IOT BASED SMART CAR PARKING SYSTEM

MINI PROJECT REPORT

Submitted by

RAGUL S- 2116230701253
AKASH N- 2116230701019

In partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024-2025

BONAFIDE CERTIFICATE

Certified that this project**"IOT BASED AUTOMATIC CAR PARKING SYSTEM**" is the bonafide work of"**RAGUL S-2116230701253 AKASH N -2116230701019"**
who carried out the project work under my supervision.

SIGNATURE
Dr.N.Duraimurugan, M.Tech., Ph.D.

Associate Professor,

Computer Science & Engineering

Rajalakshmi Engineering College
(Autonomous)
Thandalam, Chennai -602105.

Submitted for the **ANNA UNIVERSITY** practical examination Mini-Project work viva voice held on_____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work.

We also extend our sincere and hearty thanks to our Internal Guide **Dr.N.Duraimurugan, M.Tech., Ph.D.** Associate Professor, Department of Computer Science and Engineering for his valuable guidance and motivation during the completion of this project. Our sincere thanks to our family members, friends and other staff members of information technology.

RAGUL S- 2116230701253
AKASH N- 2116230701019

TABLE OF CONTENTS

CHAPTER RNO.	TITLE	PAGE NO.
	LIST OF FIGURES	1
	LIST OF ABBREVIATIONS	2
	ABSTRACT	3
1.	INTRODUCTION	4-6
	1.1 INTRODUCTION	4
	1.2 SCOPE OF THE WORK	5
	1.3 PROBLEM STATEMENT	5
	1.4 AIM AND OBJECTIVE	6
2.	SYSTEM SPECIFICATIONS	7
	2.1 HARDWARE SPECIFICATION	7
	2.2 SOFTWARE SPECIFICATION	7
3.	SYSTEM DESIGN	8-11
	3.1 ARCHITECTURE DIAGRAM	8
	3.2 USE CASE DIAGRAM	9
	3.3 ACTIVITY DIAGRAM	10
	3.4 CLASS DIAGRAM	11

4.	**MODULE DESCRIPTION**	**12-14**
	4.1 HARDWARE MODULE	**12**
	4.2 DATA COLLECTION AND PROCESSING MODULE	**12**
	4.3 ALERTING MODULE	**13**
	4.4 WEB APPLICATION MODULE	**13**
	4.5 INTEGRATION MODULE	**14**
5.	**SAMPLE CODING**	**15-16**
6.	**SCREEN SHOTS**	**17**
7.	**CONCLUSION AND FUTURE ENHANCEMENT**	**18-19**
8.	**REFERENCES**	**20**

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO.
3.1	ARCHITECTURE DIAGRAM	9
3.2	USE CASE DIAGRAM	10
3.3	ACTIVITY DIAGRAM	11
3.4	CLASS DIAGRAM	12
6.1	OUTPUT SCREENSHOT	25

LIST OF ABBREVIATION

ABBREVIATION	FULL FORM
IOT	INTERNET OF THINGS
ESP32	ESPRESSIF 32-BIT MICROCONTROLLER WITH WI-FI AND BLUETOOTH
IR	INFRARED
LED	LIGHT EMITTING DIODE
LPR	LICENSE PLATE RECOGNITION
API	APPLICATION PROGRAMMING INTERFACE
SMS	SHORT MESSAGE SERVICE
WI-FI	WIRELESS FIDELITY
AI	ARTIFICIAL INTELLIGENCE
HTML	HYPERTEXT MARKUP LANGUAGE
CSS	CASCADING STYLE SHEETS
JS	JAVASCRIPT
UI	USER INTERFACE
SSD	SINGLE SHOT MULTIBOX DETECTOR
GPS	GLOBAL POSITIONING SYSTEM (IF NAVIGATION IS INCLUDED OR PLANNED)

ABSTRACT

The IoT-Based Automatic Car Parking System is an innovative solution designed to automate and manage vehicle parking efficiently in urban and high-traffic areas. With the increasing number of vehicles, traditional parking systems are no longer effective in terms of time, space utilization, and convenience. This smart parking system uses essential components such as the **ESP32 microcontroller**, **IR sensors**, **LEDs**, **breadboard**, and a **power supply** to monitor and manage parking slots in real time. **IR sensors** are placed at each parking slot to detect whether a slot is occupied or vacant. These sensors continuously send signals to the **ESP32**, which is responsible for processing the data. The ESP32 is chosen for its built-in **Wi-Fi** capabilities, allowing it to send real-time slot availability data to a web or mobile application accessible by users. **LED indicators** are connected to show the status of each slot visually—**green** indicates availability, while **red** indicates that the slot is occupied. All components are assembled on a **breadboard** for ease of wiring and testing during development. A reliable **power supply** ensures the system runs smoothly without interruption. This automated system not only reduces the time spent searching for parking but also minimizes fuel consumption and traffic congestion. By enabling remote monitoring and real-time slot management, it provides an efficient and user-friendly parking experience. The system is ideal for deployment in shopping malls, commercial buildings, hospitals, and smart city infrastructures, making parking more intelligent and convenient for both users and administrators.

