

A
Minor Project Report
on
**ADVANCED CAR PARKING
MANAGEMENT SYSTEM**

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering
in

Computer Engineering

to

**Kavayitri Bahinabai Chaudhari
North Maharashtra University, Jalgaon**

Submitted by

**Nilesh L. Tayade
Kamesh P. Surwade
Darshit D. Puranik
Harshada L. Salunke
Anurag R. Sonawane**

Under the Guidance of

Prof Dr. K. P. Adhiya



**DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
2023 - 2024**

**SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
DEPARTMENT OF COMPUTER ENGINEERING**

CERTIFICATE

This is to certify that the minor project entitled *Advanced Car Parking Management System*, submitted by

**Nilesh L. Tayade
Kamesh P. Surwade
Darshit D. Puranik
Harshada L. Salunke
Anurag R. Sonawane**

in partial fulfillment of the degree of *Bachelor of Engineering in Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

Date: April 26, 2024

Place: Jalgaon

**Prof Dr. K. P. Adhiya
Guide**

**Dr. Manoj E. Patil
Head**

**Prof. Dr. G.K.Patnaik
Principal**

Acknowledgements

We extend our deepest gratitude to all those who played a pivotal role in the successful completion of this project report. We are immensely thankful to the Almighty for granting us the strength and resilience throughout this endeavor.

Our heartfelt appreciation goes to Principal Prof. Dr. Girish. K. Patnaik for his unwavering support and provision of essential facilities that facilitated the completion of this Project report. We express our gratitude to everyone who made it possible for us to bring this report to fruition.

We owe a debt of gratitude to Prof. Dr. Krishnakant P. Adhiya, our esteemed Project Guide, for wholeheartedly dedicating himself to guiding us and assisting in achieving our objectives. His constant encouragement has been instrumental in maintaining our progress on the right track.

Our sincere thanks also go to the Project In-charge for their valuable suggestions and guidance that significantly contributed to the project's success. Additionally, the guidance provided by other supervisors deserves our heartfelt appreciation; their insightful comments and tips immensely enhanced our presentation skills.

Lastly, we wish to express our profound gratitude to our parents and friends. Their unwavering support and encouragement were indispensable, and without them, this successful completion wouldn't have been possible.

Nilesh L. Tayade
Kamesh P. Surwade
Darshit D. Puranik
Harshada L. Salunke
Anurag R. Sonawane

Contents

Acknowledgements	ii
Abstract	1
1 Introduction	2
1.1 Background	2
1.2 Motivation	3
1.3 Problem Definition	3
1.4 Scope	3
1.5 Objectives	4
1.6 Selection of Life Cycle Model Development	4
1.7 Organization Of Report	5
1.8 Summary	6
2 Project Planning And Management	7
2.1 Feasibility Study	7
2.1.1 Economical Feasibility	7
2.1.2 Operational Feasibility	8
2.1.3 Technical Feasibility	8
2.2 Risk Analysis	9
2.3 Project Scheduling	9
2.4 Effort Allocation	10
2.5 Cost Estimation	11
2.6 Summary	12
3 Analysis	14
3.1 Requirement Collection And Identification	14
3.1.1 Requirements Collection	14
3.1.2 Requirements Identification	14
3.2 Software Requirements Specifications (SRS)	15
3.2.1 Product Feature	15

3.2.2	Operating Environment	16
3.2.3	Assumption	17
3.2.4	Fuctional Requirements	17
3.2.5	Non-Fuctional Requirements	18
3.2.6	External Interfaces (User, Hardware, Software, Communication) . . .	19
3.3	Summary	19
4	Design	21
4.1	System Architecture	21
4.2	Data Flow Diagram	22
4.2.1	Level 0 DFD	22
4.2.2	Level 1 DFD	23
4.2.3	Level 2 DFD	24
4.3	UML Diagrams	25
4.3.1	Use Case Diagrams	26
4.3.2	Sequence Diagrams	27
4.3.3	Component Diagrams	28
4.3.4	Activity Diagram	28
4.4	Summary	29
5	Coading/Implementation	31
5.1	Summary	49
6	Result and Discussion	50
7	Conclusion and Future Work	52
Bibliography		54

Abstract

In this report, a solution is presented towards the main issue and problems of vehicle parking in places as the resources such as vehicles have been popular among the people along with its usage. Due to the population, several resources, technologies, and entities are on its peak from which vehicles are more in common for day to day activity. People face the problem in order to find an appropriate place while they are parking their vehicles in places where they go, visit, and enjoy. This report is presenting a solution towards this issue by using the emerging technique Mobile Phones using the platform of Web application so the system can be automated eco-friendly and efficiently used by the end-users.

Chapter 1

Introduction

The new Car Parking Management System addresses the challenges of city parking. It simplifies and speeds up parking using advanced technology, benefiting both drivers and parking attendants.

As our reliance on cars grows, we need improved parking methods. Traditional ways are no longer sufficient, but this new system solves that issue. It's a modern solution that meets our increasing need for safer, more efficient, and easier car parking.

This system isn't just a fix for current problems; it's also prepared for the future. By employing innovative technology, it doesn't just enhance parking now but also prepares us for what lies ahead. It represents a significant shift in how we manage parking in bustling cities, making the process more straightforward and well-organized for everyone.

1.1 Background

The Car Parking Management System arose due to rising challenges in city parking caused by more cars and fewer spaces. Cities faced congestion and drivers struggled to find parking spots. Experts from different fields saw these problems and teamed up. They looked at current parking systems and saw they weren't user-friendly or efficient.

Their collaboration birthed the Car Parking Management System. It aimed not just to solve current parking issues but to predict and handle future parking needs. They delved into research on user habits and traffic.

Their goal wasn't just a parking fix but a whole new parking experience. The system aimed to use space better, improve security, offer real-time info, and be easy for everyone. They also cared about reducing traffic jams and pollution.

They worked hard, testing and refining the system to make sure it worked well in all city settings. The result? A smarter, future-ready Car Parking Management System set to change how we park cars in cities, making it simpler, safer, and more efficient for everyone involved.

1.2 Motivation

The motivation behind the Smart Car Parking System arises from urbanization and the escalating challenges associated with parking. Traditional parking methods face difficulties coping with the expanding urban demand. Manual management techniques are time-consuming and can lead to inefficient utilization of parking spaces, causing delays for both users and operators. Moreover, conventional systems lack real-time data and insights, posing challenges in making informed decisions regarding parking availability.

1.3 Problem Definition

The current car parking systems suffer from inefficiencies and lack real-time information, leading to traffic congestion and inconvenience. Traditional methods, like valet parking with tokens and manual records, are outdated and struggle during critical situations such as police inquiries or vehicle thefts.

To address these issues, the Advanced Car Parking Management System (ACPMS) is proposed. It aims to optimize space usage, enhance security, streamline payments, and promote environmental sustainability. By integrating technologies like sensors, data analytics, mobile apps, and smart payments, ACPMS offers a user-centric solution for urban parking challenges.

The goal is to reduce traffic congestion, environmental impact, and improve user satisfaction by providing a seamless and efficient parking experience. This modern approach eliminates towing issues and fines, ensuring secure parking without disruptions.

1.4 Scope

Our Car Parking Management System extends beyond parking to offer a comprehensive mobility solution.

It serves various industries, including commercial, residential, and public sectors.

The system is highly scalable, meeting the growing demands of modern urban areas. It has potential for advanced data analysis and business intelligence, aiding in decision-making and urban planning.

1.5 Objectives

- The primary goal is to enhance parking efficiency and reduce urban congestion.
- Real-time data and insights support informed decision-making and parking optimization.
- A key focus is customer satisfaction, ultimately increasing revenues for operators.
- These objectives align with our vision of convenient, sustainable, and secure urban parking.

1.6 Selection of Life Cycle Model Development

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds.

These builds are provided in iterations. Each iteration typically lasts from about one to three weeks, Every iteration involves cross functional teams working simultaneously on various areas like -

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing
- Acceptance Testing

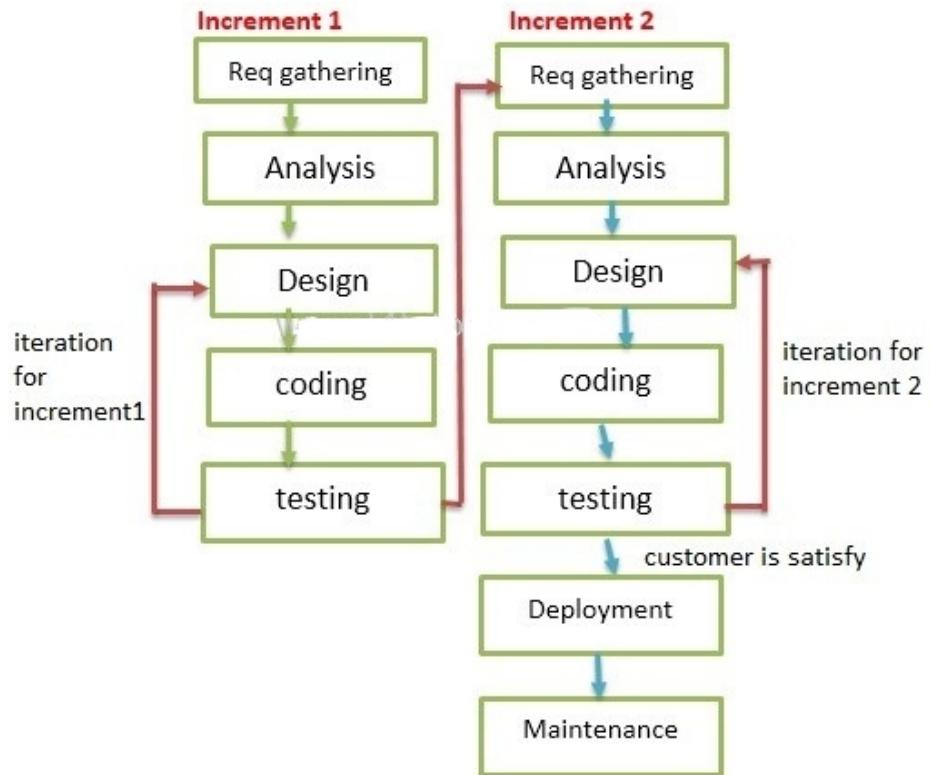


Figure 1.1: Agile Model

1.7 Organization Of Report

Chapter 1: Introduction - This section provides a general introduction of the project being submitted and shall include write up on: parking management system, the Project Definition, Scope, the Objectives to be achieved, selection of Life Cycle Model on the basis of analysis of gathered requirements, users participation, etc.

Chapter 2: Project Planning and Management - This section includes the Feasibility Study that summarizes results of the analysis and evaluations conducted to review the proposed solution and for the purpose of identifying if the project is really feasible, cost-effective and profitable. Risk analysis could include legal/contractual risks, environmental risks, revenue risks, project management risks, regulatory risks etc. A project schedule communicates what tasks need to get done and which organizational resources will be allocated to complete those tasks in what timeframe. The Gantt chart shows the project scheduling details and the effort allocation among the participants. This chapter also contains the cost

estimation analysis focus on the cost estimates, budget for the project, and means of financing and phasing of expenditure.

Chapter 3 : Analysis - This chapter focuses on the gathering of requirements/data and their identification. The Software Requirements Specification (SRS) document lists sufficient and necessary requirements, functional and non-functional requirements, their operating environment, product features, etc. For the project development.

Chapter 4 : Design - This chapter contains important aspects like the System Architecture, Data Flow Diagrams (DFDs), UML diagrams which contains the Blueprints of the system. They highlight the architecture, flow and sequential execution of the events.

1.8 Summary

The introduction to the Car Parking Management System project begins with establishing the Background, providing contextual information on the need for efficient parking solutions. The Motivation behind the project is outlined, emphasizing the driving factors for developing an advanced parking management system. The Problem Definition highlights existing challenges in parking management, setting the stage for the project's relevance. The Scope delineates the boundaries and functionalities of the proposed system. Objectives are defined to articulate the specific goals and outcomes the project aims to achieve. The Selection of the Life Cycle Model for Development is justified, outlining the chosen approach for project execution. The organization of the report is clarified, offering a structured overview of how information will be presented throughout the documentation.

In the next chapter, we will discuss about Project Planning And Management.

Chapter 2

Project Planning And Management

2.1 Feasibility Study

The feasibility study is carried out to test whether the proposed system is worth being implemented. Feasibility study is a test of system proposed regarding its work ability, its impact on the organization ability to meet user needs and effective use of resources.

The key consideration involve in the feasibility study are:

- Economical feasibility
- Operational Feasibility
- Technical Feasibility

2.1.1 Economical Feasibility

Economic feasibility refers to the evaluation of a proposed project's ability to generate economic benefits that outweigh its costs and justify its implementation. It involves analyzing the financial viability, potential returns, and overall economic justification of a project or investment.

The primary focus of our feasibility study is to evaluate the economic viability of the proposed Smart Car Parking System. Its practicality hinges on its affordability and cost-effectiveness. Since this project centers on software, it requires only a standard computer with the necessary specifications, eliminating the need for additional internet or specialized systems.

The key advantage of this system lies in its accessibility and affordability. By avoiding costly hardware and specialized infrastructure, we ensure that the solution remains financially feasible for all users. This approach not only promises cost savings for end-users but also fosters widespread adoption and usability, aligning with our goal of providing an economically viable and accessible parking management solution for everyone.

2.1.2 Operational Feasibility

Operational feasibility is the measure of how well a proposed project aligns with the operational capabilities and constraints of an organization, assessing whether it can be successfully implemented, integrated into existing processes, and sustained over time. It involves evaluating practical considerations such as compatibility with current systems, availability of resources, impact on organizational structure, and the ability to meet technical requirements, with the aim of determining the project's viability from an operational perspective.

Operational feasibility for the Advanced Car Parking Management System relies on its adept execution of essential tasks for stakeholders. Protecting individual privacy, the system provides summarized statistics, ensuring data confidentiality while delivering valuable insights.

Its design ensures smooth operations for drivers seeking spots and facility managers overseeing parking areas. By furnishing aggregated data sans personal details, the system retains operational feasibility. Employing advanced technologies without specifying any singular algorithm ensures accuracy and reliability, fortifying the system's effectiveness in managing parking spaces efficiently.

2.1.3 Technical Feasibility

Technical feasibility is the assessment of whether a proposed project or system can be successfully developed, implemented, and operated using the available technology, expertise, and resources. It involves evaluating the compatibility of the project with existing technical infrastructure, determining the availability of required technology and skills, and assessing the overall technical achievability of the proposed solution. The primary objective is to ascertain whether the project can be technically executed within the constraints and capabilities of the organization.

The assessment aims to gauge the company's technical proficiency for the project. It involves checking the hardware, software, and system needs. The portal uses PHP and Django for backend, alongside MySQL for the database. For the front end, it relies on HTML, CSS, and JavaScript.

All the required hardware and software are easily accessible, confirming the system's technical feasibility. Moreover, the Advanced Car Parking Management System doesn't need any extra hardware beyond personal systems for its operation.

2.2 Risk Analysis

Risk analysis is the process of identifying, assessing, and prioritizing potential risks or uncertainties associated with a project, decision, or activity. It involves the systematic examination of potential events or situations that could have adverse effects on the achievement of objectives. The goal of risk analysis is to provide a structured framework for understanding, managing, and mitigating risks, helping organizations make informed decisions and enhance their ability to navigate uncertainties effectively.

