IoT Based Parking System:
"Prospects, Challenges, and Beyond"

by

Shafquat Mahmud Quader
21341045
Salma Hani Nova
18101546
Saumitra Das Talukdar
18101594
Md.Shaba Sayeed
17201147
Mubtasimur Rahman Sadab
18101388

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering (C.S.E)

Department of Computer Science and Engineering
Brac University

September 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Shafquat Mahmud Quader
21341045

Salma Hani Nova
18101546

Saumitra Das Talukdar
18101594

Md.Shaba Sayeed
17201147

Mubtasimur Rahman Sadab
18101388

Approval

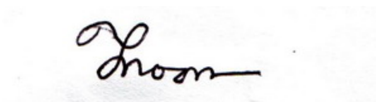
The thesis/project titled **IoT Based Parking System: "Prospects, Challenges, and Beyond"** submitted by

1. Shafquat Mahmud Quader(21341045)
2. Salma Hani Nova(18101546)
3. Saumitra Das Talukdar(18101594)
4. Md.Shaba Sayeed(17201147)
5. Mubtasimur Rahman Sadab(18101388)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on August, 2021.

Examining Committee:

Primary Supervisor:
(Member)



Ms. Jannatun Noor Mukta
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The world is getting populated over time and with that number of cars per space is increasing. Parking gets complicated for people over time. Over the development of the Internet of Things (IoT), many parking systems get proposed. By evaluating the previous works, our paper makes a comparative analysis based on user experience, hardware, and software systems. Finally, we propose a system, finding strengths and loopholes in the previous system that focuses on implementing parking systems with IoT. Our system identifies the driver who wrongly parks the vehicle and instructs the proper direction where to park the car.

Keywords:
Internet of Things (IoT), Cloud, A comparative analysis, Architecture

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our co-advisor Ms. Jannatun Noor Mukta ma'am for her kind support and advice in our work. She helped us whenever we needed help.
Thirdly, A. B. M. Alim Al Islam (Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology) sir's review helped us a lot in our later works.
And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AC Alternating-Current

ADC Analog to Digital Converter

API Application Programming Interface

APS Auxiliary Power System

AVR Alf-Egil Bogen Vegard Wollan

BaaS Business as a Service

C Connectivity

CCTV Closed-circuit television

CPMS Central parking management system

CPS Cyber-physical systems

DC Direct-Current

DIP Dual-Inline Package

E End Node

GP Global Positioning System

GPIO General-Purpose Input/Output

GUI Graphical User Interface

I/O Input/Output

IC Integrated Circuit

IDE Integrated Development Environment

IEEE the Institute of Electrical and Electronics Engineers

iOS iPhone Operating System

IoT Internet of Things

IP Internet Protocol

IR Infrared sensor

$JSON$ JavaScript Object Notation

LCD Liquid Crystal Display

LED Light Emitting Diode

$LMPS$ Localized Multi Path Selection

$LPMS$ Local parking management system

MCU Micro Controller Unit

$MISO$ Master In Slave Out

$MOSI$ Master In Slave In

OCR Optical Character Recognition

OpenCV Open Source Computer Vision Library

P Processing Node

PC Personal Computer

$PLMS$ Parking IoT monitoring system

PWM Pulse Width Modulation

RC Remote Control

$RFID$ Radio Frequency Identification

RPM Remote Patient Monitoring

$RTOS$ Real Time Operating System

SCK Serial Clock

SCL serial Clock Pin

SDK Software Development Kit

SMS Short Message Service

SPI Serial Peripheral Interface

SQL Structured Query Language

SS Slave Select

TCP Transmission Control Protocol

TTL Transistor–Transistor Logic

$UART$ Universal Asynchronous Reception and Transmission

$WiFi$ Wireless Fidelity

$WLAN$ Wireless Local Area Network

Chapter 1

Introduction

1.1 Thoughts Behind Parking System

Searching for a proper parking system is very time-consuming in the metropolitan area. It results in a traffic jam. Mostly it is seen not to park in the proper place. They park in the wrong area. As a result, it is necessary to construct a parking system. Furthermore, while implementing the parking system, a very complex setup is required, which is costly and not feasible for overpopulated and developing countries. Reviewing previous works, we could understand that there has been no feasible proposed architecture for parking systems for developing countries [51], [24], [48], [45], [27].

1.2 History of IoT Based Parking System

In 1905, Paris, France, developed an automated vehicle parking system. It was shaped like a Ferris wheel and could accommodate eight vehicles. In the 1920s, a paternoster structure, a Ferris wheel-style APS (for vehicles instead of people), gained popularity because it could stop eight vehicles in the same amount of ground space as a conventional two-vehicle APS, [21]. Mechanically simplistic with a hint of style, the paternoster was easy to use in various locations, including inside structures. Simultaneously, Kent Programmed Carports began offering APS with capacities exceeding 1,000 vehicles. The Auto Stacker, which debuted in 1961 at Woolwich, Southeast London, proved difficult to operate. Interest in APS was reignited in the 1990s in the United States, and by 2012, there were 25 significant currents and joint APS projects (representing about 6,000 stopping spots). While interest in APS remained high in the United States much further into the 1990s, by the 1970s, Europe, Asia, and Central America had adopted more sophisticated APS. In 2012, Japan had 1.6 million APS parking spots, according to reports, [21].

1.3 Motives and Objectives

The primary goal of our research topic is to develop a part of the parking system through the Internet of things (IoT) which will help build a smart city. Generally, IoT devices work as sensors to collect data by sensing and sending output for further processing. Through this research, we will learn about IoT fields and the importance of this field in science. The following section contains the primary research goals:

- To gain a huge and deep knowledge about IoT.
- To show the importance of IoT-related topics.
- To perceive the features of the IoT working process and functionalities.
- To understand how IoT devices can work to develop smart parking systems.
- To develop a smart parking system for avoiding traffic jams.

1.4 Problem Statement

Billions of people are now utilizing the Internet of Things globally to address the global problems we confront daily. Disease, education, the technological sector, IoT has spread their wings for saving humankind in every possible way. The Internet of Things has projected to reach 35 billion devices by 2025. As additional users do introduce, the vulnerability of this technology gets called into question. With the exponential growth of devices, it is critical to address these vulnerabilities and improve their security. Implementation of IoT should not revise to such an extent that it should not use for any wrongdoings. The problem we have mentioned below can partially resolve by bringing IoT, which shows in the study, [47].

Beating traffic is something people have been trying to do for years now. Traffic congestion is the primary concern in our daily life. Scrutinizing into the matter, we find that parking in the illegal road area is the central issue for this specific problem. However, it does seem that people need parking spots for their workplace, home, and everywhere. Often it is seen that multiple vehicles need the parking spot in the same area. Unfortunately, it does seem that they do not have a definite parking spot in any nearby area. Eventually, people end up parking on the roads, which ultimately initiates traffic congestion.

Then, there is high mismanagement in the available parking spots. The facilities in those parking spots are not enough to provide hassle-free parking for the users. As there is a lack of inspection about whether the vehicle is perfectly parked or not, drivers park their vehicle as they wish to park. As a result, the parking spaces are not using adequately.

Drivers tend to park vehicles in places they are not supposed to park. In some cases, they block roads and the pavement, which causes a chaotic environment in the entire area. Blocking pavement forces pedestrians to walk on the road. Therefore, road accidents are happening now and then.

Moreover, haphazard roadside parking is also creating trouble for ordinary people. When parking space is not available, drivers park their vehicle in nearby areas without any restriction that causes long queues of vehicles. This problem can consider as the sole reason for traffic congestion.

When a driver finds a specific place to park their vehicles, it is more time-consuming to reach that specific parking spot due to traffic congestion. This traffic jam problem can lengthen the trip times and increase vehicular queuing. Then again, it is also time-consuming to search for another parking spot and to go there. Eventually, the

problem gets bigger and broader with vehicles to add to the road, which shows in the study, [25].
In the upcoming days, it will be more challenging to deal with this mentioned problem. IoT came as a blessing in our life to face such challenges. Implementation of IoT in this sector can bring a long-lasting change.

1.5 Our Contributions

The previous studies do implement using high tools. However, the tools were not friendly in developing countries. For example, in Bangladesh, where many people cannot read English, implementing a project using RFID is not feasible. Moreover, using high-tech detection procedures is costlier for every vehicle and also challenging to implement. As we know, developing countries tend to spend a meagre amount of money in their Research and Development sector, so we propose an architecture using IoT that can implement using minimal resources. We make the following contributions to this study based on our research:

- We present a comparative analysis of IoT-based related to the smart parking system.
- We find strengths and weaknesses of the different IoT-based parking systems for the ease of exploring for future researchers.
- We propose a theoretical approach of a very cost-efficient and feasible IoT-based intelligent parking system for developing countries.

Chapter 2

Background

2.1 Internet of Things (IoT)

In the 1980s and 1990s, the attachment of Sensors and intelligence was considered (furthermore a few others possibly far older predecessors), but the progress was slow since the technology was not accessible at the time, except a few early attempts like an internet-connected vending machine. Things could not interact because the chips were too heavy and complicated, [9]. Figure 2.1 refers to the example fields of IoT.

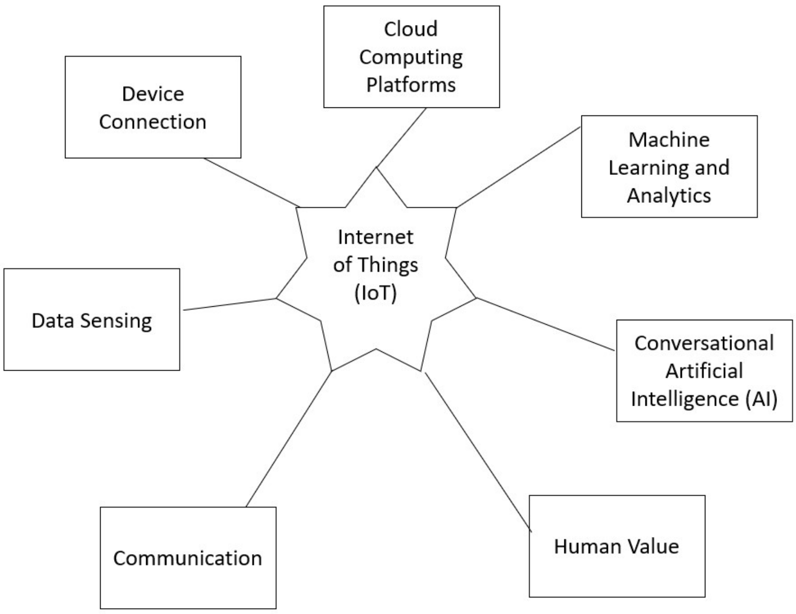


Figure 2.1: Internet of Things

Developing low-cost, power-efficient CPUs was necessary before it became possible to make billions of gadgets talk to one another. RFID tags are being utilized, which are low-power semiconductors that are wirelessly communicative, and increasing access to broadband internet and cellular and wireless networking have contributed to alleviating some of these issues, [9], [10]. In necessary for the Internet of Things to expand, IPv6 needed to be adopted. This would, among other things, provide enough IP addresses for all devices required by the planet (or possibly the whole galaxy), [7], [10]. Anything to which an Internet Protocol (IP) address may be assigned, and transfer data over The Internet of Things may include a network if it meets certain requirements. Examples include cardiac monitoring implants, farm animals with biochip transponders, and vehicles with Tire pressure sensors integrated into the vehicle, alerting the driver when the pressure drops, [7], [23].

With the Internet of Things, gadgets and objects have sensors built-in, which integrate data from many sources and use analytics to convey the most relevant information with applications customized to specific needs, [7], [23]. These cutting-edge Internet of Things (IoT) technologies can identify which data is important and which may be safely ignored. With this information, patterns may be discovered that can be utilized to offer recommendations or to anticipate issues before they occur, [9].

Nearly whatever can be operated or interacted with over the internet has the potential to be turned into an IoT device, [23]. An Internet of Things (IoT) gadget can be operated through a smartphone application, [7]. It can be a smart thermostat, a motion sensor, a streetlight, or a lightbulb. An IoT device may be as whimsical as a plush animal or as serious as a self-driving vehicle. A portion of the Internet of Things (IoT) includes a jet engine equipped with hundreds of sensors that gather and send data to ensure the system operates optimally[10], [23].

2.2 Image Processing

Image processing is a method for improving images or extracting valuable information from them. Image processing is a subcategory of signal analysis in which an image is utilized as an input, as opposed to other types of signal processing, and the result may either be the image itself or a collection of associated characteristics/features. In picture processing, converting an image to a digital format and then executing operations to improve the image and extract valuable information is described.

Various procedures are used to images in order to enhance them or extract useful information from them. Mathematics defines image processing as the computer's manipulation of a two-dimensional image, that is, an image described as a function of two actual variables, such as the function t(x,y), with a magnitude equal to the intensity of an image at a particular point of coordinates (a, b).

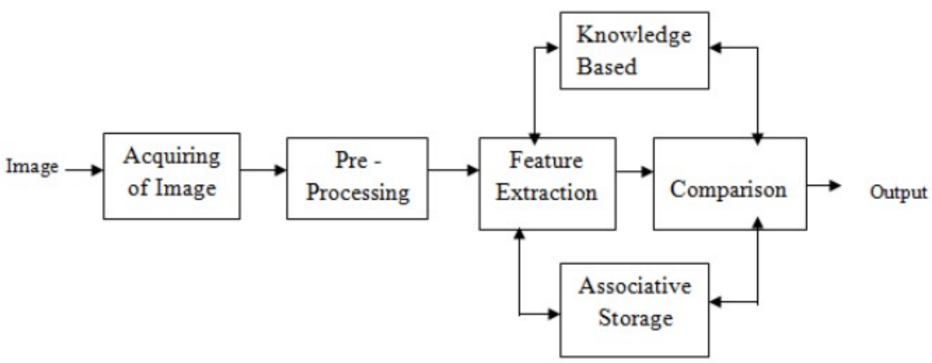


Figure 2.2: Digital Image Processing retrive from [30]

Image processing can result in either a single image or a collection of traits or attributes linked to one another. This can be classified into two categories: digital picture processing and analogue image processing, [30]. the figure 2.2 refers the digital Image Processing. Analogue image processing methods such as printing and photography may produce tangible copies such as prints and photographs. When using these visual methods and procedures, image analysts rely on a variety of interpretative foundations. Through the use of a computer, modern image processing methods enable the modification of digital images, [4].

In order to extract information from digital data, all kinds of data must go through three basic phases: pre-processing, augmentation, and presentation, as well as information extraction (if utilizing digital techniques). Those are the three major stages that all kinds of data must go through when processed via digital methods, [30]. If one wants others to replicate and validate one's findings, image processing must be conducted correctly. This process includes recording and reporting on processing stages and applying similar treatments to suitable control images to demonstrate comparability, [4].

Essentially, image processing consists of three steps: capture, manipulation, and display. In order to import the photo, image capture tools must be used. After completing an analysis and modification of the image, The result may be a modified image or a report based on the analysis of images, [4]. There are numerous applications for digital image processing that may be found in several fields, and many examples include colour processing, transmission and encoding, microscopy, machine/robot vision, and remote sensing. A large percentage of image processing techniques consider a picture to be a two-dimensional signal capable of being handled using standard signal processing methods.

The objectives of this operation may be broken down into three categories. Starting with image processing, where the input is a picture, and the output is another picture, and then there is image analysis, where the input is a picture, and the output is a measurement or dimension; and then there is computer vision, which takes pictures and converts them into text. Finally, image comprehension is a procedure in which a picture is sent into a computer, with the output being a standard description of what was seen, [4].

Image processing is utilized in various research and engineering applications, including artificial intelligence and remote sensing. Face recognition and forecasting, item identification, fingerprint detection, sorting, argument reality, microscopic imaging, the mechanism for lane departure warnings, non-photorealistic representation, medical imaging, and morphological imaging are possible applications. Image processing is one of the technologies that is now undergoing the most rapid development. It is also a significant subject of study in the engineering and computer science fields, among other disciplines, [4].

Chapter 3

Literature Review

In the modern-day, the Internet has altered our way of life. The introduction of the "Internet" to humankind in 1983 revolutionized the way people viewed the world. There are over 4.33 billion active IoT users on the planet. The Internet of Things (IoT) refers to a collection of gadgets such as smartphones, televisions, laptops, and other electronic devices that are all linked and capable of generating and sharing data via the Internet, according to study, [49].

The human race has advanced tremendously in terms of interrelationships between the physical and digital domains in the previous decade, giving rise to the "Internet of Things". The number of linked items that may participate in the IoT ecosystem does anticipate to grow exponentially to 5.8 billion by 2020. In the future, the IoT paradigm is predicted to link and affect everything from persons, groups, artefacts, products, data, and services as shown in the study, [44].

According to a study, [34], finding the availability of a parking spot is effortless with the help of low-cost cloud computing. Furthermore, the saved data may be retrieved and viewed from anywhere using conventional AIPS in this procedure. However, a smartphone is required to carry out this procedure. Furthermore, a certain parking period must choose, and the price for the application must pay. As a result, most individuals may be hesitant to utilize this intelligent facility because of the service price.

Image processing can determine the angle at which the vehicle is parked, according to study, [50]. This image processing is when the vehicle's incorrect parking angle does identify. Through the camera, they can detect the vehicle's license plate. In this procedure, the driver must have a smartphone. Someone violating the rules might circumvent the system by refusing to install the program on his computer.