CHAPTER 1
INTRODUCTION

1. Introduction

In today's rapidly urbanizing world, the number of vehicles on the road is increasing at an unprecedented rate, leading to significant challenges in managing parking spaces efficiently. One of the most common issues faced in cities is the unavailability of proper parking systems, which results in traffic congestion, fuel wastage, time loss, and driver frustration. Traditional parking systems are often manual, inefficient, and lack real-time information, making it difficult for users to find vacant slots quickly. To overcome these limitations, the integration of modern technologies such as the Internet of Things (IoT) offers a promising solution.

An IoT-Based Automatic Car Parking System is a smart system that uses sensors and microcontrollers to detect and manage parking slot availability in real time. This system makes use of components like the **ESP32 microcontroller**, **IR sensors**, **LED indicators**, a **breadboard**, and a **power supply** to build an intelligent, automated parking infrastructure. The IR sensors detect the presence or absence of vehicles in each slot, and the ESP32 processes this data and transmits it using its built-in Wi-Fi module to a cloud-based application. This information can be accessed by users through a web or mobile interface, allowing them to locate available slots before reaching the parking area.

Additionally, LED indicators placed at each slot provide visual status updates—green for available and red for occupied—making it easier for drivers to navigate. The entire system is cost-effective, energy-efficient, and scalable, making it suitable for various environments such as shopping malls, corporate campuses, hospitals, and smart cities. By automating the parking process, the system not only saves time and fuel but also enhances the overall user experience and contributes to better traffic management. This project demonstrates how IoT can be effectively utilized to solve real-world urban problems through intelligent automation.

1.2 Scope of the Work

The IoT-Based Automatic Car Parking System is an innovative solution designed to automate and manage vehicle parking efficiently in urban and high-traffic areas. With the increasing number of vehicles, traditional parking systems are no longer effective in terms of time, space utilization, and convenience. This smart parking system uses essential components such as the **ESP32 microcontroller**, **IR sensors**, **LEDs**, **breadboard**, and a **power supply** to monitor and manage parking slots in real time. **IR sensors** are placed at each parking slot to detect whether a slot is occupied or vacant. These sensors continuously send signals to the **ESP32**, which is responsible for processing the data. The ESP32 is chosen for its built-in **Wi-Fi** capabilities, allowing it to send real-time slot availability data to a web or mobile application accessible by users. **LED indicators** are connected to show the status of each slot visually—**green** indicates availability, while **red** indicates that the slot is occupied. All components are assembled on a **breadboard** for ease of wiring and testing during development. A reliable **power supply** ensures the system runs smoothly without interruption. This automated system not only reduces the time spent searching for parking but also minimizes fuel consumption and traffic congestion. By enabling remote monitoring and real-time slot management, it provides an efficient and user-friendly parking experience. The system is ideal for deployment in shopping malls, commercial buildings, hospitals, and smart city infrastructures, making parking more intelligent and convenient for both users and administrators

Problem Statement

With the rapid increase in vehicle numbers, especially in urban and commercial areas, finding an available parking space has become a major issue. Traditional parking systems are manual and do not provide real-time information, forcing drivers to spend significant time searching for free slots. This leads to increased traffic congestion, wasted fuel, and higher pollution levels. Additionally, manual management is often inefficient, costly, and prone to errors, resulting in poor utilization of parking spaces. Drivers face frustration and delays, especially during peak hours. There is a clear need for an automated parking system that can monitor parking slot occupancy in real time and provide this information remotely to users. Using IoT technology, such a system can detect available spaces, guide drivers efficiently, and reduce the time spent in searching for parking. This will improve traffic flow, save fuel, reduce emissions, and enhance the overall parking experience for users.

1.3 Aim and Objective

To design and develop an IoT-based automatic car parking system that efficiently monitors parking slot availability in real time and guides drivers to vacant spaces, thereby reducing parking time, traffic congestion, and fuel consumption.

- To implement IR sensors for accurate detection of vehicle presence in each parking slot.
- To use the ESP32 microcontroller for processing sensor data and enabling wireless communication.
- To develop a web or mobile application that displays real-time parking slot availability to users remotely.

- To integrate LED indicators for visual guidance of slot occupancy

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 IOT DEVICES

1. **ESP32 Microcontroller**

2 .IR Sensor

3. **LED**
4. **Breadboard**
5. **Power Supply**
6. **Wi-Fi Module**

2.2 SYSTEM HARDWARE SPECIFICATIONS

Component	Specification
Processor	Intel i5 11th Gen (Minimum)
Memory Size	8 GB (Minimum)
Hard Disk (HDD/SSD)	40 GB (Minimum)
ESP 32	MICROCONTROLLER