Risk analysis for the web app in the Advanced Car Parking Management System involves potential technical glitches impacting functionality.

System downtimes, cybersecurity threats, or server issues might disrupt operations. Furthermore, reliance on internet connectivity poses a risk for users in areas with poor connectivity.

Mitigation strategies include robust cybersecurity measures, regular system maintenance, and alternative solutions for internet-dependent functionalities.

2.3 Project Scheduling

Project scheduling is the process of planning, organizing, and documenting the sequence of tasks, activities, milestones, and resources required to achieve specific project objectives within a defined timeframe. It involves creating a timeline that outlines when each task or activity should start and finish, as well as their interdependencies. The purpose of project scheduling is to provide a structured framework for project management, allowing teams to allocate resources efficiently, monitor progress, and ensure that the project is completed on time.

The project schedule for the web app in the Advanced Car Parking Management System entails a phased approach. It starts with requirement gathering and analysis, followed by design, development, testing, and deployment phases.

Each phase allocates specific time frames, ensuring a systematic progression. Regular checkpoints and reviews are scheduled to monitor progress and address any deviations.

This structured approach aims to adhere to deadlines and milestones while accommodating potential adjustments for unforeseen challenges, ensuring a smooth and timely delivery of the web app. (It is important to note, however, that the schedule evolves over time) .

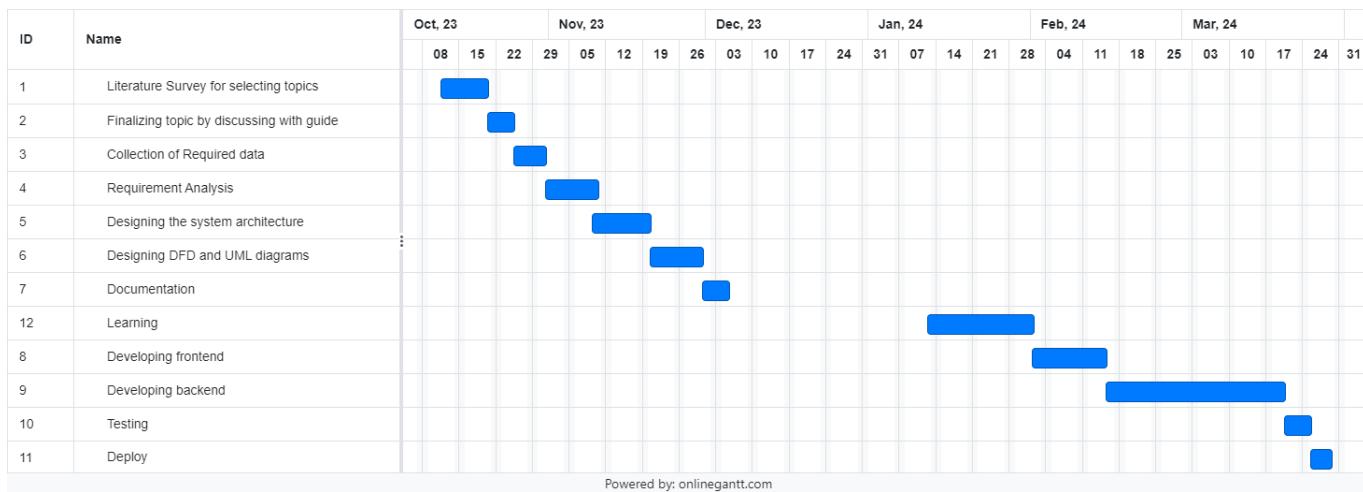


Figure 2.1: Project Scheduling

2.4 Effort Allocation

Effort allocation refers to the distribution or assignment of resources, particularly time and manpower, to various tasks or activities within a project or organization. It involves determining the proportion of effort that should be dedicated to different components or stages of a project to achieve overall objectives efficiently. Effort allocation is a critical aspect of project management, strategic planning, and resource optimization, ensuring that resources are used effectively to meet specific goals and deadlines.

Project means teamwork, Project is developed by combination of effort of team. So whole project is divided into modules and number of modules is allotted to team members.

The modules like planning, coding, gathering information, designing, implementing, combining, etc. all the modules are distributed to each member of the group itself and the implementing and coding of the project is done by all the members together.

	Nilesh Tayade	Kamesh Surwade	Harshada Salunkhe	Anurag Sonawane	Darshit Puranik
Literature Survey for selecting topics	•	•	•	•	•
Finalizing topic by discussing with guide	•	•	•	•	•
Collection of required data	•	•	•	•	•
Requirement analysis	•	•	•	•	•
Designing the system architecture	•				•
Designing DFD and UML diagrams	•				•
Documentation	•	•			•
Learning	•	•	•	•	•
Developing Frontend		•			•
Developing Backend	•		•	•	
Testing	•		•	•	
Deploy	•				

2.5 Cost Estimation

Cost estimation is the process that takes direct costs, indirect costs and other factors into account, and calculates a budget that meets the financial commitment necessary for a successful project. The following table shows the rough cost estimation for the proposed system w.r.t COCOMO Model.

COCOMO (Constructive Cost Model) is a widely used software cost estimation model developed by Barry Boehm. It provides a set of equations for estimating effort, duration, and staffing for software projects. There are three variations of COCOMO: Basic COCOMO, Intermediate COCOMO, and Detailed COCOMO.

Certainly, here are the simplified formulas and approximate calculations using Intermediate

COCOMO for a Car Parking Management System project:

For your case, assuming $a = 2.4$, $b = 1.05$, $c = 2.5$, $d = 0.38$, and $EAF = 1$ (no adjustments), and using the previously mentioned $KLOC = 1.5$, we can calculate:

1. Effort Estimation:

$$\text{Effort} = 2.4 \times (KLOC)^{1.05} \times 1$$

$$E = 2.4 \times (1.5)^{1.05} \times 1$$

$$E \approx 3.18 \text{ person-months}$$

2. Development Time Estimation:

$$\text{Development Time} = 2.5 \times (\text{Effort})^{0.38}$$

$$D = 2.5 \times (3.18)^{0.38}$$

$$D \approx 6.07 \text{ months}$$

3. Staffing Estimation:

$$\text{Number of People (Staffing)} = \frac{\text{Effort}}{\text{Development Time}}$$

$$N = \frac{3.18}{6.07}$$

$$N \approx 0.52 \text{ person}$$

Keep in mind that the number of people is fractional because COCOMO provides an average number and real-world situations may require rounding up to the nearest whole person. Additionally, these calculations are estimates, and actual results may vary based on specific project characteristics and team dynamics.

2.6 Summary

The car parking management system project begins with a thorough Feasibility Study, examining economic, operational, and technical aspects to ensure viability. An Economical Study delves into financial considerations, assessing costs and benefits. Operational Feasibility evaluates how seamlessly the system integrates into existing operations, while Technical Feasibility ensures compatibility with available technology. Risk Analysis identifies and addresses potential project risks, enhancing overall success. Project Scheduling establishes a timeline for milestones, promoting an organized development process. Effort Allocation

strategically assigns tasks among team members, optimizing workforce capabilities. Cost Estimation calculates financial resources needed, providing a comprehensive overview of project expenses. These elements collectively form a robust foundation for successful planning and management of the car parking management system.

In the next chapter, we will discuss about Analysis of Advance Car Parking Management System.

Chapter 3

Analysis

3.1 Requirement Collection And Identification

Requirement collection is the process which is used to gather, analyze, and documentation and reviews the requirements. Requirements describe what the system will do in place of how.

3.1.1 Requirements Collection

Collecting Parking Spot Data: The project gathered information about available parking spots. We collected details like where the spots are, if they're occupied or free, and how easy they are to access. We got this data from city records, satellite images, and by visiting the parking areas ourselves. The aim was to create a complete and accurate database. This database helps users find parking spots quickly when using the system.

3.1.2 Requirements Identification

here are the requirement identifications presented pointwise with simplified language and correct grammar:

- User Authentication: Determine the need for secure access methods to ensure only authorized users can utilize the system.
- Spot Reservation System: Identify the necessity for a feature allowing users to reserve parking spots in advance through the system.
- Real-time Spot Availability: Establish the requirement for instant updates on parking spot availability to help users find open spots quickly.
- User-Friendly Interfaces: Specify the importance of easy-to-use interfaces for both users and administrators to navigate the system effortlessly.

- Robust Database: Determine the need for a strong database to store accurate and comprehensive parking spot information.
- Scalability: Recognize the importance of the system's ability to adapt and grow as the demand for parking management increases.
- Responsiveness: Ensure the system's swift and efficient performance to meet user expectations and avoid delays.
- Integration with Payment Gateways: Consider the integration of payment methods to facilitate seamless transactions within the system.

These identified requirements form the basis for developing a functional and efficient Car Parking Management System.

3.2 Software Requirements Specifications (SRS)

It provides requirements, needs of project and those things which help to complete project. System requirement describe a system from a technical perspective, which describe the essential characteristics of the hardware and software that will meet those needs. It specifies the capabilities, capacities and characteristics of the system in both Qualitative and Quantitative terms.

3.2.1 Product Feature

1. **Parking Spot Availability:** Display real-time availability of parking spots for users to identify open spaces.
2. **Reservation System:** Allow users to reserve parking spots in advance through the system.
3. **User Authentication:** Implement secure login methods to ensure authorized access to the system.
4. **User-Friendly Interface:** Design intuitive interfaces for both users and administrators to navigate the system effortlessly.
5. **Database Management:** Utilize a robust database to store accurate and comprehensive parking spot information.
6. **Notification System:** Provide alerts or notifications regarding spot availability or reservation confirmations to users.

7. **Integration with Payment:** Include a secure payment gateway for users to make parking reservations or payments within the system.
8. **Scalability:** Design the system to adapt and expand with the growing demand for parking management solutions.

These product features serve as the core functionalities of the Car Parking Management System, aiming to deliver an efficient and user-friendly experience for both users and administrators.

3.2.2 Operating Environment

The Car Parking Management System operates optimally under the following environment:

Hardware Requirements

- Minimum Processor: Dual-core 2 GHz or higher
- RAM: 4 GB or higher
- Storage: Minimum 100 GB HDD/SSD space
- Network: Ethernet or Wi-Fi connectivity
- Display: Minimum resolution of 1280x720 pixels

Software Requirements

- Operating System: Windows 10, macOS 10.14+, Ubuntu 18.04+ or compatible
- Web Browsers: Google Chrome (latest), Mozilla Firefox (latest), Safari (latest), Microsoft Edge (latest)
- Database: MySQL 8.0+ or compatible
- Programming Language: PHP and JavaScript

Other Requirements

- Internet connectivity for real-time updates and notifications
- Secure HTTPS connection for data transmission
- Regular maintenance and updates for optimal performance and security

It is essential to ensure that the Car Parking Management System is deployed in an environment meeting these requirements for its efficient operation.

3.2.3 Assumption

- **Reliable Internet Connectivity Assumption:** Users are assumed to have consistent and reliable internet access for effective system utilization.
- **User Awareness Assumption:** Users are presumed to possess basic technological familiarity, enabling them to navigate the system independently.
- **Parking Spot Sensor Assumption:** The assumption relies on the availability and integration of sensors for real-time data collection on parking spot occupancy.
- **Database Accuracy Assumption:** It is assumed that the initial database provided with parking spot information is accurate and regularly updated.
- **System Compatibility Assumption:** The system is assumed to be compatible with commonly used devices and web browsers for easy user access.
- **Data Privacy Measures Assumption:** Adequate data privacy measures are presumed to be in place to protect user information and transactions.
- **Administrative Access Assumption:** Administrators are assumed to have appropriate access and permissions to manage the system efficiently.
- **Payment Processing Assumption:** The assumption relies on smooth and secure transaction processing within the system through integrated payment gateways.

These are the assumptions lay the foundation for the successful deployment and operation of the Car Parking Management System, subject to further adjustments or validations as the project progresses.

3.2.4 Functional Requirements

Functional requirements are the functions which are expected from the software or platform. Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behaviour.

The functional requirements of the Car Parking Management System are as follows:

- **User Registration and Authentication:** Secure account creation and verification for user access.

- **Parking Spot Availability:** Real-time display of available spots across different locations.
- **Spot Reservation:** Advance booking of parking spots via the system.
- **Spot Identification:** Easy spot location and identification features for users.
- **Notification System:** Alerts for spot availability and reservation updates.
- **Payment Integration:** Secure payment gateway for reservation transactions.
- **Administrative Dashboard:** User-friendly interface for overseeing system activities.
- **Reporting and Analytics:** Generation of insights on usage and system performance.
- **System Maintenance:** Regular updates and database management for system reliability.

These requirements collectively define the anticipated functions and behavior of the Car Parking Management System, aiding in system design and development.

3.2.5 Non-Fuctional Requirements

Non-functional Requirement is mostly quality requirement. That stipulates how well the portal does, what it has to do. Other than functional requirements in practice, this would entail detail analysis of issues such as availability, security, usability and maintainability.

Non-functional requirements are as follows: The non-functional requirements for the Car Parking Management System are as follows:

- **Availability:** Ensure the system maintains high availability, minimizing downtime to offer continuous user access.
- **Security:** Implement robust security measures to protect user data, employing encryption, authentication, and access control protocols.
- **Usability:** Provide an intuitive and user-friendly interface for easy navigation and accessibility across various user skill levels.
- **Scalability:** Design the system to scale effectively, accommodating increased users and parking spot data without performance degradation.
- **Reliability:** Demonstrate consistent reliability by delivering accurate parking spot information and functionalities.

- **Performance:** Optimize system performance for quick response times and efficient processing of user requests.
- **Compliance:** Ensure adherence to legal and regulatory standards regarding data privacy, parking regulations, and payment security.
- **Maintainability:** Enable easy maintenance and updates, facilitating swift bug fixes, feature enhancements, and system upgrades.

These non-functional requirements are essential for maintaining the system's quality, reliability, and effectiveness in operating the Car Parking Management System.

3.2.6 External Interfaces (User, Hardware, Software, Communication)

The Car Parking Management System requires specific hardware and software interfaces for its operation:

- **Hardware Requirements:**
 - **Processor:** Intel Core i3 processor or higher
 - **RAM:** 4 GB or higher
 - **Input Device:** User-operated interface such as keyboards or touchscreens
 - **Output Device:** Monitor or display for presenting parking spot information
- **Software Requirements:**
 - **Backend:** PHP(for server-side scripting and database management)
 - **Frontend:** HTML, CSS, JavaScript (for user interface and interaction)
 - **Database:** MySQL or similar (for storing parking spot information and user data)
 - **Development Tools:** IDEs like Visual Studio Code and Xammp for coding and debugging

3.3 Summary

In the Project Analysis phase for the Car Parking Management System, Requirement Collection and Identification are pivotal. Requirement Collection involves gathering essential information for the project, while Requirement Identification pinpoints specific needs and

functionalities. This lays the foundation for the Software Requirement Specification (SRS). The SRS outlines crucial aspects such as Product Features, Operation Environment, Assumptions, Functional Requirements, Non-functional Requirements, and External Interfaces (User, Hardware, Software, Communication). These elements collectively define the comprehensive set of specifications guiding the development of the system, ensuring a clear and structured framework for the Car Parking Management System project.