According to the study, [21], the cloud vision API technique assisted the system in solving the training data set, resulting in vehicle registration plate accuracy. When one vehicle gets stolen from a parking space, the license plate gets scanned and translated to text, and information may be used to court to identify the vehicle and driver. This identity helps to improve the security of the parking system and the surrounding region.

According to a study, [50], a fingerprint-based prototype was incorporated into the design to prevent unauthorized entry to the vehicle. This fingerprint-based prototype guarantees that only the vehicle's original owner will be able to start it. It is impossible to duplicate a fingerprint since each individual is unique. On the contrary, some potential users are unable to join this program. Furthermore, in certain situations, capturing complete and accurate fingerprints is challenging.

3.1 Internet of Things(IoT)

Kevin Ashton was the first to use the Internet of Things, initially launched in 1999 at auto-ID centres. This latest technology promises to connect all of our surrounding objects to a network and allow them to communicate with one another with little human intervention. The Internet of things is still in its early stages, and there is no standard architecture in place as the study of [47].

When we talk about the Internet of Things, we are talking about situations where network connection and computational power are extended to non-computer objects, sensors, and ordinary things, enabling these devices to produce, trade, and consume data with little to no human involvement. Even though research, [44] shows that there is no one universal definition.

In a nutshell, IoT can control its whole from anywhere at any time. The Internet of Objects (IoT) is research devoted to determining how things function with just one point of contact. End Node(E), Processing Node(P), Connectivity(C), and Application (Physical devices that are linked to a network and may interact with one another in the Internet of Things) are the four building elements that make up the Internet of Things. In IoT processes and services, interconnected network devices and physical objects do link to data management, organization, collection, and analysis, according to study, [49].

A considerable number of parking attendants would suggest a significant financial outlay, which would be a significant problem for a parking system. According to the researchers of the study, [22], using IoT to enhance parking system capabilities might help reduce or eliminate parking fines.

The Internet of Things was created for remote tracking, management, and monitoring communication equipment through computers connected to the Internet. The Internet and Things are the two foundations of the IoT strategy. The Internet is a massive network of connected servers, computers, mobile devices, and other devices. Things are above problems in parking systems. The three primary components of their proposed IoT-based parking infrastructure are a parking control framework, a parking user application, and IoT platforms as the study of [35].

3.2 Image Processing

Image processing is a broad phrase in today's world that refers to various processing or enhancing photographs. In several definitions, this phrase refers to mathematical operations or algorithms used to process images. It is an example of visual signal processing. The input is a picture, and the output may be the image itself or a collection of its characteristics or qualities.

According to a study, [20], Image processing has been identified as the best fit for this suggested system. By evaluating video images and determining the differences between subsequent frames, video image processing identifies automobiles. It can also monitor several lanes, zones, or lanes at a cheaper maintenance cost than conventional sensors. Aside from that, video image processing is typically a cost-effective approach that is simple to install and modify for future usage, and it covers a wide range of detection zones that are necessary inside the camera's field of vision.

A complete whole image processing system is being developed, according to study, [20]. MAT LAB software does use to create image processing software that uses the template matching approach. All of the character templates will be defined using a test point. Those templates will compare to the figure who desires to identify. The character in that image will be the template with the highest matching point. The technology is capable of automatically identifying license plate numbers. Following identification, the plate number will compare to a database of plate numbers.

3.3 IoT Architecture

Any physical object that can connect to the Internet and identify itself among other devices is considered part of the Internet of Things (IoT) [48], [13], [1]. Other capabilities such as data exchange and remote control are also possible. A wide range of devices and equipment are constantly active in local and global networks through embedded sensors, according to the study, [52], including machines, sensors, electronics, collections, as well as humans [26]. Moving on to Smart City, Smart Parking for Citizens is an excellent example of how we can effectively and efficiently use the Internet of Things (IoT) in our daily lives to provide different services to different users, [12]. The intelligent parking system is easy to use and can also use by non-technical people [26]. With this system, users can use free parking spaces anywhere in the world.

In the study, [49], study, [22], and study, [35], provide Wi-Fi-based module architecture while in the study, [22], made special use of RFID tags of study, [44] made the use of the Multi-Agent system in its architecture. In the study, [20], made the use of image processing in its architecture. It made use of the A* algorithm to find the shortest path for finding the vacant space. It makes the use of Image Processing to find the vehicle. It uses CCTV and RFID tags for proper security surveillance [36], [33], [32]. A study of paper, [47], made use of a mobile-based architecture system. The study, [35], uses parking meters to find parking spots in the city. Furthermore, it uses Webcams and Image Processing systems. By using Automated License Plate Algorithm, it finds out if the vehicle does wrongly placed.

3.4 Cloud Server

In study, [34], makes use of Cloud-based architecture while in study, [22], study, [21], study, [34], study, [35] uses it for its privacy system for Data. The study, [21], stores all of its data in a firebase server. For the study, [49] data is available in the cloud server. The study, [34], uses a cloud-based error reducing system. Here, the user has access to an admin-based cloud database. Similarly, in the study, [35] the user has access to the Cloud. The study of paper, [47], uses an admin database control system. Here, the administrator has access to the cloud server.

3.5 Smart Parking

Parking happens to be expensive in the current era. For this, a new system is required. In the modern world, people demand more intelligent services for which smart parking is becoming more popular. A study [15] points out the availability of parking systems in real-time. This study provides a solution to park the vehicle in the urban area. Then again, the study [14] points out a secure and intelligent parking system. It finds the problem in the current standard parking system. The modern designs are not enough to fit so many vehicles in an area. Therefore, smart parking is required for the appropriate parking of vehicles. It provides a cost-effective parking system by making a graphical representation comparison between the conventional parking system and the intelligent parking system. Based on simulation time, the number of occupied slots, it makes a probability analysis of the parking systems. Based on that, they make a probable cost per parking system. They eventually prove that the smart parking system they proposed is cost-friendly.

Chapter 4

Comparative Analysis

Reviewing eight papers related to intelligent parking, architecture, data privacy, data ownership, data integrity, data availability, unauthorized data, processor, system application, and components of the papers.

Table 4.1: Theoretical Comparative Study - 01

Contents/ Papers	Data Privacy	Unauthorized Data	Data Ownership
Widyasari et.al [49].	Administrator based	Possibility of access to unauthorized data since have access to user credentials	Data is shared through the process of crowdsensing.
Patro et.al [22]	Cloud Based	No possibility of unauthorized data	Access to the cloud gives ownership.
LookMuang et.al [21]	Cloud Based	No possibility	Access to the cloud gives ownership.
Saarika et.al [34].	Cloud Based	No possibility	Access to the cloud gives ownership.
Pomaji et.al [47]	Administrator based	Possibility of unauthorized access to user credentials	Administrator has ownership of data.
Sadhukan et.al [35]	data not found	No possibility	User having access to cloud has ownership
Ali et.al [50]	Cloud based	Possibility if anyone has access to editing database	Administrator
Belkhala et.al [44]	Administrator based	No possibility	Administrator

Table 4.1 shows the comparison of our studies for data privacy, unauthorized data, and data ownership.

4.1 Data Privacy

The study, [49], does not give a brief explanation about the data privacy of the system. After logging in, an application uses the parking management system with login, registration, forget the password, and other features. A Crowd-sensing feature introduces for data accuracy. The Application itself has a particular account role, such as an administrator will be able to read and edit account management. Afterwards, according to a study, [22], an IoT cloud is used as one of the components to implement the parking system. In this study, no user's direct information does acquire as the system did not require personal information. Therefore the possibility of a data breach can be ignored. The information about the occupancy and non-occupancy of vehicles in the parking lot stores is in the central database. Then, in the study, [21], the authors used a cloud database for the overall data storage, maintenance, and other data-related activities. Cloud Firebase for recording user databases such as name and number plate information. Also, cloud vision API uses for image analysis. As the Data is cloud-based, the entire database is encrypted and challenging for any outsider to access. As stated in the study "Smart transportation system using IoT," a cloud server will maintain a database with parking slots and availability information. As in one of the previous studies, There has been no direct collection of user information, and privacy will be protected. Moreover, in the study, [47], three modules were introduced to the software platform of the vehicle parking system. The administrator module was one of the three models that work as an operative module of the Application. This module maintains all the necessary Data, booking price, user details, and others. The administrator can only make any modifications to the data. Besides, in the study, [47] proposed a system that uses an integrated component called a parking meter. The proposed model consists of four models, and one of the models is GUI (Graphical User Interface) which uses from the driver's end to reserve parking slots and payment options.

4.2 Unauthorized Data

According to the study, [49], Although the firebase server gets utilized, every piece of data becomes made private. Since the server has a user name and password, therefore unauthorized access can happen anytime. Also, if the server is not secured, it may release some info by unauthorized access. In the study, [22], unlike the previous study, there will be no server in this database. Therefore, unauthorized access cannot take place in the implemented work of this study. According to a study, [34], the database has been used in the cloud server. Therefore, unauthorized data cannot have any access to the database. This study also cannot have any unauthorized access like the previous study. However, this study uses a cloud system to save the database. According to a study, [35], unauthorized access is possible just like study, [49]. In the study, the unauthorized data can not access that due to too much information needed to log in, including driver authentication and vehicle number. This Data is what separates the outcome of this study from the study, [21] and the study, [47]. In the study, [35] LMPS method gets used, and also they use auto license plate recognition. SMS messages get eventually routed for manual input. As a consequence, illegal data input is rendered impossible.

4.3 Data Ownership

From the study, [49], crowdsensing distributes data in such a manner that either the user and the administrator have access to the database. Nevertheless, in the study, [49], Data is kept in the Cloud and shared between users and the device in the research. So in this study, Users and those using the device are the data owners. In the subsequent two studies, Smart Parking Using the Cloud is used to store data. However, in these two studies, the person who has access to the cloud server owns the data. In the study, [47], the administrator stores data in the user database. So, unlike other studies, in implementing this study, the administrator is the owner of the user database. Just like the studies, a few previous studies in the study, [47], a cloud-based server gets used, and anyone with access to the cloud database gets ownership of the data.

Table 4.2: Theoretical Comparative Study - 02

Contents/ Papers	Architecture	Data Integrity	Data Availability
Widyasari et.al [49]	WiFi Module based Architecture using Sensors	Requires specific time for specific crowdsensing	Available in the cloud server
Patro et.al [22]	WiFi Module using RFID	Requires specific time for specific crowdsensing	Available by Cloud Computing
LookMuang et.al [21]	Mobile Application using sensors based architecture	Real Time innovative parking system	Recorded by Firebase Database
Saarika et.al [34]	Cloud based Architecture	Cloud based error reducing system	Available in Cloud Server
Pomaji et.al [47]	Mobile Application based architecture	Admin Database Controlled System	Available in Administrator Module
Sadhukan et.al [35]	WiFi and WLAN based architecture using sensors	Mobile Agent controlled Parking system	Available by GUI Module
Ali et.al [50]	Sensors based smart car parking availability prediction architecture	Internet of Things (IoT), cloud technology, and sensor networks.	Available in cloud Database
Belkhala et.al [44]	Multi Agent System and AI based Architecture	Crowdsensing based on Image Processing System	Available to user and Subscriber

The comparison of our studies for architecture, data integrity, and data availability show in Table 4.2.

4.4 Architecture

According to a study, [49], the architecture is focused on IoT to find the speed of the vehicle using a WiFi Module. Similarly, in an IoT-based smart parking system, a WiFi module is used: A Proposed Algorithm and Model for Detecting Vehicles Using RFID Tags In the study, [21], the WiFi Module is not used as in the previous studies. However, this study uses ultrasonic sensors for license plate detection to find the nearest location. In this study, though, the mobile Application gets used to finding the license plate. The study, [34], WiFi Module is not used like the first two studies. This design has some similarities to the research study, [21], which also focuses on finding free slots. However, in this study, cloud servers have been used with the help of hardware and software modules which makes the works indistinguishable from other studies. In the study, [47], the use of Android phones is seen just like the study, [21]. Here, Android phones get used to finding the slot number. However, in this study, the use of a technology called Parking Control Unit is used. Now, coming to the subsequent study, [35], just like the studies, [49], and [22], WiFi and WLAN get used in its core architecture. Moreover, like the study, [21], ultrasonic sensors get used in this study. This study makes use of these devices to find if the vehicle is appropriately parked or not. In the next part, completely new technology gets seen in this study. The vehicle's entry time gets recorded, and a snapshot gets taken. The image gets sent to LMPS. After parking, the image of the slot gets sent to the client application. In the study, [50], deep learning is used to make the architect, while in the study, [44], AI is used to make the architecture using multi-agents. Just like other architectures in the study, [50] is also cloud-based architecture.

4.5 Data Integrity

According to the study, [49], there is a requirement of a specific time for crowd-sensing. However, if things do not do at this specific time, there might be a data corruption problem. In the study, [22], Data may be processed in real-time for extremely responsive applications when using a combination of applications. Besides that, the Cloud offers an expandable solution to IoT, where the resource's size fluctuates on an ongoing basis. In other words, the system is open to having anything added to it or taken away. In the study, [21], Mobile applications provide real-time (available/unavailable) information about parking spots. The detection of the number plate needs to do between the range of 30 cm to 70cm. The limitation of the data performance of this study is between this specific distance. If it exceeds 70 cm, then there might be a reduction in data performance. In this study, we also find slight data corruption like the study, [49]. Next, coming to the study, [34], to reduce errors, temperature sensors get used, and most values of sensors get stored in the Cloud. In this study, a different approach of using a temperature sensor get used to reduce errors. These approaches of reducing errors get also used in the study, [22]. However, this approach is entirely different but used for the same purpose. Next, in the study, [47], the administrator maintains all the modules related to the database. With the booking module, it can cancel booking within 20 minutes. In the study, [35], a vehicle-to-parking facility communications system that utilizes mobile agents to reduce data transmission while simultaneously maximizing vehicle-to-parking facility communication. In the study, [50], IoT, sensor networks, and cloud integration are applied, whereas, in the study, [44], crowdsensing based on image processing gets used. In all eight studies, different approaches are made to enhance data performance and reduce data corruption.

4.6 Data Availability

According to the study, [22], The combination of cloud computing and Since IoT has so many information resources generates enormous quantities of semi-structured and unstructured data as a result. IoT collects, retrieves, processes, and visualizes vast volumes of data. Simultaneously, the Cloud enables low-cost, on-demand, limitless computing space to create the best and most cost-effective data management solutions created by the Internet of Things With the assistance of norm. We can use Api's to view and simulate data stored in the Cloud, and cloud-based services are available. Therefore, according to a study, [21], Driver datasets, such as name and license plate, may be stored in the Cloud using Firebase. Moreover, the study, [34], using a cloud server that maintains a database that contains information about parking slots and their availability. Furthermore, the study, [47], mentioned about the Administrator Module is mainly used for managing the database and performing various operations on it. Finally, according to study, [35], For the most part, the GUI module is used to get information on the availability of different parking facilities in cities, as well as the option to book a parking spot via the internet. In the study, [50], Data is available in cloud databases like the study, [49], study, [22], and the study, [34] while in the study, [44] data is available to the user.

4.7 System Application

According to the study, [49], the author suggested a three-part IoT-based parking management system: an application for parking management, an application for parking lot users, and finally, an IoT platform. A web service will connect the Internet of Things (IoT) and smartphone apps in the parking database. Services will be able to update the database with the supplied Data easily. According to the study, [22], the author used IoT cloud, Parking sensors, and Processing unit to implement the intelligent parking management system. This intelligent parking module helps track the accessibility of every single vacant place and helps to transfer data live using a WiFi module. According to the study, [21], the author used computer vision for detecting the vehicle number plate for security purposes and also designed a payment procedure system using mobile phones to reduce time and traffic congestion. According to the study, [34], the author proposed an intelligent transportation system that combines the IoT for parking management with a signboard that displays valuable information about the location, such as weather and distance. The parking management system and the intelligent signboard are the two aspects of this system. It can get implemented through sensors and a cloud-connected web application. According to the study, [47], the author suggested a system that directs several vehicles to the nearest parking spot based on parking space availability at any given time. The Android operating system gets used to implement the vehicle parking system(CPS). Three Modules get used for this system:

User Module, Administrator Module, Booking Module. According to the study, [35], the author introduced an integrated component called a parking meter for real-time detection of improper parking, estimating the duration of each vehicle's parking lot usage along with an automatic collection of parking charges. It consists of 4 modules and are Parking IoT monitoring system (PLMS), Local parking management system (LPMS), Central parking management system (CPMS), and lastly, parking availability information. According to the study, [50], the author proposed a system based on a deep long short term memory network to anticipate parking availability using IoT, cloud technology, and sensor networks. In this research study, three types of studies get conducted to anticipate free parking spaces' availability. Based on geography, based on days of the week, and lastly, Based on working hours of the day.

4.8 Processor

According to the study, [49], the author proposed to implement the system using an Arduino as a micro-controller because the activity of the design is straightforward and completed repeatedly. On the contrary, according to the study, [22], the author used Arduino Uno that can be inserted into the WiFi module to establish an internet connection. Specifically, Arduino Mega 2560 gets implemented in the processing unit composed of an Arduino Uno, a processor on the chip. Afterwards, In the study, [21], the author used Raspberry Pi3 as an embedded controller in the system architecture. Compared to the study, [49], and study, [22], in this study, a different variety of microcontrollers gets implemented. Apart from this, according to the study, [34], the author uses a Raspberry Pi single-board computer to obtain information from the Cloud and pass the data to the display device. The author then uses the ESP8266 microcontroller in the research of study [47], which needs a command from the user or administrator. Moreover, according to a study, [35], Arduino Mega 2560 is used in the proposed system's PLMS (Parking IoT Monitoring System). This architecture is similar to the microcontroller of the proposed architecture in the study, [22]. Then in the study, [50], the author does not mention the use of a processor or microcontroller. Also, in the architecture of study, [44], the processor part was not mentioned by the author.

