL, and Verilog.

PCB Layout: Proteus lets in users to create PCB layouts of digital circuits the use of a graphical interface. It supports numerous PCB layout rules, which includes trace width, clearance, and thru length.

3-d Modeling: Proteus offers a 3-D modeling environment that allows users to visualize and examine the bodily properties of digital circuits. It helps various 3-D fashions, inclusive of enclosures, connectors, and cables.

Library: Proteus consists of a huge library of electronic additives, such as resistors, capacitors, transistors, ICs, and microcontrollers, that may be utilized in electronic designs.

Proteus is widely used in various industries, inclusive of car, aerospace, scientific, and purchaser electronics, for designing and simulating electronic circuits. It is to be had in diverse editions, together with Proteus 8 Professional, Proteus 8 Lite, and Proteus eight SCH, that cater to exclusive design wishes and budgets.

Simulation is the process of creating a digital version of a device or a system and jogging experiments on it to analyze its conduct and overall performance. In the context of electronic circuits, simulation refers back to the procedure of making a digital model of an digital circuit and going for walks experiments on it to analyze its electric conduct and performance.

4.2 SIMULATION

Simulation is an vital tool in digital design and evaluation, as it allows designers to check and optimize digital circuits before building them. Simulation can assist designers to pick out and fix layout mistakes, compare the overall performance of the circuit beneath various situations, and optimize the circuit for value, energy consumption, and different factors.

There are diverse simulation tools and software available for electronic circuits, such as Proteus, LTspice, PSpice, and Multisim. These gear offer a graphical interface for growing and modifying digital schematics, in addition to a simulation engine for analyzing the electric behavior of the circuit.

Simulation may be used to research numerous electric parameters of the circuit, including voltage, modern, strength, frequency, and impedance. It also can be used to investigate the transient response of the circuit, which includes the start-up conduct, the reaction to enter indicators, and the steadiness of the circuit.

Simulation can be accomplished at numerous degrees of abstraction, together with circuit-level simulation, system-level simulation, and behavioral simulation. Circuit-stage simulation includes modeling the individual additives of the circuit, including resistors, capacitors, and transistors, and analyzing their electric conduct. System-level simulation

includes modeling the complete gadget, such as a digital or analog circuit, and studying its behavior and performance.

Behavioral simulation entails modeling the excessive-level behavior of the circuit, which includes its functionality and overall performance, without modeling the man or woman components.

Simulation is a powerful device in electronic design and analysis, as it permits designers to test and optimize electronic circuits before constructing them, reducing the time, cost, and risk related to physical prototyping and checking out.

The automotive exhaust flow diagram begins by defining the theoretical parameters and objectives.

The first step is a realistic design of the parking lot and exit. Then, the simulation software starts the parking with a fixed number of vehicles. Each vehicle is assigned specific attributes such as entry time, length of stay and payment terms. The simulation then runs at discrete time intervals, typically representing seconds or minutes. Over time, cars start leaving the parking lot based on the scheduled exit time. The simulation tracks the movement of each vehicle towards the exit. Analyzes traffic to identify potential dead ends or obstacles. The software takes into account factors such as vehicle speeds, distances between vehicles and lanes. If a car has a valid exit ticket, it moves through the exit smoothly.

CHAPTER 5

SIMULATION CIRCUIT AND RESULTS

5.1 SIMULATION CIRCUIT

The simulation of the project was carried out in Proteus simulation. parking system was achieved. Simulation Circuit Diagram for parking system is also shown in fig.4.1.

The figures in the subsequent pages show the following how they work and what are the components used in simulation setup.

The Proteus simulation uses the following features.

UNO: This is the Arduino Uno board, which is a microcontroller board based on the ATmega328P microcontroller. It is used to control parts of the circuits.

SOR2: This is a 4-channel relay module that can be controlled by an Arduino Uno board. It is used to control garage doors and other high-powered devices.

R: This is a 12V DC power supply for the relay module.

ARD1: This is the Arduino Uno board connector used to connect the board to the breadboard.

UT4: This is an ultrasonic sensor module used to measure the distance between the sensor and the object. It consists of two parts: a transmitter (UT) and a receiver (UR).