In the next chapter, we will discuss about the Design about Car Parking Management System.

Chapter 4

Design

System design provides the understanding and procedural details necessary for implementing the system. Design is an activity concerned with making major decisions, often of a structural nature. Design builds coherent, well planned representations of programs that concentrate on the interrelationships of parts at the higher level and the logical operations involved at the lower levels. Software design is the first of the three technical activities designs, coding and test which are required to build and verify the Web Application.

4.1 System Architecture

The System Architecture provides the details of how the components or modules are integrated. Figure 4.1 is indicating the system architecture of the Advanced Car Parking Management System. This architecture will give complete description of input and outputs of each process.

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages.

The purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other.

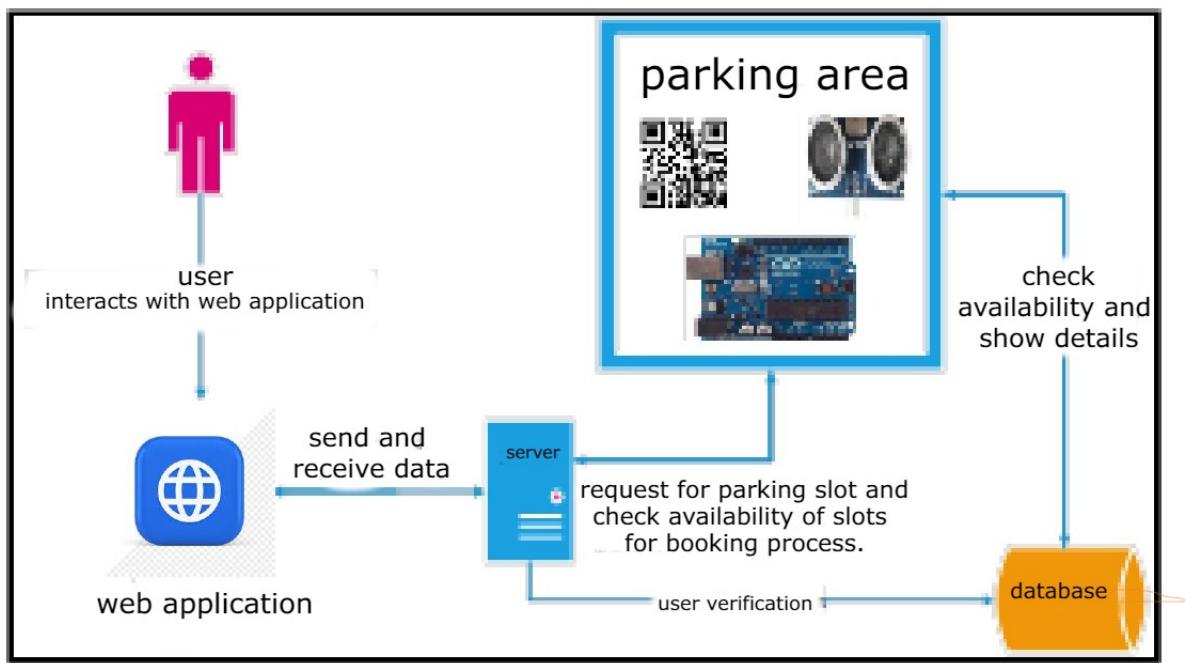


Figure 4.1: System Architecture

4.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the ‘flow’ of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

4.2.1 Level 0 DFD

Level 0 contains one input and one output. The system provides information to the user means system is input and the user is output. Figure 4.2 shows Level 0 DFD of project.

It is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows. As this is the basic and very first level of dfd data is processed and send to system.

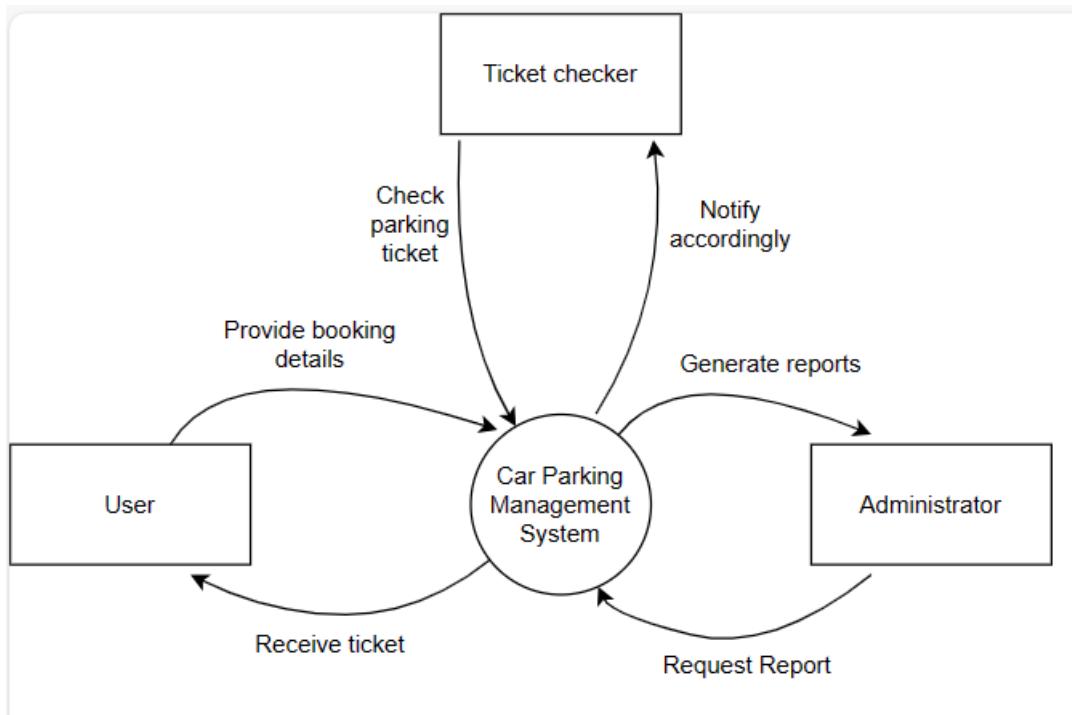


Figure 4.2: Level 0 Data flow diagram

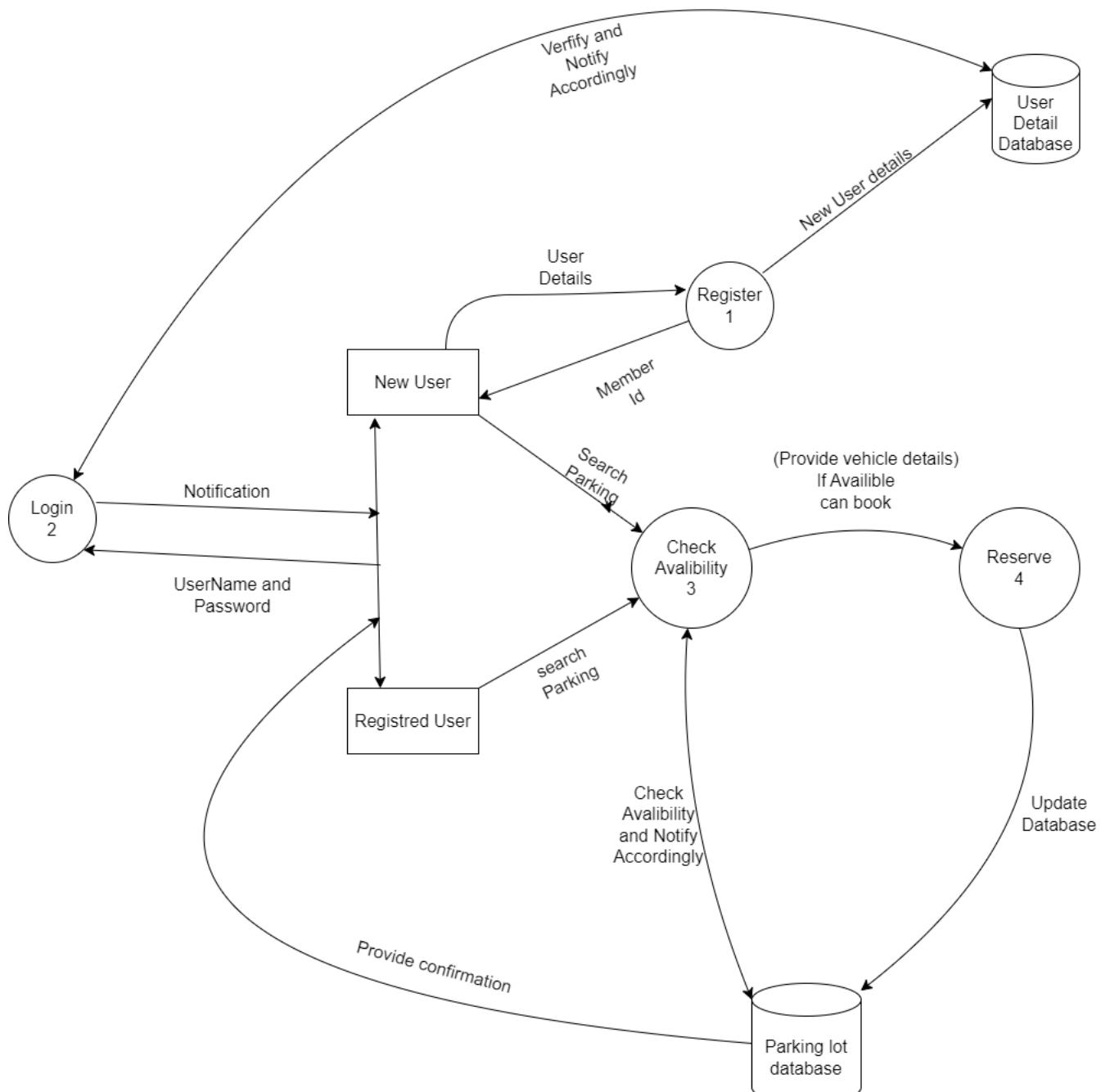
4.2.2 Level 1 DFD

A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an "exploded view" of the context diagram. Figure 4.3 shows Level 1 DFD of project.

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes.

The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information. It provides a more detailed view of the Context Level Diagram.

Here, the main functions carried out by the system are highlighted as we break into its subprocesses.



DFD LEVEL 1

Figure 4.3: Level 1 Data flow diagram

4.2.3 Level 2 DFD

A level 2 data flow diagram offers a more detailed look at the processes that make up an information system than a level 1 DFD does. It can be used to plan or record the specific makeup of a system. Figure 4.4 shows Level 2 DFD of project. This DFD is for admin side.

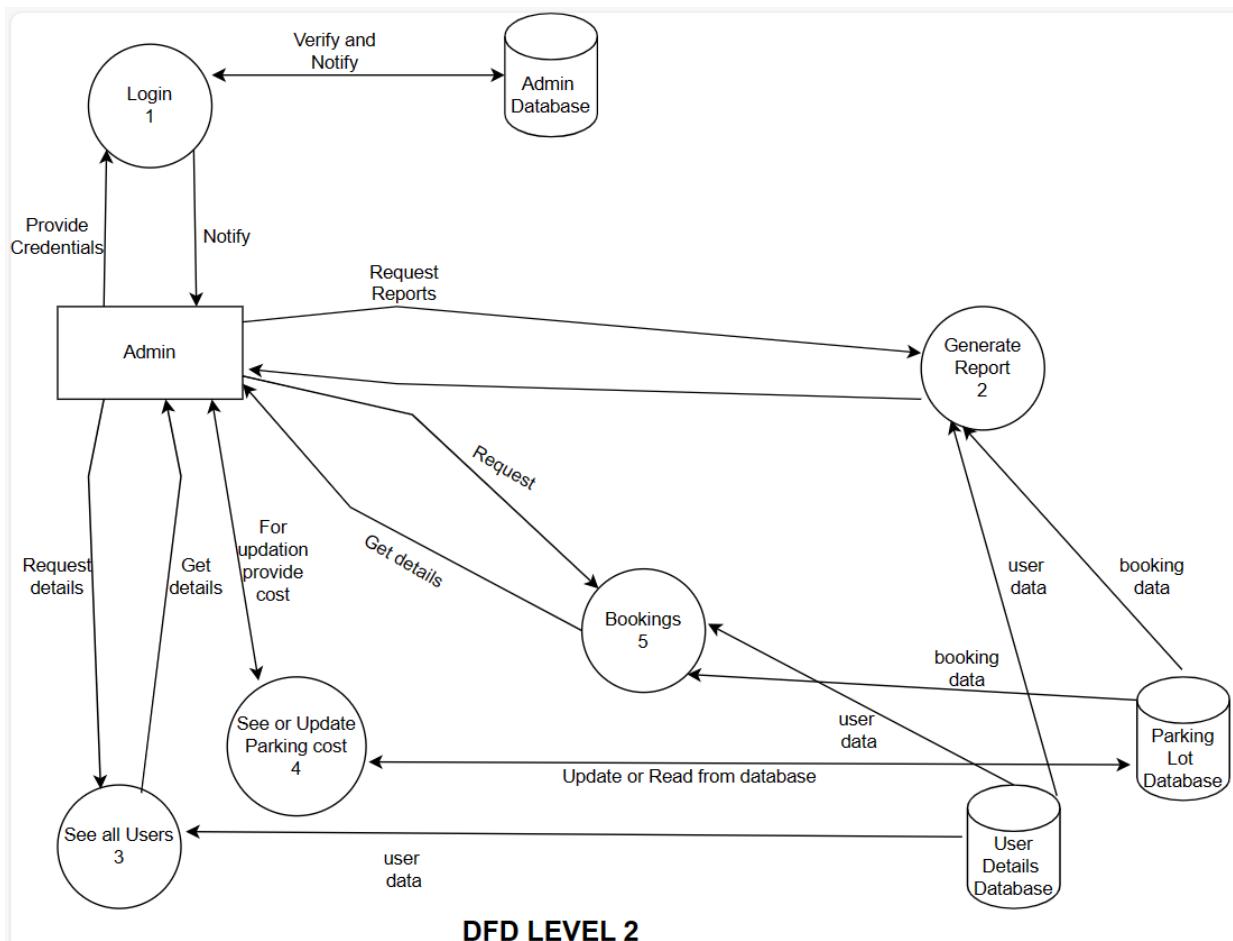


Figure 4.4: Level 2 Data flow diagram

4.3 UML Diagrams

The UML is a language for:

- Visualizing: The structures which are transient can be represented using the UML
- Specifying: The UML addresses the specification of all the important analysis, design and implementation decisions that must be made in developing and deploying a software-intensive system
- Constructing: The UML is not a visual programming language, but its models can be directly connected to a variety of programming languages
- Documenting: The UML addresses the documentation of a system's architecture and all of its details

4.3.1 Use Case Diagrams

A use-case model describes a system's functional requirements in terms of use cases. It is a model of the system's intended functionality (use cases) and its environment (actors). Use cases enable you to relate what you need from a system to how the system delivers on those needs. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, a number of use case diagrams are used.



Figure 4.5: Use Case Diagram

4.3.2 Sequence Diagrams

Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. Figure 4.6 shows sequence diagram for Use cases.