Table 4.3 shows the comparison of our studies for the processor, system application, and components.

Table 4.3: Comparative Study of Application

Contents /Papers	Processor	System Application	Components
Widyasari et.al [49]	Arduino	parking management, parking IoT users, IoT platform	Ethernet shield, Ultrasonic sensor
Patro et.al [22]	Arduino Uno	track the accessibility, transfer data	Infrared Sensor, Ethernet shield, Ultrasonic sensor
LookMuang et.al [21]	Raspberry Pi3	detecting the vehicle number plate, payment procedure	Optical Character Recognition (OCR), Ultrasonic sensor, Camera
Saarika et.al [34]	Raspberry Pi	parking management, displays valuable information about the location, weather, distance	Intelligent Signboard, WiFi Module, Display device
Pomaji et.al [47]	ESP8266	User Module, Administrator Module, Booking Module	RFID Tag, Sensor, RFID Reader
Sadhukan et.al [35]	Arduino Mega 2560	Parking lot monitoring , Local parking, Central parking, parking availability information	Ultrasonic Sensor, Camera, WiFi Module, Display device, Alarm IC
Ali et.al [50]	No data found	to anticipate the availability of free parking spaces	Acoustic Sensor, Ultrasonic Sensor, IR Sensor, CCTV, Drone, Laser Sensor
Belkhala et.al [44]	No data found	No data found	Magnetic Sensor, fixed RFID reader, end device, OCR camera, LCD Screen, Laser Sensor

4.9 Visual Representations of Papers

To make an easier understanding of the paper, we make a visual representation of the papers. All the architecture of the studies can get pictured by taking a look at the following figures. Here, we look at the ten papers we compare based on Hardware, Software, and functionalities. We take a look at the flowcharts of the architecture. It shows how the architecture functions. The workflow of the previous architecture and its strengths and weaknesses can get understood from these representations. In Figure 4.1, Figure 4.2, Figure 4.3, Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.7, Figure 4.8 are showing the procedures of the architectures from the comparative study of the eight research papers.

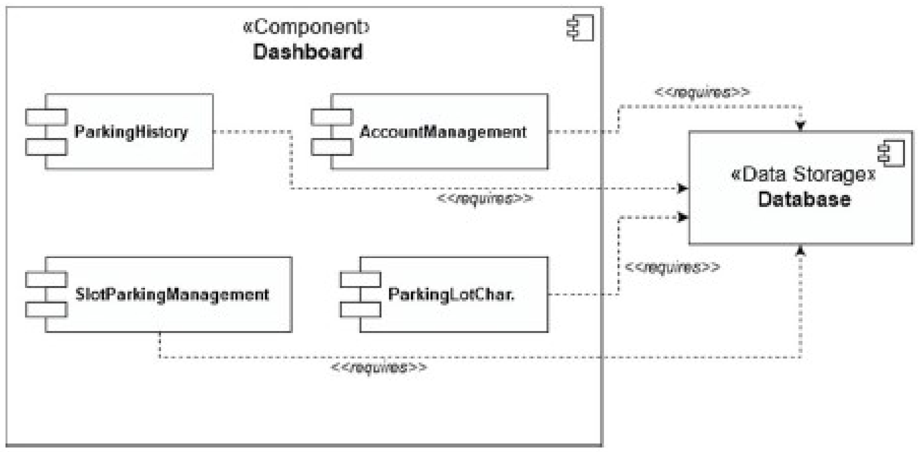


Fig. 2. Parking Management Application Architecture

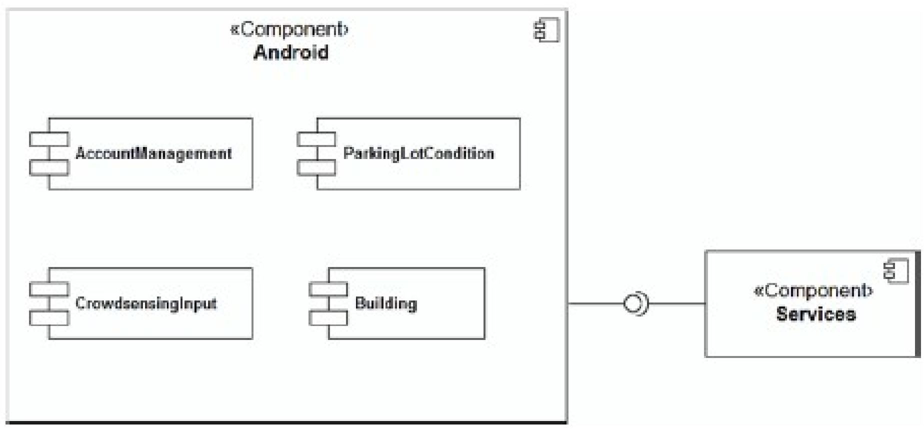


Fig. 3. Parking Lot Usage Application Architecture

Figure 4.1: The Architecture from study [49]

A. Proposed Flow Chart :

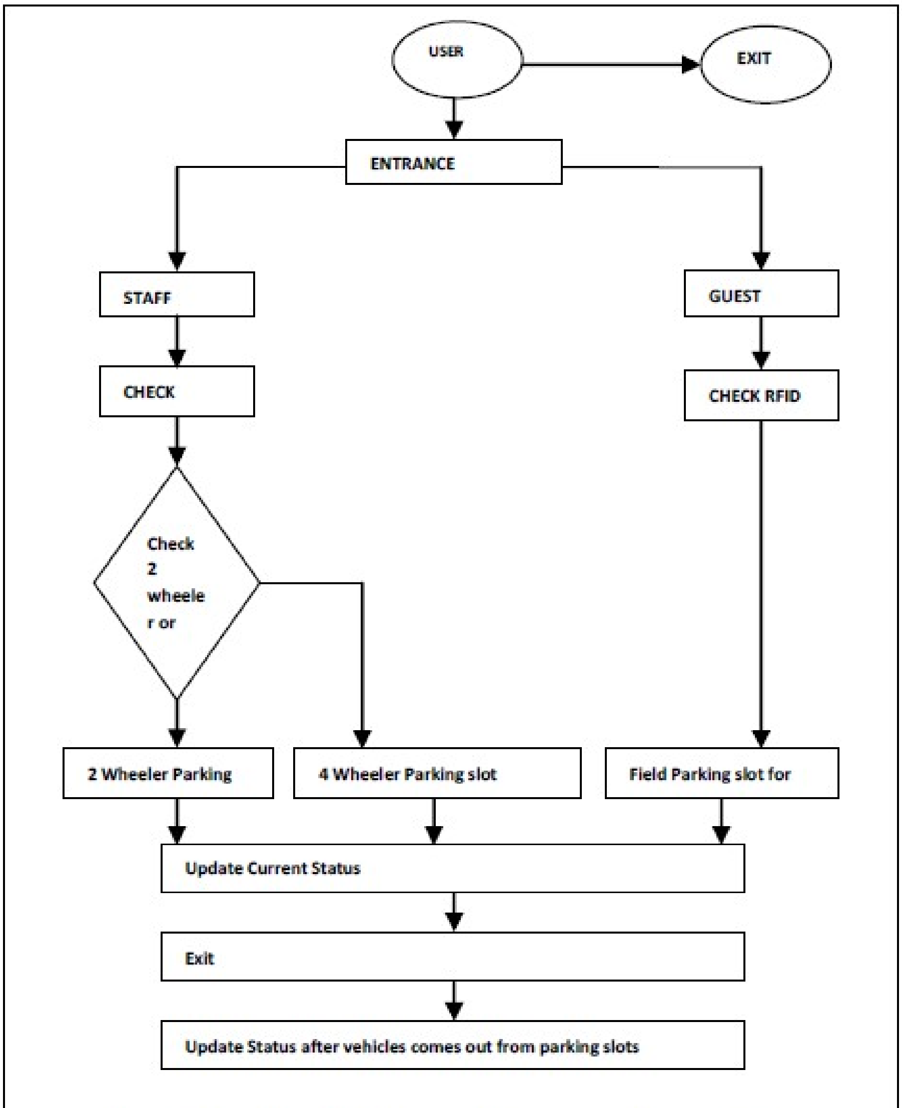


Fig. 5. Flowchart for IoT Based Smart car parking system

Figure 4.2: The Architecture from study [22]

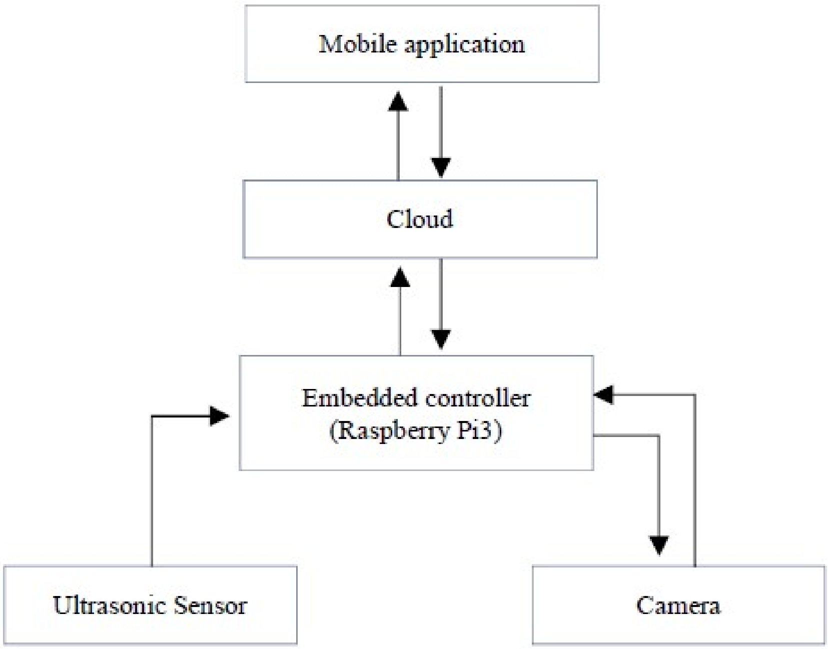


Figure 4.3: The Architecture from study [21]

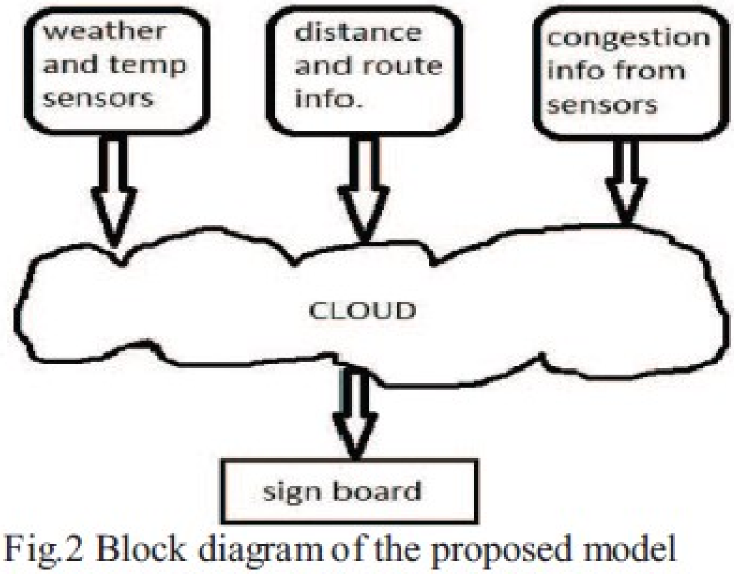


Figure 4.4: The Architecture from study [34]

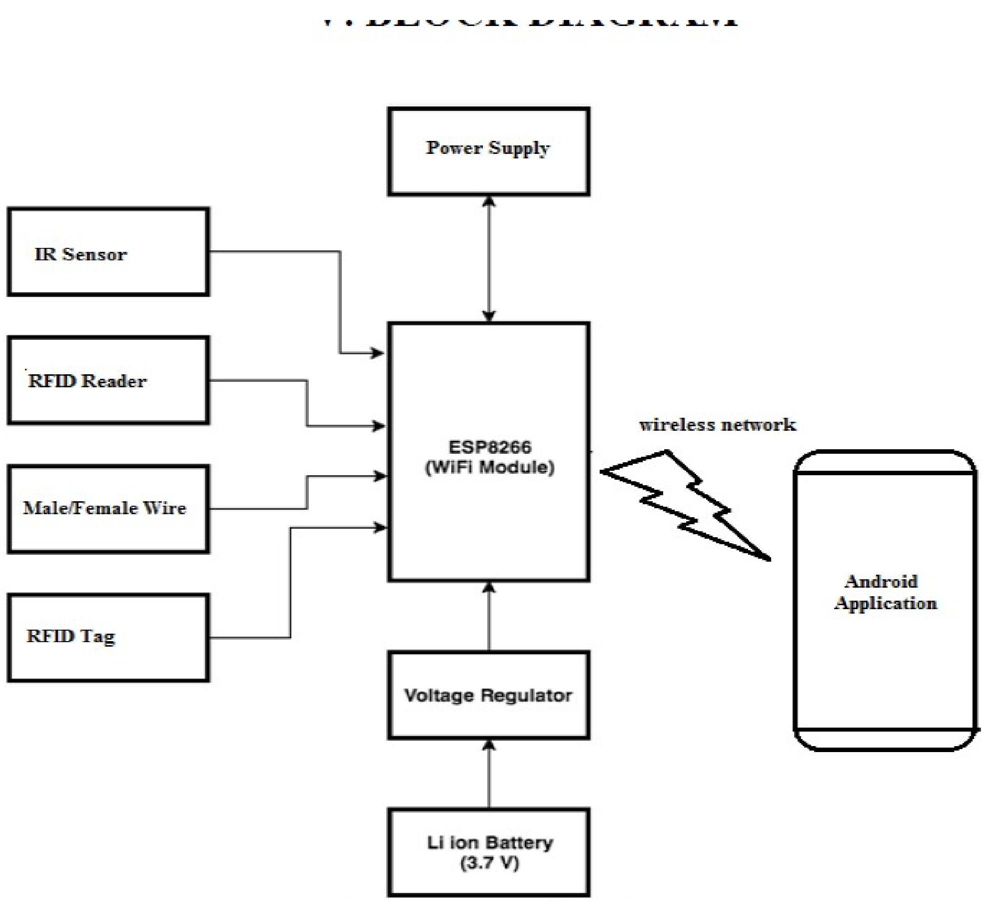


Fig 2. System block diagram

Figure 4.5: The Architecture from study [47]

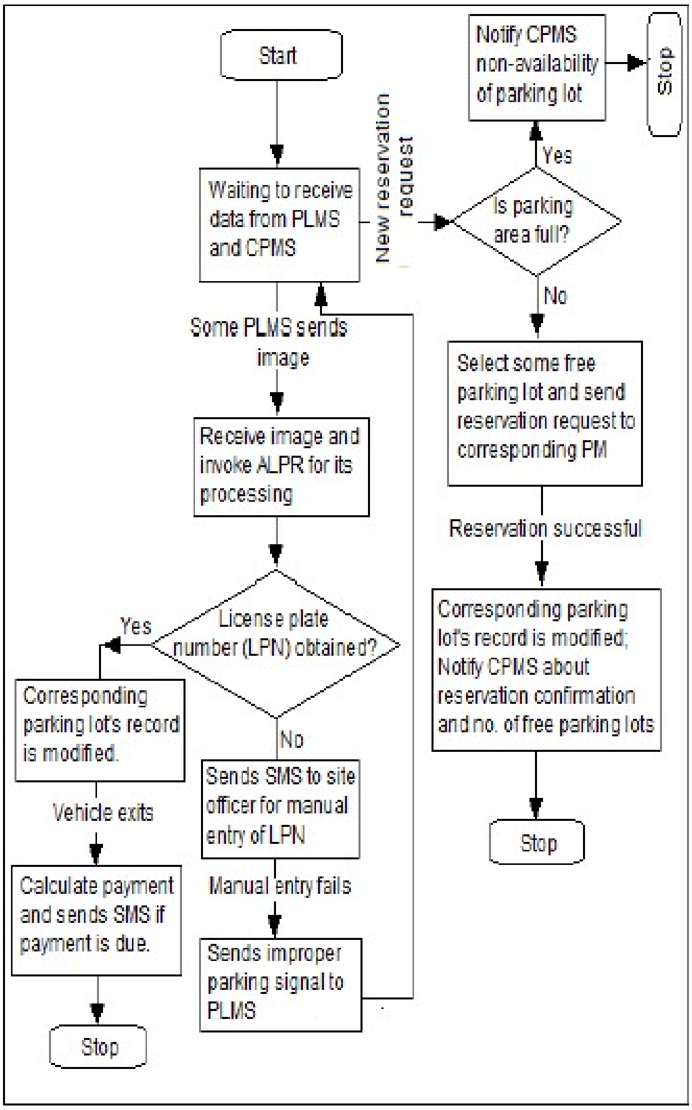


Fig. 4. Work flow diagram for Local Parking Management System

Figure 4.6: The Architecture from study [35]

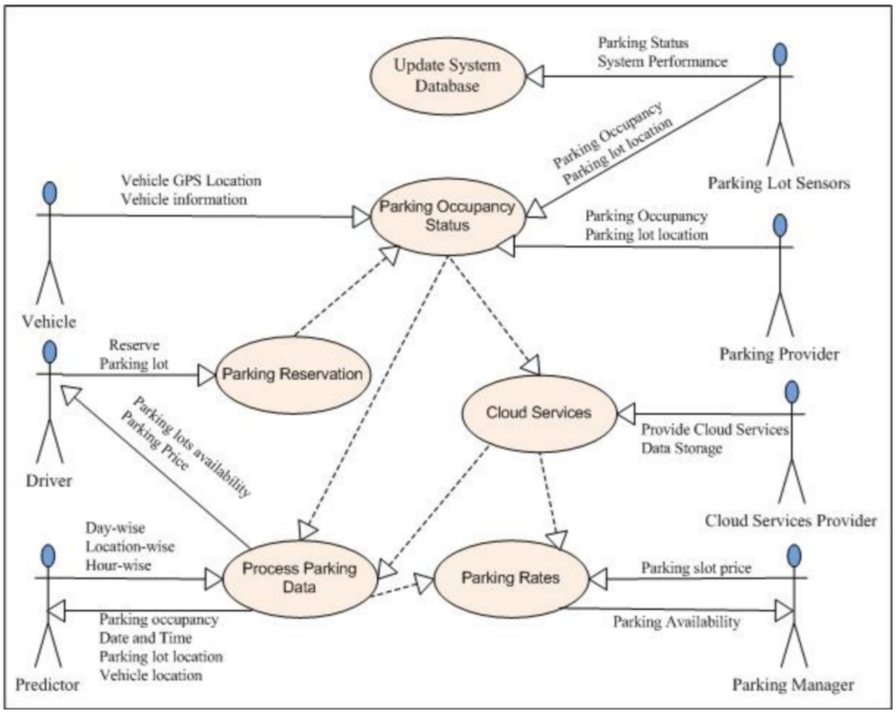


Figure 4.7: The Architecture from study [50]

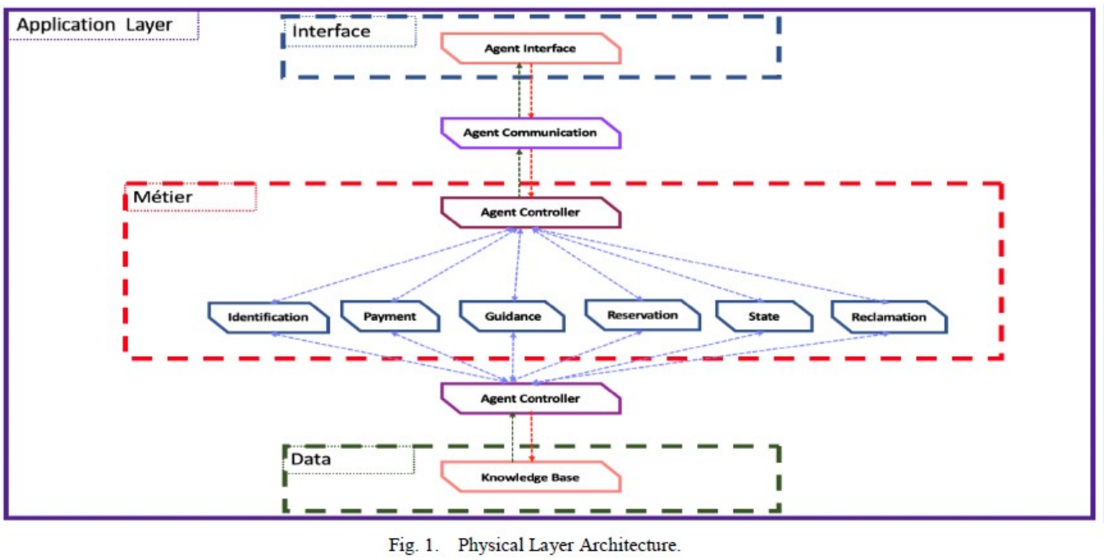


Figure 4.8: The Architecture from study [44]

Chapter 5

Architecture of Proposed System

5.1 Local Data Management

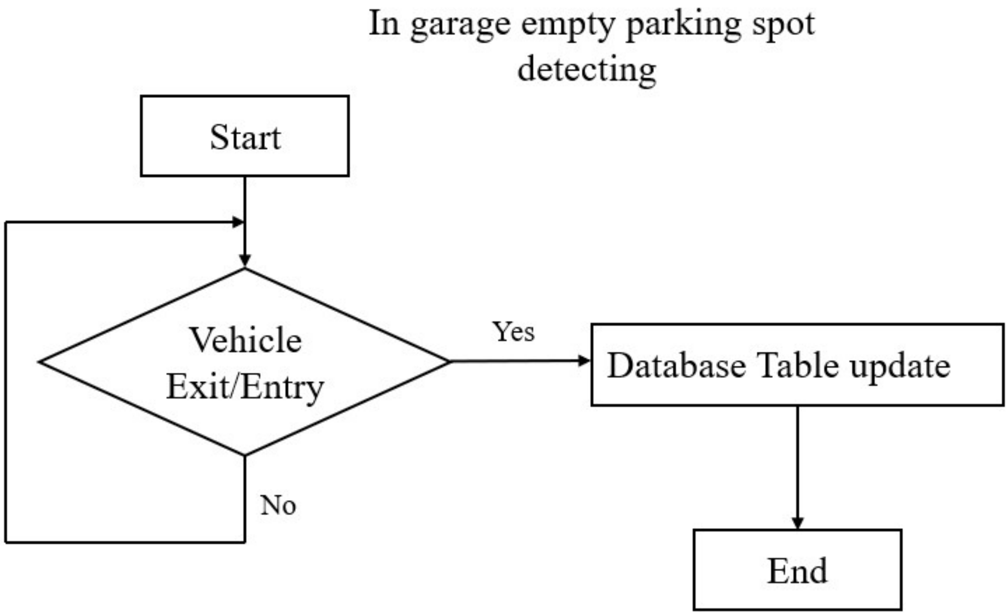


Figure 5.1: Flow chart of system in garage

In the garage, there will be a system that has an IR sensor, the esp8266 (WiFi module) and Arduino, [24], [20]. The esp8266 of the system gets connected to the database. The system detects how many spaces are in a particular garage and how many empty spaces the vehicle is. It just counts the amount of free space within the parking distance and updates the database. Additionally, it will detect the free space in real-time and send data to the cloud. This system will work as local data [31]. This Figure 5.1 shows the process for each local garage.

5.2 Cloud-based Data Management

There will be one database for each area, which will gather information from local sources. It is a cloud database that offers data access as a service while running on the cloud. A virtual machine image may be used to self-host databases in the cloud, or a database service can be purchased from a cloud database provider. Some cloud databases are SQL-based, some are No SQL-based,[2]. In our system database will collect data from each public parking space from local data. It will collect data from local data. Also will collect data from the scanner, which is attached with the machine in "No parking space", [25].

5.3 Hardware View

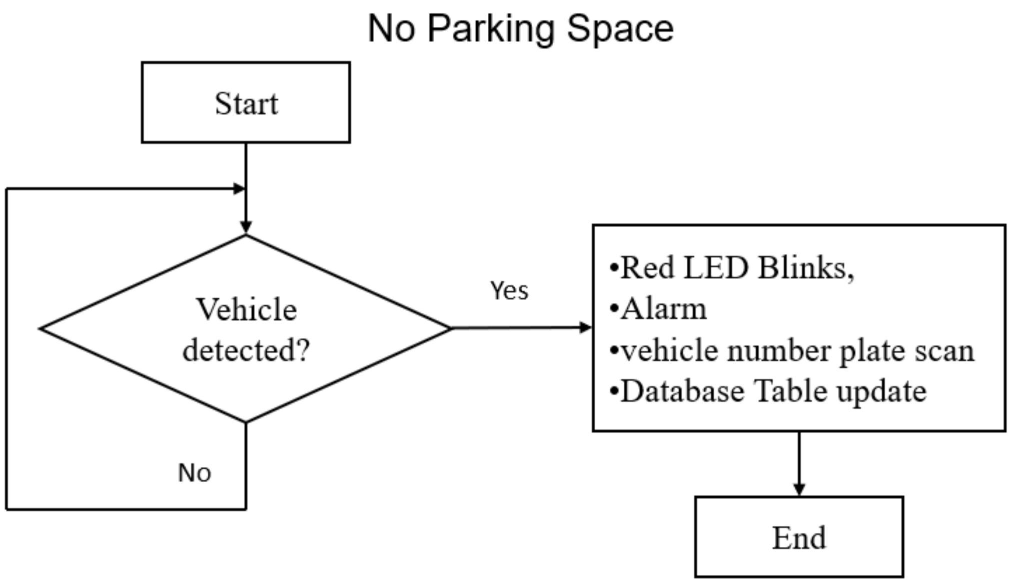


Figure 5.2: Flow chart of system in Non-Parking Space

A machine is built here with an LCD, Scanner, and a camera. The LCD will use to show the database which is managing cloud-based data. The database will appear on the LCD screen of our machine in the "No parking" zone when a driver tries to park his vehicle in a non-parking spot. Also, the attached alarm of the machine will also make sound for notifying driver that this is not the right place for parking vehicle. Moreover, the database will show the garage location. It will help the people to get an available parking space. On the other hand, the scanner of this machine will scan the vehicle number of those who will try to park the vehicle in the "No Parking" area, and the data will store in a database. This machine gets set in all non-parking areas. The system gets portrayed using a flow-chart in Figure 5.2.

Chapter 6

Hardware Components

As hardware we use Arduino UNO R3, resistors, servo motor, WiFi module, LCD Screen, and Camera.

6.1 Resistors

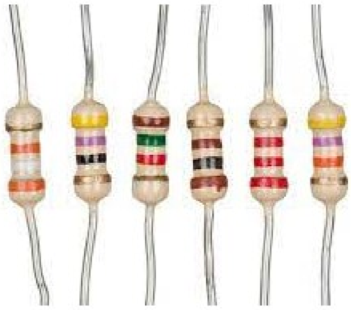


Figure 6.1: Resistors

A resistor is an electronic element that restricts or opposes the passage of electrical current in an electronic circuit. Additionally, resistors may use in conjunction with an active device that generates a particular voltage, such as a transistor. The remaining components may moderate. The current flowing through a DC (direct current) circuit is directly proportional to the voltage and inversely proportional to the resistance. This formula is the well-known Ohm's Law [52]. OHM'S law probably applies in alternating-current (AC) circuits as long as the resistor is free of inductance or capacitance. Resistors get manufactured in a variety of methods. The most often used resistor in electrical devices and systems is the carbon-composition resistor. Carbon-composition resistors are made by combining fine granulated carbon (graphite) with clay and hardening it. The quantity of resistance is proportional to the carbon content of the clay, [8]. The greater this ratio, the less resistance there is. Picture of resistors gets shown in Figure 6.1.

6.2 Liquid Crystal Display (LCD)

A liquid crystal display (LCD) is almost like an electronic screen that uses liquid crystals. A 16x2 LCD is an introductory module that does extensively used in a variety of devices and circuits, [6]. A 16x2 LCD has two lines and can display 16 characters per line. Each letter does show in a 5x7 pixel grid on this LCD. The 224 letters and symbols may show on the 16 x 2 intelligent alphanumeric dot matrix display. Command and Data are the two registers on the LCD, [6].

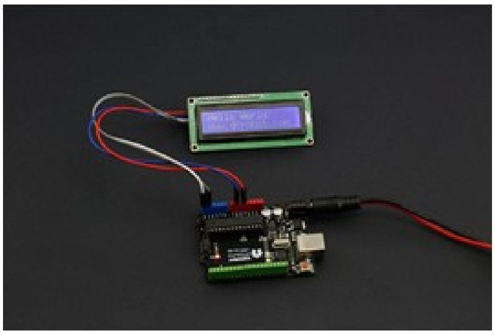


Figure 6.2: The way LCD takes input

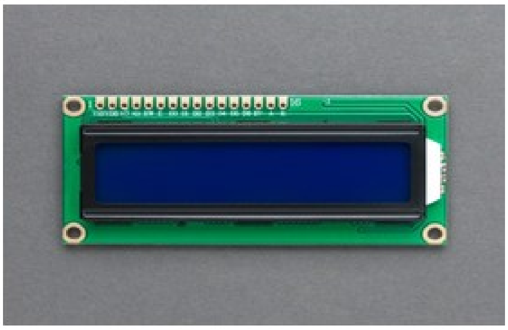


Figure 6.3: Liquid Crystal Display

The command registers include information about the different commands sent to the display. The data register gets used to store the data that will be present. The method of controlling the display entails first loading the data registers with the data that comprise the picture to gets shown and then loading the instruction register with instructions [46]. The display's contrast may get changed by changing the potentiometer linked to the VEE pin. Figure 6.2 and Figure 6.3 represents the process of the LCD taking inputs and the display used to show the inputs.

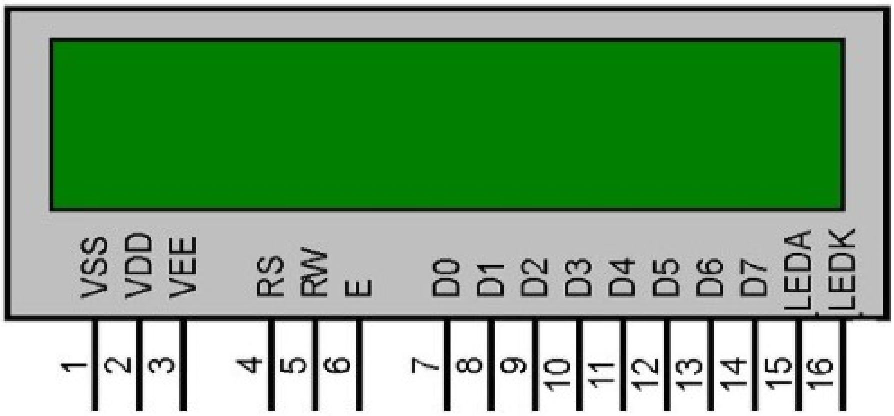


Figure 6.4: Pins of LCD

Table 6.1: LCD Pin Description

Pin Number	Pin Name	Pin Type	Pin Connection	Pin Description
1	Ground	Source	Ground pin of LCD	Links to the MCU's / power source's ground
2	VCC	Source	The LCD's supply voltage pin	Linked to the power source's supply pin
3	V0/VEE	Control	The LCD's contrast is modified.	links to a variable power supply capable of supplying 0-5V
4	Select Register	Control	Transitions between the Command and Data Registers	linked to an MCU pin, this variable receives either 0 or 1. 0 means Command Mode 1 write Data Mode
5	Read/ Write	Control	Changes the LCD's operation mode between reading or writing	linked to an MCU pin, this variable receives either 0 or 1. 0 means Write Operation 1 means Read Operation
6	Enable	Control	Must be very high in order to execute Operation Read/Write	linked to the MCU
7-14	Data Bits	Data/ Command	to transmit instructions or data to the LCD display.	4-Wire Mode -> 4 pins (0-3) get linked to MCU 8-Wire Mode -> 8 pins(0-7) get linked to MCU
15	LED Positive	LED	A standard LED-like procedure gets performed to turn the LCD on.	linked to +5V
16	LED Negative	LED	The LCD linked to GND operates normally in the manner of an LED.	linked to ground

Figure 6.4 shows multiple pins of the LCD display and Table 6.1 is showing the description of LCD pins. It means a 16x2 LCD that can display 16 characters per line, and there are two such lines. The adaptive alphanumeric circuit board display with a 16 x 2 can display 224 distinct letters and symbols. This LCD contains two registers, which are called the Command and Data registers. The Command register stores a variety of commands that sends to the display, [6].

LCD Modes: 4-bit and 8-bit

The LCD may operate in four-bit mode and eight-bit mode in two distinct modes. Furthermore, data is transmitted in 4-bit form nibble by nibble, starting with the top nibble and finishing with the bottom nibble, [6]. This bit is composed of four bits, the top four bits (D4-D7) of a byte from the upper nibble and the bottom four bits (D0-D3) of a byte from the lower nibble. This nibble enables us to transmit data in an 8-bit format, [6]. In 8 bit mode, on the other hand, the 8-bit data is sent in a single stroke due to all eight data lines. Thus, the 8-bit mode is more stable and quicker than the 4-bit option. The primary disadvantage is that it requires eight data lines to communicate with the microcontroller. This option will cause our MCU to run out of I/O pins. As a result, the 4-bit mode gets frequently utilized. These modes do not make use of control pins.

LCD Modes: Reading and Writing

As previously stated, the accurate LCD does comprise an interface IC. This interface IC may be read or written by the MCU. If necessary, information such as the cursor's location, status completion interruptions, and so on. The Interface IC present in most of the LCDs is HD44780U. LCD commands are shown in Table 6.2.

Table 6.2: LCD Commands

Hex Code	LCD Instruction Register Commands	Hex Code	LCD Instruction Register Commands
0F	LCD ON, cursor ON	C0	Move pointer to second line
01	Clear display screen	38	2 lines and 5×7 matrix
02	Return home	83	Cursor line 1 position 3
04	Decrement cursor (shift cursor to left)	3C	Activate line 2
06	Increment cursor (shift cursor to right)	08	Display OFF, cursor OFF
05	Shift display right	C1	2nd line leap, position 1
07	Shift display left	OC	Display ON, cursor OFF
0E	Display ON, cursor blinking	C1	2nd line leap, position 1
80	Move pointer to start line	C2	Jump to the second line, position 2

6.3 The WiFi Module

Espressif Systems, headquartered in Shanghai, China, manufactures the ESP8266, a low-cost WiFi microprocessor equipped with a complete TCP/IP stack and microcontroller capabilities. In August 2014, the ESP-01 module, developed by a third-party manufacturer AI-Thinker, first brought the chip to the attention of Western manufacturers, [41], [40]. Using IEEE 802.11 bgn, the ESP8266 module allows microcontrollers to connect to 2.4 GHz WiFi. It may get used to giving WiFi connectivity to external host MCUs using ESP-AT firmware, or it can be utilized as a self-contained MCU (Micro Controller Unit) using an RTOS-based SDK. 10th of August, 2020 [13], [40].

The ESP8266 Arduino compatible module is a low-cost WiFi chip with complete TCP/IP capability. Furthermore, this tiny board has an MCU (Micro Controller Unit) to control I/O digital pins using a simple, pseudo-code-like programming language. ESP8266 module is shown Figure 6.5.

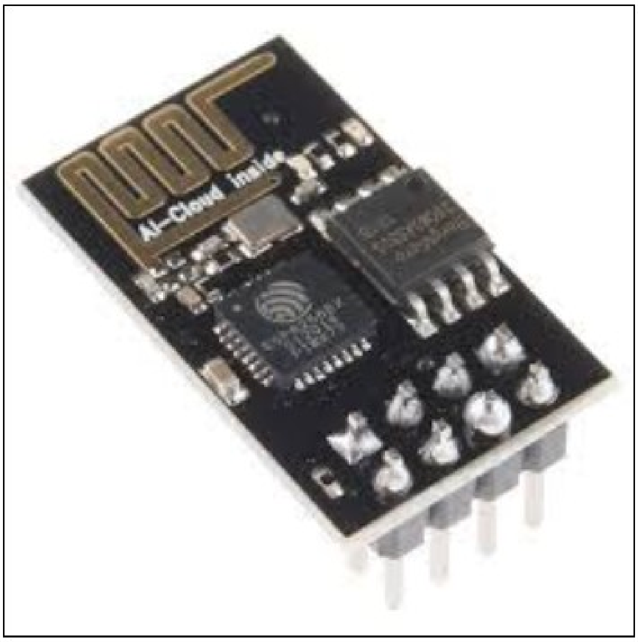


Figure 6.5: ESP8266

A WiFi module, also known as a serial to WIFI module, is an IoT transmission layer module. The function embeds a WiFi is a wireless network protocol and the IEEE802 wireless network protocol into a serial port or TTL level embedded module[41]. In Table 6.3 shows the pin description of ESP8266 and the Figure 6.6 shows the arrangement of ESP8266 pins.

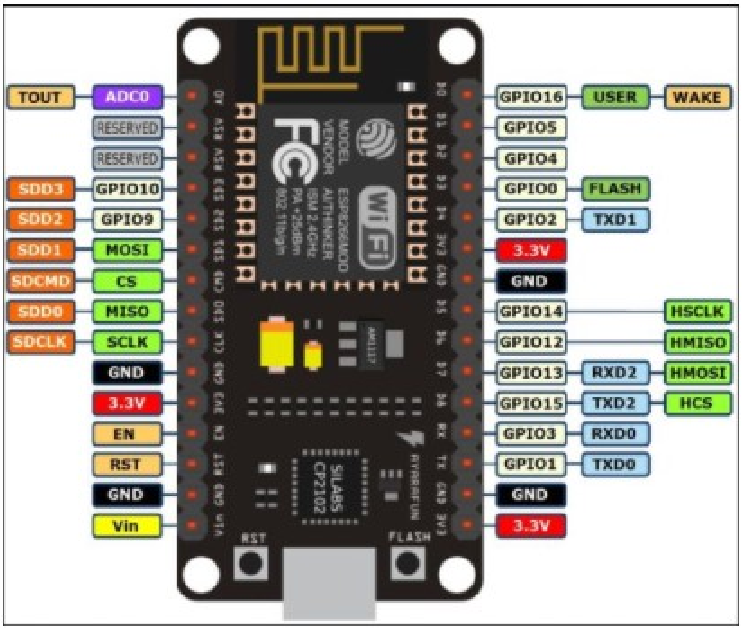


Figure 6.6: Pins of ESP8266

Table 6.3: Pin Description of ESP8266

Pins Name	Function	Restrictions and Usage
GPIO 0	Boot mode select	No Hi-Z
GPIO 1	TX0	Not used during serial transmission
GPIO 2	Boot mode select (TX1)	Cannot link to ground during boot time At boot time, sends debug data.
GPIO 3	RX0	Unsuitable for use during serial transmission
GPIO 4	SDA (I^2C)	
GPIO 5	SCL (I^2C)	
GPIO 6 - GPIO 11	Flash connection	Neither functional, nor separatable
GPIO 12	MISO (SPI)	
GPIO 13	MOSI (SPI)	
GPIO 14	SCK (SPI)	
GPIO 15	SS (SPI)	Unusable pull-up resistor
3V3		voltage source of other components
GND		connecting to the ground
Vin		internal voltage input voltage range: 4.75V to 10V
A0	ADC0 (Analog Input)	input voltage(0 to 1V)

Boot modes

Some I/O pins have a special function during boot, and they select 1 of 3 boot modes. Table 6.4 shows selected boot modes for the I/O pins.

Table 6.4: Boot modes

GPIO 15	GPIO 0	GPIO 2	Mode
0V	0V	3.3V	UART Bootloader
0V	3.3V	3.3V	Boot sketch (SPI flash)
3.3V	x	x	SDIO mode (not used for Arduino)

PWM

Unlike other ATMEL CHIPS (ARDUINO), the ESP8266 does not support hardware PWM. However, software PWM is accessible on all digital pins. PWM range is set to 10-bits @ 1kHz by default, although this can gets modified up to >14-bit@1kHz.

A high power WiFi module has an RF transceiver, security, ARM7 based processors, PHY functions, and 802.11 MAC, which works with IEEE regulatory domains. A high Power WiFi module is:

- Cost-effective.
- Used in embedded devices.
- High power.
- Flexible.
- Used in applications (thermostats and wireless sensors).

The ESP8266 WiFi Module is a self-contained system (SOC) with an inbuilt TCP/IP protocol stack that enables any microcontroller to connect to a WiFi network. The ESP8266 is capable of hosting programs and offloading all WiFi networking operations to a separate application processor [13].

6.4 Arduino Uno R3

Arduino Uno R3 is a microcontroller board that gets powered by an ATmega328 AVR microprocessor contained in a detachable dual-inline package (DIP). Twenty digital input/output pins do provide, six of which are analogue and six of which are PWM. This Arduino is suitable for a wide variety of applications, [18]. The Arduino computer software, which is easy to use, supports the loading of applications. Arduino is an open-source platform for developing electronic creations [52], , [18]. It is often composed of a part of the software and a physical programmable circuit board (commonly referred to as a microcontroller) or IDE (Integrated Development Environment) that runs on a computer and is used to create and upload code the computer's physical board. The Arduino platform has gained popularity among those just starting to start in electronics, and for a good reason, [29], [18]. Figure 6.7 shows Arduino Uno R3.

Figure 6.7: Arduino Uno R3

Unlike most previous programmable circuit boards, the Arduino requires separate hardware (dubbed a programmer) to load new code into the board, [18]. Rather than that, a USB cable is all that gets required in this instance. The Arduino IDE makes learning to program easier by using an abbreviated version of the C++ language. Arduino is a typical example of a factor since it consolidates the functionality of a complicated microcontroller into a more manageable packaging. Arduino Uno R3 is a microcontroller board based on the ATmega328P [52]. This board contains everything necessary to support the microcontroller. To get started, connect it to a PC through a USB connection and power it using a battery or an AC-DC converter, [18]. The name Uno refers to "one" in the "Italian language," and was selected to commemorate the release of Arduino's IDE 1.0 software. The R3 Arduino Uno is the third and most recent upgrade of the Arduino Uno. Its board and IDE software are current versions of the Arduino standard. The Uno is the standard model of a series of USB-Arduino boards, [29].

Table 6.5: Arduino Pins Description

Pin Name		Pin No.	Pin Description
Power Supply	Vin		- voltage pin (input) - input supply of external power source
	5V		- power supply voltage - board components
	3.3V		- provide 3.3V - board's voltage regulator
	GND		ground
Reset			resetting microcontroller
Analog		A0 - A5	- analog input - range of 0-5V
Digital		0 - 13	digital input or output
Serial (UART)	TX	0	- Arduino board communicate with end devices. - To send and receive data using an end device
	RX	1	
External Interrupt		2, 3	producing External interrupt
PWM		3, 5, 6, 9, 10, and 11	To transform a digital input to an analog input by modulating the pulse
SPI (Serial Peripheral Interface pin)	SS	10	Slave Select
	MOSI	11	Master Out Slave In
	MISO	12	-Master In Slave Out
	SCK	13	Serial Clock
LED Pin		13	When this digital pin gets high, an LED illuminates.
AREF (analog reference)			to focus on providing a standard voltage from a separate power source

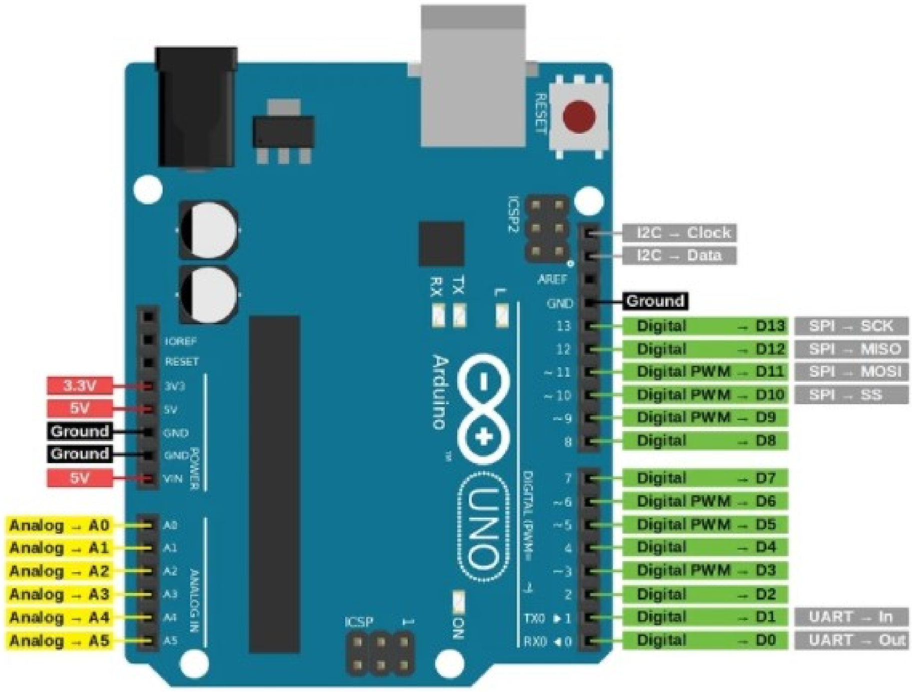


Figure 6.8: Pins of Arduino Uno R3

In Table 6.5 and Figure 6.8 shows the structure of Arduino UNO R3 and the pin description. The Uno is an excellent introductory Arduino board. This board includes 14 digital I/O pins (six of which may use as PWM outputs), six analogue inputs, a USB connection, a power connector, a reset button, and many other features, which means it has everything we need to get started and nothing we do not need, [18].

Popularity-wise, the Arduino UNO board dominates everything else in the Arduino product line because another computer's USB port may link the board. As a serial device, it connects the board to a computer system as well as providing power [38].

6.5 Camera

Figure 6.9: Image Processing Camera

The Internet of things (IoT) has a diverse range of uses in our everyday lives. It gets used in both commercial and industrial environments all around the world. Every day, IoT devices are making the world a little bit better and wiser. With the emergence of the Internet of Things (IoT), [28], the global parking system has taken on a new dimension. In the twenty-first century, traffic congestion is an issue that that has taken away the tranquillity of people's minds. Unplanned vehicle parking is the primary explanation for this. Even after searching numerous parking lots, drivers are unable to locate their desired parking spot, [3]. Therefore, parking spaces are underutilized. There were articles on identifying the availability of vehicles and detecting the correct positioning of the vehicle in prior studies. On the other hand, those studies cannot come up with the concept of inappropriate vehicle parking, which has proven to be a significant issue. For solving this problem, a smart parking system may get developed. Sensors will identify when the vehicle gets parked in the incorrect location[3]. If the vehicle is parked inappropriately, the sensor will sound an alarm, which will continue to sound until the vehicle is moved from the spot and sent to the owner's vehicle. Eventually, this will make everyone aware of parking in the correct spot, and the drivers will not park in the incorrect location. As a result of this system, traffic congestion will reduce to some extent. Figure 6.9 shows a picture of an image processing camera[3].

6.6 Infrared Sensors (IR sensor)

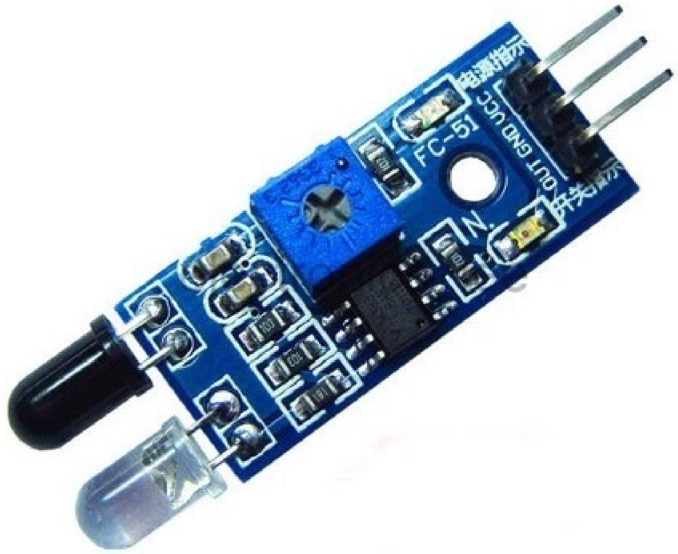


Figure 6.10: IR Sensors

The Infrared (IR) sensor gets used to detect obstructions in front of the sensor or distinguish between colours, depending on the sensor's design. The operation of an infrared sensor acting as an Object Detection Sensor gets explained in detail. When the IR transmitter produces radiation, part of the radiation reaches the object, and some of the radiation reflects the IR receiver, which is called infrared radiation [42]. The sensor's output gets determined by the strength of the reception received by the infrared receiver. The selection of the infrared sensor to be used in the vehicle parking system gets intended to draw attention to a possible solution that may aid in the improvement of the process of locating an empty parking spot, which has become a source of frustration for the majority of vehicle users [51]. Picture of Infrared Sensor gets shown in Figure 6.10.

6.7 Servo Motor

A servo motor is similar to a motor that rotates precisely owing to its high degree of precision. The majority of the time, this kind of motor gets equipped with a control circuit that provides feedback on the present position of the motor shaft. This information may use to assist the servo motors in rotating with extreme accuracy. A servo motor gets used to rotate an item at a particular angle or distance from the centre. It does the makeup of nothing more than an essential motor that operates via a servo mechanism. Power is supplied to a DC servo motor by a DC power source [5].

An alternating current servo motor, on the other hand, is powered by an alternating current. Furthermore, in addition to these major categories, there are many other types of servo motors depending on the gear arrangement and operational characteristics that may find. A servo motor with a gear arrangement enables us to manufacture a servo motor with a very high torque while maintaining compact and lightweight packaging. They get utilized in various applications, including toy vehicles, radio-controlled helicopters and aeroplanes, robotics, and so on. Servo motors get measured in kilograms per centimetre, which is the unit of measurement for them. [5].

Figure 6.11 shows the picture of a servo motor. Most servo motors have a force rating of 3kg/cm, 6kg/cm, or 12kg/cm, depending on the manufacturer. This kg/cm number indicates how many times a servo motor can lift in a given amount of time. For example, a 6kg/cm Servo motor should be capable of lifting 6kg if the load is hanging 1cm from the vehicle's shaft; the longer the distance between the load and the motor's shaft, the lower the body-weight capacity citeLin. An electrical pulse determines a servo motor's position, and the motor's electronics get located next to the motor.

Figure 6.11: Servo Motor

Working Mechanism

It comprises of three parts:

- Controlled device
- Output sensor
- Feedback system

A positive feedback system is utilized in a closed-loop to regulate motion and the ultimate position of the shaft. The device gets controlled through a feedback signal produced when the output signal gets compared to the reference input signal. As a result, the feedback system generates the third signal by measuring and comparing the reference and output signals. As an input signal, this third signal controls the device. No matter how many times the input and output signals differ, this signal will be present as long as there is a feedback loop in place. As a result, the principal purpose of servomechanism is to keep a system's output constant in the face of noise. A servo is composed of a direct current or alternating current motor, a potentiometer, a gear assembly, and a control circuit. To begin, we use gear assembly to reduce motor RPM and increase motor torque. Consider the following scenario: the potentiometer knob is positioned at the beginning of the servo motor shaft, ensuring that no electrical current gets produced at the inverter output of the potentiometer.

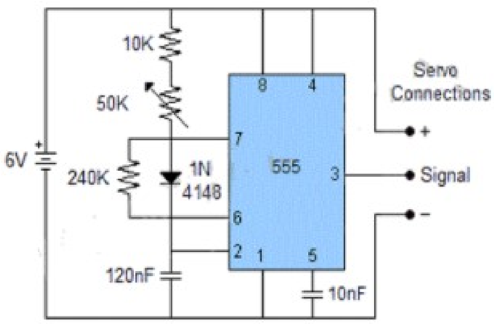


Figure 6.12: Servo Motor Controller

In order to determine the error, an electrical signal must now deliver to the amplifier's other input port. Then, using a feedback controller, a reference voltage is generated by evaluating the difference between two signals, one from the attenuator and one from external sources. This error signal gets received as an input by the motor, which then begins to rotate. After connecting the potentiometer to the motor shaft, each time the motor turns, the potentiometer turns as well, resulting in the generation of an electrical signal—likewise, the output feedback signal changes in response to the angular location of the potentiometer. After a while, the potentiometer's output equals the external signal. Due to the lack of difference between the signal received from external sources and the signal produced at the potentiometer, there will be no output signal from the amplifier to the motor input in this scenario, and the motor will stop spinning when this occurs.

Figure 6.12 represents the controller of the servo motor. A servo motor's most significant advantage is its ability to provide high torque levels at high speeds, which stepper motors cannot. They are also (80 to 90) per cent efficient. Servo motors are vibration and resonance-free and can get driven in either an AC or DC mode.

Chapter 7

Platforms, Languages, and Libraries

7.1 Google Firebase

In 2011, Andrew Lee and James Templin co-founded Envolve, a startup company that would later become known as Firebase. Google bought Firebase in October 2014, converting it into the Google BaaS platform (Business as a Service). It is a software development platform that enables developers to build mobile apps for iOS, Android, and the web. To provide a unified platform for mobile developers, Google has expanded Firebase's services inside the company. Analytics, authentication, database administration, setup, file storage, and push messaging are just a few available features. The services are cloud-based and scalable, requiring little to no effort on the part of the developers, [2].

Firebase Google Analytics

With the introduction of the most recent Google Analytics app reports that link with Firebase - Google's integrated application development platform [19]. The phrase "Google Analytics for Firebase" refers to the most current Google Analytics app reports that appear in the Firebase interface, as well as other Google app-centric products that interact through Firebase, [16]. The Firebase SDK collects primary app use data for us without needing us to write any additional code. It allows everyone to use the same data and statistics for an app, whether Google Analytics or the Firebase interface. We learn how users engage with the app and how successful our marketing efforts are. Furthermore, the reports allow us to develop audiences and connect to third-party networks, acting on results immediately. Analytics integrates with all Firebase features and allows users to create an unlimited number of reports on up to 500 unique events specified through the Firebase SDK. Analytics reports provide a user with a clear understanding of how users behave, allowing them to make informed marketing and performance optimization decisions [19].

Firebase Cloud Storage

Google also provides cloud storage via their Firebase platform. With powerful, easy, and cost-effective object storage designed for Google scale, it can quickly store and distribute pictures, music, video, and other user-generated material. The Firebase SDK allows developers to connect with Firebase programmatically, offering cloud storage features. It enables them to download and upload files from and to consumers in real-time. It also has features for resuming or retrying transfers in the case of a poor connection.

Firebase Authentication

We took advantage of Firebase's authentication module to make dealing with our users as secure and straightforward as possible. Users' sign-in and onboarding experiences are improved thanks to Firebase Authentication, which makes building secure authentication systems more accessible and faster. A complete identity solution gets provided by this feature, which includes support for email and password accounts and logins to Google, Facebook, GitHub, and Twitter accounts.

Firebase Real-time Database

The Firebase Realtime Database is a NoSQL cloud database that enables real-time data synchronization across many clients while maintaining offline capabilities. In the real-time database, data gets stored in JSON format, and all connected clients share a single instance of the database, which gets constantly updated with the most current information. All customers' data is synchronized in real-time and available even if an app is not running on the client's device. User information, parking spot location and condition, user subscription history, and passcodes are all stored on the server, [16].

Firebase Crashlytics

To minimize time spent troubleshooting, create an easily consumable list of all the issues that have happened. A user can immediately determine which issues need immediate attention by immediately showing the user impact on the Crashlytics dashboard [16]. If a user gets real-time alerts, they will be able to maintain their bearings. Crashlytics, which may be accessed here, can be used to report a crash.

7.2 Programming Language

A programming language is a set of commands, instructions, and syntax used to build computer programs. The programming languages that programmers use to create code gets referred to as “high-level languages”. This code can be compiled into a “low-level language,” which is recognized directly by computer hardware.

Arduino Language

Arduino has its programming language, which is similar to C++, with several unique methods and functions. C++ is an easy-to-understand programming language. Creating an Arduino sketch (code file) gets examined and compiled into machine language. Arduino, on the other hand, can be utilized with Python or another high-level programming language. If a user already has a working knowledge of Python, they may start Arduino by controlling it with Python, [17].

Python

Python is a high-level, general-purpose programming language. In recent days, it has become one of the most popular programming languages around the world. Python first appeared back in 1991, thanks to the Python Software Foundation. This language was created to construct various applications and is not specialized for any specific problem. It gets primarily used in software development, internet development, machine learning, and other related fields [16], [43].

7.3 Open CV

OpenCV stands for Open Source Computer Vision Library, a collection of mainly programming functions for real-time computer vision. It gets developed to provide an everyday basis for computer vision applications and accelerate machine perception incorporation into commercial products. Intel created it at first. Later on, supported by Willow Garage and Itseez afterwards. Real-time computer vision software and applications use OpenCV as an image processing library. Images get transformed into multidimensional arrays in OpenCV, which considerably simplifies their handling. An image can alter by altering its pixels with the help of OpenCV functions, [37].

Chapter 8

Methodological Analysis

8.1 Vehicle Detection and Space Counting in garage and Storing Data in Firebase

Esp 8266 works on a local-based server. To work with this module, we need a local IP address. So that is why we used firebase as our local database. It will read data through the IR sensor (infrared sensor) will show data on the database. The set-up of this part gets shown in Figure 8.1.

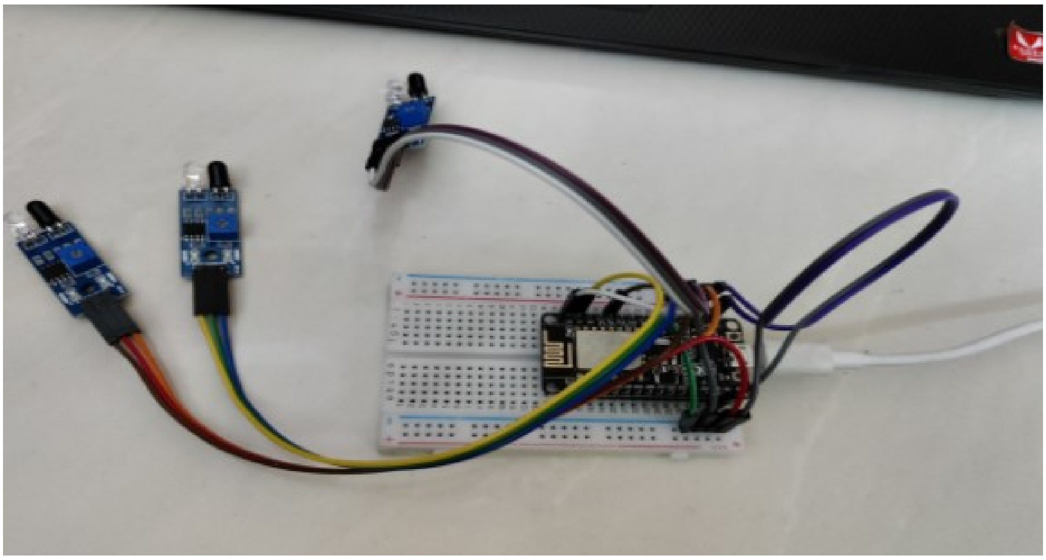


Figure 8.1: WiFi Module and IR sensors setup to Detect Available Space

Here, in firebase, the red arrow detects whether that garage has any available space or not. It is getting data from ESP8266 which is a WiFi module. Here, 1 means space is available in the garage for parking. It will turn 0 automatically when there will be no space. This is shown in Fiqure 8.2.

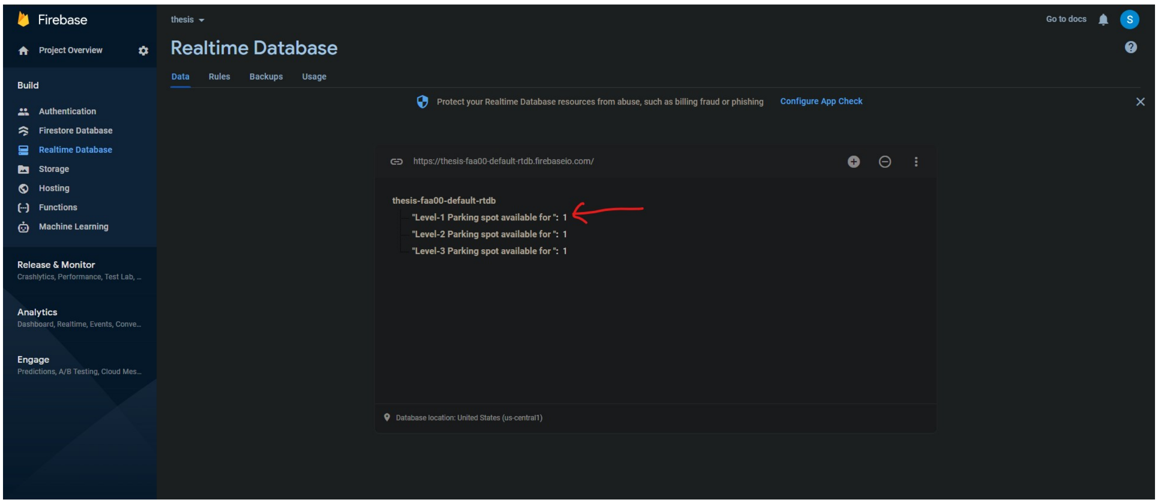


Figure 8.2: Database of Local Server

This database information will show in LCD of Figure 8.3 which will locate at a no parking space.

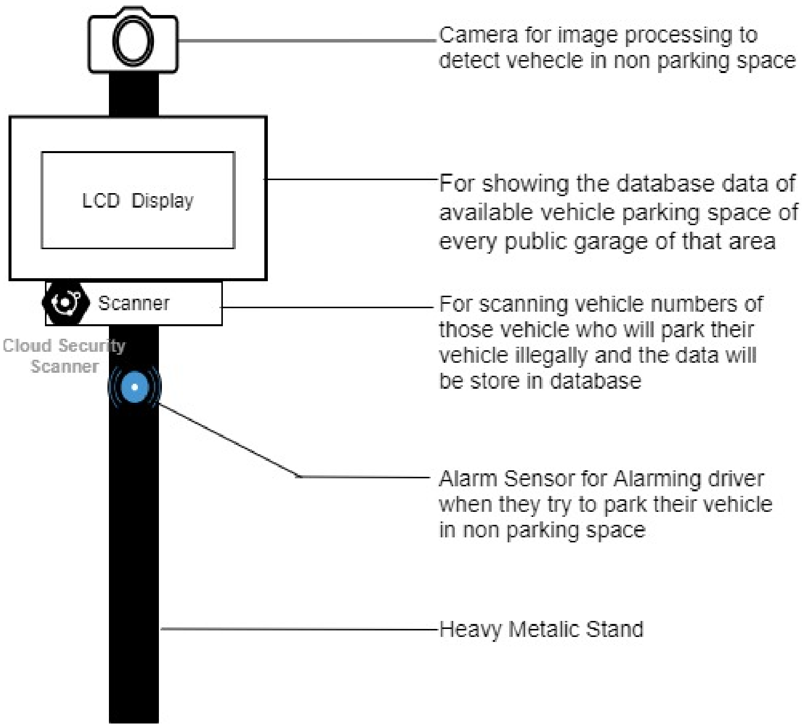


Figure 8.3: Machine in “No-Parking Space”.

When we need to read analogue data, we need Arduino. Through this Arduino, we can easily show the available space and total space.

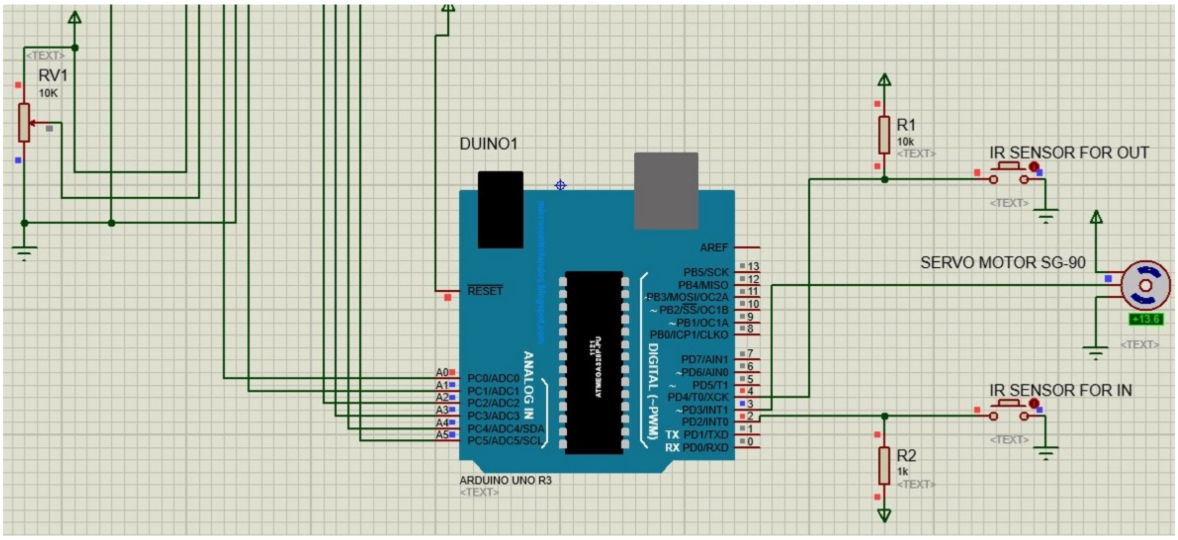


Figure 8.4: Through Arduino Garage space counting.

For this, we need an Arduino servo motor and IR sensor, and WiFi module. Here Ir sensor and Arduino will detect vehicles and will count available space. Then, through the WiFi module, data will go to the firebase. And then, the data will show on LCD. This Arduino simulation gets shown in Figure 8.4 and the data will show in LCD as Figure 8.5.

Figure 8.5: Data is showing in LCD

Here, data can be sent from the local base to the cloud using Arduino, WiFi module (ESP8266), and IR sensors.

8.2 Vehicle Detection in No-Parking Space through image processing

We have all of the required libraries installed. This same technique works through training a model using different picture characteristics associated with the identified item (in this instance, a vehicle), which we then utilize to identify the object in our target. Consider it similar to the train and test data sets of any machine learning model.

Extensible Markup Language (XML) is a mark-up language that allows for the encoding of documents by defining a set of rules in a machine- and human-readable format. It is an extension of SGML (Standard Generalized Markup Language) that enables the specification of the text's structure. Custom tags can be defined in XML documents. We can also transmit information using XML as a standard format, which is becoming increasingly popular.

Sections of XML documents are denoted by similar components and specified by starting and ending tags. A tag is a component of markup that begins with the letter and ends with the character >. The content of an element gets defined as the characters appearing between the start- and end-tags if any. Elements can include markup, which can include other elements. The root element is the most significant, top-level element, and it contains all of the other components. When working with an XML tree structure, navigation, modification, and removal are straightforward to accomplish programmatically. Python comes with a built-in library called Element Tree, which functions to read and manipulate XML documents. This dataset can be taken as the train and test datasets of any image processing model. In this case, .xml files which capture the image details the target object.

Here, .xml files, which capture the target object's image details set as train data set and test data set is the video of a moving vehicle on the road. For visualization in Python, we import the CV2 library. Then we take the information from the camera detecting the vehicle, and we use the link "cap = cv2.VideoCapture('0')". We use a pre-trained .xml file which contains information on vehicles built from individual images using "car cascade = cv2.CascadeClassifier('cars.xml')", which is shown in Figure 8.6.

Figure 8.6: Detects vehicles

The capturing image gets divided into frames, and the code hardly reads one at a time. Using the APIs that we imported previously, we detect the location of the vehicle in each frame. We locate the coordinates of each detected vehicle, draw a rectangle around it.

```
"ret, frames = cap.read()
   gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)"
```

Figure 8.7: The video is read frame by frame.

The captured video that we took using the camera is read frame by frame. The frame gets then converted to a greyscale, which aids in the detection of the vehicle. The image is converted to greyscale because the trained data set is built-in greyscale to reduce file size. Figure 8.7 shows the algorithm of how image processing read data frame by frame.

```
   for (x,y,w,h) in vehicles:
      plate = frames[y:y + h, x:x + w]
      cv2.rectangle(frames,(x,y),(x +w, y +h) ,(51 ,51,255),2)
      cv2.rectangle(frames, (x, y - 40), (x + w, y), (51,51,255), -2)
      cv2.putText(frames, 'vehicler', (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255,
255, 255), 2)
      cv2.imshow('vehicle',plate)
```

Figure 8.8: Detects and Stores the Coordinates of the Vehicles

The first section of code detects and stores the coordinates of the vehicles in the frame (x, y axes, and the width and height of the vehicle). The font accompanying the text may easily modify, and the code (51,51,255) represents the B-G-R sequence's colour code for the rectangle and text.

```
   frames = cv2.resize(frames,(600,400))
      cv2.imshow('vehicle Detection System', frames)
      k = cv2.waitKey(30) & 0xff
      if k == 27:
         break
```

Figure 8.9: Generated Image in Loop

The viewer gets shown the generated image (frame), and the loop repeats until the user presses the Enter key on the keyboard. Figure 8.8 and 8.9 shows the algorithms of video processing

8.3 Vehicle Number Plate detection

This program's complete code gets written in Python. This program accepts an image as input and automatically identifies one or more number plates as well as the number as a string.

Figure 8.10: Original

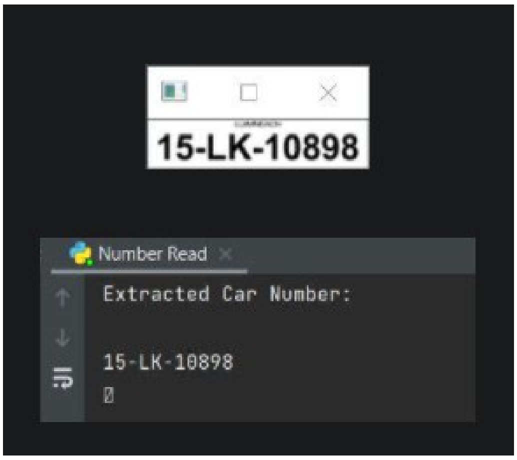


Figure 8.11: Output Results

First of all, three specific module needs to install to run this program. As our program deals with image processing, we have used the Open-CV library to implement the program. Moreover, Python-Tesseract gets used for character recognition in the input image. In addition, Python-Tesseract gets utilized to recognize characters in the input picture. Imutils library is also required because the software requires image manipulation such as scaling, rotating, charting, and identifying edges. The use of Open-CV and Imutils makes the job considerably easier. Figure 8.10 shows the input image of number plate recognition process

At first, the program will read the image file into Python using Open-CV to apply further operation in the image. The first step is to transform the picture to a different format to greyscale to minimize the image's complexity. When a picture is in RGB, the processing of the image becomes more difficult. Grays-calling simplifies the procedure and reduces the amount of time it takes to finish it. It takes several steps to hand over the final output. Figure 8.11 shows the final output of the process. The program will apply a filter after the greyscaling operation to make the image smooth and noise-free. Also, it will find the edges of the entire image for localization after processing it. We have used a canny algorithm to detect the edges. Edge detection will help us find the contours, which will eventually help find where the number plate is within the image. After applying the canny algorithm, detected edges get shown in Figure 8.12 and After applying the canny algorithm, detected edges get shown in Figure 8.13.

```python
image = cv2.imread("image/car1.jpg")
image = imutils.resize(image, width = 500)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

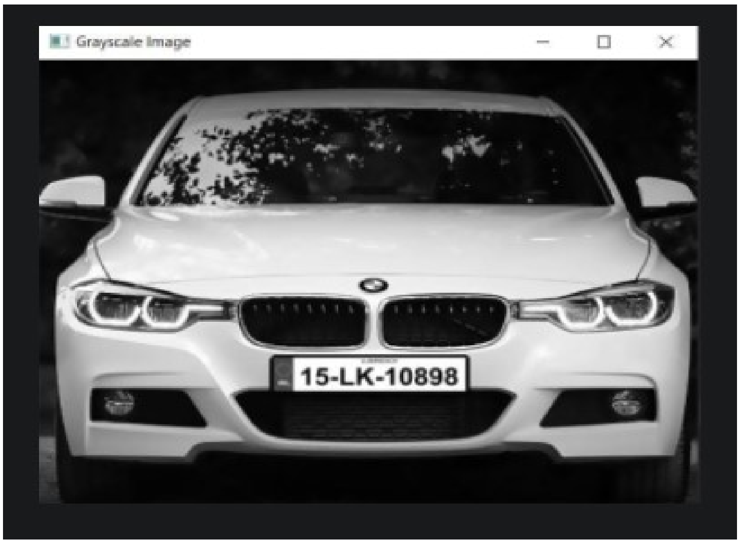


Figure 8.12: After Gray-scaling

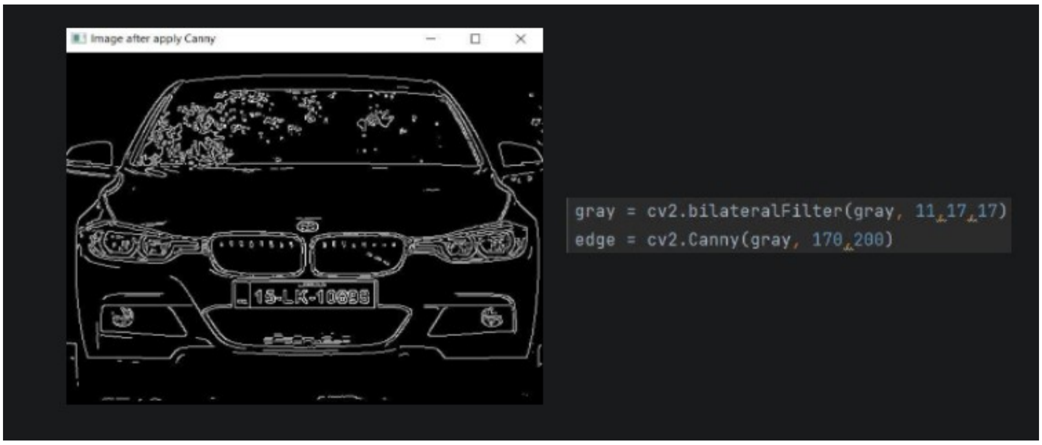


Figure 8.13: After Finding Edges

After applying the canny algorithm, the program will perform contour detection. This algorithm will recognize lines in the picture and polygons within those lines. It will look for a four-pointed contour to determine a rectangle, which is the form of a number plate. The result we obtain after completing all of the procedures gets mentioned below. Figure 8.14 shows the detected number plate, [11].

Figure 8.14: Number Plate

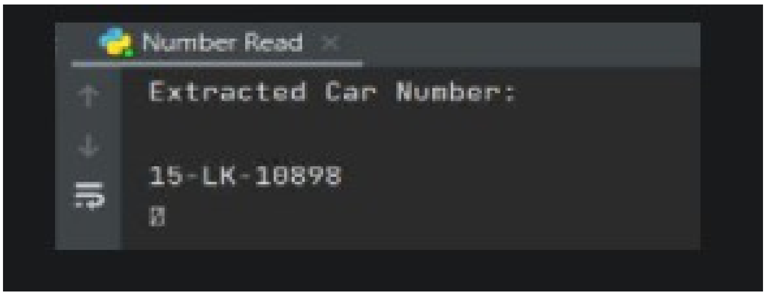


Figure 8.15: Final Output

Finally, the program will utilize tesseract OCR to extract the number as text from the picture after identifying the number plate from the image. Essentially, the program will conduct an operation to convert the picture to a string and display it as the program's final result. Figure 8.15 shows the string converted from the detected number plate.

Chapter 9

Result Evaluation

9.1 Results

The output we get from our proposed Architecture-

- It determine the number of vacant spaces that are present in the garage.
- It determine if any vehicle parked in a “No Parking” Area.
- It capture number plate of the vehicle who try to park in non parking space.
- It show the firebase data on LCD.
- The cloud stores all data with Realtime.

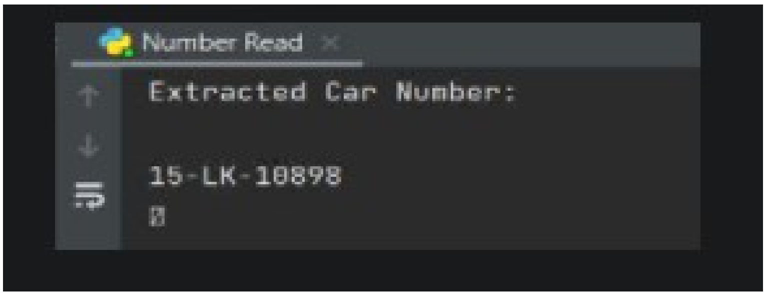


Figure 9.1: Scanning Output

Figure 9.2: Output in LCD

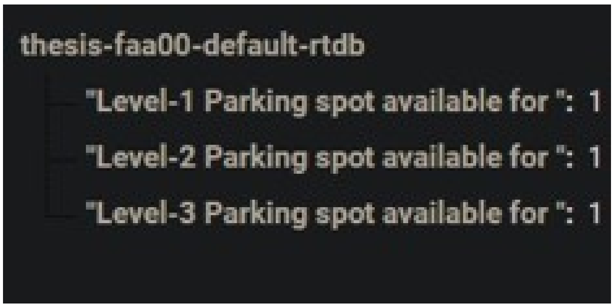


Figure 9.3: Firebase Output

The results are showing in Figure 9.1, Figure 9.2, Figure 9.3

9.2 Time and Space Complexity

We calculate time and space complexity by using clock and simulator. When we run the simulation in software we get some load complexity which is shown in Figure 9.4.

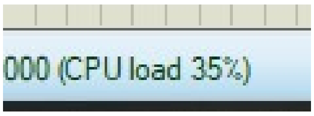


Figure 9.4: Simulation Load

On a test basis, we calculate the overall time required to run the algorithm and the space required to run the algorithm. When we run the algorithm, Arduino itself calculates the total memory required while running the algorithm [22], [50].

Time Complexity:.
Run-time of code 2 seconds.
Firebase gets data in Realtime.
Firebase sends data to LCD in Realtime.
Vehicles number plate detection time 5 seconds.
Image Processing time 3 seconds.

Space Complexity:
1928 bytes for sketch and 196 for dynamic variables.
2124 byte is total space complexity.

Chapter 10

Epilogue

10.1 Why this Proposed Architecture?

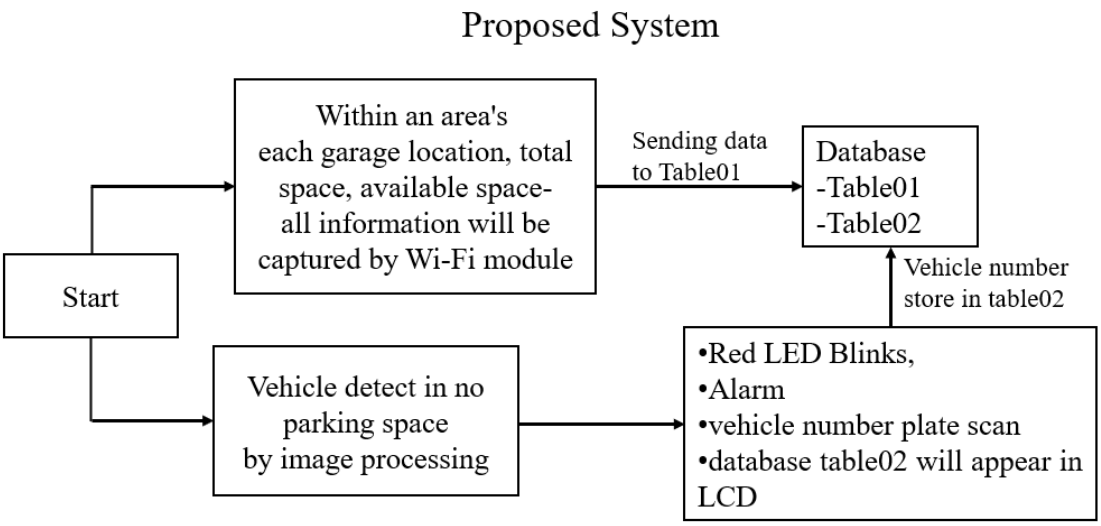


Figure 10.1: The Whole Procedure

In Figure 10.1 The whole procedure is shown in a flow chart. For the following reasons, the proposed architecture gets required:

- It helps to save our time.
- In the existing studies, there is nothing related to the "No Parking" spot, and even if we try to implement the previous architectures for the "No Parking" spot, it is not compatible with the place.
- By collecting data, it gives people real-time information of where the vehicle could go.
- In other studies, information comes automatically, but in this study, they come manually.

10.2 Advantages of this System

The advantage of this proposed architecture is by this proposed architecture, and people will get data automatically. When the user parks the vehicle, the sensor scans the vehicle's license plate. Additionally, the camera detects the vehicle, and the alarm sensor is ready to alert the motorist if they park in the wrong spot. People would not have to find the available parking space because they would get the information from the database of LCD.

10.3 Limitations

In our proposal, there have some limitations they are:

- There is no booking parking system. By the time people go there, the slot gets booked. By the time people go into the parking spot, the location may fill up.
- The architecture does not implement at a large scale.
- Google Map does not still include in the work.

10.4 Insufficient of resources

Now, as a popular topic, The Internet of Things (IoT) is gaining popularity in technology. It is a great concept where it brings physical objects to the Internet. It is also getting popular in education and also research. About the growth of IoT products, many vendors and manufacturers have commented how the future might evolve in this sector. Numerous types of lightweight protocols get developed to establish communication among various physical objects. However, in the absence of undefined information, it is difficult to provide correct and flawless information among this wrong information [45]. Hence, correct information is one such issue that needs to address to transform speculations into reality. There is much information that we found which confused and also, we thought that the information is not sufficient to research -

- Many IoT systems today are minorly described on the Internet and implemented, using different protocols and technologies that confuse and create complex configurations.
- Lack of mature information on IoT technologies and business processes. There is currently little focus on establishing strategies for establishing situational knowledge of an organization's IoT asset security posture.
- There is a lack of instructions for IoT device maintenance and management.
- There are no standards for IoT device authentication and authorization.
- For IoT-based incident response efforts, there are no best practices.

- There are no auditing or logging standards for IoT components.
- IoT devices with restricted interfaces can interact with security devices and applications.
- There were fewer research papers about the topic. So, for that, the information we get from the research was deficient.
- The data analysis does not properly found on the Internet. For this, it was hard to know the various information and procedures about data analysis.
- many papers had the same research topic. As for the same topic, there were more, but the information gets found to be less.

While implementing the demo, we did not have sufficient funding. Therefore, we could make a model with cheaper products that work initially. The advanced version of the kits could not yet get found.
As everyone said, we have gone under constant pressure to uncover new information that will allow us to achieve the impossible: fit the information's limited resources while delivering excellent results [39].

10.5 Future Work

We have worked on a model. However, it is not sufficient to implement in the practical field. In the future, we plan to implement the whole model on a large scale. For example, we will add a booking system feature, make a mobile application, and so on. We plan to move the demo architecture to a GPS-based architecture. It will not be just a model but be implemented for actual vehicles. The demo will be implemented all over the globe in a developing country, making this a cost-efficient project. We are on our way to get new funds to continue this work in the practical field. For this, we need a better model of kits.

10.6 Conclusion

To conclude, on the Internet, we have got many articles regarding IoT-based smart parking systems. Nevertheless, we did not get any satisfactory solution to unruly parking on the roads. Also, we did not get enough solutions regarding people's unawareness in maintaining discipline in parking their vehicles or other transports in non-parking areas. So, to solve some portion of this problem, we came up with the idea that indicates developing a smart parking system. If we can implement our planning about smart parking systems in real life, it will bring some rules and discipline to roads and streets. Also, this implementation will help to reduce road accidents, and traffic jams [39]. Those who own a vehicle will get benefits as well. They will be able to find parking spaces efficiently, which will save their time and energy. In this manner, our research paper will fill some parts of the smart parking system gap and help build a smart city.

Bibliography

[1] P. Trahanias and L. J, "Vector directional filters-a new class of multichannel image processing filters," *IEEE Transactions on Image Processing*, vol. 2, 1993, 18296237. DOI: 10.1109/83.242362. eprint: https://doi.org/10.1109/83.242362. [Online]. Available: https://doi.org/%2010.1109/83.242362.

[2] M. Egenhofer, "Spatial sql: A query and presentation language," *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, no. 1, pp. 86–95, 1994. DOI: 10.1109/69.273029. eprint: https://doi.org10.1109/69.273029. [Online]. Available: https://doi.org10.1109/69.273029.

[3] J.-S. Lee, Y.-Y. Jung, B.-S. Kim, and S.-J. Ko, "An advanced video camera system with robust af, ae, and awb control," *IEEE Transactions on Consumer Electronics*, vol. 47, no. 3, pp. 694–699, 2001. DOI: 10.1109/30.964165. eprint: https://doi.org10.1109/30.964165. [Online]. Available: https://doi.org10.1109/30.964165.

[4] T. Lehmann, C. Gonner, and K. Spitzer, "Addendum: B-spline interpolation in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 660–665, 2001. DOI: 10.1109/42.932749. eprint: https://doi.org10.1109/42.932749. [Online]. Available: https://doi.org10.1109/42.932749.

[5] C.-M. Lin and C.-F. Hsu, "Adaptive fuzzy sliding-mode control for induction servomotor systems," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 362–368, 2004. DOI: 10.1109/TEC.2003.821859. eprint: https://doi.org10.1109/TEC.2003.821859. [Online]. Available: https://doi.org10.1109/TEC.2003.821859.

[6] H.-P. Shieh, "Transflective display by hybrid oled and lcd," pp. 650–651, 2005. DOI: 10.1109/LEOS.2005.1548173. eprint: https://doi.org10.1109/LEOS.2005.1548173. [Online]. Available: https://doi.org10.1109/LEOS.2005.1548173.

[7] B. Nath, F. Reynolds, and R. Want, "Rfid technology and applications," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 22–24, 2006. DOI: 10.1109/MPRV.2006.13. eprint: https://doi.org10.1109/MPRV.2006.13. [Online]. Available: https://doi.org10.1109/MPRV.2006.13.

[8] A. D. Grasso, D. Marano, G. Palumbo, and S. Pennisi, "Improved reversed nested miller frequency compensation technique with voltage buffer and resistor," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 5, pp. 382–386, 2007. DOI: 10.1109/TCSII.2007.892217. eprint: https://doi.org10.1109/TCSII.2007.892217. [Online]. Available: https://doi.org10.1109/TCSII.2007.892217.

[9] Y.-h. Han, H. Jang, J. Choi, B. Park, and J. Mcnair, “A cross-layering design for ipv6 fast handover support in an ieee 802.16e wireless man,” *IEEE Network*, vol. 21, no. 6, pp. 54–62, 2007. DOI: 10.1109/MNET.2007.4395111. eprint: https://doi.org10.1109/MNET.2007.4395111. [Online]. Available: https://doi.org10.1109/MNET.2007.4395111.

[10] J. Lu and X. Duan, “Comparative analysis of intel pentium 4 and ieee/emc tc-9/acem cpu heat sinks,” pp. 1–6, 2007. DOI: 10.1109/ISEMC.2007.86. eprint: https://doi.org10.1109/ISEMC.2007.86. [Online]. Available: https://doi.org10.1109/ISEMC.2007.86.

[11] M. Bergeron, J. Cadorette, J.-F. Beaudoin, M. D. Lepage, G. Robert, V. Selivanov, M.-A. Tetrault, N. Viscogliosi, J. P. Norenberg, R. Fontaine, and R. Lecomte, “Performance evaluation of the labpet apd-based digital pet scanner,” *IEEE Transactions on Nuclear Science*, vol. 56, no. 1, pp. 10–16, 2009. DOI: 10.1109/TNS.2008.2010257. eprint: https://doi.org10.1109/TNS.2008.2010257. [Online]. Available: https://doi.org10.1109/TNS.2008.2010257.

[12] A. Khanna and R. Anand, “Iot-based smart parking system,” *Information Technology*, vol. 2, no. 2, pp. 97–112, 2009, PMID: 27442303. DOI: 10.1089/big.2014.0018. eprint: https://doi.org/10.1089/big.2014.0018. [Online]. Available: https://doi.org/10.1089/big.2014.0018.

[13] H. Wang and W. He, “A reservation-based smart parking system,” *INFCOMW*, 2011. DOI: 10.1109/INFCOMW.2011.5928901. eprint: https://doi.org/10.1109/INFCOMW.2011.5928901. [Online]. Available: https://doi.org/10.1109/INFCOMW.2011.5928901.

[14] G. Yan, W. Yang, D. B. Rawat, and S. Olariu, “Smartparking: A secure and intelligent parking system,” *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 1, pp. 18–30, 2011. DOI: 10.1109/MITS.2011.940473. eprint: https://doi.org10.1109/MITS.2011.940473. [Online]. Available: https://doi.org10.1109/MITS.2011.940473.

[15] E. Polycarpou, L. Lambrinos, and E. Protopapadakis, “Smart parking solutions for urban areas,” in *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2013, pp. 1–6. DOI: 10.1109/WoWMoM.2013.6583499. eprint: https://doi.org10.1109/WoWMoM.2013.6583499. [Online]. Available: https://doi.org10.1109/WoWMoM.2013.6583499.

[16] L. Wu, Y. Chamming, Y. Lin, and S. Tung, “Justiot internet of things based on the firebase real-time database,” *SMILE*, pp. 16–40, 2013. DOI: 10.1109/SMILE.2018.8353979. eprint: https://doi.org/10.1109/SMILE.2018.8353979. [Online]. Available: https://doi.org/10.1109/SMILE.2018.8353979.

[17] A. Adriansyah and A. W. Dani, “Design of small smart home system based on arduino,” pp. 121–125, 2014. DOI: 10.1109/EECCIS.2014.7003731. eprint: https://doi.org10.1109/EECCIS.2014.7003731. [Online]. Available: https://doi.org10.1109/EECCIS.2014.7003731.

[18] Y. A. Badamasi, “The working principle of an arduino,” pp. 1–4, 2014. DOI: 10.1109/ICECCO.2014.6997578. eprint: https://doi.org10.1109/ICECCO.2014.6997578. [Online]. Available: https://doi.org10.1109/ICECCO.2014.6997578.

[19] D. A. Filvà, M. J. C. Guerrero, and M. A. Forment, "Google analytics for time behavior measurement in moodle," pp. 1–6, 2014. DOI: 10.1109/CISTI.2014.6877095. eprint: https://doi.org10.1109/CISTI.2014.6877095. [Online]. Available: https://doi.org10.1109/CISTI.2014.6877095.

[20] M. Idris and Z. Razak, "Smart parking system using image processing techniques in wireless sensor network environment," *Big Data*, vol. 2, no. 2, pp. 97–112, 2014, PMID: 27442303. DOI: 10.1089/big.2014.0018. eprint: https://doi.org/10.1089/big.2014.0018. [Online]. Available: https://doi.org/10.1089/big.2014.0018.

[21] R. Lookmuang, K. Nambut, and S. Usanavasin, "Smart parking using iot technology," *ICBIR*, vol. 2, no. 2, pp. 97–112, 2014, PMID: 27442303. DOI: 10.1109/ICBIR.2018.8391155. eprint: https://doi.org/10.1109/ICBIR.2018.8391155. [Online]. Available: https://doi.org/10.1109/ICBIR.2018.8391155.

[22] S. Patro, M. Sepathy, N. Padhy, and R. Sah, "Iot based smart parking system: A proposed algorithm and model," *ICCSEA*, vol. 2, no. 2, pp. 97–112, 2014, PMID: 27442303. DOI: 10.1109/ICCSEA49143.2020.9132923. eprint: https://doi.org/10.1109/ICCSEA49143.2020.9132923. [Online]. Available: https://doi.org/10.1109/ICCSEA49143.2020.9132923.

[23] J. Monsalve, A. Landwehr, and M. Taufer, "Dynamic cpu resource allocation in containerized cloud environments," pp. 535–536, 2015. DOI: 10.1109/CLUSTER.2015.99. eprint: https://doi.org10.1109/CLUSTER.2015.99. [Online]. Available: https://doi.org10.1109/CLUSTER.2015.99.

[24] O. Orrie, B. Silva, and G. P. Hancke, "A wireless smart parking system," *IECON*, vol. 2, no. 2, pp. 97–112, 2015, PMID: 27442303. DOI: 10.1109/IECON.2015.7392741. eprint: https://doi.org/10.1109/IECON.2015.7392741. [Online]. Available: https://doi.org/10.1109/IECON.2015.7392741.

[25] T. N. Pham, M.-F. Tsai, D. B. Nguyen, C.-R. Dow, and D.-J. Deng, "A cloud-based smart-parking system based on internet-of-things technologies," *IEEE Access*, 2015. DOI: 10.1109/ACCESS.2015.2477299. eprint: https://doi.org/10.1109/ACCESS.2015.2477299. [Online]. Available: https://doi.org/10.1109/ACCESS.2015.2477299.

[26] I. Chatzigiannakis, A. Vitaletti, and A. Pyrgelis, "A privacy-preserving smart parking system using an iot elliptic curve-based security platform," *Compute.Commun*, 2016. DOI: 10.1016/j.comcom.2016.03.014. eprint: https://doi.org/10.1016/j.comcom.2016.03.014. [Online]. Available: https://doi.org/10.1016/j.comcom.2016.03.014.

[27] M. Friefer and M. Fernstrom, "Investigation of smart parking systems and their technologies," *Thirty Seventh International Conference on Information Systems, Dublin 2016*, 2016.

[28] T. Le, J. Famey, and S. Latre, "Evaluation of the ieee 802.11ah restricted access window mechanism for dense iot networks," *WoWMoM*, 2016. DOI: 10.1109/WoWMoM.2016.7523502. eprint: https://doi.org10.1109/WoWMoM.2016.7523502. [Online]. Available: https://doi.org10.1109/WoWMoM.2016.7523502.

[29] M. Matijevic and V. Cvjetkovic, “Overview of architectures with arduino boards as building blocks for data acquisition and control systems,” pp. 56–63, 2016. DOI: 10.1109/REV.2016.7444440. eprint: https://doi.org10.1109/REV.2016.7444440. [Online]. Available: https://doi.org10.1109/REV.2016.7444440.

[30] A. D. Africa and J. Velasco, “Development of a urine strip analyzer using artificial neural network using an android phone,” *ARPN Journal of Engineering and Applied Sciences*, vol. 12, pp. 1706–1713, Mar. 2017.

[31] M. Bahruddin, R. Kassim, and N. Buniyamin, “Development of fire alarm system using raspberry pi and arduino uno,” *ICEESE*, pp. 16–40, 2017. DOI: 10.1109/ICEESE.2013.6895040. eprint: https://doi.org/10.1109/ICEESE.2013.6895040. [Online]. Available: https://doi.org/10.1109/ICEESE.2013.6895040.

[32] Y. Joshi, P. Gharate, C. Ahire, N. Alai, and S. Sonavane, “Smart parking management system using rfid and ocr,” *IICECT*, 2017. DOI: 10.1109/ICESA.2015.7503445. eprint: https://doi.org/10.1080/23335777.2017.1358765. [Online]. Available: https://doi.org/10.1080/23335777.2017.1358765.

[33] W. Langsin, Q. Wu, M. Yang, B. Li, and P. Symeon, “Design and implementation of smart parking management system based on rfid and internet,” *IICECT*, 2017. DOI: 10.1109/ICCECT.2012.12. eprint: https://doi.org/10.1080/23335777.2017.1358765. [Online]. Available: https://doi.org/10.1080/23335777.2017.1358765.

[34] P. S. Saarika, K. Sandhaya, and T. Sudha, “Smart transportation system using iot,” *SmartTechCon*, vol. 2, no. 2, pp. 97–112, 2017, PMID: 27442303. DOI: 10.1109/SmartTechCon.2017.8358540. eprint: https://doi.org/10.1109/SmartTechCon.2017.8358540. [Online]. Available: https://doi.org/10.1109/SmartTechCon.2017.8358540.

[35] P. Sadhukhan, “An iot parking system for smart cities,” *ICACCI*, vol. 2, no. 2, pp. 97–112, 2017, PMID: 27442303. DOI: 10.1109/ICACCI.2017.8125982. eprint: https://doi.org/10.1109/ICACCI.2017.8125982. [Online]. Available: https://doi.org/10.1109/ICACCI.2017.8125982.

[36] E. Tsiropoulou, J. Baras, and P. Symeon, “Rfid-based smart parking management system,” *Cyber Physical Systems*, vol. 3, 2017. DOI: doi.org/10.1080/23335777.2017.1358765. eprint: https://doi.org/10.1080/23335777.2017.1358765. [Online]. Available: https://doi.org/10.1080/23335777.2017.1358765.

[37] G. Chandan, A. Jain, H. Jain, and Mohana, “Real time object detection and tracking using deep learning and opencv,” pp. 1305–1308, 2018. DOI: 10.1109/ICIRCA.2018.8597266. eprint: https://doi.org10.1109/ICIRCA.2018.8597266. [Online]. Available: https://doi.org10.1109/ICIRCA.2018.8597266.

[38] P. Chippalkatti, G. Kadam, and V. Ichake, “I-spark: Iot based smart parking system,” *ICACCT*, 2018. DOI: 10.1109/ICACCT.2018.8529541. eprint: https://doi.org/10.1109/ICACCT.2018.8529541. [Online]. Available: https://doi.org/10.1109/ICACCT.2018.8529541.

[39] P. Mangwani, “Smart parking system based on internet of things,” *International Journal of Applied Engineering Research*, vol. 13, pp. 10 281–10 285, 2018.

[40] J. Mesquita, D. Guimarães, C. Pereira, F. Santos, and L. Almeida, "Assessing the esp8266 wifi module for the internet of things," vol. 1, pp. 784–791, 2018. doi: 10.1109/ETFA.2018.8502562. eprint: https://doi.org10.1109/ETFA.2018.8502562. [Online]. Available: https://doi.org10.1109/ETFA.2018.8502562.

[41] S. Thakare and P. Bhagat, "Arduino-based smart irrigation using sensors and esp8266 wifi module," pp. 1–5, 2018. doi: 10.1109/ICCONS.2018.8663041. eprint: https://doi.org10.1109/ICCONS.2018.8663041. [Online]. Available: https://doi.org10.1109/ICCONS.2018.8663041.

[42] M. M. Abd Kadir, M. N. Osman, N. A. Othman, and K. A. Sedek, "Iot based car parking management system using ir sensor," *jcrinc*, 2019. doi: 10.24191/jcrinn.v5i2.151. eprint: https://doi.org/10.24191/jcrinn.v5i2.151. [Online]. Available: https://doi.org/10.24191/jcrinn.v5i2.151.

[43] J. Alcalá-Fdez, J. M. Alonso, C. Castiello, C. Mencar, and J. M. Soto-Hidalgo, "Py4jfml: A python wrapper for using the ieee std 1855-2016 through jfml," pp. 1–6, 2019. doi: 10.1109/FUZZ-IEEE.2019.8858811. eprint: https://doi.org10.1109/FUZZ-IEEE.2019.8858811. [Online]. Available: https://doi.org10.1109/FUZZ-IEEE.2019.8858811.

[44] S. Belkhala, S. Benhadou, K. Boukhdir, and Medromi, "Smart parking architecture based on multi-agent system," *IJACSA*, vol. 2, no. 2, pp. 97–112, 2019, PMID: 27442303. doi: 10.14569/IJACSA.2019.0100349. eprint: https://doi.org/10.14569/IJACSA.2019.0100349. [Online]. Available: https://doi.org/%2010.14569/IJACSA.2019.0100349.

[45] M. Chandran, N. Mahrom, T. Sabapathy, M. Jusoh, M. Osman, M. Yasin, N. Hambali, R. Jamaluddin, N. Ali, and Y. A. Wahab, "An iot based smart parking system," *ijisa*, 2019. doi: 10.1088/1742-6596/1339/1/012044. eprint: https://doi.org/10.1088/1742-6596/1339/1/012044. [Online]. Available: https://doi.org/10.1088/1742-6596/1339/1/012044.

[46] V. Moganaa, A. Vijayadevi, V. Prithee, and A. Vaishnavi, "Smart parking system using internet of things (iot)," *IJIREEICE*, 2019. doi: 10.17148/IJIREEICE.2019.7104. eprint: https://doi.org/10.17148/IJIREEICE.2019.7104. [Online]. Available: https://doi.org/10.17148/IJIREEICE.2019.7104.

[47] A. Pomaji, S. Boinmand, S. Wankhede, and B. Singh Pushpendra Dhakulkyar, "Smart parking management system," *IJCSE*, vol. 2, no. 2, pp. 97–112, 2019, PMID: 27442303. doi: 10.26438/ijcse/v7i5.12041208. eprint: :https://doi.org/10.26438/ijcse/v7i5.12041208. [Online]. Available: https://doi.org/10.26438/ijcse/v7i5.12041208.

[48] M. Reza, M. Ismail, and A. Rokoni, "Smart parking system with image processing facility," *ijisa*, 2019. doi: 10.5815/ijisa.2012.03.06. eprint: https://doi.org/10.5815/ijisa.2012.03.06. [Online]. Available: https://doi.org/10.5815/ijisa.2012.03.06.

[49] R. Widyasari, M. Z. Catur Candra, and S. Akbar, "Iot-based smart parking system development," *ICoDSE*, vol. 2, no. 2, pp. 97–112, 2019, PMID: 27442303. doi: 10.1109/ICoDSE48700.2019.9092707. eprint: https://doi.org/10.1109/ICoDSE48700.2019.9092707. [Online]. Available: https://doi.org/10.1109/ICoDSE48700.2019.9092707.

[50] A. .Ali Glowacz and C. Martis, "Iot based smart parking system using deep long short memory network," *electronics*, vol. 2, no. 2, pp. 97–112, 2020, PMID: 27442303. DOI: 10.3390/electronics9101696. eprint: https://doi.org/10.3390/electronics9101696. [Online]. Available: https://doi.org/10.3390/electronics9101696.

[51] D. Ashok, A. Tiwari, and V. Jirge, "Smart parking system using iot technology," *ETITE*, 2020. DOI: 10.1109/ic-ETITE47903.2020.457. eprint: https://doi.org/10.1109/ic-ETITE47903.2020.457. [Online]. Available: https://doi.org/%2010.1109/ic-ETITE47903.2020.457.

[52] A. Maharjan and A. Elocouemi, "Smart parking utilizing iot embedding fog computing based on smart parking architecture," *CITISIA*, 2020. DOI: 10.1109/CITISIA50690.2020.9371848. eprint: https://doi.org/10.1109/CITISIA50690.2020.9371848. [Online]. Available: https://doi.org/10.1109/CITISIA50690.2020.9371848.