2.3 SOFTWARE SPECIFICATIONS

Operating System	Windows 11
Browser	Google Chrome
IDE	Arduino IDE

CHAPTER 3

SYSTEM DESIGN

3.1 ARCHITECTURE DIAGRAM

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components

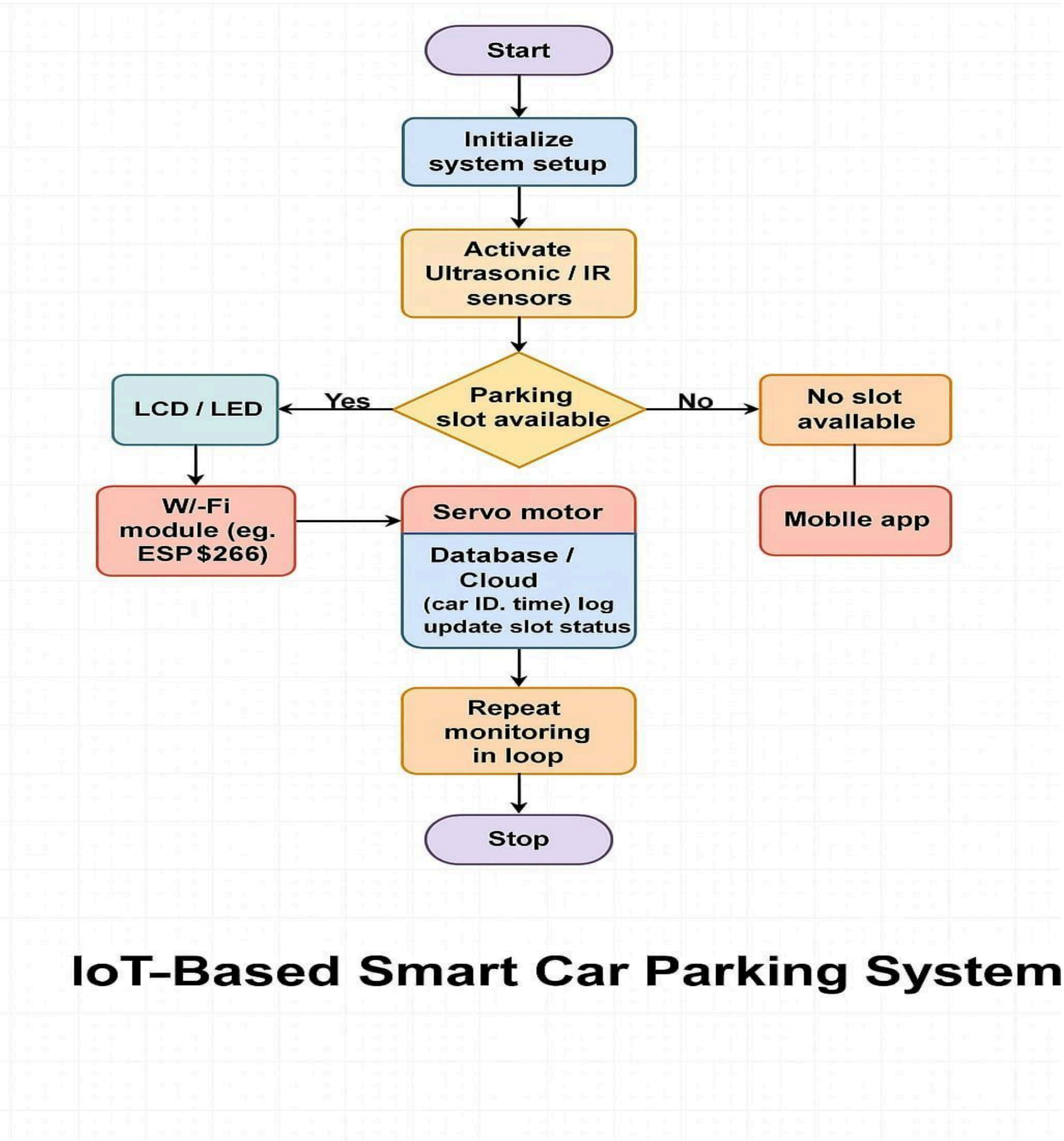

Figure 3.1 Architecture Diagram

From the above Figure 3.1, the architecture of the system is well understood.

3.2 USE CASE DIAGRAM

The use case diagram shows interactions between users and the automatic car parking system. The main actors are the Driver and System Admin. Drivers check parking slot availability through a mobile or web app. The system detects slot occupancy using IR sensors and updates the status. LED indicators guide drivers to free slots. The admin monitors and maintains the system. This diagram helps visualize key system functions and user roles.

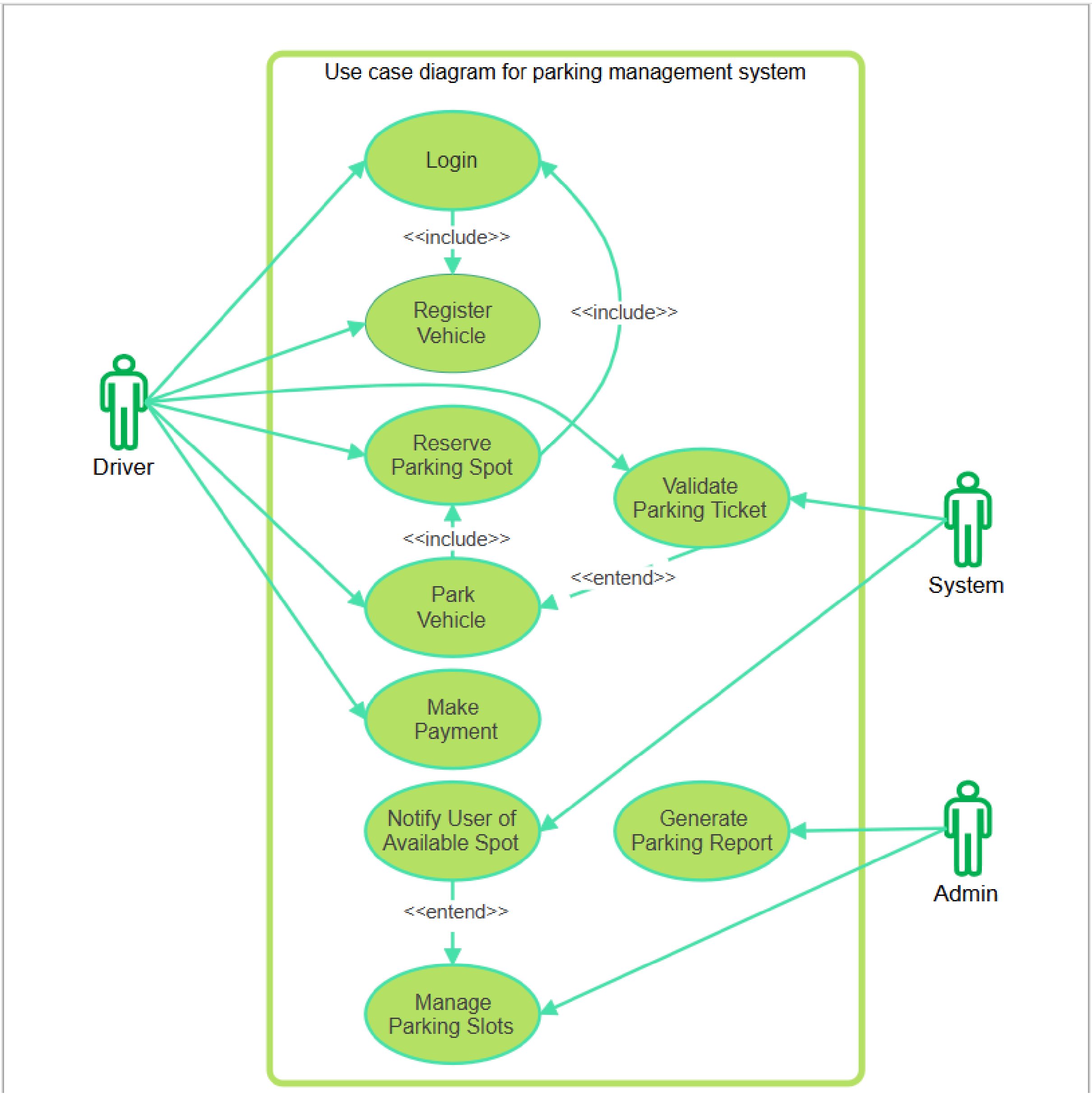

Figure 3.2 Use case diagram

3.3 ACTIVITY DIAGRAM

The activity of the IoT-Based Automatic Car Parking System begins with the IR sensors detecting the presence or absence of vehicles in each parking slot. This data is processed by the ESP32 microcontroller, which updates the slot status accordingly. The information is then transmitted via Wi-Fi to a mobile or web application, allowing users to view real-time slot availability

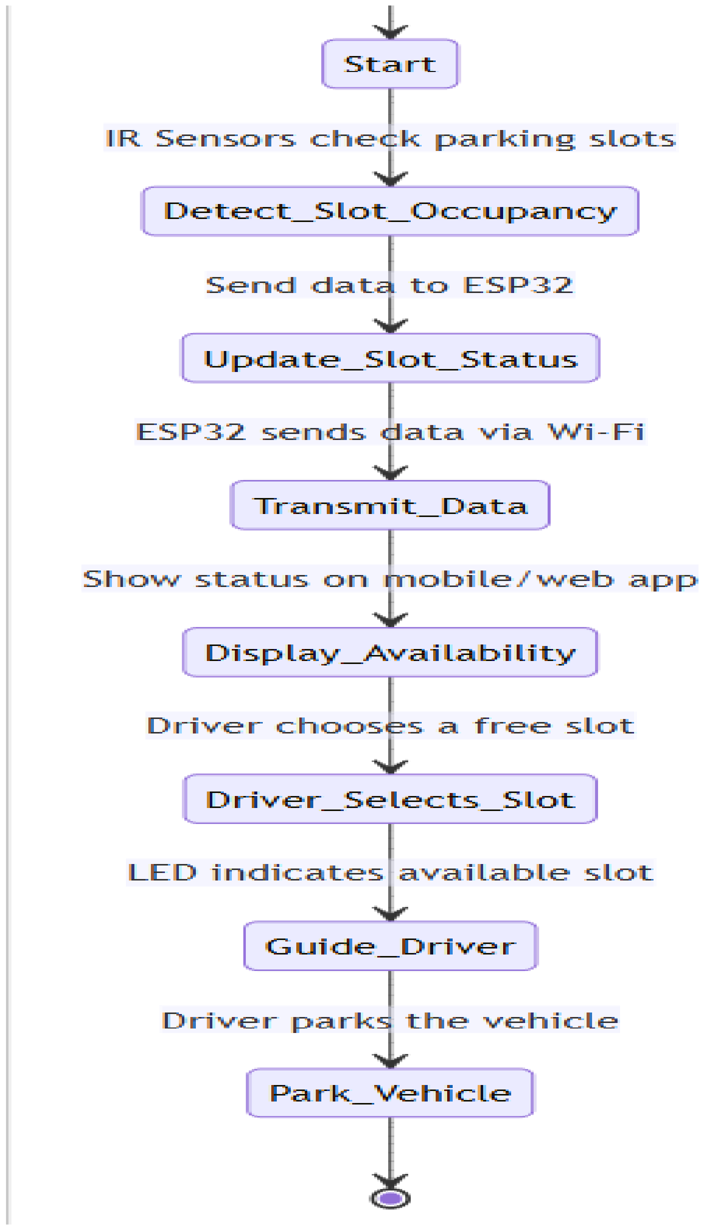

Figure 3.3 Activity Diagram

3.4 CLASS DIAGRAM

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

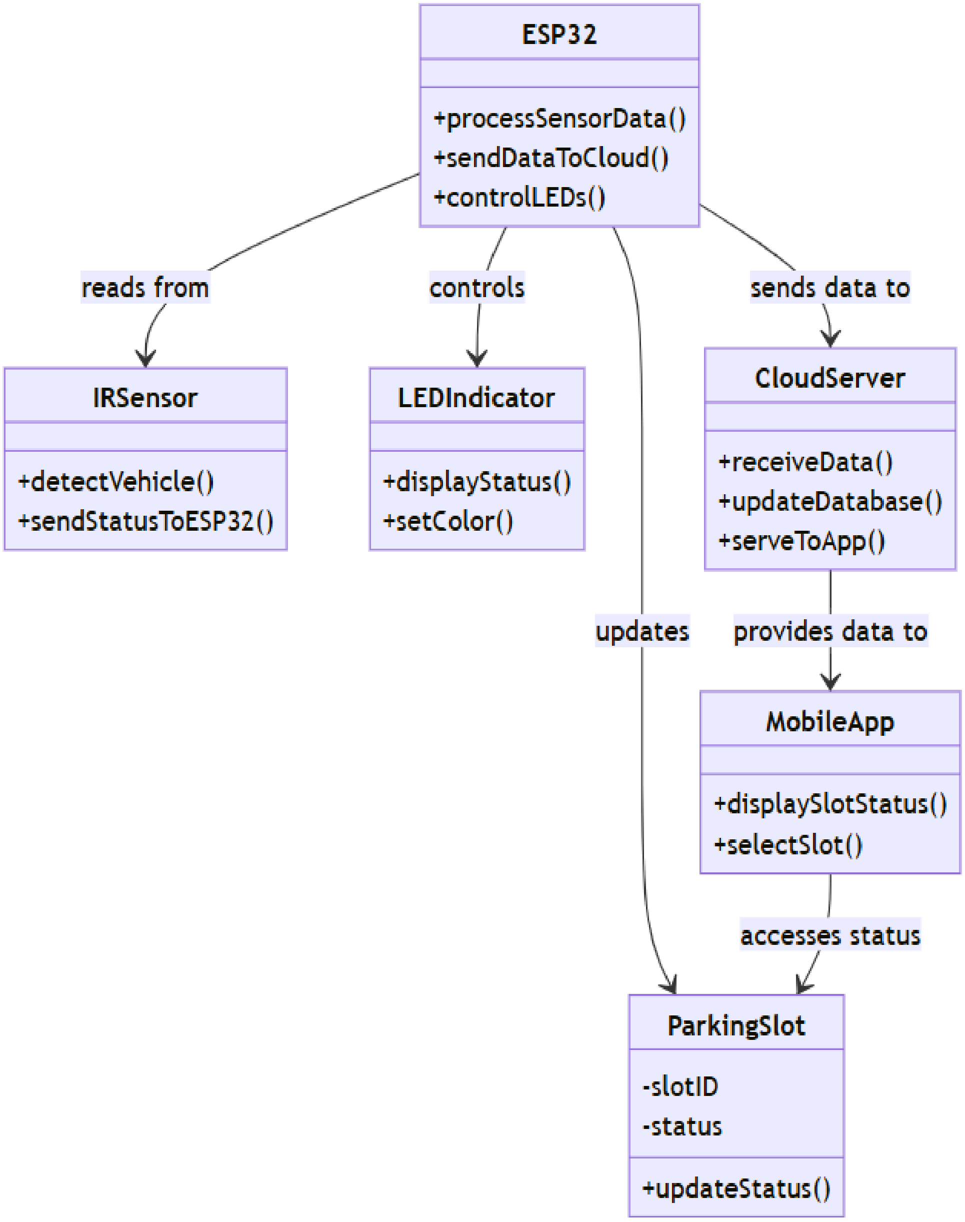

Figure 3.4 Class Diagram

CHAPTER 4

MODULE DESCRIPTION

The system consists of several key modules working together to automate parking. The sensor module uses IR sensors to detect whether a parking slot is occupied. The control module, managed by the ESP32, processes this data and updates the LED indicators to show slot availability. The communication module sends real-time data to a mobile or web application using Wi-Fi. Finally, the user interface module allows drivers to view available slots and park efficiently.

4.1 HARDWARE MODULE

The hardware module of the IoT-Based Automatic Car Parking System includes essential electronic components that work together to detect and display parking slot status. The core of the system is the ESP32 microcontroller, which processes input data and controls communication. IR sensors are used to detect the presence or absence of vehicles in each parking slot. Based on sensor input, LED indicators (typically red and green) display the occupancy status of each slot to guide drivers visually. All components are connected using a breadboard for prototyping, and powered through a stable power supply module. The ESP32's built-in Wi-Fi module enables real-time data transmission to cloud platforms or mobile applications, forming the backbone of the smart parking system. This hardware setup ensures efficient monitoring, fast processing, and clear communication between the system and users.

4.2 DATA COLLECTION AND PROCESSING MODULE

The Data Collection and Processing Module is responsible for gathering real-time information about the status of each parking slot and processing it for further actions. IR sensors are deployed in each parking slot to detect the presence or absence of a vehicle. This raw sensor data is then transmitted to the ESP32 microcontroller, which serves as the system's processing unit. The ESP32 reads inputs from all sensors, interprets the data to determine which slots are free or occupied, and updates this information in real time. It also controls the LED indicators accordingly—turning green for available and red for occupied slots. Additionally, the processed data is sent via the ESP32's Wi-Fi module to a cloud server or directly to a user interface, enabling remote monitoring through a mobile or web application. This module ensures accurate data handling, fast response, and seamless communication within the system. It reduces human effort and error by automating slot detection and status updates. The system also allows for scalability by adding more sensors and slots as needed. Overall, this module forms the intelligent core of the parking system and enhances user convenience through smart technology

4.3 ALERTING MODULE

The Alerting Module is designed to notify users and administrators about important parking events and system statuses. When the system detects that all parking slots are occupied, it can send an alert to users via the mobile or web application, informing them that no spaces are currently available. Additionally, if any sensor malfunctions or communication errors occur, the system generates alerts to notify the administrator for prompt maintenance. The ESP32 microcontroller plays a crucial role in monitoring these conditions and triggering alerts through Wi-Fi connectivity. Alerts can be sent as push notifications, SMS, or email, ensuring timely information delivery. This module improves user experience by preventing unnecessary waiting and enhances system reliability by enabling quick responses to technical issues.

COMPONENTS MODULE

1. Sensor Module

This module uses **IR sensors** to detect the presence or absence of vehicles in each parking slot. It continuously monitors each slot and sends status data to the microcontroller. Each sensor is assigned to a specific slot.

2. Control Module (ESP32 Microcontroller)

The ESP32 collects data from the IR sensors and processes it. It decides which slots are occupied or available and sends this data to the cloud or app through its built-in Wi-Fi. It also controls the LED indicators based on the slot status.

3. LED Indicator Module

This module helps guide drivers visually by showing the status of each parking slot. A **green LED** indicates that the slot is available, while a **red LED** means the slot is occupied.

4. Communication Module

Using the ESP32's Wi-Fi capability, this module transmits real-time slot data to a cloud server or directly to a web/mobile app interface. This enables remote monitoring of parking availability.

5. User Interface Module

This module allows drivers to view available slots through a **mobile or web application**. It displays real-time slot status, helping users quickly find and navigate to free parking spaces.

4.4 WEB APPLICATION MODULE

The **Web Application Module** provides users with a user-friendly interface to access real-time parking information remotely. It displays the current status of parking slots—showing which are occupied and which are available—using data received from the ESP32 via the cloud server. Users can view slot availability, select parking spots, and receive alerts about slot status or parking conditions. The application also allows system administrators to monitor overall system health, sensor statuses, and manage settings. Built using web technologies like HTML, CSS, and JavaScript, the module communicates with backend servers through APIs to fetch and update parking data. This module enhances user convenience by offering easy access to parking information anytime and anywhere. It supports multiple user roles with different access levels to ensure security. The interface is designed to be responsive, supporting various devices such as smartphones, tablets, and desktops. Real-time updates reduce delays in information, improving user experience. The module also logs historical parking data for analytics and system optimization. Overall, it acts as a critical link between the hardware system and the end-users

4.5 INTEGRATION MODULE

The **Integration Module** serves as the central unit that connects and coordinates all other modules within the automatic car parking system. It ensures seamless communication between the hardware components—such as IR sensors, ESP32 microcontroller, and LED indicators—and the software elements like the cloud server and web application. This module manages data flow, ensuring that sensor inputs are accurately processed, status updates are sent to the user interface, and alerts are triggered when necessary. It handles communication protocols, including Wi-Fi connectivity and API interactions, to synchronize real-time parking data across all platforms. By integrating the physical devices with cloud services and user interfaces, this module enables smooth operation and enhances the overall efficiency and reliability of the parking system. The module also facilitates scalability, allowing additional sensors or devices to be easily added without disrupting existing operations. It performs error handling to maintain data consistency and prevent system failures. Furthermore, it supports security measures such as data encryption and authentication to protect sensitive information. This integration layer acts as the backbone that ties together all system components into a cohesive smart parking solution

CHAPTER 5

SAMPLE CODING

PROJECT CODE:

```
#define BLYNK_TEMPLATE_ID "TMPL37d1-Xx33"
#define BLYNK_TEMPLATE_NAME "parking2"
#define BLYNK_AUTH_TOKEN
"JYMB5gYoWSaP8w6lgwXBqrgZK2LmPBEq"

#include<Wire.h>
#include<WiFiClient.h>
#include<BlynkSimpleEsp32.h>
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0*27, 16, 2);
#define SENSOR1 32
#define SENSOR2 33
#define SENSOR3 25
#define SENSOR4 26

// Blynk Timer
BlynkTimer timer;

// WiFi Credentials
char auth[] = "JYMB5gYoWSaP8w6lgwXBqrgZK2LmPBEq";
char ssid[] = "ragul";     // Your WiFi name
char pass[] = "62226222";    // Your WiFi password

void setup() {
  Serial.begin(115200);
```

```
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print( "Welcome To");
  lcd.setCursor(0, 1);
  lcd.print("JustDoElectronic");
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  pinMode(SENSOR1, INPUT_PULLUP);
  pinMode(SENSOR2, INPUT_PULLUP);
  pinMode(SENSOR3, INPUT_PULLUP);
  pinMode(SENSOR4, INPUT_PULLUP);
}
void sensor() {

  int L1 = digitalRead(SENSOR1);
  int L2 = digitalRead(SENSOR2);
  int L3 = digitalRead(SENSOR3);
  int L4 = digitalRead(SENSOR4);

  if (L1 == 1){
    Serial.println(" IR Sensor 1 detected");
    lcd.setCursor(0, 0);
    lcd.print("1:Full.");
    WidgetLED LED1(V0);
    LED1.on();



  } else {
    Serial.println(" 1 === ALL clear");
```

```
  lcd.setCursor(0, 0);
  lcd.print("1:Empty");
  WidgetLED LED1(V0);
  LED1.off();
}
if (L2 == 1) {
  Serial.println(" IR Sensor 2 detected");
  lcd.setCursor(9, 0);
  lcd.print("2:Full.");
  WidgetLED LED2(V1);
  LED2.on();
}else {
  Serial.println(" 2 === ALL clear");
  lcd.setCursor(9, 0);
  lcd.print("2:Empty");
  WidgetLED LED2(V1);
  LED2.off();
}
if (L3 == 1) {
  Serial.println(" IR Sensor 3 detected");
  lcd.setCursor(0, 1);
  lcd.print("3:Full.");
  WidgetLED LED3(V2);
  LED3.on();
}else {
  Serial.println(" 3 === ALL clear");
  lcd.setCursor(0, 1);
  lcd.print("3:Empty");
```

```
  WidgetLED LED3(V2);
  LED3.off();
  }
  {
   if (L4 == 1) {
  Serial.println(" IR Sensor 4 detected");
  lcd.setCursor(9, 1);
  lcd.print("4:Full.");
  WidgetLED LED4(V3);
  LED4.on();
 }else {
  Serial.println(" 4 === ALL clear");
  lcd.setCursor(9, 1);
  lcd.print("4:Empty");
  WidgetLED LED4(V3);
  LED4.off();
  }
 }
}

 void loop() {
  sensor();
  Blynk.run();
  delay(200);
 }
```

CHAPTER 6

SCREEN SHOTS

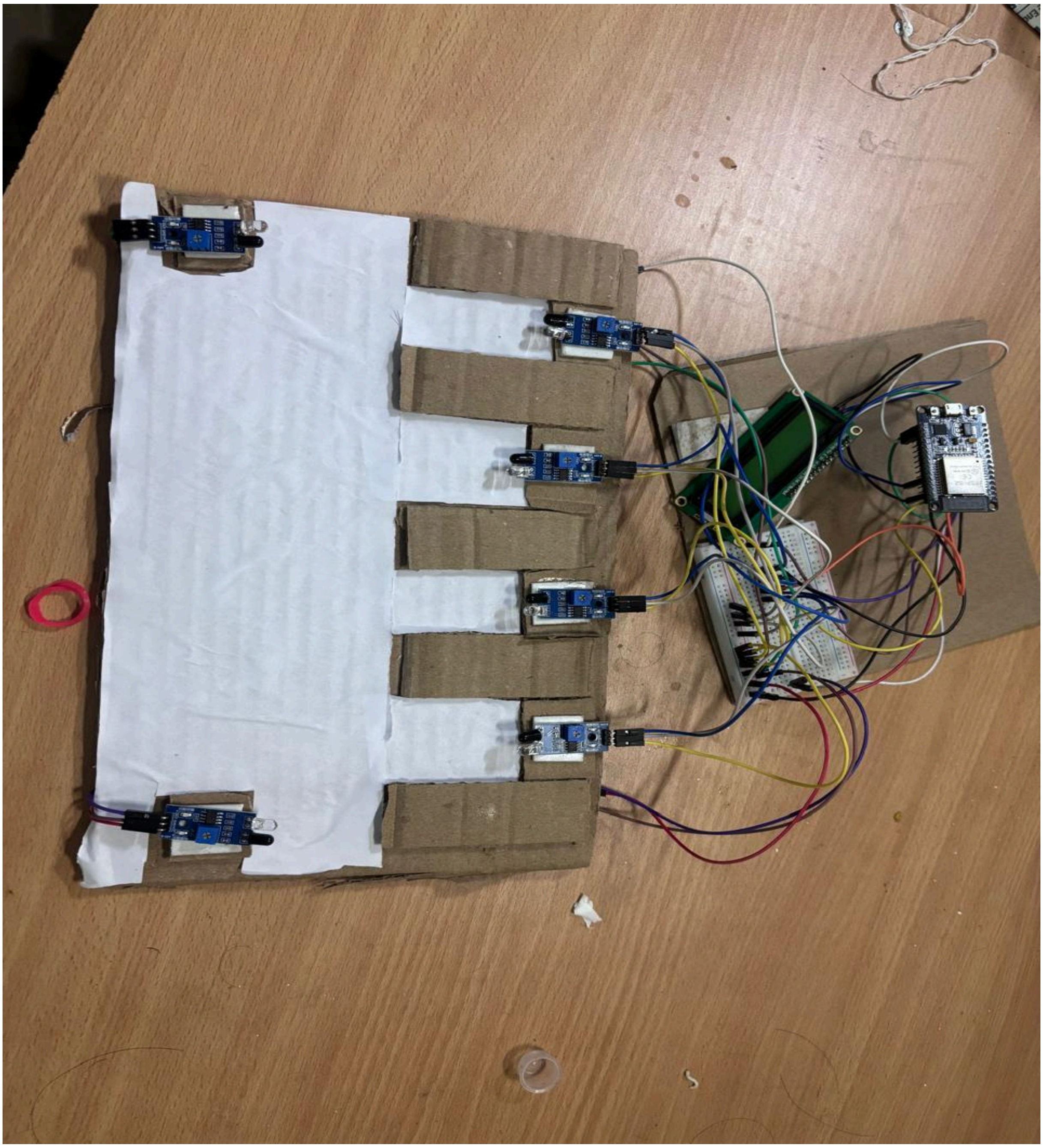

Figure 6.1

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

The IoT-Based Automatic Car Parking System is a smart and efficient solution to address the growing challenge of parking management in urban environments. By combining IR sensors, an ESP32 microcontroller, LED indicators, and a web-based interface, the system accurately detects vehicle presence and displays real-time parking slot availability. This significantly reduces the time spent searching for parking, minimizes traffic congestion, and enhances user convenience. The integration of hardware and software ensures reliable communication and smooth operation across all modules. Through automation and real-time monitoring, the system eliminates the need for manual intervention, making it cost-effective and user-friendly. It also demonstrates the potential of IoT technology in developing smart city infrastructure. The modular nature of the system allows for easy scalability and future enhancements. Overall, this project provides a practical, innovative approach to urban parking issues and lays the foundation for more advanced smart parking solutions in the future.

The project promotes digital transformation in public services and showcases how simple components can be used to create impactful smart solutions. It is also energy-efficient and suitable for both small-scale and large-scale parking setups. This system can be further developed to support automated billing, advanced security, and integration with navigation systems. Overall, it offers a sustainable and scalable approach to solving modern-day parking issues and contributes to smarter, more efficient cities.

Future Enhancement:

Several enhancements can further improve the system:

- Integrate **automatic gate barriers** that open only when a slot is available.
- Use **ultrasonic or camera-based sensors** for improved vehicle detection accuracy.
- Develop a **mobile application** with real-time notifications and navigation support.
- Add a **payment gateway** for advance slot booking and cashless transactions.
- Implement **AI-based analytics** to predict parking demand and optimize space usage.
- Enable **voice guidance** and **accessibility features** for a better user experience.
- Expand the system to support **multi-level or large-scale parking structures**
- Implement **solar-powered components** to make the system energy-efficient and eco-friendly.
- Add **license plate recognition (LPR)** for automatic vehicle identification and security.
- Include **SMS/email alert integration** for users without smartphones or apps.
- Develop a **dashboard for administrators** to view analytics, reports, and manage the system remotely.

8.REFERENCES

References

1. **K. Jain, A. Kumar, and M. Sharma, "IoT Based Smart Parking System," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2018, pp. 0233-0237.**
   https://ieeexplore.ieee.org/document/8524286

2. **P. Arulmozhivarman and R. Karthikeyan, "Smart Parking System Using IoT," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 262-266.**
   https://ieeexplore.ieee.org/document/8893538

3. **P. V. T. R. Lakshmi and M. P. Deepa, "IoT Based Smart Parking System Using ESP8266," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 0387-0391.**
   https://ieeexplore.ieee.org/document/9245536

4. **M. S. Chavan and R. D. More, "IoT based smart parking system," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2017, pp. 42-45.**
   https://ieeexplore.ieee.org/document/8058374

5. **S. Jayapriya and R. V. Renuka, "Automated Parking System Using IoT," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 392-396.**
   https://ieeexplore.ieee.org/document/937614