A simplified sequence diagram for a typical reservation process within the system. This sequence diagram illustrates the interactions between a user, the parking system, and a parking lot during the reservation process:

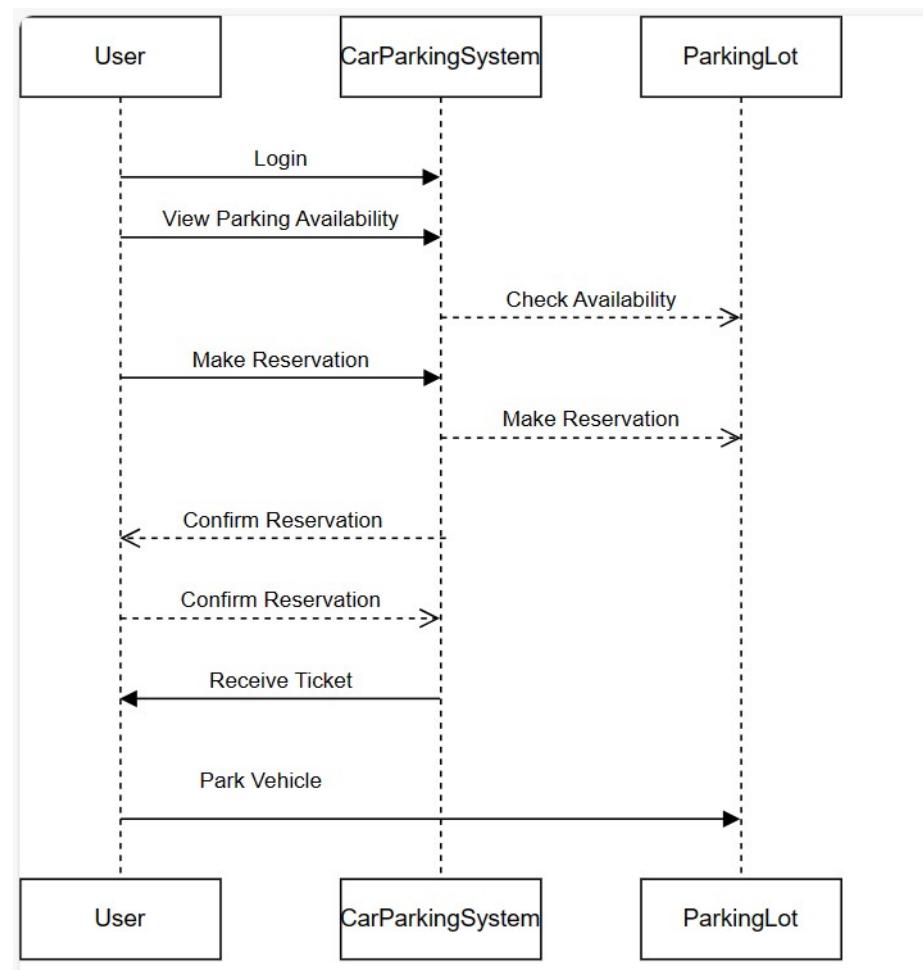


Figure 4.6: Sequence Diagram 2

In this sequence diagram: The user logs in to the system. The user views parking availability. The Car Parking System checks parking availability with the ParkingLot. The user makes a reservation through the system. The Car Parking System sends the reservation request to the ParkingLot. The Car Parking System confirms the reservation and notifies

the user. The user confirms the reservation. The user parks their vehicle at the ParkingLot.

4.3.3 Component Diagrams

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development. Figure 4.14 shows component diagram for car parking management system.

In Unified Modeling Language (UML), a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.



Figure 4.7: Component Diagram Of Car parking management system

4.3.4 Activity Diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing various decision paths.

The process starts with the user, who is either registered or not registered. If the user is registered, they can book a parking slot by specifying the location information.

The platform will then send a QR code to the user's phone. The user can then scan the QR code to enter the parking lot.

If the user is not registered, the platform will notify them of all the possibilities for parking.

The user can then choose a parking slot and the platform will send them a QR code. The user can then scan the QR code to enter the parking lot.

Once the user enters the parking lot, the platform will keep sending them notifications about the status of their parking. For example, the platform may notify the user when their parking time is about to expire.

The process ends when the user leaves the parking lot.

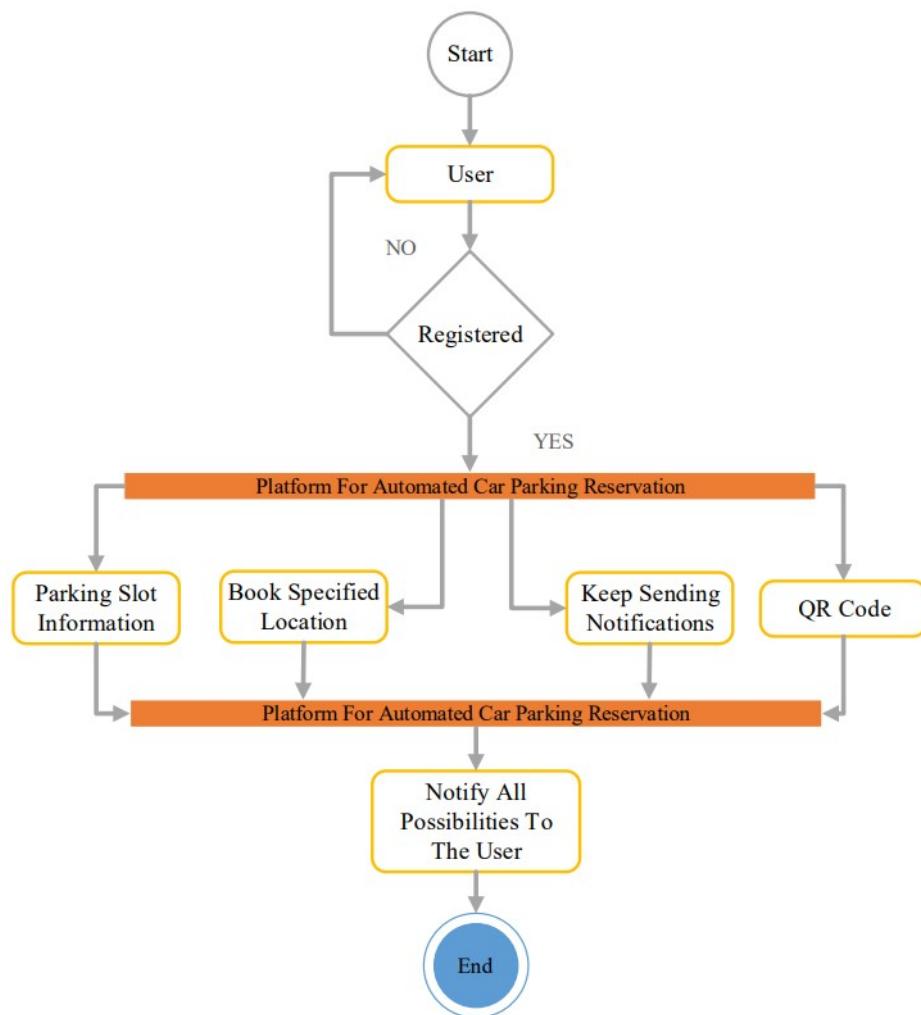


Figure 4.8: Activity Diagram

4.4 Summary

In the design phase of the Car Parking Management System project, the System Architecture is established, providing a high-level overview of the system's structure and components. Data Flow Diagrams (DFDs) play a crucial role, with a Level 0 DFD offering a broad depiction of system interactions, followed by more detailed Level 1 and Level 3 DFDs that

break down processes into finer levels of detail. UML Diagrams further enhance the design, including a Use Case Diagram to illustrate system-user interactions, a Sequence Diagram to depict the flow of processes over time, a Component Diagram illustrating system components and their relationships, and an Activity Diagram detailing the dynamic aspects of system behavior. These design components collectively serve to provide a comprehensive and visually intuitive blueprint for the Car Parking Management System.

Chapter 5

Coading/Implementation

In this Chapter we are going to discuss about the implementation of the project by using Html,Css,Php and Mysql using Xampp and Visual Studio code. We break our System into following module to understand the working and workflow accordingly..

Module Descriptions with Key Highlights

Admin Module

- **Manage User Details:** Admin can view and manage all user details registered in the system.
- **Parking Cost Management:** Admin has the authority to set and manage parking costs for different types of vehicles or parking durations.
- **Booking Management:** Admin can view and manage all parking bookings made by users, including approving or rejecting bookings.
- **Parking Logs Management:** Admin can access and review logs of parking activities for auditing and monitoring purposes.
- **Revenue Report Generation:** Admin can generate reports to analyze the revenue generated from parking bookings over a specific period.

Ticket Checker Module

- **Ticket Verification:** Ticket checker has the ability to verify parking tickets presented by users to ensure they are valid.
- **Parking Logs Viewing:** Ticket checker can access logs of parking activities to monitor parking compliance and detect any irregularities.

User Module

- **Registration and Login:** Users can register for an account and log in to access the system.
- **Check Parking Availability:** Users can check the availability of parking slots for their desired date and time.
- **Reserve Parking:** Users can make bookings for parking slots for specific dates and times.
- **Receive Notifications:** Users receive notifications regarding their bookings, payment status, and any updates related to parking availability.
- **Make Payment:** Users can securely make payments for their parking bookings through the system.
- **Download Ticket:** Users can download and print parking tickets after successfully booking a parking slot.

5.1 Admin Module

The Admin Module is responsible for managing the overall operation of the car parking management system. It provides administrative functionalities to ensure smooth operation and efficient management of parking spaces, bookings, and users. Key highlights include:

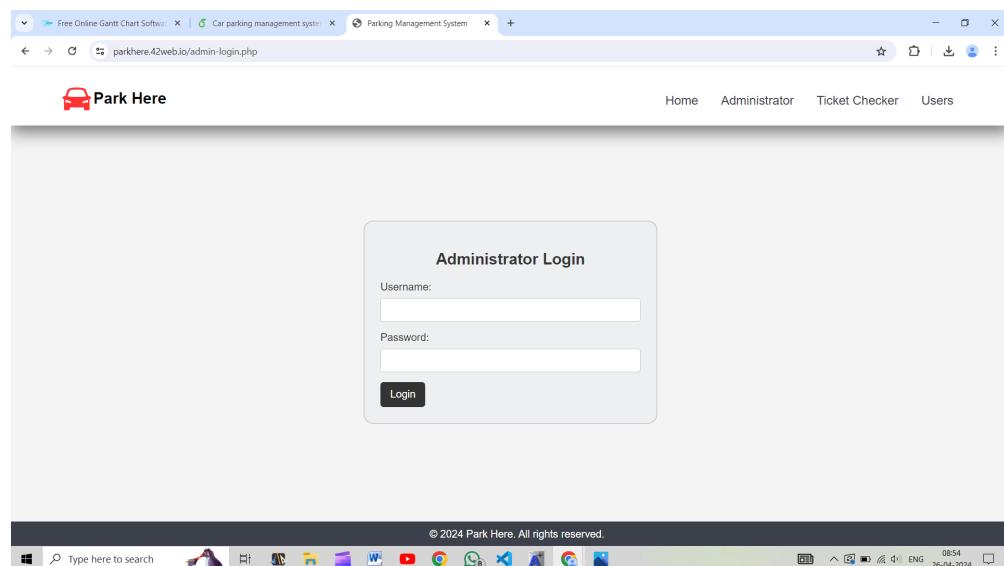


Figure 5.1: Admin Login Interface

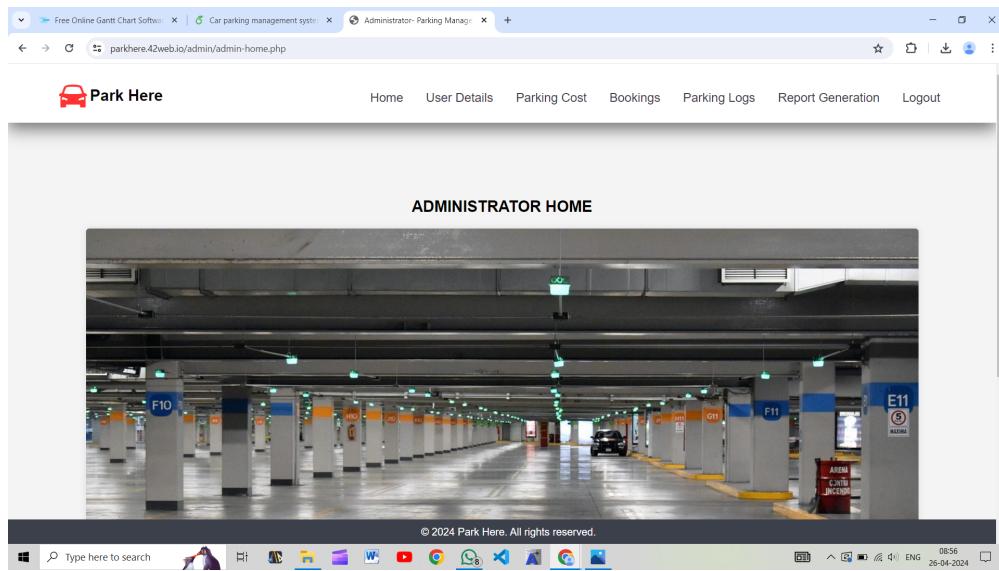


Figure 5.2: Admin Index

- User Details: Admin can view and manage user details such as registration information, booking history, and contact information and if wanted then can search also.

The screenshot shows the 'USER DETAILS' page. At the top, there's a navigation bar with links: Home, User Details, Parking Cost, Bookings, Parking Logs, Report Generation, and Logout. Below the navigation bar is a search bar labeled 'Search: Enter search term...'. A table displays user information with columns: User ID, Name, Email, DOB, Address, Gender, and Phone No. The data is as follows:

User ID	Name	Email	DOB	Address	Gender	Phone No
3	Nilesh Liladhar Tayade	nilesh@gmail.com	2003-10-28	Ashababa nagar,jalgaon 425001	male	9673466829
4	Harshada Liladhar Salunke	harshadasalunke21117@gmail.com	2003-07-01	xyz	female	1234567890
5	Kamesh	kameshsurwade@gmail.com	2003-03-31	Police Line 144 Building No 03/5	male	8605870030
6	anurag	anurag@gmail.com	2024-04-16	khotenagar jalgaon	male	7249560449
7	KAMESH SURWADE	kamesh@gmail.com	2003-03-11	XYZ	male	86056700680

Figure 5.3: User Details

- Parking Cost Management: Admin has the authority to set and modify parking costs based on various factors such as location, time of day, and vehicle type.