PAI: This is a 10k potentiometer used to adjust the sensitivity of the ultrasonic sensor.

PAZ: This is a 10k resistor used to pull up the input pin of an ultrasonic sensor.

PR3: This is a 1N4007 diode used to protect the circuit from voltage spikes generated by the relay.

R4: This is a 1k resistor used to stop the current flowing through the diode.

5.2 APPLICATION RESULTS

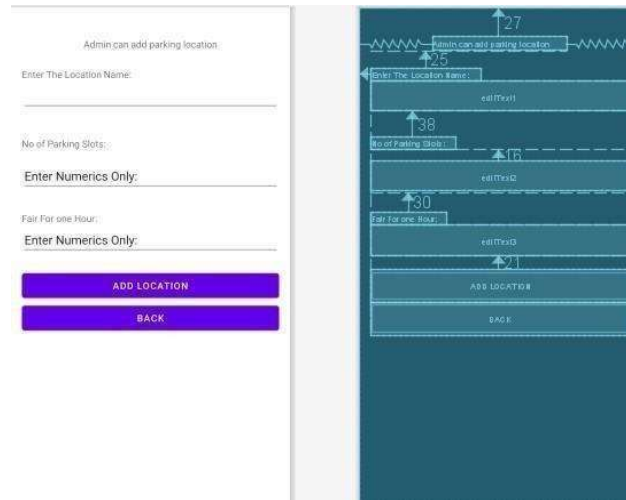


Fig 5.2 Admin add parking slot page

The image above shows admin add parking location, given in the first location, the admin enters the location to add followed by the slot to add. Appropriateness estimates will be based on establishment time and will be disclosed accordingly.



Fig 5.3 Admin home page

Above determine indicates admin's home page .All the specs of the software may be to be had in this page, admin can upload parking locations ,if admin desires to control location, can control within the next block, if admin desires to view all of the parking places available may be in a position to check in the view places choice.

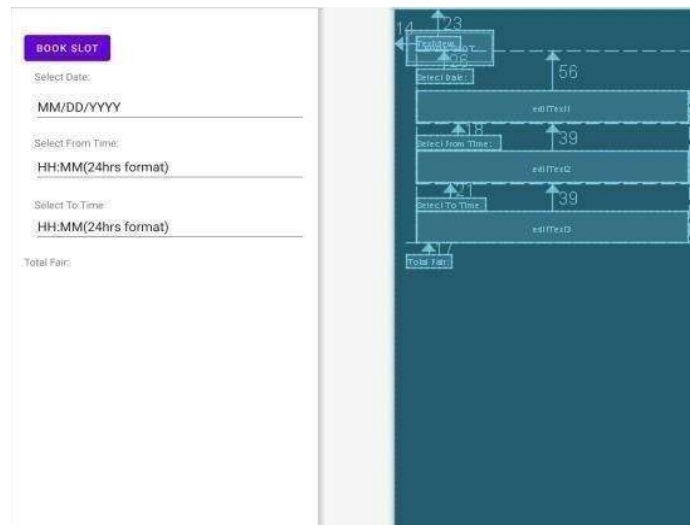


Fig 5.4 Slot booking page

Above figure suggests reserving slot web page, date in format of MM/DD/YYYY of the unique day while the person want to book the slot. Next consumer want to enter the time for which he wants to park the car. After coming into all the required fields straight away, total fair will be displayed on the display screen based totally on parking hours.

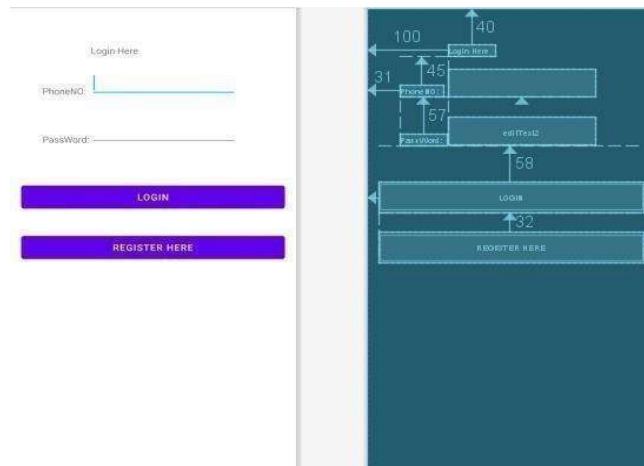


Fig 5.5 Login page

If person has already registered inside the app then consumer want to go into the login information to undergo the app and book the slot for parking the car. If person is new to the app, first consumer need to sign up within the app by entering the non-public cellular number and password to login.

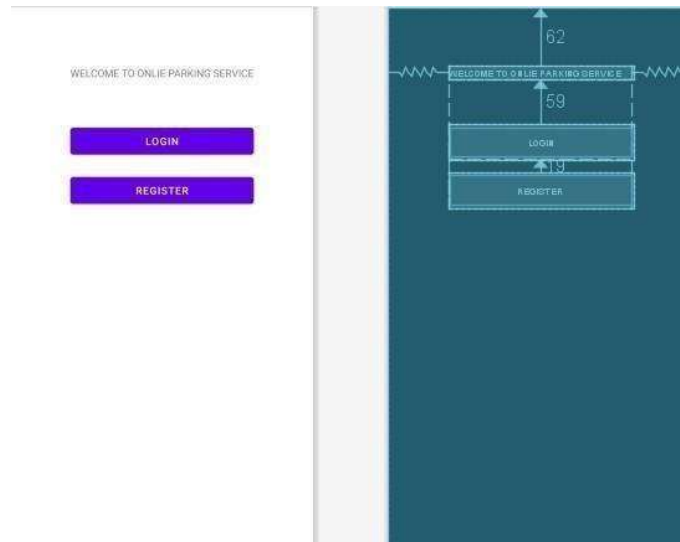


Fig 5.6 Welcome page

While launching the app, Login page display as shown in the image above. If the user is a registered user, they only need to login to continue booking the slot, otherwise they need to register to book the slot.

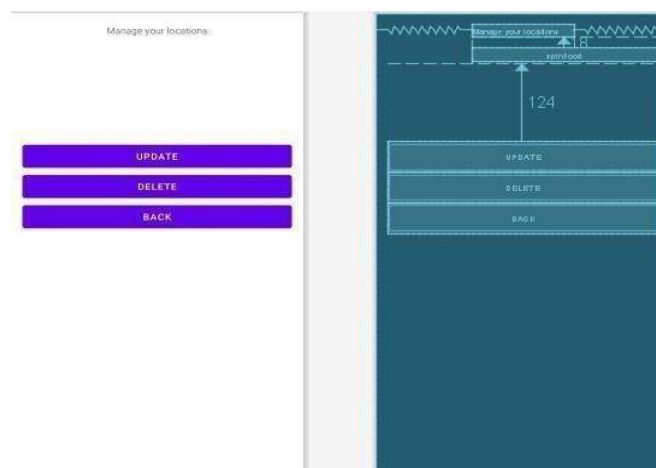


Fig 5.7 Manage your locations

While managing slots, if the administrator wants to add any slots, he/she can add new slots from various locations using the available update option and if he/she wants to delete any slots from specific locations, administrator can be deleted using the delete option provided.

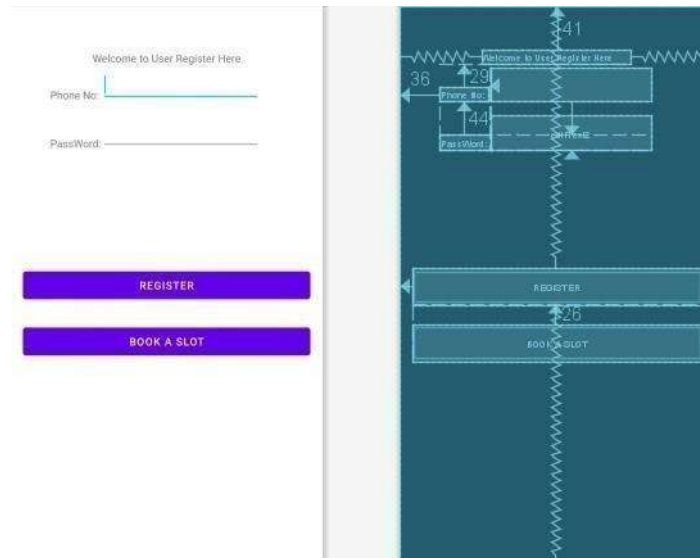


Fig 5.8 Registration page for new user

If the administrator is a new user of the application, you need to log in and register with the app in order to book a slot. To register, enter the password for the user's personal mobile number and click on register. You are now registered in the app and do not need to login again because the user can enter his driving location will be established directly.



Fig 5.9 Text view

In this Text View Page, there are four options which are view slots, book slots, view booking and cancel booking. In view slots we can see whether slots are available for bookings. In book slots we can book any slot according to our choice and comfort. After booking we can view our booking slot for parking the vehicle and we can cancel the booking slot in necessary conditions within provided time.

5.3 HARDWARE RESULTS

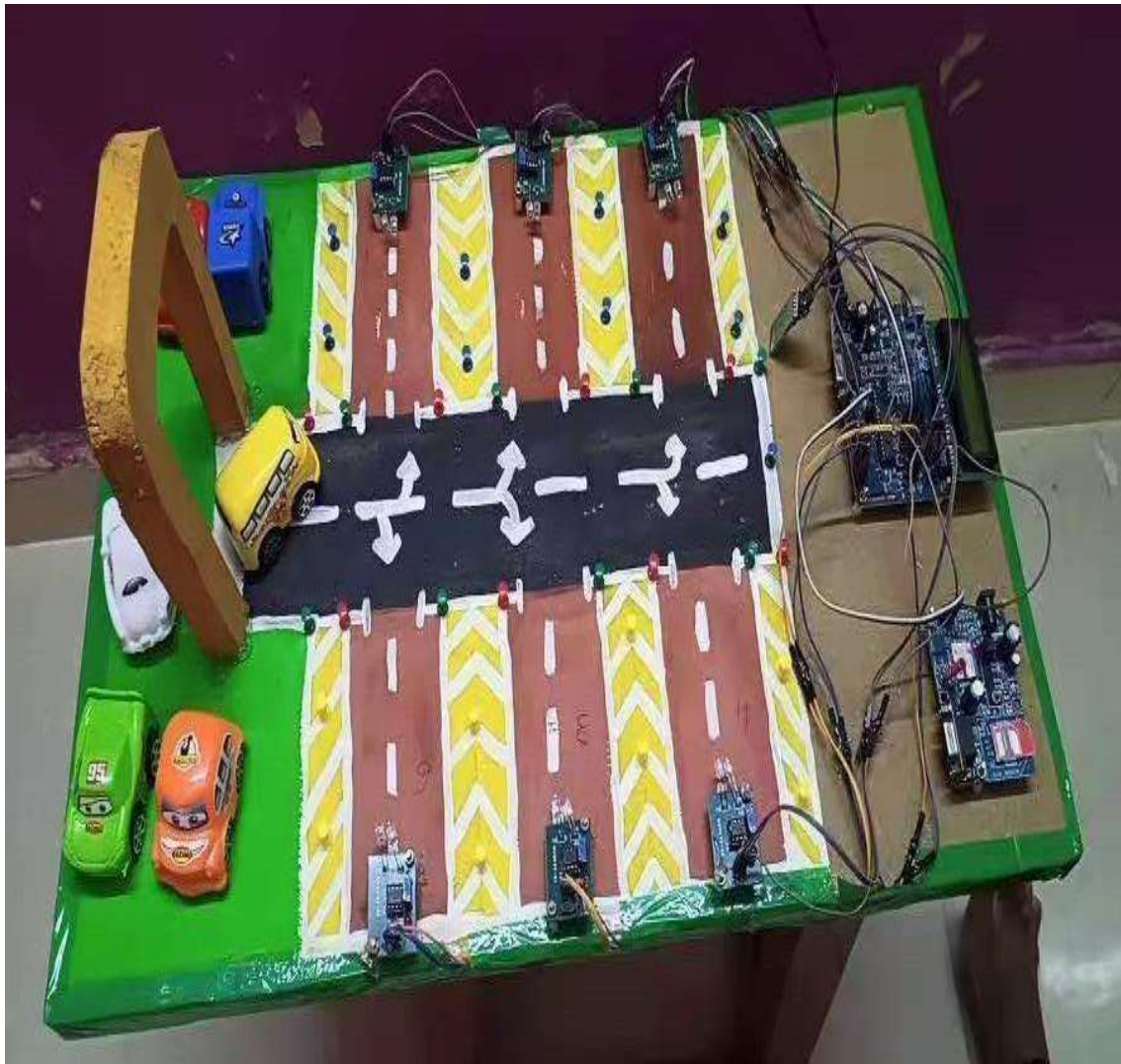


Fig. 5.10 Photocopy of Hardware circuit

Using IoT technology, the parking system can use sensors embedded in parking lots that communicate with a central system to provide real-time availability information. Drivers can use this information to quickly locate available parking spaces, reducing traffic congestion, jams, and contamination. Integrating mobile apps can enable payments, reservations, and other conveniences, making the parking experience seamless and efficient.

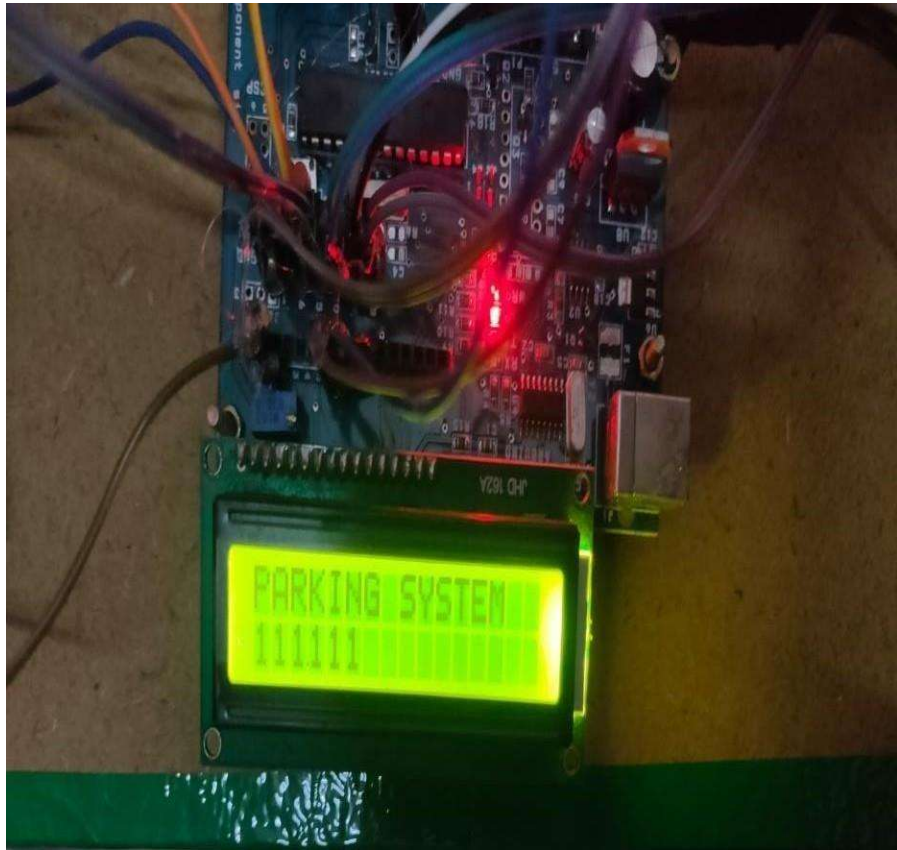


Fig 5.11 Display output

Using IoT technology, parking systems can provide real-time parking information, making it easier for drivers to find a parking spot. IoT technology can also help manage parking lots by monitoring usage patterns and occupancy, enabling employees to make informed decisions about parking design and management.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

6.1 CONCLUSION

The Intellectual Parking System (IPS) allows users to park in parking spaces with an Android smartphone with minimal effort. The buyer will check the availability of the parking space and reserve the parking space in advance. This will address a number of issues that have arisen due to poor traffic management. This design was developed for the Android mobile operating system because mobile computing has proven to be a very dynamic learning environment for researchers in recording and data distribution. This device can be used anywhere due to its ease of use and because of its convenience. The cloud will play an important role in this project as, the data of all parking lots will be available in the cloud. Cloud computing is the provision of computer services such as servers, storage, databases, networking, software, analytics, intelligence, etc. Through the cloud, cloud computing provides an alternative to on-premises datacenters. When we have a data center on campus, we have to manage everything, including buying and installing hardware, virtualization, installing operating systems, and other critical applications, setting up networks, firewalls setting up, setting up data storage and our responsibility is to maintain through it for life. This common problem nowadays is congested parking spaces. Traffic continues to exceed existing parking spaces, clogging the roads. Violence around residents, distorted traffic due to lack of space, charges for parking fees are some of the problems. Smart parking development refers to systems based on IoT which transmits data about free parking spaces via web/mobile applications.

6.2 FUTURE SCOPE

With the assist of existing methodologies and proposed methodology, the future paintings may be executed on automating the app as well. In this proposed technique, the user has to cancel the booked slots if he is not capable attain the vacation spot in specific time. Instead of this, this may be changed as automated cancelling and for this reason consequences inside the automation of app.

REFERENCES

- [1] . Jung , H. G. , Choi , C. (1999). G. , Kim , D.K. S. , & Yoon , P. (1999). J. 2006. "System Configuration of Intelligent Parking Assistant Systems," Proc. 13th World Congress. Intelligence. Transp. Syst. Works, London, UK, October).
- [2] . Kianpisheh, A., Mustaffa, N., Limtrairut, P., and Keikhosrokiani, P. 2012. "Smart Parking System (SPS) Architecture Using Ultrasonic Detector," International Journal of Software Engineering and Applications (6:3), pp. 55-58.
- [3].Panaippan , R. , Trivedi , J.M. , Studer , A. , & Perrig , A. Age . "A VANET-Based Approach to Available Vehicle Parking," Proceedings of the 4th ACM International Conference on Vehicular Ad Hoc Networks,ACM2007, pp. 1-11, 75-76.
- [4] . Pham , T. (1999). N. , Tsai , M.-F. , Nguyen , D.K. B. , Dow , C.-R. , & Deng , D.-J. 2015. "Cloud-based smart parking system based on Internet of Things technology," IEEE Access (3), pp 1581-1591.
- [5] . Solik , P. , Marasovich , I. , Stefanizzi , M. (2010). L. , Patrono , L. , & Manetti , L. (1999). "RFID- based efficient vehicle parking detection method," Software, Telecommunications and Computer Communications (SoftCOM),2015 23rd International Conference on, IEEE2015, pp.108.
- [6] . Song , I. , Govan , K. , Neri , J. , Han , H. , Sheng , T. , Lee , H. , and Currey , F. Year . "Intelligent Parking System Design Using FPGA," 2006 International Conference on Field Programmable Logic and Applications,IEEE2006, pp. 1-12, 1-6.
- [7] . Xu , J. , Chen , G. , and Xie , M. Year . "Visually-guided automated parking for intelligent vehicles,"Proceedings of the IEEE Intelligent Vehicle Symposium2000, pages 725-730.

- [8] . Yang , L. , Rongguo , M. , and Longfei , W. Year . "Intelligent Parking Communication Based on Agent Technology," Information Engineering, 2009. ICIE'09. Proceedings of the WASE International Conference on, IEEE2009, pp. 265-268.
- [9] . Ye , H.-T. , Chen , B.-C. , and Wang , B.-X. 2016. "Urban parking integration system integrated with cloudcomputing technology and smart mobile devices," Eurasia Journal of Mathematics, Science and Technology Education (12:5), pp. 1-13, 1231-1.
- [10]. Zhao , Y. , & Collins , E.J. G. 2005. "Robust Automated Parallel Parking in Tight Spaces by Fuzzy Logic," Robotics and Autonomous Systems (51:2), pp. 1-11. 111-127.
- [11] . Zheng , Y. , and Cao , J. Year . "Intelligent vehicle parking systems based on wireless sensor networks," 2006 First International Conference on Computing and Widespread Applications, IEEE2006, pp. 65-70.