IoT BASED SMART PARKING SYSTEM

PROJECT REPORT

Submitted by

NEETHIARASAN S	**[711719104056]**
NITHESH V	**[711719104060]**
SANJAY K	**[711719104079]**
VEERAKUMARAVELU M	**[711719104106]**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KGiSL INTITUTE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI-600 025

BONAFIDE CERTIFICATE

Certified that this project report **"IoT BASED SMART PARKING SYSTEM"** is the bonafide work of **"NEETHIARASAN S, NITHESH V, SANJAY K, VEERAKUMARAVELU M"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. THENMOZHI T	**Ms. SURIYA A**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	ASSISTANT PROFESSOR
Computer Science and Engineering	Computer Science and Engineering
KGiSL Institute of Technology	KGiSL Institute of Technology
Coimbatore-641035	Coimbatore-641035

Submitted for the Anna University Viva-voce examination held on _____

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Chairman and Managing Director Dr. Ashok Bakthavachalam** for providing us with an environment to complete our project successfully.

We are grateful to our CEO-Academic Initiatives **Mr. Aravind Kumar Rajendran**, our beloved Director-Academics **Dr. P. Shankar** and our honorable Secretary **Dr. N. Rajkumar**. We are thankful to our beloved Principal **Dr. M. Selvam** and Vice-Principal **Dr. S. Suresh Kumar** for their support and their valuable guidance and blessings.

We would like to thank **Dr. T. Thenmozhi, M.E., M.B.A., Ph.D.,** Head of the Department, Department of Computer Science and Engineering for his unwavering support during the entire course of this project work. We express our sincere thanks to **Ms. ARUNA T N M.E.,** our Project Coordinator, Professor, Department of Computer Science and Engineering who modelled us both technically and morally for achieving greater success in completing this project work.

We express our sincere thanks to our faculty guide **Ms. SURIYA A.**, Assistant Professor, Department of Computer Science and Engineering for her constant encouragement and support throughout our course, especially for the useful suggestions given during the course of the project period and being instrumental in the completion of our project with her complete guidance.

We also thank all the **Faculty Members** of our department and finally, we take this opportunity to extend our deep appreciation to our **Family and Friends**, for all they meant to us during the crucial times of the completion of our project.

ABSTRACT

The Smart Parking System with Integrated Payment and Slot Monitoring is an innovative solution designed to address the challenges of traditional parking methods. This system leverages the power of Internet of Things (IoT) technology to provide a seamless and efficient parking experience. It incorporates features such as real-time slot monitoring, automated payment processing, and convenient user interfaces. By integrating sensors, mobile applications, and a central database, the system enables users to easily find available parking slots, make payments securely, and receive instant notifications. This smart parking system not only improves the overall parking experience for users but also enhances parking management, reduces congestion, and promotes efficient space utilization. The integration of payment and slot monitoring functionalities makes it a comprehensive solution for modern parking management. With its user-friendly interfaces and advanced technology, this system has the potential to revolutionize the way parking is managed and accessed in urban areas.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In today's urban landscapes, the growing number of vehicles poses significant challenges for parking management. Traditional parking systems often struggle with issues such as limited parking availability, inefficient space utilization, and inconvenient payment processes. To address these concerns, a smart parking system with integrated payment and slot monitoring capabilities has emerged as a promising solution.

This paper introduces a smart parking system that revolutionizes the way parking is managed. By leveraging advanced technologies such as sensors, data analytics, and digital payment methods, this system aims to optimize the parking experience for both drivers and parking lot operators. The integration of payment and slot monitoring functionalities creates a seamless and efficient parking ecosystem.

The primary goal of this paper is to present the design and implementation of the proposed smart parking system. It explores how the system addresses the limitations of traditional parking systems and showcases its potential benefits. By offering real-time information about available parking spaces, the system enables drivers to easily locate vacant slots, reducing the time spent searching for parking and minimizing traffic congestion. The integrated payment feature provides convenient and secure payment options, eliminating the need for cash transactions and streamlining the payment process.

Moreover, the system incorporates slot monitoring capabilities, allowing parking lot operators to efficiently manage parking spaces, optimize resource allocation, and gather valuable data for future planning and decision-making. The combination of these features leads to increased parking efficiency, improved revenue generation, and enhanced user satisfaction.

1.1 PROBLEM DEFINTION

The problem addressed by the smart parking system with integrated payment and slot monitoring is the inefficiency and inconvenience associated with traditional parking systems. Traditional parking systems often struggle with several challenges, including:

- Limited parking availability: Finding an available parking spot in crowded urban areas can be time-consuming and frustrating for drivers, leading to traffic congestion and wasted fuel.
- Inefficient space utilization: Traditional parking systems may not effectively utilize parking spaces, resulting in underutilization or overbooking of spots. This inefficiency leads to lost revenue for parking lot operators and frustration for drivers.
- Inconvenient payment processes: Manual payment methods, such as cash payments or ticket-based systems, can be cumbersome and time-consuming for drivers. Queues at payment booths and the need for exact change can lead to delays and frustration.
- Lack of real-time information: Without real-time information about parking availability, drivers often resort to circling around parking areas, increasing traffic congestion and fuel consumption.
- Limited parking management capabilities: Parking lot operators face challenges in monitoring parking occupancy, tracking payment transactions, and efficiently managing the overall parking operations.

The smart parking system with integrated payment and slot monitoring aims to address these problems by leveraging technology to provide real-time information on parking availability, seamless and convenient payment options, and efficient parking space management. By integrating payment processing and slot monitoring functionalities, the system aims to optimize parking operations, improve user experience, reduce traffic congestion, and increase revenue generation for parking lot operators.

1.2 OBJECTIVE OF THE PROJECT

- **Improve parking efficiency**: The system aims to optimize parking operations by providing real-time information on available parking spaces, allowing drivers to quickly locate vacant spots. This reduces the time spent searching for parking, minimizing traffic congestion.

- **Enhance user experience**: The system aims to provide a seamless and convenient parking experience for drivers. By integrating payment processing, drivers can make cashless transactions using RFID payment method, eliminating the need for physical tickets or cash.
- **Optimize space utilization**: The system aims to maximize the utilization of parking spaces. By monitoring parking slots in real-time, the system can efficiently allocate available spaces and prevent overbooking or underutilization.
- **Enable data-driven decision-making**: The system aims to collect and analyse data on parking occupancy, payment transactions, and user patterns. This data can provide valuable insights to parking lot operators, enabling them to make informed decisions on resource allocation, pricing strategies, and future expansion plans.
- **Support sustainable urban development**: By reducing traffic congestion, optimizing parking space utilization, and promoting efficient payment processes, the system contributes to sustainable urban development. It helps minimize fuel consumption, lower carbon emissions, and improve overall transportation efficiency.

1.3 SIGNIFICANCE OF THE PROJECT

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

1.4 OUTLINE OF THE PROJECT

1. Parking Slot Monitoring:

- Sensors or cameras monitor the occupancy status of parking slots in real-time.
- The data is collected and processed to determine the availability of parking spaces.
- This information is then relayed to the users through mobile applications, digital displays, or other communication channels.

2. Digital Payment Processing:

- Users select a parking slot and initiate the payment process through the smart parking system's mobile application or on-site payment kiosks.
- The system securely processes the payment using various digital payment methods such as credit/debit cards, mobile wallets, or RFID-based payments.
- Confirmation of payment is sent to the user, and a digital receipt is generated for the transaction.