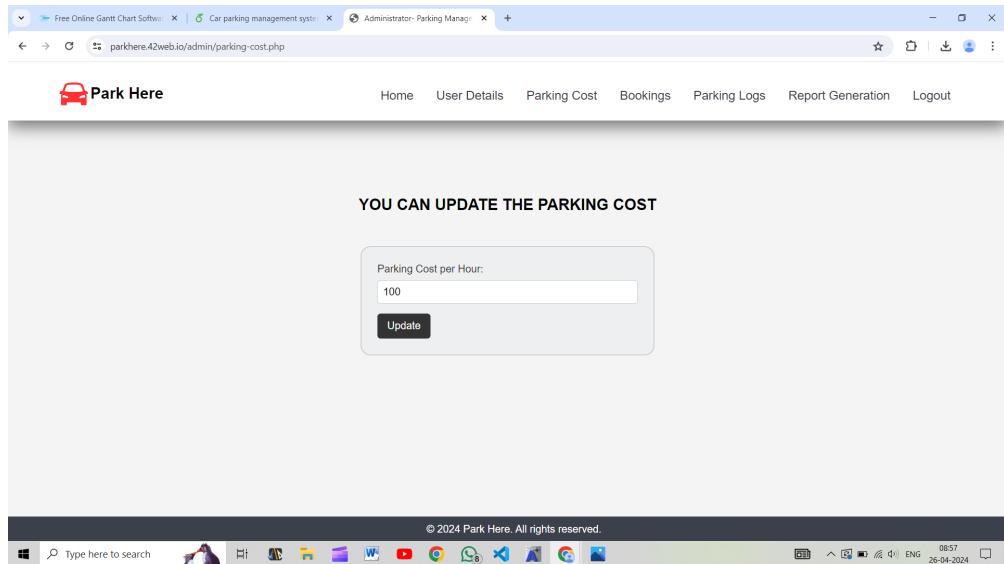


Figure 5.4: Parking Cost Management

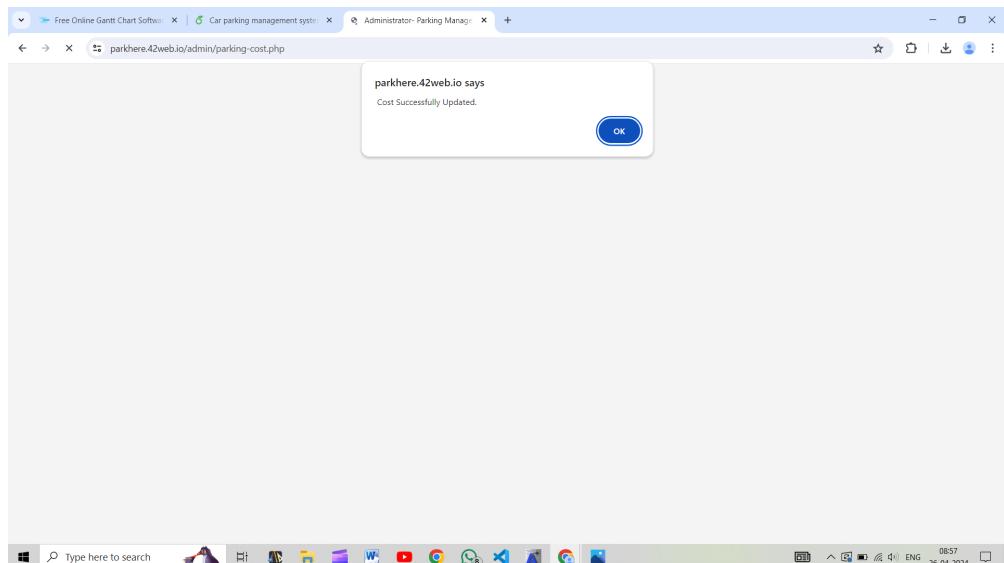


Figure 5.5: Parking Cost Management

- View Bookings: Admin can View all bookings having necessary details with parking status yes means booking done and parking also used and no means the booking is done but parking not used.

The screenshot shows a web browser window titled 'Administrator- Parking Manager'. The main content area is titled 'BOOKING DETAILS'. It features a search bar and a table with columns: Parking Date, Start Time, End Time, Slot Name, Booking Time, Parking Cost, and Parking Status. The table contains data for various bookings from April 16 to April 25, 2024. The status column indicates whether parking was used ('Yes') or not ('No').

Parking Date	Start Time	End Time	Slot Name	Booking Time	Parking Cost	Parking Status
2024-04-16	2:00 PM	4:00 PM	A1	2024-04-15 23:11:12	40	Yes
2024-04-17	7:00 AM	10:00 AM	A5	2024-04-15 23:21:39	60	Yes
2024-04-16	4:00 PM	7:00 PM	A2	2024-04-16 00:13:35	60	No
2024-04-17	11:00 AM	1:00 PM	A10	2024-04-16 02:32:07	100	Yes
2024-04-22	6:00 AM	7:00 AM	A6	2024-04-21 12:03:05	10	No
2024-04-11	6:00 AM	7:00 AM	A5	2024-04-22 13:16:09	888	No
2024-04-25	11:00 AM	1:00 PM	A8	2024-04-24 07:40:06	1776	Yes

Figure 5.6: View Bookings

- View Parking Logs: Admin can access logs of parking activities, including entries, exits, and duration of parking for auditing and analysis purposes.

The screenshot shows a web browser window titled 'Administrator- Parking Manager'. The main content area is titled 'PARKING LOGS'. It features a search bar and a table with columns: ID, Ticket ID, Vehicle Number, Username, Slot Name, and Timestamp. The table contains data for various parking events from April 15 to April 25, 2024. The timestamp column shows the exact time of each log entry.

ID	Ticket ID	Vehicle Number	Username	Slot Name	Timestamp
3	OahBUTGt	MH 19 AD 8697	Kamesh	A1	2024-04-15 23:14:54
4	PuLnjoGR	Mh 19 JK 5555	Harshada Liladhar Salunke	A5	2024-04-15 23:25:16
5	1NDkevfw	mh 19 vd 9988	Nilesh Liladhar Tayade	A10	2024-04-16 02:36:27
6	gW25Vn1b	MH 19 CC2211	Nilesh Liladhar Tayade	A8	2024-04-24 07:48:06
7	yLbmVO3h	MH 19 KS 1696	Kamesh	A1	2024-04-25 12:33:06

Figure 5.7: View Parking Logs

- Revenue Report Generation: Admin can generate reports to track revenue generated from parking fees over specific periods, aiding in financial analysis and planning.

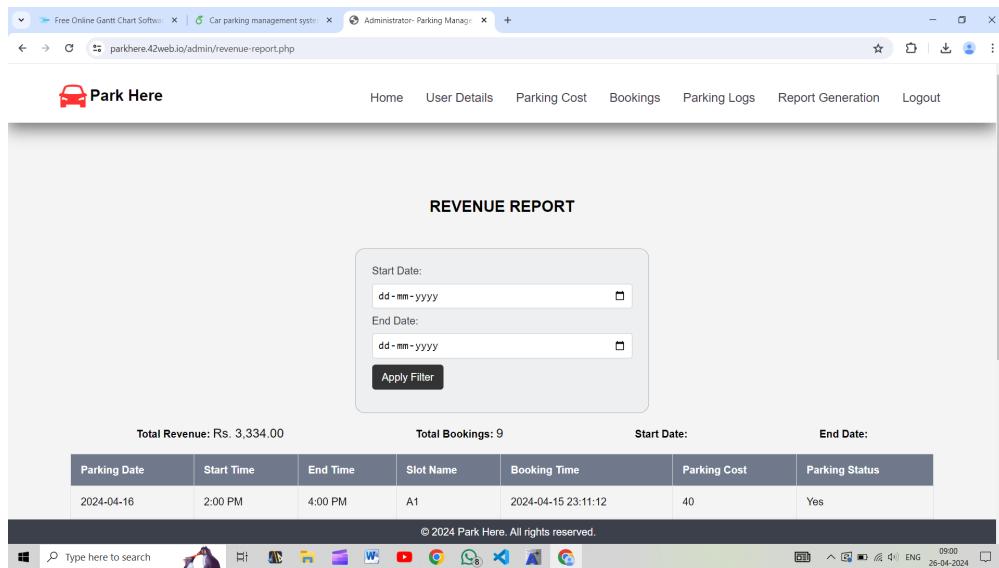


Figure 5.8: Revenue Report Generation

Ticket Checker Module

The Ticket Checker Module is designed for personnel responsible for verifying parking tickets and ensuring compliance with parking regulations. Key highlights include:

- Login Feature: Ticket checker can log in to the system to access ticket verification and parking logs viewing functionalities securely.

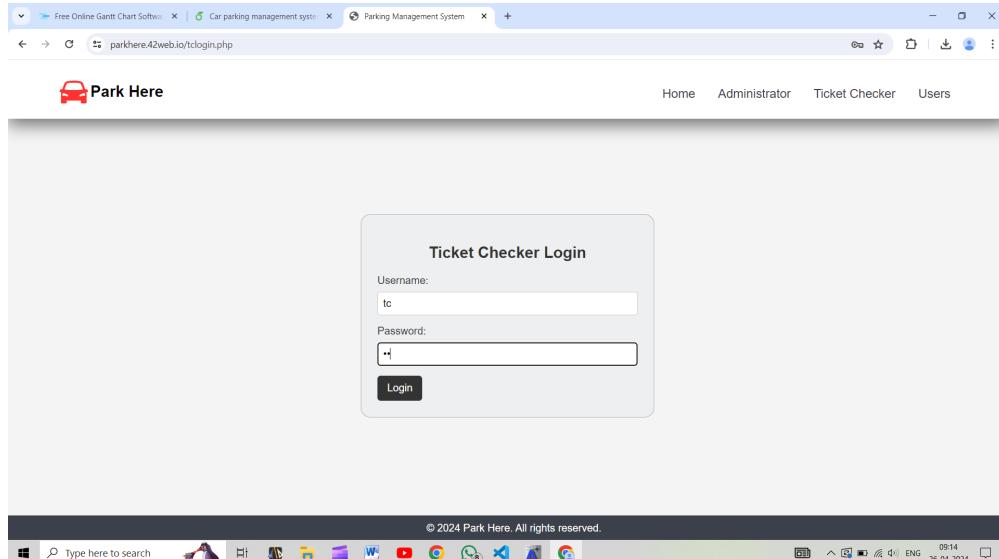


Figure 5.9: Ticket Checker Login

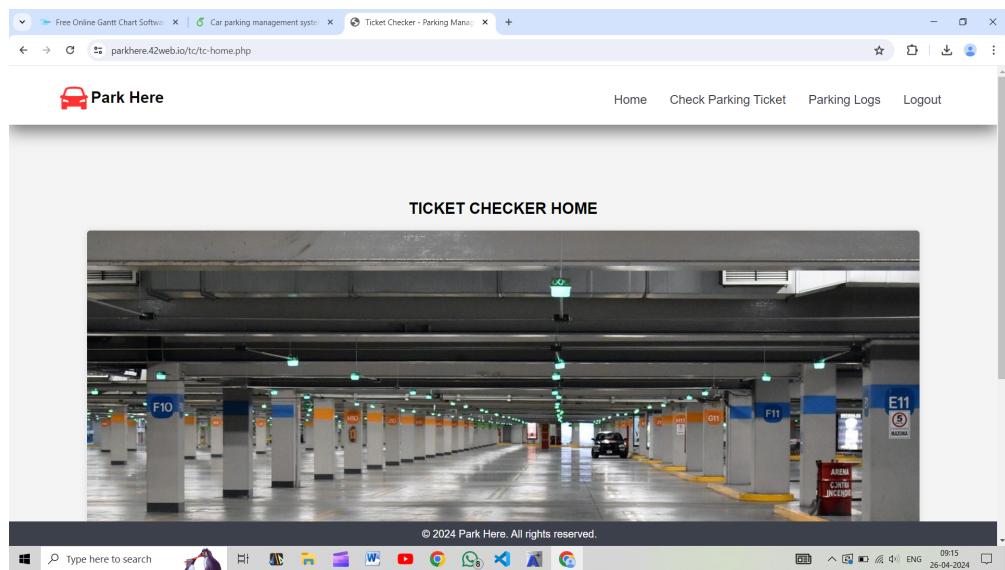


Figure 5.10: Ticket Checker Home

- Ticket Verification: Ticket checkers can validate parking tickets issued to users, ensuring they are genuine and valid for the corresponding parking session.

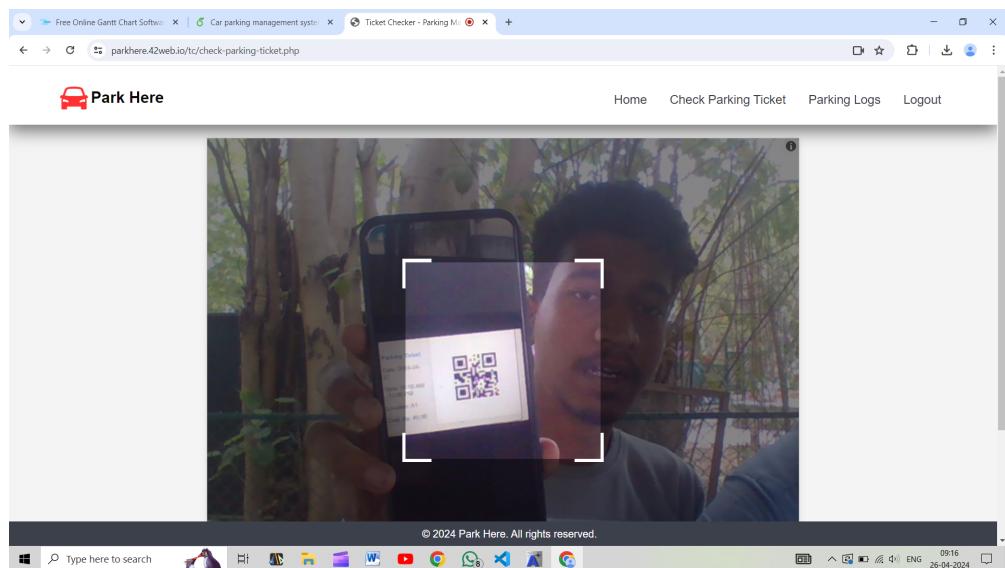


Figure 5.11: Ticket Verification

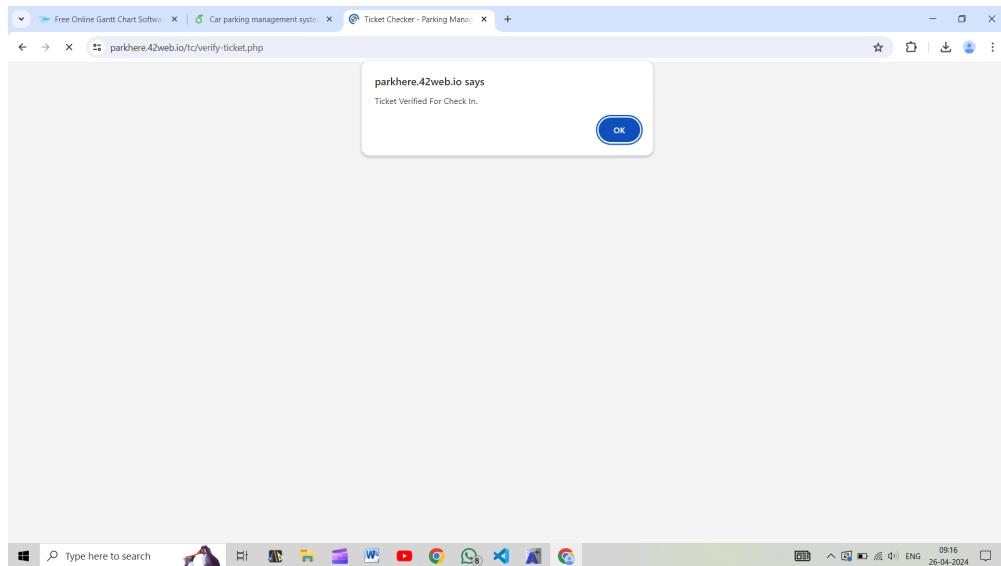


Figure 5.12: Ticket Verification

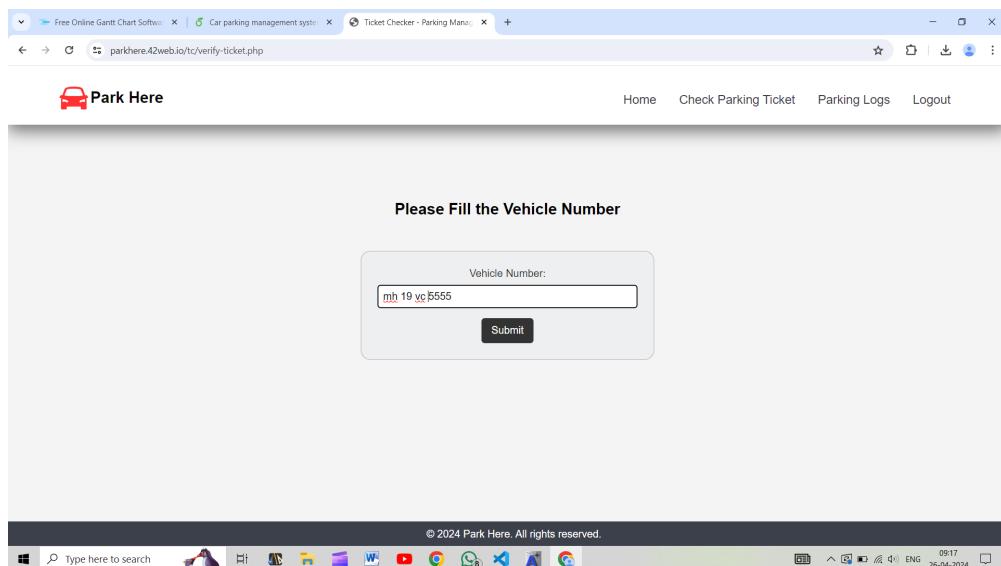


Figure 5.13: Ticket Verification

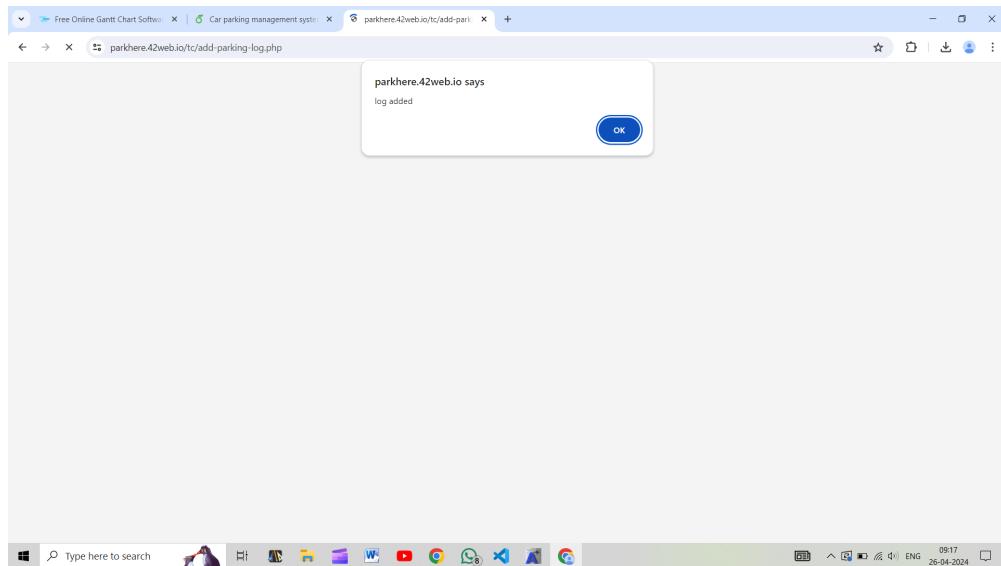


Figure 5.14: Ticket Verification

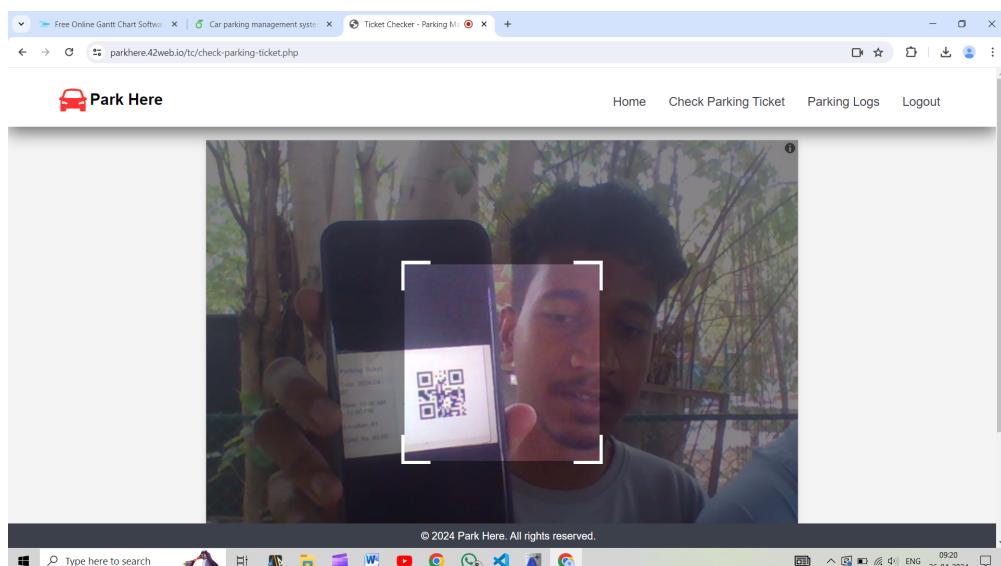


Figure 5.15: Ticket Verification

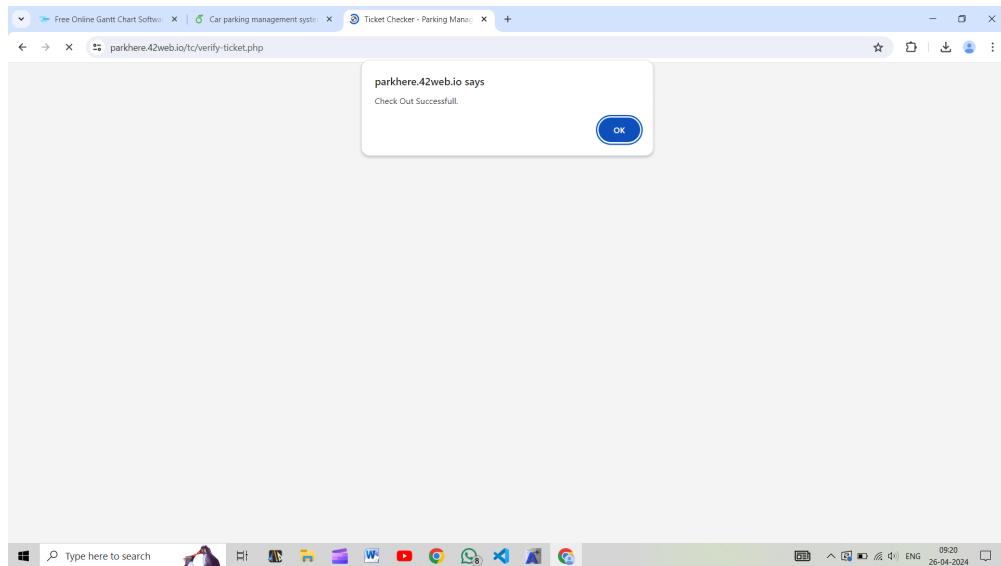


Figure 5.16: Ticket Verification

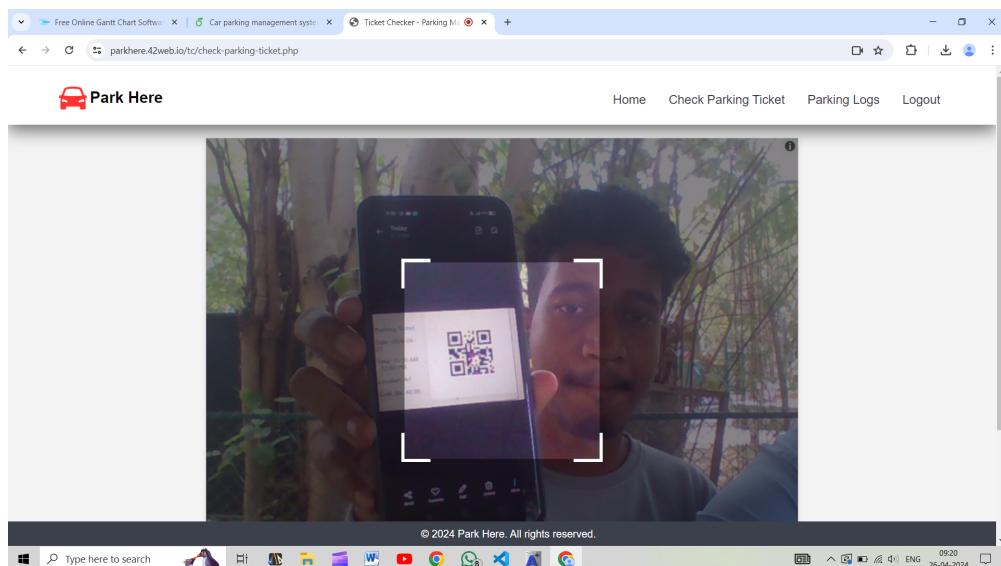


Figure 5.17: Ticket Verification

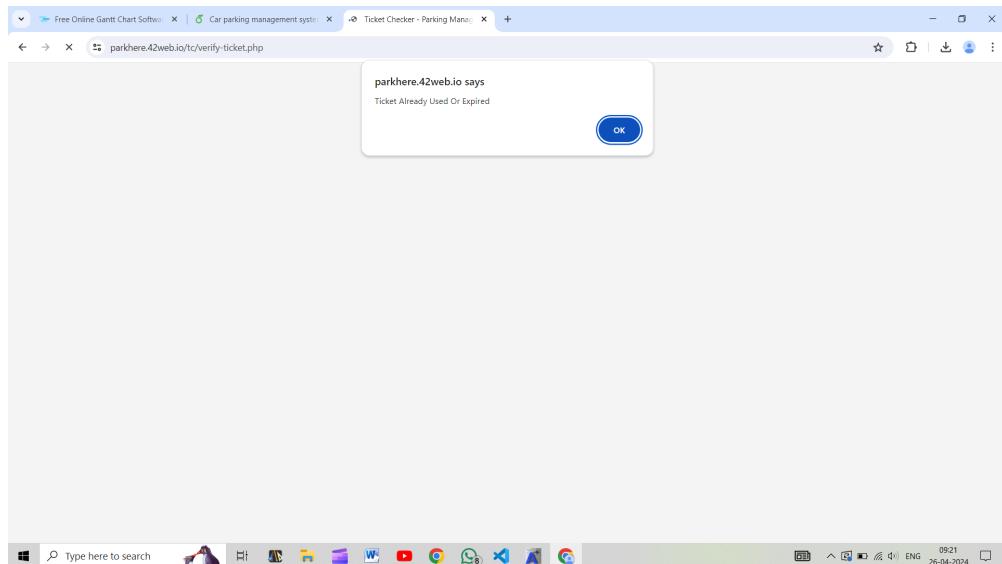


Figure 5.18: Ticket Verification

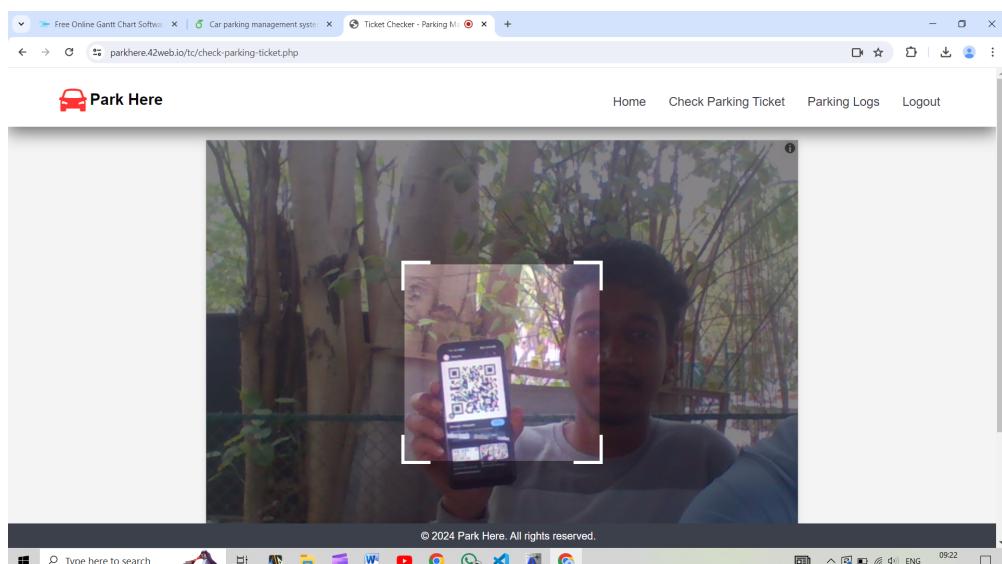


Figure 5.19: Ticket Verification

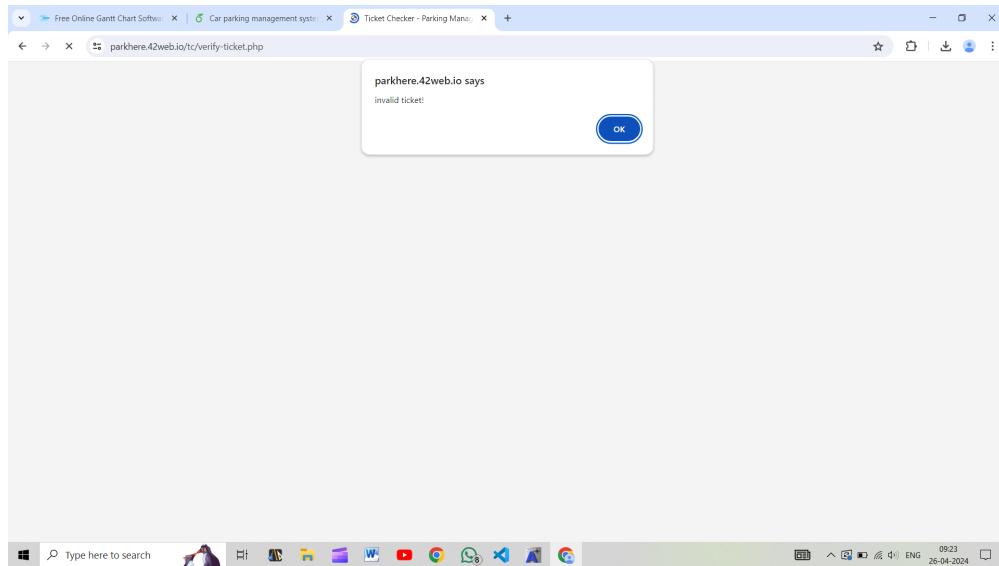


Figure 5.20: Ticket Verification

- **Parking Logs Viewing:** Ticket checkers have access to parking logs to monitor parking activities, including entries, exits, and any anomalies or violations.