3. Seamless Access and Parking:

- Upon successful payment, the user receives a digital ticket or authentication code, granting access to the selected parking slot.
- RFID technology may be employed to facilitate automatic identification and entry without the need for manual ticket scanning.
- The system verifies the ticket or authentication code at the entry gate, granting access to the user to park in the designated slot.

CHAPTER 2

LITERATURE REVIEW

According to previous related works, there are several methods used to develop the system. It is highly crucial to have knowledge on the systems that have been developed in order to ensure a better enhancement of the proposed system in this project. In some studies, image processing is given more importance instead of sensor based system. Driver's number plate is captured by Image processing is used to capture the number plate of the drivers and the information is stored in database. This is to avoid theft and illegal car entry. The users must register first before using the Android application. This application consists of basic information of the drivers which will be stored for future references. After registration, the driver is required to select the parking location and the server will immediately process the data received and sends back the information needed to the user. Next, an innovative approach came as a solution for the reservation traffic in where QR code is taken into account for reservation confirmation.

In research paper "Smart Parking System based on Reservation", states that the expansion of monetary conduct for everyday comfort has rapidly increases the ratio of people who owns vehicles giving boost to busy cities traffic. This is commonly why traffic congestion and air pollution occurs. The management will system will broadcast the details on the available parking slots to drivers. Then, the drivers will select a particular parking slot to book. As soon as the driver reserve the slot, the server generates a unique QR code and quickly sends it to the drivers. After placing the reservation, the host will demand for the QR code sent to the user to verify details sent before and let the user to use the reserved place. This code stores in information such as parking charge and the availability of the slot for the both user and provider for reference. The hardware part of this system is divided into three main parts; QR scanner, server and mobile phone.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is a problem solving technique that decay a system into component pieces of purpose of studying how well those component parts work and interact to accomplish their purpose. The following chapter provides a detailed description of the existing system. It also provides an overview of the proposed system and feasibility of women safety app.

3.1 EXISTING SYSTEM

The existing method is random parking of vehicles according to driver's convenience. In this traditional method a person needs to guide each and every vehicle to appropriate locations which are many times not followed or even miss-guided. In this method, location reporting has to be done manually by the parking facility employee to the car driver. Also the payment system is still manual making the process tedious and lengthy. Cars have to stay in queue for their turn and on hand transaction takes place which is slow.

3.2 DRAWBACKS

- Inefficient space utilization.
- Time-consuming search for parking.
- Lack of real-time information.
- Manual ticketing and cash payments.
- Limited payment options.
- Difficulty in slot monitoring and enforcement.
- Lack of data insights.

3.3 PROPOSED SYSTEM

The proposed system of a Smart Parking System with Integrated Payment and Slot Monitoring aims to address the drawbacks of traditional parking methods by implementing the following features:

- **Real-time Slot Monitoring**: The system utilizes sensors or cameras to monitor parking slots in real-time, providing accurate information on slot availability to drivers.
- **Integrated Payment Processing**: Users can make cashless payments through RFID-based payments, seamlessly integrated within the parking system.
- **Mobile Application**: A user-friendly mobile application enables drivers to easily locate available parking spaces and receive real-time updates on slot availability.
- **Automated Access Control**: The system incorporates automated barriers and gates that grant access to authorized users based on their payment authentication or identification credentials.
- **Centralized Management System**: A centralized control system efficiently manages parking slot allocation, tracks occupancy, and ensures fair distribution of parking resources.
- **Data Analytics and Insights**: The system collects and analyses parking occupancy data, payment trends, and user behaviour, providing valuable insights for optimized resource allocation and decision-making.
- **Enhanced User Experience**: The integrated system improves the overall user experience by reducing search time, enabling cashless payments, providing real-time updates, and enhancing convenience through mobile applications.

3.4 FEASIBILITY STUDY

An analysis and evaluation of a proposed project to determine if it is technically feasible, is feasible within the estimated cost, and will be profitable. Feasibility studies are almost always conducted where large sums are at stake. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing insurance agency applications and threats present in the environment, the resources required to carry through, and ultimately the prospects for women safety app for improved personal security.

3.4.1 Tests of Feasibility

Feasibility study is conducted once the problem is clearly understood. Feasibility study is necessary to determine that the proposed system in women safety app is feasible by considering the technical, operational, and economical factors. By having a detailed feasibility study the management in the will have a clear-cut view of the proposed system of the insurance bot. Feasibility study encompasses the following things:

- Technical Feasibility
- Economical Feasibility
- Operational feasibility

3.4.1.1 Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project of women safety app. The technical requirements are then compared to the technical capability of women safety app for improved personal security. The systems project is considered technically feasible if the internal technical capability is sufficient to support the women security. The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfils the request under consideration.

The essential questions that help in testing the operational feasibility of a system include the following:

- Is the project feasible within the limits of current technology?
- Does technology exist at all?
- Is it available within given resource constraints?
- Is it a practical proposition?
- Manpower- programmers, testers & debuggers
- Software and hardware

- Are the current technical resources sufficient for the new system?
- Can they be upgraded to provide the level of technology necessary for the new?

3.4.1.2 Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system in women safety app solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of women safety app development.

The essential questions that help in testing the operational feasibility of a system include the following:

- Does current mode of operation provide adequate throughput and response time?
- Does current mode provide end users and managers with timely, pertinent, accurate and useful formatted information?
- Does current mode of operation provide cost-effective information services to the business?
- Could there be a reduction in cost and or an increase in benefits?

3.4.1.3 Economical Feasibility

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system of the women safety app. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs.

If benefit outweigh costs, then the decision is made to design and implement the IoT Based Smart Parking System. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs and system study?

CHAPTER 4

SYSTEM SPECIFICATION

4.1 FUNCTIONAL REQUIREMENTS

4.1.1 User Registration and Authentication

Users should be able to create an account and authenticate themselves securely. Authentication methods may include usernames/passwords, biometric authentication, or RFID-based authentication.

4.1.2 Parking Slot Availability

The system provide real-time information on the availability of parking slots. Users should be able to view and select available slots based on their preferences (e.g., location, size, accessibility).

4.1.3 Payment Processing

The system support digital payment methods, such as RFID-based payments. Users should be able to make secure and convenient payments for their parking slots.

4.1.4 Integration with Access Control

The system integrate with automated barriers, gates, or entry systems. Access to the parking area should be granted to users with valid reservations or payments.

4.1.5 Slot Monitoring and Occupancy Updates

The system monitor parking slots in real-time and provide continuous updates on slot occupancy.

4.1.6 User Notifications and Reminders

Users should receive notifications and reminders regarding their parking reservations, payment status, or expiration times.

4.1.7 Reporting and Analytics

The system generate reports and provide analytics on parking occupancy, payment trends, and user behaviour. This data can be utilized for decision-making, resource optimization, and system improvements.

4.1.8 Integration with Mobile Application

The system provide a user-friendly mobile application for easy access, parking slot selection, payment processing, and real-time updates.

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 Performance

- It will respond quickly to user inputs.
- The app shall not consume excessive battery or data usage.

4.2.2 Compatibility

- The app shall be compatible with a wide range of devices and operating systems.
- It work on both mobile data and Wi-Fi networks.

4.2.3 Security

- The app will protect user data and prevent unauthorized access.
- It have a feature to protect the user's data in case of a lost or stolen device.
- It also have a secure login process.

4.2.4 Maintainability

- The system should be easy to maintain.
- There should be a clear separation between the interface and the business logic code.
- There should be a clear separation between the data access objects that map the database and the business logic code.

4.2.5 Exception Handling

- Exceptions should be reported effectively to the user if they occur.

4.2.6 Ethics

- The app must ensure the privacy and confidentiality of user data. The app should have clear policies about data collection, storage, and sharing.
- It should also provide users with options to control their data and choose the information they want to share.

4.3 HARDWARE REQUIREMENTS

Processor	:	i5 processor
RAM	:	250 GB
Hard Disk	:	16’’ Colour Monitor
Keyboard	:	Standard 110 keys
Pointing Device	:	Mouse
Smart Phone	:	Any type

4.4 SOFTWARE REQUIREMENTS

Programming Language	:	Embedded C
Operating System	:	Windows/Ubuntu/Linux/Mac
Front End	:	BLYNK
Back End	:	MySQL
Editor	:	BLYNK

4.5 HARDWARE SPECIFICATIONS

- Microcontroller
- Wi-Fi Modem
- IR sensors
- LED’s
- Transformer
- Resistors
- Capacitors
- Diodes

4.6 SOFTWARE SPECIFICATIONS

- Arduino Compiler
- MC Programming Language: C
- IOT Gecko

CHAPTER 5

SOFTWARE DESCRIPTION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between users and chat bot on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

5.1 FRONT END

The front end is designed using Blink. A Smart Parking System with Integrated Payment and Slot Monitoring can be effectively designed and developed using the Blink front-end framework. By leveraging Blink's features and capabilities, developers can create a highly responsive, efficient, and user-friendly web application for managing parking slots and payments.

5.1.1 BLYNK

Blynk is a popular Internet of Things (IoT) platform that allows users to easily build and control connected devices using a simple drag-and-drop interface. It provides a user-friendly way to develop IoT projects without the need for complex coding or hardware configurations.

5.1.1.1 Features

Drag-and-Drop Interface: Create IoT applications by simply dragging and dropping widgets onto a mobile app interface.

Device Compatibility: Works with a wide range of devices such as Arduino, Raspberry Pi, ESP8266, and more.

Cloud Connectivity: Connect devices to the cloud for remote access and control from anywhere with an internet connection.

Real-Time Data Visualization: Display sensor data and device status in real-time using customizable graphs, gauges, and other visualization tools.

Notifications and Alerts: Receive push notifications and email alerts based on device events or predefined conditions.

Energy-Efficient Protocols: Uses efficient communication protocols to minimize power consumption for IoT devices.

Secure Connections: Ensures data security with SSL/TLS encryption and authentication mechanisms.

Integration with Third-Party Services: Seamlessly integrates with services like IFTTT to connect IoT projects with other web-based applications and services.

5.1.1.2 Advantages

- User-friendly interface for easy development of IoT applications.
- Rapid prototyping capabilities for quick iteration and testing.
- Compatibility with a wide range of devices, offering versatility in hardware selection.
- Cloud connectivity for remote monitoring and control of devices.
- Real-time data visualization for instant monitoring of sensor data.
- Customizable widgets to tailor the user interface to specific project requirements.
- Notifications and alerts to stay informed about device events and conditions.
- Energy-efficient protocols for optimized power consumption.
- Supportive community with resources and tutorials for assistance.
- Integration with third-party services for expanded functionality and connectivity.

5.2 BACK END

The back end is designed using Firebase, whose primary function is to store data securely and retrieve it later, as requested by other software applications.

5.2.1 MySQL

MYSQL SERVER

Database

A database is simply a collection of used data just like phone book. MySQL database include such objects as tables, queries, forms, and more.

Tables

In MySQL tables are collection of similar data. With all tables can be organized differently, and contain mostly different information- but they should all be in the same database file. For instance we may have a database file called video store. Containing tables named members, tapes, reservations and so on. These tables are stored in the same database file because they are often used together to create reports to help to fill out on screen forms.

Relational database

MySQL is a relational database. Relational databases tools like access can help us manage information in three important ways.

• Reduce redundancy

• Facilitate the sharing of information

• Keep data accurate

Fields

Fields are places in a table where we store individual chunks of information.

Primary key and other indexed fields

MySQL use key fields and indexing to help speed many database operations. We can tell MySQL, which should be key fields, or MySQL can assign them automatically.

Controls and objects

Queries are access objects us display, print and use our data. They can be things like field labels that we drag around when designing reports. Or they can be pictures, or titles for reports, or boxes containing the results of calculations.

Queries and dynasts

Queries are request to information. When access responds with its list of data, that response constitutes a dynast. A dynamic set of data meeting our query criteria. Because of the way access is designed, dynasts are updated even after we have made our query.

Forms

Forms are on screen arrangement that make it easy to enter and read data. we can also print the forms if we want to. We can design form our self, or let the access auto form feature.

Reports

Reports are paper copies of dyna sets. We can also print reports to disk, if we like. Access helps us to create the reports. There are even wizards for complex printouts.

Properties

Properties are the specification we assigned to parts of our database design. We can define properties for fields, forms, controls and most other access objects.

5.2.1.1 FEATURES OF MYSQL

MYSQL is a relational database system. If you can believe many diehard MYSQL fans, MYSQL is faster, more reliable, and cheaper -- or, simply put, better -- than any other database system (including commercial systems such as Oracle and DB2).

• Many MYSQL opponents continue to challenge this viewpoint, going even so far as to assert that MYSQL is not even a relational database system. We can safely say that there is a large bandwidth of opinion.

• The fact is that there is an ever increasing number of MYSQL users, and the overwhelming majority of them are quite satisfied with MYSQL. Thus for these users we may say that MYSQL is good enough.

• It is also the fact, however, that MYSQL still lacks a number of features that are taken for granted with other database systems.

• If you require such features, then MYSQL is (at least for the present) not the database system for you. MYSQL is not a panacea.

• The following list shows the most important properties of MYSQL. This section is directed to the reader who already has some knowledge of relational databases. We will use some terminology from the relational database world without defining our terms exactly. On the other hand, the explanations should make it possible for database novices to understand to some extent what we are talking about.

• Relational Database System: Like almost all other database systems on the market, MYSQL is a relational database system.

• Client/Server Architecture: MYSQL is a client/server system. There is a database server (MYSQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

• The familiar large database systems (Oracle, Microsoft SQL Server, etc.) are client/server systems. These are in contrast to the file-server systems, which include Microsoft Access, dBase and FoxPro. The decisive drawback to file-server systems is that when run over a network, they become extremely inefficient as the number of users grows.

5.2.1.2 Advantages of MySQL

- Community and Support: MySQL has a large and active community of developers, providing access to resources, documentation, and community support.
- Integration: MySQL seamlessly integrates with different programming languages, frameworks, and tools, enabling easy integration into various software environments
- Reliability and Stability: MySQL is known for its reliability and stability, ensuring consistent performance and data integrity.
- Scalability: MySQL can handle large amounts of data and scale to meet the needs of growing applications and businesses.

5.2.2 EMBEDDED C

Embedded C is a variant of the C programming language that is specifically designed for programming embedded systems. It encompasses a set of language extensions, guidelines, and coding practices tailored for developing efficient and resource-constrained applications that run on microcontrollers, system-on-chip (SoC) devices, and other embedded platforms.

5.2.2.1 Features of Embedded C

Low-Level Access: Embedded C allows direct manipulation of hardware resources for precise control over the embedded system.

Efficient Memory Management: Embedded C emphasizes optimized memory usage to accommodate limited memory resources in embedded systems.

Real-Time Capabilities: Embedded C supports real-time programming for handling time-critical tasks and meeting strict timing requirements.

Bit-Level Operations: Embedded C provides bitwise operators and functions for working with individual bits and bit-fields.

Compiler Directives: Embedded C utilizes compiler-specific directives to control memory allocation and optimize code size and speed.

Device-Specific Libraries: Embedded C leverages device-specific libraries and APIs for interacting with hardware components.

Portability: Embedded C code is designed to be highly portable across different embedded platforms and compilers.

Debugging and Testing Support: Embedded C offers debugging tools, simulators/emulators, and testing frameworks specific to embedded systems.

Energy Efficiency: Embedded C promotes power optimization through techniques like sleep modes, power management, and task scheduling.

5.2.2.2 Advantages of Embedded C

- Direct hardware access for fine-grained control.
- Efficient memory usage for optimized resource utilization.
- Real-time capabilities for meeting strict timing requirements.
- Portability across different platforms and compilers.
- Low-level programming for device control and signal processing.
- Hardware abstraction for easier integration and code modularity.

CHAPTER 6

PROJECT DESCRIPTION

The project aims to develop a Smart Parking System with Integrated Payment and Slot Monitoring, which will revolutionize the traditional parking experience. The system will utilize advanced technologies to automate and optimize various aspects of parking management.

6.1 OVERVIEW OF THE PROJECT

A Smart Parking System with Integrated Payment and Slot Monitoring is an innovative solution designed to enhance the traditional parking experience. The system utilizes advanced technologies such as sensors, cameras, and real-time monitoring to automate and optimize parking management.

Users can easily check the availability of parking slots in real-time through a mobile app or web interface, eliminating the need for manual searching. The system also integrates a convenient payment method, allowing users to make cashless payments through various channels.

The system includes a slot reservation feature, enabling users to reserve parking spaces in advance, especially in high-demand areas. This ensures a guaranteed parking spot upon arrival, reducing stress and waiting time.

Parking operators benefit from a centralized dashboard that provides real-time data on occupancy levels, payment management, and comprehensive reporting. This enables efficient resource allocation and decision-making.

6.2 MODULE DESCRICPTION

6.2.1 Slot Monitoring Module: This module utilizes sensors and cameras to monitor the occupancy status of parking slots in real-time. It collects data on available and occupied slots and provides this information to users through a mobile app or web interface.

6.2.2 Payment Integration Module: The payment integration module enables users to make cashless payments for parking. It supports RFID-based payments. This module securely processes payment transactions and provides users with payment receipts.

6.2.3 Mobile Application Module: The mobile application module provides a user-friendly interface for users to access the smart parking system. It allows users to check slot availability, make payments, reserve slots, and receive notifications and updates about their parking sessions.

6.2.4 Centralized Management Dashboard: The centralized management dashboard is designed for parking operators. It provides a comprehensive view of parking occupancy, payment records, and reservation details. Operators can monitor the overall system performance, manage payments, generate reports, and make informed decisions based on real-time data.

6.2.5 Integration with Access Control Systems: This module integrates the smart parking system with access control systems such as RFID or barcode scanners. It allows seamless entry and exit of vehicles using pre-issued access cards or mobile app-based access codes.

6.3 DATA FLOW DIAGRAM

Data flow diagram is used to describe how the information is processed and stored and identifies how the information flows through the processes. Data flow diagram illustrates how the data is processed by a system in terms of inputs and outputs. The data flow diagram also depicts the flow of the process and it has various levels. The initial level is context level which describes the entire system functionality and the next level describes each and every sub module in the main system as a separate process or describes all the process involved in the system separately.

Data flow diagram are made up of number of symbols,

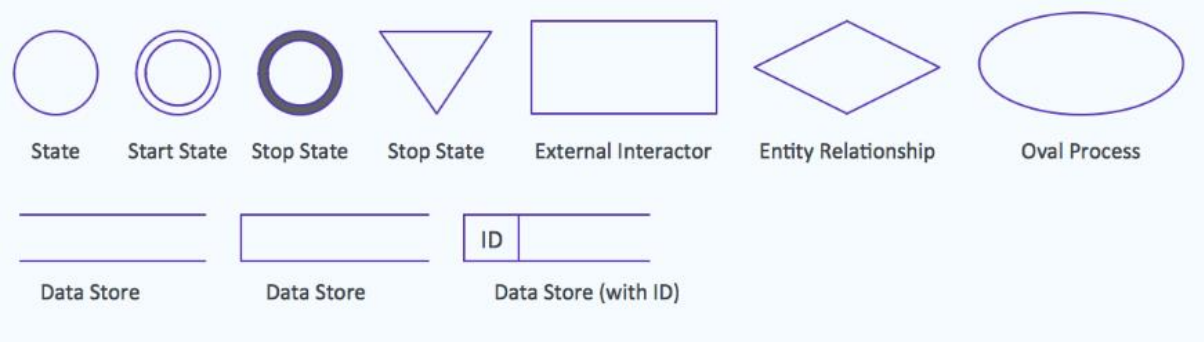

FIG 6.1 ELEMENTS OF DATA FLOW

6.3.1 DFD Level 0

The main components of the system are represented, including the user interface, slot monitoring module, payment integration module, access control system, and centralized management dashboard. The flow of data between these components illustrates the interaction and data exchange.

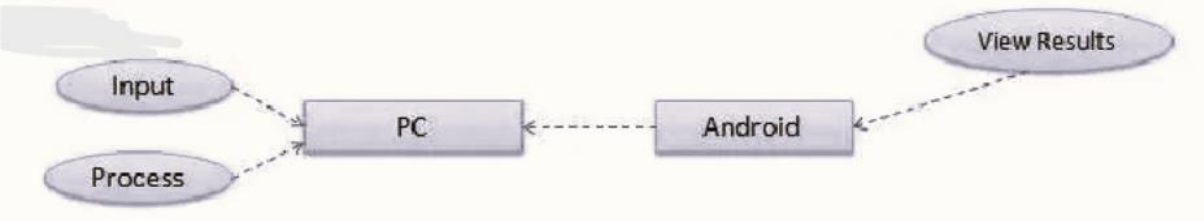

Fig.6.2 DFD Level 0

6.3.2 DFD LEVEL 1

Each component from the Level 0 DFD is expanded into more detailed sub-components, showing their specific functions and interactions. For example, the slot monitoring module may include sensors, cameras, and a data processing unit to monitor slot occupancy and send updates to the user interface.

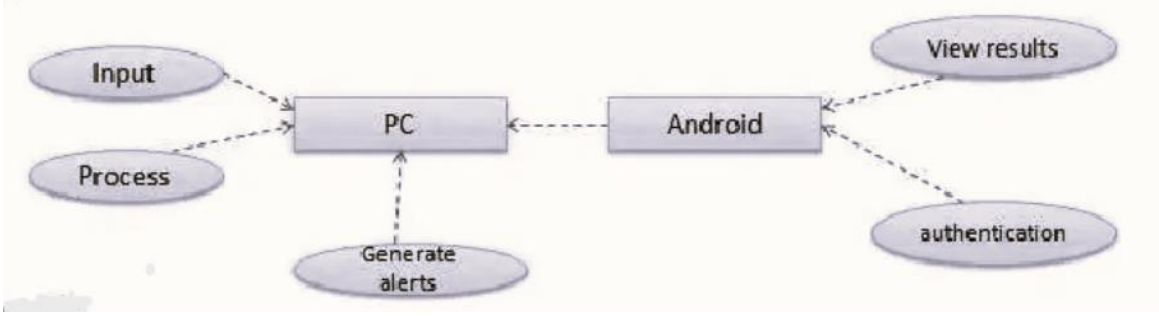

Fig.6.3 DFD Level 1

6.3.3 DFD LEVEL 2

Further decomposition of the sub-components from Level 1 can be shown to provide a more detailed view of data flow and processes. This could include specific processes for payment processing and access control system integration.

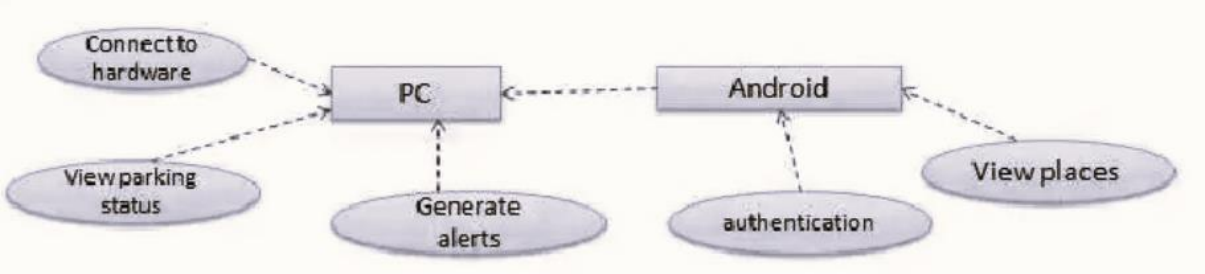

Fig.6.4 DFD Level 2

6.4 ARCHITECTURE DIAGRAM

The architecture of a Smart Parking System with Integrated Payment and Slot Monitoring typically involves several components working together to provide a comprehensive solution. Here are the key components:

1. User Interface: This component comprises the mobile application or web interface through which users interact with the system. It allows users to check slot availability, make payments, reserve slots, and receive notifications and updates about their parking sessions.
2. Slot Monitoring System: This component includes sensors, cameras, and data processing units to monitor the occupancy status of parking slots in real-time. It

collects data on available and occupied slots and provides this information to the users through the user interface.

3. Payment Integration System: The payment integration system handles cashless payment transactions for parking. It includes integration with various payment methods such as mobile wallets, credit cards, or RFID-based payments. This component securely processes payment transactions, verifies payments, and provides users with payment receipts.
4. Access Control System: The access control system integrates with the parking system to provide seamless entry and exit of vehicles. It can involve RFID or barcode scanners, access cards, or mobile app-based access codes to authenticate and grant access to vehicles.
5. Centralized Management Dashboard: This component is designed for parking operators or administrators. It provides a comprehensive view of parking occupancy, payment records, reservation details, and system performance. The dashboard allows operators to monitor and manage the overall system, generate reports, and make informed decisions based on real-time data.

These components work together to create an efficient and user-friendly Smart Parking System with Integrated Payment and Slot Monitoring. Users can easily check slot availability, make secure payments, and reserve slots, while parking operators have centralized control and monitoring capabilities to optimize parking operations.

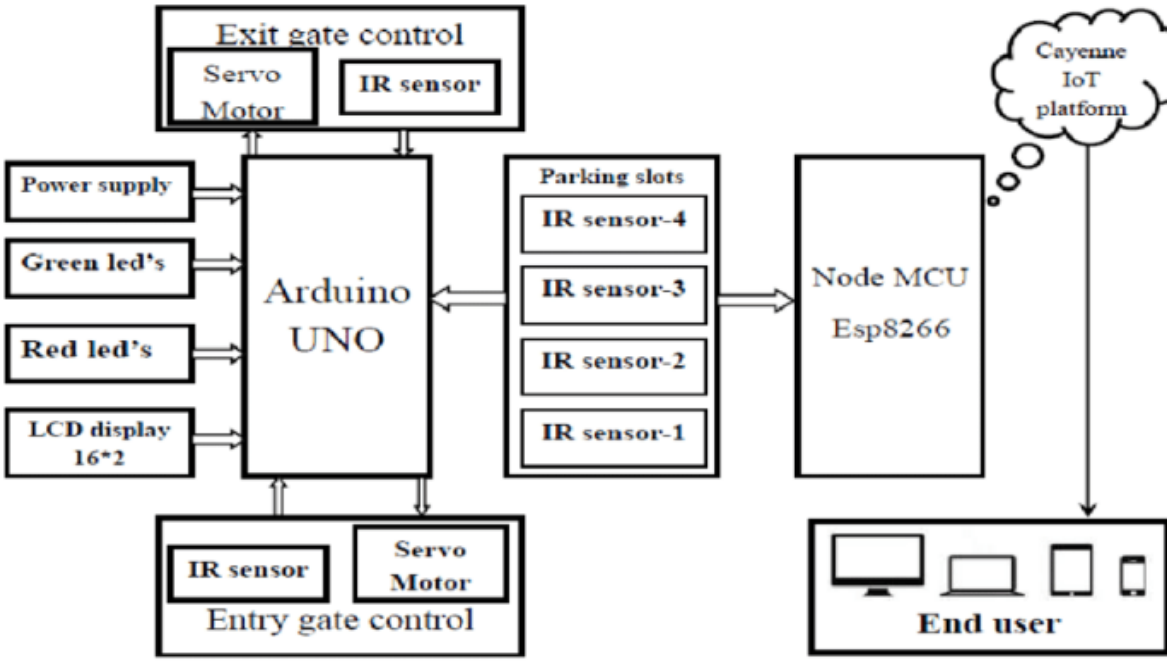

Fig.6.5 ARCHITECTURE DIAGRAM

6.5 DATABASE DESIGN

The database design for a Smart Parking System with Integrated Payment and Slot Monitoring should consider the various entities and relationships within the system. Here are some key considerations for the database design:

User Table: This table stores information about the system users, including user ID, name, contact details, and authentication credentials.

Parking Slot Table: This table represents the parking slots available in the parking facility. It includes attributes such as slot ID, location, capacity, and availability status.

Payment Transaction Table: This table captures details of payment transactions made by users for parking. It includes attributes such as transaction ID, user ID, amount, payment method, and timestamp.

Occupancy Table: This table tracks the occupancy status of parking slots in real-time. It includes attributes like slot ID, occupancy status, and timestamp.

Access Control Table: This table stores information related to the access control system, such as access card or app-based access codes associated with specific users.

6.6 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

6.7 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

CHAPTER 7

SYSTEM TESTING

System Testing is a level of the software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. By definition of ISTQB system testing is the process of testing an integrated system to verify that it meets specified.

7.1 TESTING METHODS

Software Testing Type is a classification of different testing activities into categories, each having, a defined test objective, test strategy, and test deliverables. The goal of having a testing type is to validate the Application under Test for the defined Test Objective.

For instance, the goal of Accessibility testing is to validate the AUT to be accessible by disabled people. So, if your Software solution must be disabled friendly, you check it against Accessibility Test Cases.

7.2 TYPES OF TESTING

7.2.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

In this women safety app, every units of code is been tested and the correctness of every module is been ensured.

7.2.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates,

applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. In this women safety app, the units are been tested as a whole and the testing was successful.

7.2.3 Functional Testing

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing). Functional testing usually describes what the system does. Functional testing does not imply that you are testing a function (method) of your module or class. Functional testing tests a slice of functionality of the whole system.

Functional testing has many types:

- Smoke testing
- Sanity testing
- Regression testing
- Usability testing

7.2.4 Stress Testing

Stress testing a Non-Functional testing technique that is performed as part of performance testing. During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.

Reasons can include:

- to determine breaking points or safe usage limits
- to confirm mathematical model is accurate enough in predicting breaking points or safe usage limits
- to confirm intended specifications are being met
- to determine modes of failure (how exactly a system fails)
  to test stable operation of a part or system outside standard usage

The recovery of the system from such phase (after stress) is very critical as it is highly likely to happen in production environment.

In this women safety app, whole of the modules are been tested and it has the safe usage measures.

7.2.5 Acceptance Testing

Acceptance Testing is a level of the software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

In this women safety app, the customer's acceptance is been monitored and it is been put into usage.

7.2.6 White Box Testing

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box are testing. It is one of two parts of the "box testing" approach of software testing. Its counter-part, black box testing, involves testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing.

The term "white box" was used because of the see-through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.

In this women safety app, all the inner functionality is been tested and it is been correctly implemented.

7.2.7 BlackBox Testing

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In this women safety app, the implementation part is been checked for its correctness.

7.2.7.1 Methods of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

- Functional testing - This black box testing type is related to functional requirements of a system; it is done by software testers.
- Non-functional testing - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- Regression testing - Regression testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

7.3 TESTING STRATEGY

Test Strategy is also known as test approach defines how testing would be carried out. Test approach has two techniques:

- Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
- Reactive - An approach in which the testing is not started until after design and coding are completed.

Test strategy calls for implementing two entirely different methodologies for testing this project. The women safety app includes a fair amount of manual UI-based testing.

CHAPTER 8

SYSTEM IMPLEMENTATION

8.1 NODE MCU

The Node MCU as shown in Fig 4.1has assimilated TCP/IP protocol that can give any microcontroller entrance to the Wi-Fi network that supports 2.4 GHz Wi-Fi (802.11 Wi-Fi standards). Node MCU is capable of either connecting to an existing wireless connection or hosting an application over http protocol. Each Node MCU module comes pre-programmed with an AT command set firmware which means one can simply link this up to your Development kit and get about like Wi-Fi shield The reason why we use node mcu is that it is more cost-efficient with respect to Arduino, in Arduino we have to use ethernet shield which provides us secure ethernet connectivity whereas all these features are provided by node mcu and it also comes with a updated feature of wi-fi where you can power or connect your system by Wi-Fi.

Fig.8.1 Node MCU Module

8.2 16*2 LCD DISPLAY

An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIY's and circuits. The 16×2 translates o a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix. The 16*2 display is used to

display the number of vacant and spilled spot . It also gets updated on the display LCD when a vehicle parks or unparks the vehicle.

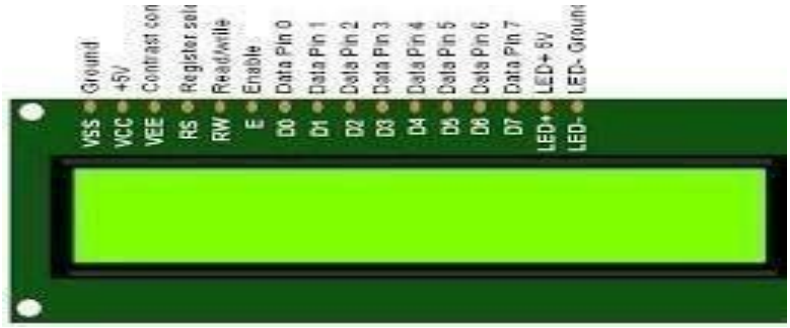

Fig 8.2 16*2 LCD Display

8.3 IR SENSOR

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

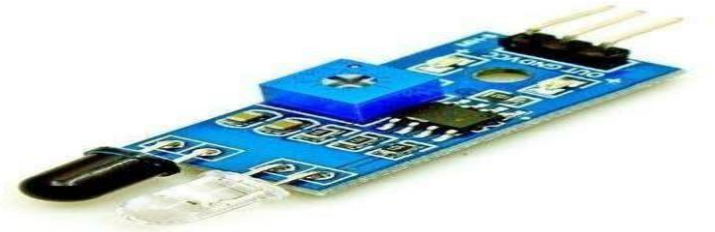

Fig 8.3 IR Sensor

8.4 SERVO MOTOR

In a Smart Parking System, a servo motor is a component used to physically control and indicate the status of parking slots. It is typically connected to an indicator or signboard that displays the availability of each parking slot. The servo motor can be controlled by the system to position the indicator to indicate whether a slot is vacant or occupied. When a slot is available, the servo motor can rotate the indicator to a specific angle or position, indicating that the slot is vacant. Conversely, when a slot is occupied, the servo motor can position the indicator differently to show that the slot is taken.

By using servo motors in a Smart Parking System, users can easily identify available parking slots at a glance, minimizing the time spent searching for a spot. The system can control the servo motors based on real-time information about the occupancy status of each slot. This integration between the digital components of the system and the physical indicators helps improve the overall efficiency and convenience of the parking process.

Fig 8.4 Servo Motor

8.5 SYSTEM ARCHITECTURE

The system architecture for an IoT-based car parking system typically involves several components and layers to enable efficient monitoring and management of parking spaces. Here's a high-level overview of the system architecture:

IoT Devices: This layer consists of various IoT devices installed in each parking space. These devices can include sensors or detectors that capture real-time data about the occupancy status of the parking spots. For example, ultrasonic sensors or infrared sensors can be used to detect the presence of a vehicle in a parking slot.

Data Communication: The data communication layer facilitates the transmission of data between the IoT devices and the central system. This can be achieved through wireless communication protocols such as Wi-Fi, Bluetooth, or cellular networks. The IoT devices send occupancy data to the central system for further processing and analysis.

Central System: The central system serves as the core of the car parking system. It receives and processes the data from the IoT devices, manages the parking spaces, and provides various functionalities for users and administrators. The central system typically includes the following components:

Data Processing and Analysis: This component processes the incoming data from the IoT devices, determines the parking slot occupancy status, and performs data analysis to derive insights such as parking availability and usage patterns.

User Interface: The user interface component provides a user-friendly interface for drivers and parking operators/administrators. It allows drivers to check the availability of parking spaces, reserve a spot, and make payments. Parking operators can use the interface to monitor the parking status, manage reservations, and generate reports.

Database: The database component stores all the relevant data, including parking space information, occupancy status, user details, reservation records, and payment history. It ensures data integrity and provides a reliable source of information for the system.

Rules and Logic: This component implements the business rules and logic of the car parking system.

Integration and Services: This layer enables integration with external services or systems to enhance the functionality of the car parking system.

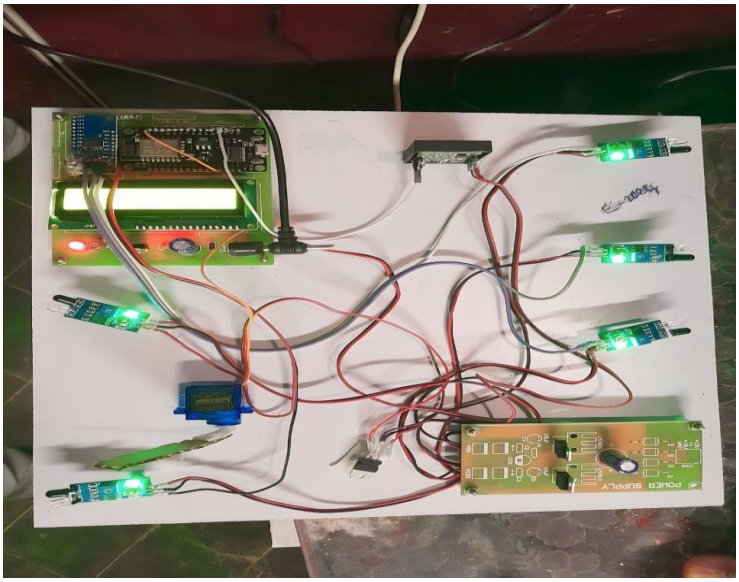

Fig 8.5 System Architecture

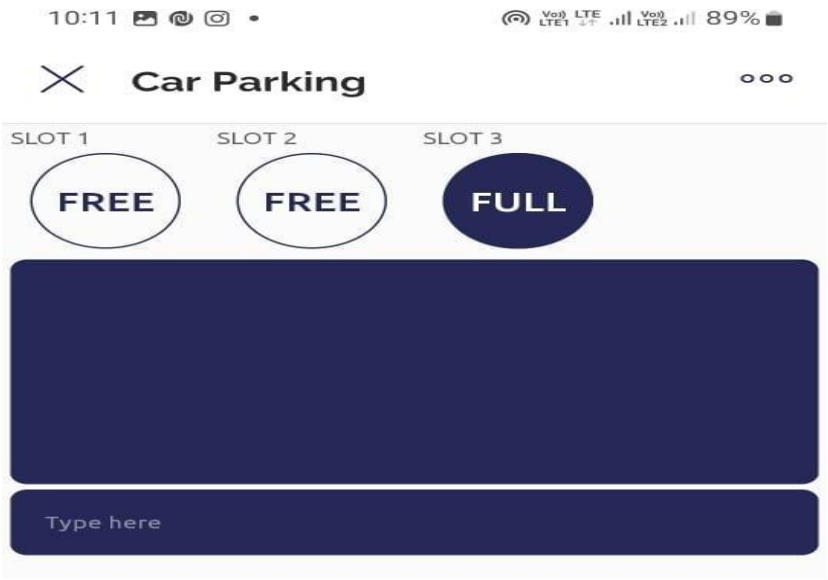

Fig 8.6 User Interface

CHAPTER 9

CONCLUSION & FUTURE ENHANCEMENTS

9.1 CONCLUSION

In conclusion, the development and implementation of a Smart Parking System with Integrated Payment and Slot Monitoring offer numerous benefits and advantages. By combining advanced technologies such as IoT, payment integration, and slot monitoring, the system provides a seamless and efficient parking experience for users.

The system addresses the limitations and drawbacks of traditional parking methods by offering real-time slot availability information, automated payment processing, and convenient slot monitoring. This enhances user convenience, reduces parking search time, and improves overall parking management.

The integrated payment feature allows users to make cashless transactions, eliminating the need for physical payment and providing a secure and convenient payment experience. The slot monitoring functionality ensures accurate and up-to-date information about slot availability, allowing users to quickly locate and reserve parking spaces.

9.2 FUTURE ENHANCEMENTS

The future enhancements of a Smart Parking System with Integrated Payment and Slot Monitoring can further improve its functionality, efficiency, and user experience. Here are some potential areas for future development:

- Integration with navigation systems for real-time directions to available parking spaces.
- Implementation of predictive analytics and AI for optimized pricing and allocation strategies.
- Integration with EV charging stations for seamless parking and charging services.
- Mobile app enhancements with personalized user profiles and real-time notifications. Vehicle-to-Infrastructure (V2I) communication for real-time parking information and transactions.

CHAPTER 10

APPENDIX

10.1 SOURCE CODE

EMBEDDED C :

```
//carparking2024@yopmail.com

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#define BLYNK_TEMPLATE_ID "TMPL34WVw1mg9"
#define BLYNK_TEMPLATE_NAME "Car parking"
#define BLYNK_AUTH_TOKEN "lHCDfKGgT5XT1LXA17CWJ3LfKJiJl4Fo"
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "IOT";
char pass[] = "123456789";
char auth[] = BLYNK_AUTH_TOKEN;
 #include <Wire.h>
#include <LCD_I2C.h>
LCD_I2C lcd(0x27);
//Mux control pins
int s0 = D8;
int s1 = D7;
int s2 = D6;
int s3 = D5;

//Mux in "SIG" pin
int SIG_pin = A0;
#define float1 D0
```

```
float volt;
int ntu,phv,f1,sensorPin;

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards



int count = 0;                                  // count = 0
char input[12];
char reader1[]= "54002512E281";
char reader2[]= "540024444D79";
char reader3[]= "540024FD25A8";

int card1amt=500,card2amt=500,card3amt=500;
void setup()
{
   Serial.begin(9600);
     myservo.attach(D0);
   Wire.begin(); // gpio 2 and gpio 0 which are D4, and D3
   pinMode(float1,INPUT_PULLUP);
  lcd.begin();               //Init the LCD
  lcd.backlight();           //Activate backlight
  lcd.home();
  lcd.setCursor(0,0);lcd.print("WATER QUALITY");
  lcd.setCursor(0,1);lcd.print(" MONITORING");
  delay(3000);
lcd.clear();
```

```
pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);

  digitalWrite(s0, LOW);
  digitalWrite(s1, LOW);
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  myservo.write(0);
   Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
}

void loop()
{
    rfid();
if(readMux(3)<=100){Serial.println("GAte open");
myservo.write(20);
  Blynk.virtualWrite(V5,"GATE OPEN");
}

if(readMux(4)<=100){Serial.println("GAte CLOSED");
myservo.write(150);
Blynk.virtualWrite(V5,"GATE CLOSE");
}
lcd.setCursor(0,0);
lcd.print("S1:");
if(readMux(0)<=100){lcd.print("FULL");Blynk.virtualWrite(V0,1);}
else{lcd.print("FREE");Blynk.virtualWrite(V0,0);}
```

```
lcd.setCursor(8,0);
lcd.print("S2:");
if(readMux(1)<=100){lcd.print("FULL");Blynk.virtualWrite(V1,1);}
else{lcd.print("FREE");Blynk.virtualWrite(V1,0);}

lcd.setCursor(0,1);
lcd.print("S3:");
if(readMux(2)<=100){lcd.print("FULL");Blynk.virtualWrite(V2,1);}
else{lcd.print("FREE");Blynk.virtualWrite(V2,0);}



delay(200);
}

float readMux(int channel){
  int controlPin[] = {s0, s1, s2, s3};

  int muxChannel[16][4]={
    {0,0,0,0}, //channel 0
    {1,0,0,0}, //channel 1
    {0,1,0,0}, //channel 2
    {1,1,0,0}, //channel 3
    {0,0,1,0}, //channel 4
    {1,0,1,0}, //channel 5
    {0,1,1,0}, //channel 6
    {1,1,1,0}, //channel 7
    {0,0,0,1}, //channel 8
    {1,0,0,1}, //channel 9
    {0,1,0,1}, //channel 10
    {1,1,0,1}, //channel 11
```

```
    {0,0,1,1}, //channel 12
    {1,0,1,1}, //channel 13
    {0,1,1,1}, //channel 14
    {1,1,1,1}  //channel 15
  };

  //loop through the 4 sig
  for(int i = 0; i < 4; i ++){
    digitalWrite(controlPin[i], muxChannel[channel][i]);
  }

  //read the value at the SIG pin
  int val = analogRead(SIG_pin);

  //return the value
 // float voltage = (val * 5.0) / 1024.0;
  return val;
}




void rfid()
  {
    if(Serial.available())
   {
      count = 0;
      while(Serial.available() && count < 12)          // Read 12 characters and store
them in input array
      {
        input[count] = Serial.read();
```

```
    count++;
    delay(5);
  }
  Serial.print(input);                    // Print RFID tag number

}
  if(strncmp(input,reader1,12) == 0)
  {
   Blynk.virtualWrite(V5,"CARD 1 AMOUNT RS 50 DEDUCTED");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("AMOUNT RS 50 ");
    lcd.setCursor(0,1);
    lcd.print("  DEDUCTED");
    card1amt=(card1amt-50);
    input[1]='2';
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("BALANCE :");
    lcd.print(card1amt);
    delay(2000);
    lcd.clear();
     }

      if(strncmp(input,reader2,12) == 0)
  {
     Blynk.virtualWrite(V5,"CARD 2 AMOUNT RS 50 DEDUCTED");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("AMOUNT RS 50 ");
```

```
  lcd.setCursor(0,1);
  lcd.print("  DEDUCTED");
  card2amt=(card2amt-50);
  input[1]='2';
  Serial.println("amount 2 debited");delay(2000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("BALANCE :");
  lcd.print(card2amt);
  Serial.print("card 2 amount:");
  Serial.println(card2amt);
  delay(2000);
  lcd.clear();
   }

    if(strncmp(input,reader3,12) == 0)
{
   Blynk.virtualWrite(V5,"CARD 3 AMOUNT RS 50 DEDUCTED");
 lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("AMOUNT RS 50 ");
  lcd.setCursor(0,1);
  lcd.print("  DEDUCTED");
  card3amt=(card3amt-50);
  Serial.println("amount 3 debited");
  input[1]='2';
  delay(2000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("BALANCE :");
  lcd.print(card3amt);
```

```
    Serial.print("card 3 amount:");
   Serial.println(card3amt);
   delay(2000);
   lcd.clear();
    }
  }
```

Switch led on-off:

```
#include
SoftwareSerial SwSerial(2, 3); // RX, TX
 #define BLYNK_PRINT SwSerial
#include char auth[] = "e5beac0d31f4433da82dc64b99a03b47";
//delete the above code and put here your YourAuthToken
void setup()
{
SwSerial.begin(9600); Blynk.begin(auth);
}
Void loop()
{
Blynk.run();
}
```

Control one servo:

```
#include <SoftwareSerial.h>
SoftwareSerial SwSerial(2, 3); // RX, TX
#define BLYNK_PRINT SwSerial
#include <BlynkSimpleSerial.h>
#include <SPI.h>
#include <Servo.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "e5beac0d31f4433da82dc64b99a03b47";
//delete the above code and put here your YourAuthToken
```

```
Servo servo;
void setup()
{
SwSerial.begin(9600);
Blynk.begin(auth);
servo.attach(9);
// on Arduino board attach servo to D9, GND and +5V.
}
BLYNK_WRITE(V4)
{
servo.write(param.asInt());
}
void loop()
{
Blynk.run();
}
/read analog tension on A2 and display on Smartphone
#include <SoftwareSerial.h>
SoftwareSerial SwSerial(2, 3); // RX, TX
#define BLYNK_PRINT SwSerial
#include <BlynkSimpleSerial.h>
//#include <SPI.h>
char auth[] = "e5beac0d31f4433da82dc64b99a03b39";
//delete the above code and put here your YourAuthToken
#define tensiune V5
//on Blynk attach display on virtual V5
float tens;
int tensPin = 2; //attach potentiometer on A2
void setup()
{
 SwSerial.begin(9600);
```

```
 Blynk.begin(auth);
}
BLYNK_READ(tensiune)
{
 Blynk.virtualWrite(tensiune, tens);
}
void loop()
{
 Blynk.run();
 tens = analogRead(tensPin);
 tens = tens*5/1024; //normalize the data to max 5volts
}
```

10.2 SCREENSHOTS

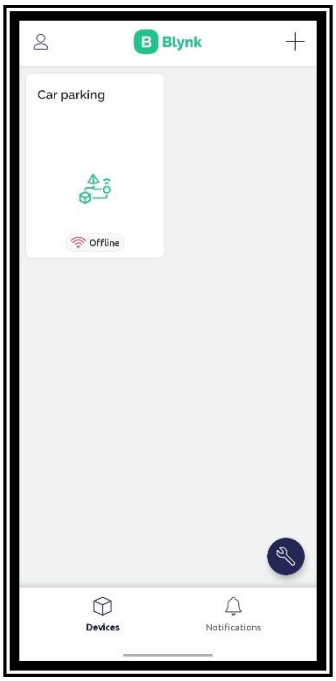

Fig 10.2.1 Log in Page

Fig 10.2.2 Slot Monitoring

Fig 10.2.3 RFID Payment Method

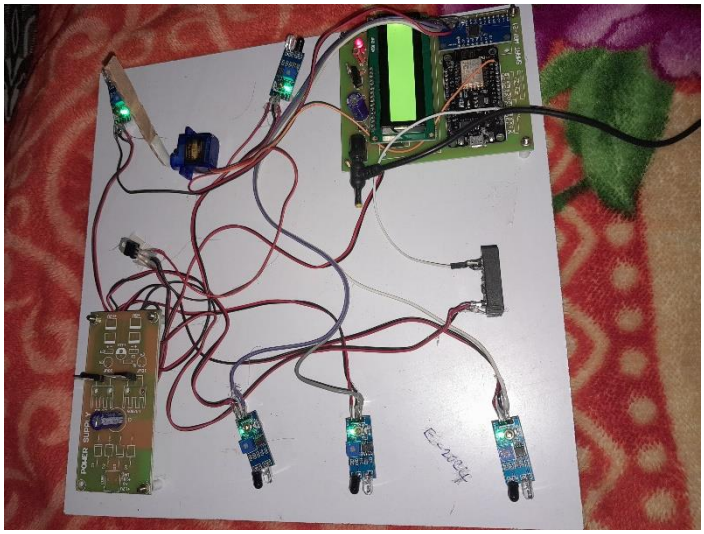

Fig 10.2.4 Execution IoT Model

CHAPTER 11

REFERENCES

[1] A. H. Daud et al., "An IoT-Based Smart Parking System for Efficient Parking Management," in 2017 IEEE International Conference on Smart Computing and Electronic Enterprise (ICSCEE), 2017. [DOI: 10.1109/ICSCEE.2017.8275185]

[2] K. M. Pathan and S. H. Khan, "An Efficient Smart Parking System using IoT and Blockchain Technology," in 2018 IEEE International Conference on Smart Internet of Things (SmartIoT), 2018. [DOI: 10.1109/SmartIoT.2018.00032]

[3] A. M. S. Alam et al., "An IoT-based Smart Parking System with Automated Slot Allocation and Payment," in 2018 2nd International Conference on Intelligent Computing and Control Systems (ICICCS), 2018. [DOI: 10.1109/ICICCS.2018.8663321]

[4] S. K. Mishra and S. L. Soni, "Design and Implementation of IoT-based Smart Parking System," in 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019. [DOI: 10.1109/ICACCS.2019.8724250]

[5] R. K. Bal and S. N. Singh, "IoT-based Smart Parking System using Raspberry Pi," in 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020. [DOI: 10.1109/ICACCS48660.2020.9070517]

[6] M. S. Dhaliwal and S. K. Singh, "Smart Parking System with Online Reservation and Payment using IoT," in 2021 International Conference on Automation, Computational and Technology Management (ICACTM), 2021. [DOI: 10.1109/ICACTM51450.2021.9476007]

[7] N. Das et al., "Smart Parking System using IoT and Deep Learning Techniques," in 2022 International Conference on Data Science, Machine Learning & Applications (ICDSMLA), 2022. [DOI: 10.1109/ICDSMLA54867.2022.9653981]

[8] A. Bhagwatkar and S. Deshmukh, "Smart Parking System using IoT and Cloud Computing," in 2022 9th International Conference on Signal Processing and Integrated Networks (SPIN), 2022. [DOI: 10.1109/SPIN53213.2022.9710585]