ID	Ticket ID	Vehicle Number	Username	Slot Name	Timestamp
3	OanBUTGt	MH 19 AD 8697	Kamesh	A1	2024-04-15 23:14:54
4	PuLnjoGR	Mh 19 JK 5555	Harshada Liladhar Salunke	A5	2024-04-15 23:25:16
5	1NDkevfw	mh 19 vd 9988	Nilesh Liladhar Tayade	A10	2024-04-16 02:36:27
6	gW25Vn1b	MH 19 CC2211	Nilesh Liladhar Tayade	A8	2024-04-24 07:48:06
7	yLbmVO3h	MH 19 KS l696	Kamesh	A1	2024-04-25 12:33:06
8	7c5w1Tje	mh 19 vc 5555	Nilesh Liladhar Tayade	A1	2024-04-25 23:47:25

Figure 5.21: Parking Logs Viewing

User Module

The User Module is tailored for individuals utilizing the car parking management system to find, reserve, and pay for parking spaces conveniently. Key highlights include:

User Registration: Users can register for an account, providing necessary details such as personal information and vehicle registration.

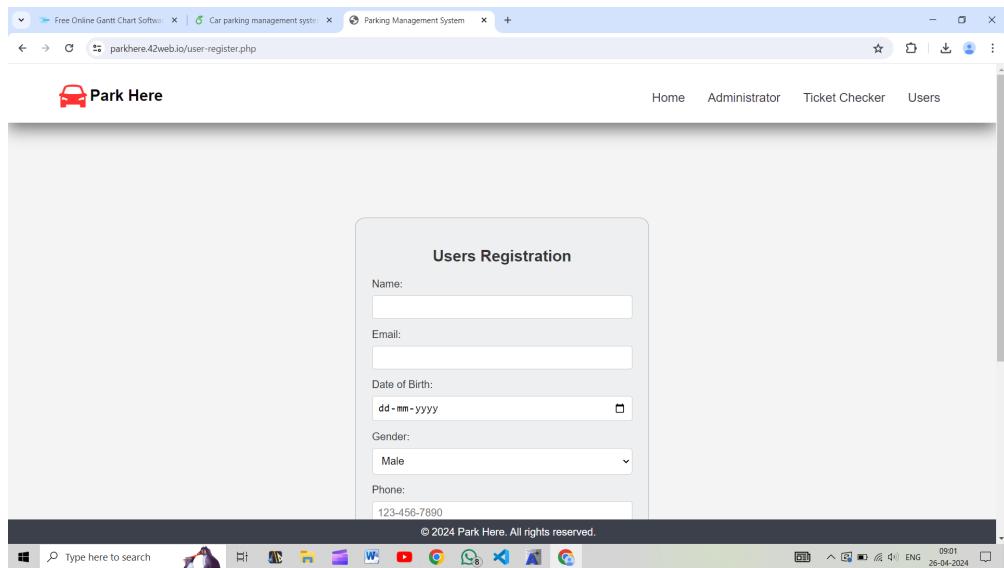


Figure 5.22: User Registration

User Login: Registered users can log in securely to access the system, manage their profile, and make parking reservations.

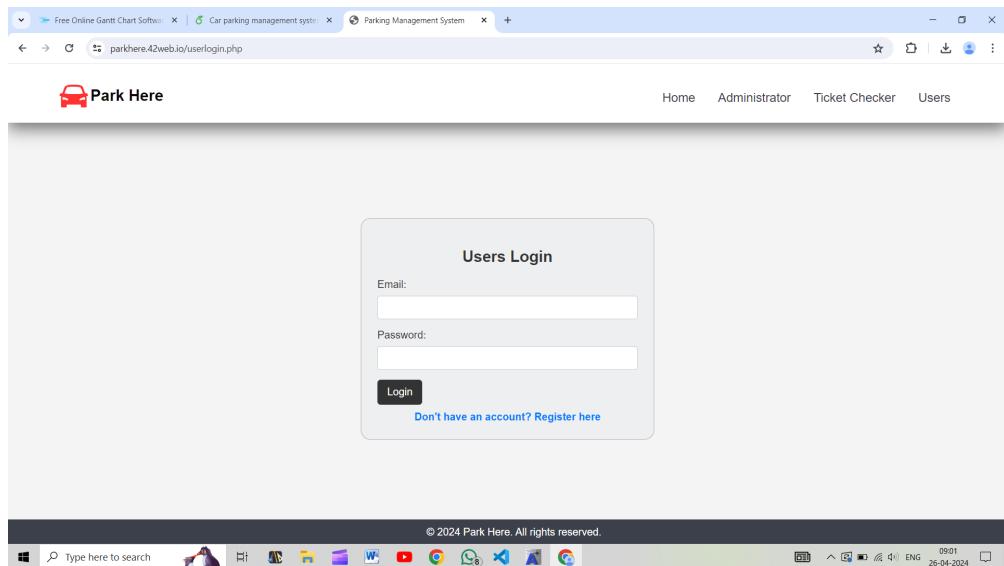


Figure 5.23: User Login

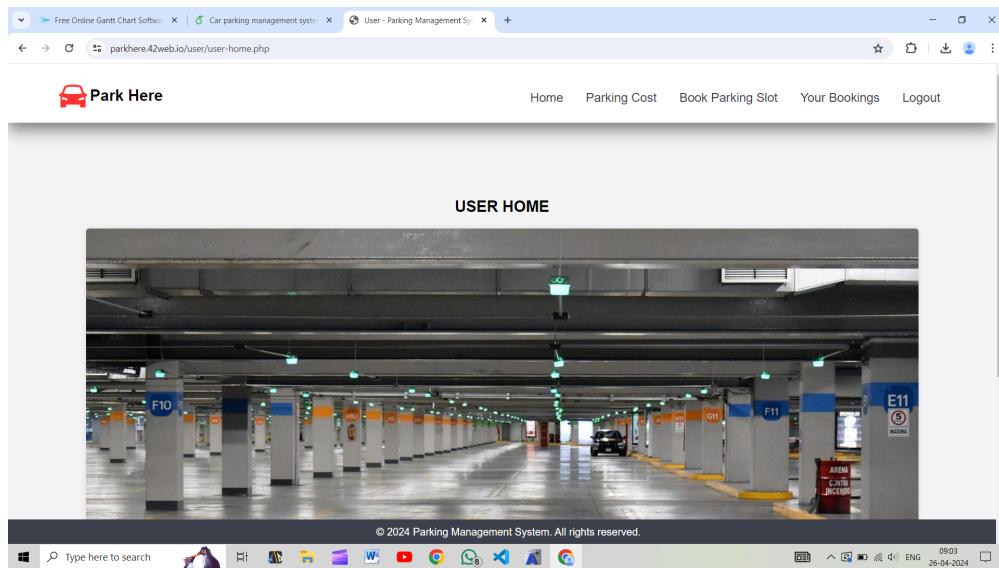


Figure 5.24: User Home

Check Parking Cost: Users can check the parking cost per hour set by admin.

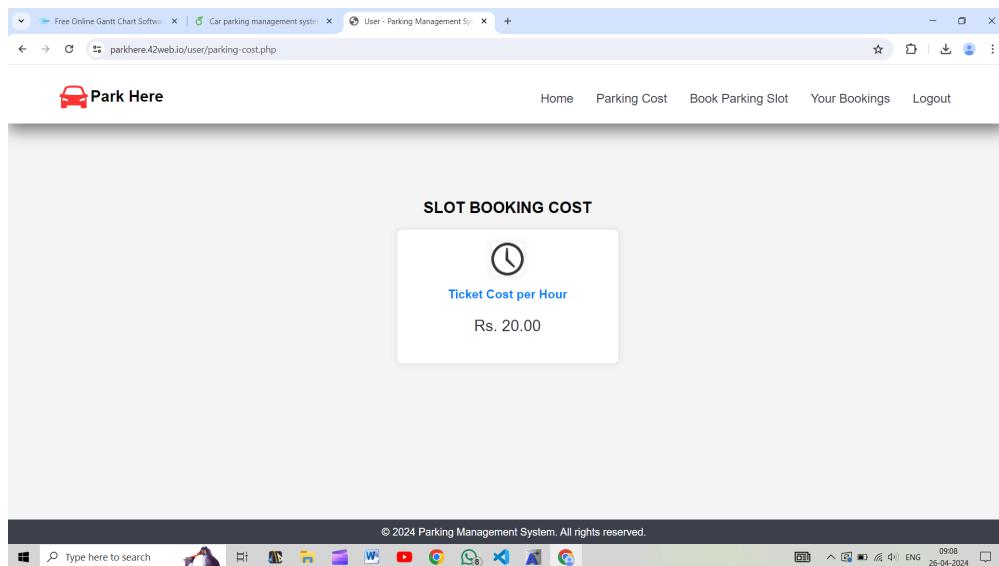


Figure 5.25: Parking Cost

Check Parking Availability and Reserve Parking: Users can check the availability of parking spaces at their desired location and time, helping them plan their parking in advance.

Users can reserve parking slots for specific dates and times, ensuring their parking space is guaranteed upon arrival.

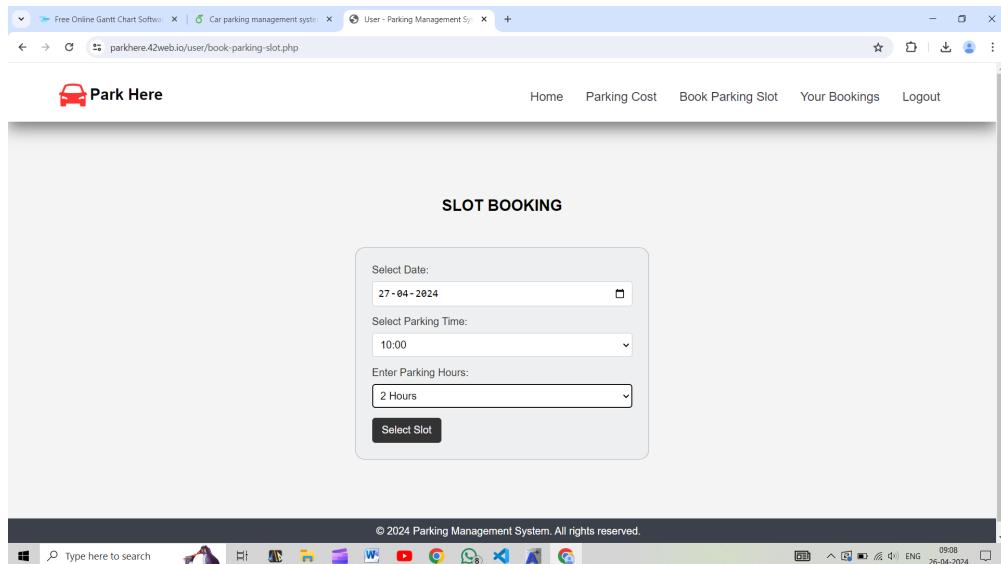


Figure 5.26: Check Parking Availability

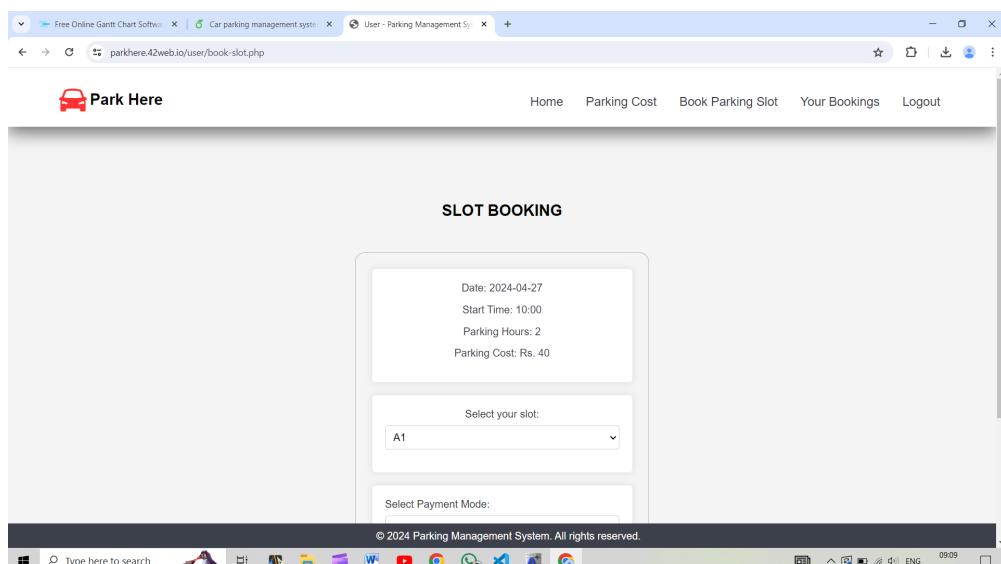


Figure 5.27: Check Parking Availability

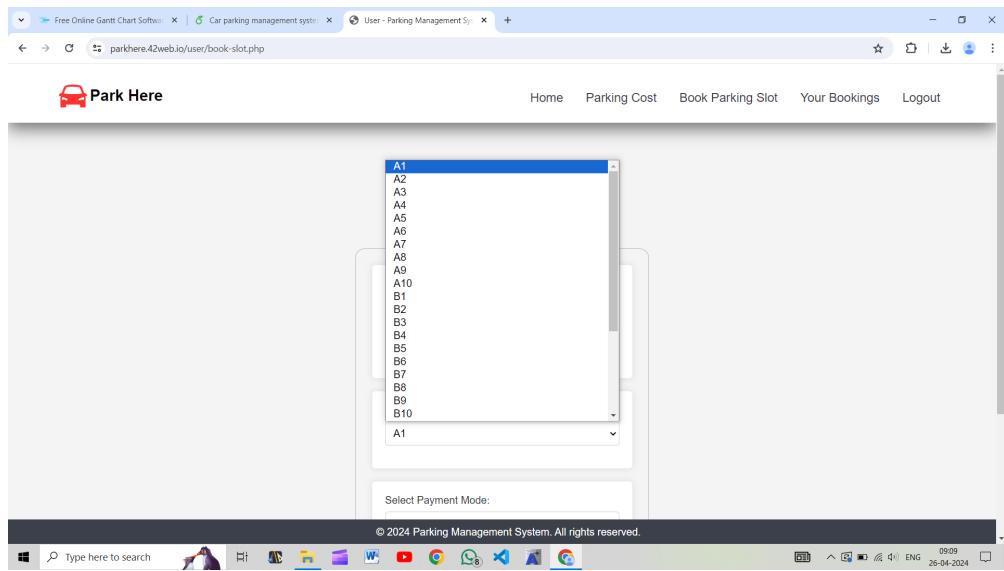


Figure 5.28: Available slot details

Make Payment: Users can securely make payments for their parking reservations through various payment methods supported by the system.

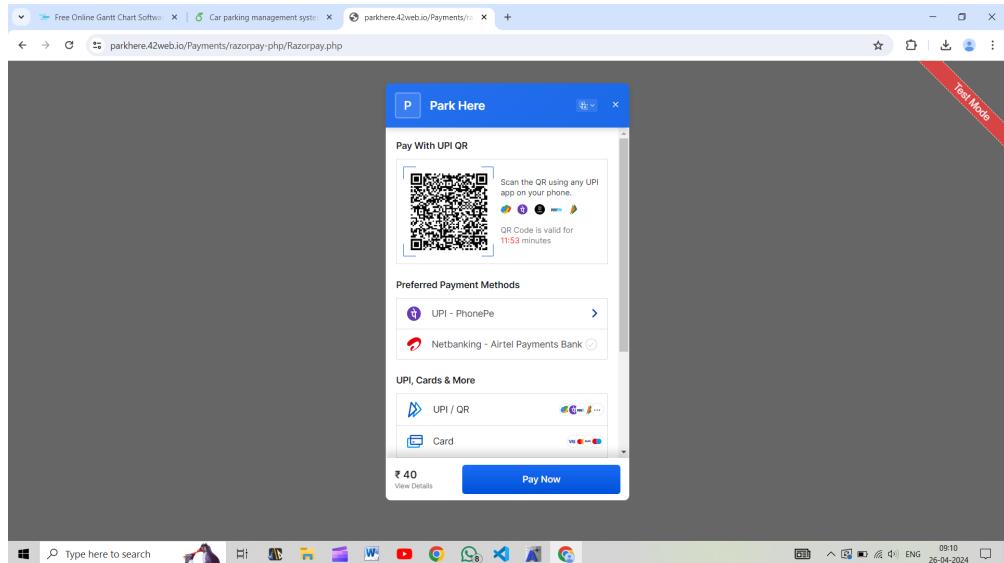


Figure 5.29: Make Payment

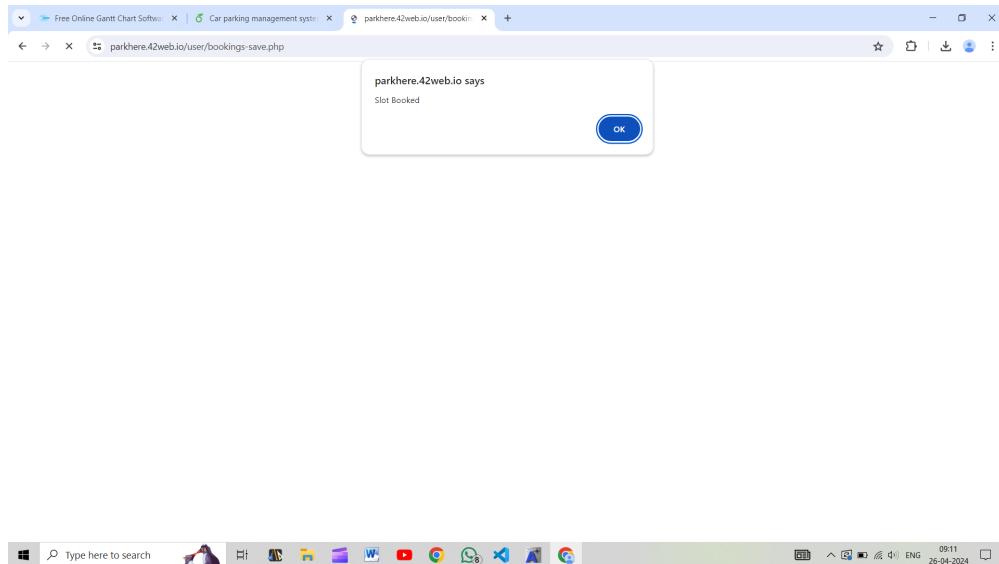


Figure 5.30: Slot booked confirmation

Generate Ticket for Booking: Upon successful reservation and payment, users can generate digital or printable parking tickets for verification purposes.

A screenshot of a web application titled 'Park Here'. The main menu includes Home, Parking Cost, Book Parking Slot, Your Bookings, and Logout. The 'Your Bookings' section is active. It displays a table titled 'YOUR BOOKINGS' with the following data:

Parking Date	Start Time	End Time	Slot Name	Booking Time	Parking Cost	Download Ticket
2024-04-16	4:00 PM	7:00 PM	A2	2024-04-16 00:13:35	RS.60.00	<button>Download</button>
2024-04-17	11:00 AM	1:00 PM	A10	2024-04-16 02:32:07	RS.100.00	<button>Download</button>
2024-04-22	6:00 AM	7:00 AM	A6	2024-04-21 12:03:05	RS.10.00	<button>Download</button>
2024-04-11	6:00 AM	7:00 AM	A5	2024-04-22 13:16:09	RS.888.00	<button>Download</button>
2024-04-25	11:00 AM	1:00 PM	A8	2024-04-24 07:40:06	RS.1776.00	<button>Download</button>
2024-04-27	10:00 AM	12:00 PM	A1	2024-04-25 23:40:45	RS.40.00	<button>Download</button>

Figure 5.31: Generate Ticket for Booking

Download Ticket: Users can download and save their parking tickets for easy access and presentation during parking ticket verification.

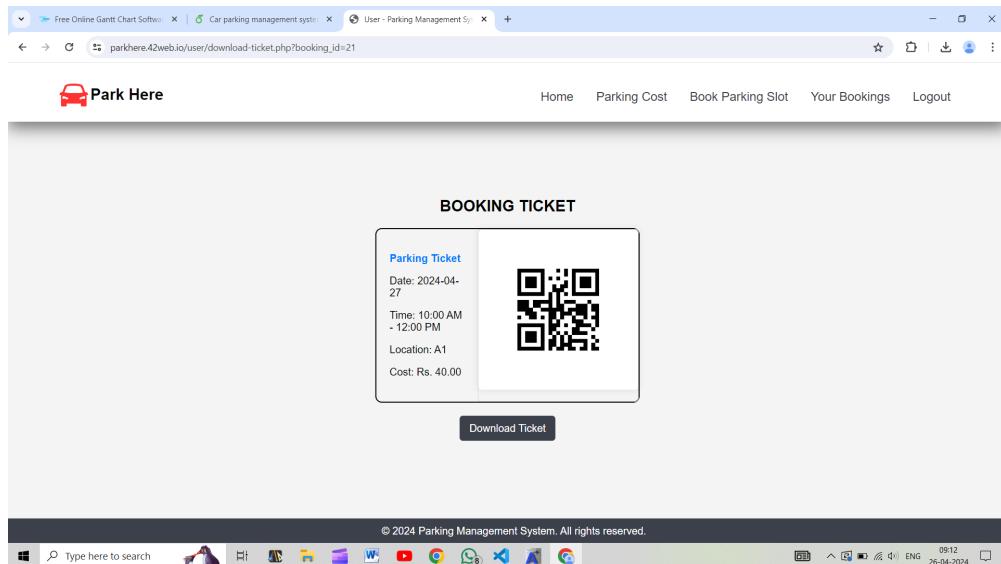


Figure 5.32: Download Ticket

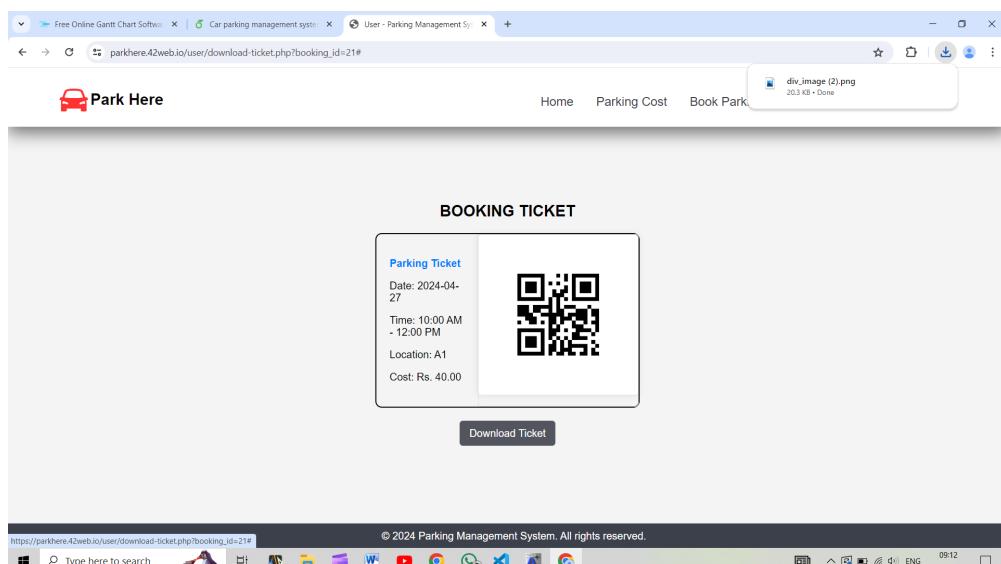


Figure 5.33: Ticket Downloaded

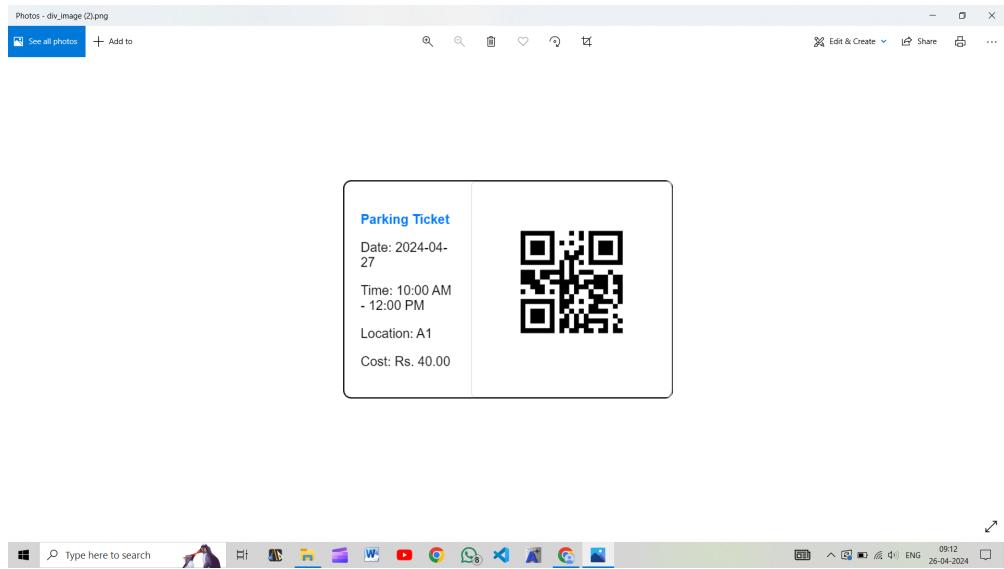


Figure 5.34: Ticket in PNG format

5.1 Summary

The implementation phase focused on developing modules for user management, parking management, admin dashboard, and ticket checking. Technologies like HTML, CSS, JavaScript, PHP, and MySQL were used. User management allowed registration, login, and profile management. Parking management facilitated booking, payment, and ticket generation. The admin dashboard enabled user and booking management, cost control, and revenue reporting. Ticket checkers could validate tickets and access logs. Rigorous testing ensured system reliability and usability, aiming to enhance parking efficiency and user experience.

Chapter 6

Result and Discussion

Result

The QR-based car parking management system has been successfully implemented using HTML, CSS, JavaScript, PHP, and MySQL. The system provides a user-friendly interface for users to reserve parking slots, make payments, and receive notifications. Administrators have access to comprehensive management tools for overseeing user details, parking bookings, and generating revenue reports. The ticket checker module allows for efficient ticket verification and monitoring of parking activities.

Key Features

Throughout the development process, several key features have been implemented to enhance the functionality and usability of the system:

- User-friendly Interface: The system interface is intuitive and easy to navigate, providing a seamless experience for users.
- Real-time Parking Availability: Users can check the availability of parking slots in real-time, enabling them to make informed decisions when reserving parking.
- Secure Payment Gateway: The system integrates a secure payment gateway, ensuring the confidentiality and integrity of user transactions.
- Comprehensive Administrative Tools: Administrators have access to a range of tools for managing user details, parking bookings, and generating revenue reports, facilitating efficient management of the parking system.
- Efficient Ticket Verification: The ticket checker module enables efficient verification of parking tickets, streamlining the parking enforcement process.

Discussion

The implementation of the QR-based car parking management system has proven to be successful, meeting the requirements and objectives outlined in the project scope. The system provides an effective solution for managing parking spaces, enhancing user convenience, and optimizing parking utilization. Moving forward, further enhancements could be made to improve system scalability, performance, and security. Additionally, user feedback and testing should be conducted to identify areas for improvement and refine the system based on user needs and preferences.

Chapter 7

Conclusion and Future Work

Future Work

The QR-based car parking management system has laid a solid foundation for further expansion and enhancement. Several avenues for future work and improvement have been identified:

- *Expansion to Different Parking Lots*: One of the primary objectives for future work is to expand the system to cater to multiple parking lots. This expansion would involve integrating the system with additional parking facilities, each with its own set of parking slots and administrative requirements.
- *Integration of License Plate Recognition (LPR) Technology*: Implementing License Plate Recognition (LPR) technology would enhance the efficiency of the ticket checker module. By automatically capturing vehicle license plate numbers using sensors, the system can streamline the ticket verification process and improve parking enforcement capabilities.
- *Real-world Deployment and Testing*: The next phase of the project involves deploying the system in real-world parking environments and conducting extensive testing to evaluate its performance and reliability under real-world conditions. This testing will involve collaborating with parking lot operators and users to gather feedback and validate the system's effectiveness.
- *Enhanced Security Measures*: Implementing advanced security measures, such as encryption and authentication protocols, will be crucial for ensuring the integrity and confidentiality of user data and financial transactions. Strengthening security measures will enhance user trust and confidence in the system.
- *Scalability and Performance Optimization*: As the system expands to serve multiple parking lots and a growing user base, optimizing scalability and performance will be

essential. This includes optimizing database queries, server configurations, and system architecture to handle increased traffic and workload efficiently.

- *User Experience Enhancement*: Continuously improving the user experience is vital for the long-term success of the system. Future work will focus on refining the system interface, incorporating user feedback, and implementing additional features to enhance user satisfaction and usability.

These future work initiatives will contribute to the ongoing development and evolution of the QR-based car parking management system, ensuring its continued relevance and effectiveness in meeting the needs of parking operators and users in real-world scenarios.

Conclusion

In summary, the QR-based car parking management system offers a comprehensive solution for efficient parking facility management. Utilizing modern web technologies, the system ensures a seamless experience for users while providing administrators with powerful tools for monitoring and control. With its successful implementation, the project demonstrates the potential of technology to enhance parking operations and improve overall efficiency. Future enhancements could further augment the system's capabilities, paving the way for even greater innovation in parking management solutions.

Bibliography

[1] Journal Paper Preferred

- Hongwei Wang and Wendo He, “A Reservation-based Smart Parking System” 2011 International Journal of Computer Science and Software Engineering (IJCSSE), Volume 8, Issue 10, October 2019 258
- P. Sheelarani, S. Preethi Anand, S. Shamili and K. Sruthi, “Effective Car Parking Reservation System Based on Internet of Things Technologies”. WCFTR’16, 2016

[2] Books Preferred

- Software Engineering by K. K. Aggarwal Yogesh Singh, New Age International, 3rd Edition, 2008.
- Fundamentals of Software Engineering by Rajib Mall, PHI, 4th Edition, 2014.

[3] Web Resources

- “UML - Standard Diagrams”, www.tutorialspoint.com/uml/uml_standard_diagrams.htm. [Online]. Available: https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm.