CHAPTER 1

INTRODUCTION

1.1 About the Project

A **Vehicle Parking Management System** is a software application designed to help manage the operation of parking facilities such as parking lots, garages, or multi-story parking structures. The primary goal of this system is to automate the process of tracking vehicles as they enter and exit, managing parking space availability, calculating parking fees, and generating reports. This system improves the overall efficiency of parking operations and offers convenience to both the users (vehicle owners) and parking administrators.

1.1.1 Objectives of the Project

1. **Automated Vehicle Tracking**: The system automatically tracks the entry and exit of vehicles, allowing administrators to have real-time data on the availability of parking spaces.
2. **Space Management**: Efficiently allocates parking spaces to vehicles and keeps track of space occupancy.
3. **Fee Calculation**: Calculates parking fees based on the duration of the vehicle's stay.
4. **User-Friendly Interface**: Provides an easy-to-use interface for both parking lot users and administrators.
5. **Report Generation**: Generates detailed reports on parking space utilization, revenue collection, and other performance metrics.
6. **Payment Management**: Enables the recording of parking fee payments and facilitates different types of payment methods (manual, online, etc.).

1.1.2 Key Features

Vehicle Entry and Exit Management:

- The system tracks when a vehicle enters and exits the parking lot.
- The entry time is recorded, and the exit time is recorded when the vehicle leaves.

Parking Space Availability:

- The system keeps track of the available and occupied parking spaces.
- It can show users how many spaces are available in real time.

Parking Fee Calculation:

- The system calculates parking fees based on the duration of the vehicle's stay (e.g., hourly, daily).
- Different rates can be applied for different types of vehicles (cars, motorcycles, trucks)

Payment Management:

- Once the parking duration is completed, the system calculates the total parking fee.
- It generates a bill for the user to make payment.
- Multiple payment options can be integrated, such as cash, card, or online payment (integration with gateways like PayPal or Stripe).

Reporting:

- The system generates daily, weekly, or monthly reports of:
    - Total revenue collected.
    - Number of vehicles parked.
    - Available parking spaces

1.2 Proposed System

The **Vehicle Parking Management System (VPMS)** is designed to streamline the operations of parking facilities, including parking lots, multi-story garages, and other types of parking structures. The system aims to automate and optimize key parking functions such as vehicle tracking, parking space management, fee calculation, and payment handling. It also provides features for reporting and space availability monitoring, benefiting both parking lot administrators and users.

The following sections outline the **proposed system design**, including the main objectives, features, architecture, and user requirements.

1.2 1 Objectives of the Proposed System

- **Automation of Parking Operations**: Automate the vehicle entry and exit processes, parking space allocation, and parking fee calculations.
- **Parking Space Management**: Track the availability of parking spaces and assign vehicles to available spaces in real time.
- **Fee Calculation and Payment**: Automatically calculate parking fees based on the vehicle’s duration of stay and offer multiple payment options (cash, card, online).
- **User Convenience**: Provide an easy-to-use interface for both users (vehicle owners) and administrators, offering functionalities like parking space search, reservation, and payment tracking.
- **Real-Time Monitoring**: Enable administrators to monitor parking space occupancy, vehicle movements, and revenue generation in real time.
- **Reporting and Analytics**: Generate reports on parking occupancy, revenue, and system usage to help administrators make informed decisions.

1.2.2 Key Features of the Proposed System

For Users (Vehicle Owners):

1. **Vehicle Registration**: Allows users to register their vehicle details (e.g., vehicle number, type) into the system.
2. **Parking Space Search**: Users can view real-time parking space availability and find an available spot.
3. **Parking Reservation**: Option to reserve parking spots in advance for a specified time, ensuring availability upon arrival.
4. **Fee Calculation**: The system calculates the parking fee based on the vehicle’s parking duration.
5. **Payment Options**: Users can pay via multiple payment methods (cash, card, or online payment options like PayPal or Stripe).
6. **History & Receipts**: View previous parking history, fees paid, and generate payment receipts.

For Administrators:

1. **Real-Time Parking Space Monitoring**: Admins can track the availability of parking spaces in real-time and ensure efficient space allocation.
2. **Vehicle Entry and Exit Management**: Track vehicles as they enter or exit the parking lot, including timestamp recording and fee calculation.
3. **Fee Management**: Set parking rates (by time, vehicle type, etc.), offer discounts, and generate invoices.
4. **Revenue and Reporting**: Admins can generate daily, weekly, or monthly reports on revenue collected, parking occupancy, and vehicle count.
5. **User Management**: Create, update, or delete user accounts. Admin can also set user roles (admin, user).
6. **Payment Tracking**: Admins can track all payments made and monitor the status of unpaid fees.

Additional Features:

1. **Mobile App/ Web Interface**: A mobile or web interface for users to check parking availability, make reservations, and pay parking fees.
2. **SMS/Email Notifications**: Notify users about parking space availability, reservation confirmation, payment receipts, and parking fee due reminders.

CHAPTER 2

LITERATURE SURVEY

Vehicle parking management is a critical aspect of urban planning and infrastructure. With increasing vehicular traffic, efficient parking systems are essential to optimize space utilization, reduce congestion, and improve the overall driving experience. Several approaches, technologies, and methodologies have been explored in the design and implementation of parking management systems. The following survey outlines relevant research, existing systems, and innovations in this field.

1. K. Kar et al. (2015):

Title: Proposed a fully automated vehicle parking management system based on sensors and cameras.

The system used a combination of RFID and Bluetooth technologies to monitor parking space occupancy in real time and manage parking fees dynamically. The study demonstrated that automated systems can increase the efficiency of parking management in congested urban areas.

2. M. A. Khan et al. (2017):

Title: Developed an automated parking system using an intelligent parking assistant.

It integrated Internet of Things (IoT) sensors to detect available parking spots and convey real-time data to vehicle owners via a mobile application.

3. P. H. Shih et al. (2018):

Title: Designed a smart parking system utilizing IoT sensors for vehicle detection and cloud computing to store and process parking data.

The system dynamically allocated parking spaces based on demand and traffic patterns, improving space utilization and reducing congestion. Their system analyzes historical data to predict peak hours and optimize parking space usage in real-time.

4. L. S. Zhang et al. (2019):

Title: Introduced a mobile-based parking system that allows users to book parking spots in advance.

The system uses GPS and real-time parking availability data to guide users to the nearest available parking space. These systems allow drivers to find, reserve, and pay for parking spaces via smartphones, creating a seamless and convenient parking experience.

5. Patel et al. (2021):

Title: Focused on implementing a mobile application integrated with a smart parking system that notifies users of space availability in real time.

Users can also reserve spots and make payments directly through the app. Their system used a time-series forecasting model to predict parking space usage in different areas of the city. The framework adapts to changing traffic patterns and parking usage behaviors.

6. **X. Wang et al. (2020)**:

Title: Proposed a machine learning-based approach for predicting parking demand based on historical parking data.

Their system used a time-series forecasting model to predict parking space usage in different areas of the city. Developed an intelligent parking management framework based on big data and machine learning to optimize parking space allocation.

CHAPTER 3

SOFTWARE AND HARDWARE SPECIFICATIONS

HARDWARE REQUIREMENTS:

Processor	:	Intel core i7 or above
Speed	:	2.0. GHz
RAM	:	1GB
Hard Disk	:	40GB to 80GB

SOFTWARE REQUIREMENTS

Database	:	MySQL
Server	:	Apache
Frontend	:	HTML
Scripting language	:	Java Script
IDE	:	Sublime
Technology	:	PHP

CHAPTER 4

SYSTEM REQUIREMENT SPECIFICATION

AND ANALYSIS

4.1 Functional Requirement

A **Vehicle Parking Management System (VPMS)** aims to streamline and optimize the process of parking vehicles, ensuring smooth operations for both parking space owners and vehicle users. Below are the key functional requirements for such a system:

- Users can register using their email or phone number.

- Users can update their contact information and vehicle details.

- A secure login and authentication mechanism should be in place (e.g., password-based, OTP, or biometric authentication).

Number Of Modules

1. User Module

This module is designed for end-users (vehicle owners) who interact with the parking system to find, reserve, and pay for parking spaces. It includes:

- **User Registration & Authentication**:
  - Users can create accounts, log in, and manage profiles with personal and vehicle details.
  - Includes options for secure login (password, OTP, or biometric authentication).
- **Search for Parking Spaces**:
  - Users can search for available parking spots based on location, parking type, and availability.
  - Filters for special parking needs (e.g., accessible spaces, EV charging stations).

- **Reservation System**:
  - Users can reserve parking spaces ahead of time.
  - Specify parking duration and get a reservation confirmation.
  - Users can modify or cancel reservations.
- **Real-time Parking Space Availability**:
  - Displays live occupancy data, showing which spaces are available or occupied.
- **Payment Integration**:
  - Users can pay for parking via multiple payment methods (credit/debit cards, mobile wallets, etc.).
  - The system calculates parking fees based on duration and dynamic pricing.
  - Generate payment receipts and historical records.
- **Navigation and Directions**:
  - Real-time navigation to guide users to their reserved parking spot.
  - Provides walking or driving directions with route optimization.
- **Parking History and Reports**:
  - Users can view their parking history, past reservations, and payment details.

2. Admin Module

The Admin module is responsible for managing the overall parking system, including the parking spaces, user reservations, dynamic pricing, and overall system monitoring. It typically includes:

- **Admin Authentication**:
  - Admins can log in with credentials and multi-factor authentication to ensure secure access.
- **Parking Space Management**:
  - Add, update, or remove parking spaces from the system.
  - Categorize spaces (e.g., EV charging, accessible parking, regular spots).
  - Monitor space occupancy in real-time and ensure availability.
- **Reservation Management**:
  - View and manage all user reservations (view active, upcoming, or canceled reservations).

  - Modify or cancel user reservations when necessary (e.g., overbooked spaces, maintenance).
- **Dynamic Pricing Management**:
  - Admins can define pricing models based on time, demand, or location.
  - Set dynamic pricing for peak times, holidays, or special events.
  - Adjust prices based on parking lot occupancy and demand fluctuations.
- **Transaction and Payment Management**:
  - View detailed reports of transactions, including payment methods, amounts, and user details.
  - Process refunds or adjustments for incorrect payments.
  - Monitor payment history for security and accuracy.
- **System Alerts & Notifications**:
  - Set up and manage automated notifications for users (e.g., upcoming reservation reminders, payment confirmations, parking expiry warnings).
  - Send bulk notifications for system updates, maintenance, or promotional offers.
- **Maintenance Management**:
  - Track and schedule parking space maintenance (e.g., cleaning, repair).
  - Handle maintenance requests and issue resolutions.

4.2 Non Functional Requirements

Non-functional requirements define how the **Vehicle Parking Management System (VPMS)** should perform, focusing on aspects such as performance, security, scalability, and user experience. These requirements ensure that the system is robust, efficient, and can handle real-world conditions effectively. Below are the key non-functional requirements for a VPMS.

. 1. Performance Requirements

- **Response Time**:
  - Real-time updates for parking availability should be reflected with **minimal delay** (no more than **5 seconds**).

- **Throughput**:
  - The system should handle high traffic, with a capability to process at least **1000 simultaneous user requests** during peak hours.

2. Scalability

- **Horizontal Scalability**:
  - The system should be able to expand without significant redesign, supporting **multiple cities or regions**.
- **Capacity**:
  - The system must be able to handle a growing user base, with the capacity to support **millions of users** and **thousands of parking spaces** across various locations.

3. Availability

- **System Uptime**:
  - The system should ensure an uptime of **99.9%** or higher, meaning it should be available at all times except during scheduled maintenance periods.
- **Disaster Recovery**:
  - The system should be able to recover from failures and restore full functionality within **1 hour**.

4. Security Requirements

- **Authentication and Authorization**:
  - The system must ensure strong authentication (e.g., two-factor authentication) for both users and administrators.
- **Data Encryption**:
  - All sensitive data (e.g., user payment details, personal information) must be **encrypted** both in transit (using HTTPS/TLS) and at rest.
  - Payment transactions should be handled securely, complying with standards such as **PCI-DSS** for payment card data security.

5. Usability Requirements

- **User Interface (UI)**:
  - The system should provide an **intuitive and user-friendly interface**, with easy navigation for both mobile and web platforms.

6. Maintainability

- **Code Maintainability**:
  - The system must follow best practices for software development (e.g., version control, continuous integration, and testing).

7. Load Testing and Stress Testing

- **Capacity Under Load**:
  - The system should be able to handle peak usage periods without degradation in performance, such as during special events, holidays, or rush hours.
  - The system should be stress-tested to handle up to **10x the expected peak load** without significant performance degradation or failure.

8. Backup and Data Recovery

- **Automated Backups**:
  - The system should regularly back up all critical data (user information, parking reservations, payment data) in an automated manner, at least **once a day**.
- **Disaster Recovery Plan**:
  - The system must have a disaster recovery plan in place, ensuring the ability to restore full service within **1 hour** after any major failure.

4.3 Class Diagram

The class diagram shows a set of classes, interfaces, collaborations and their relationships.

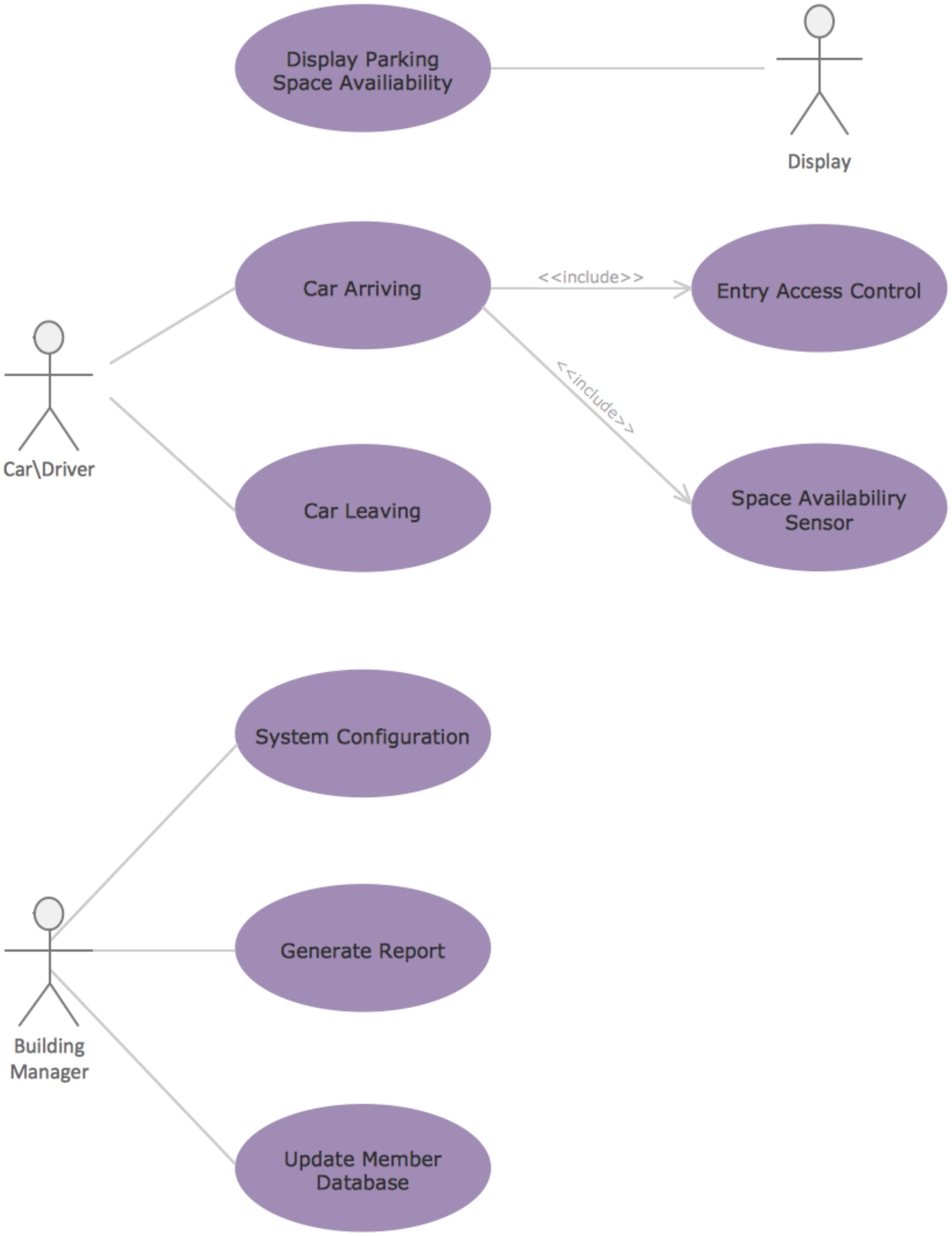
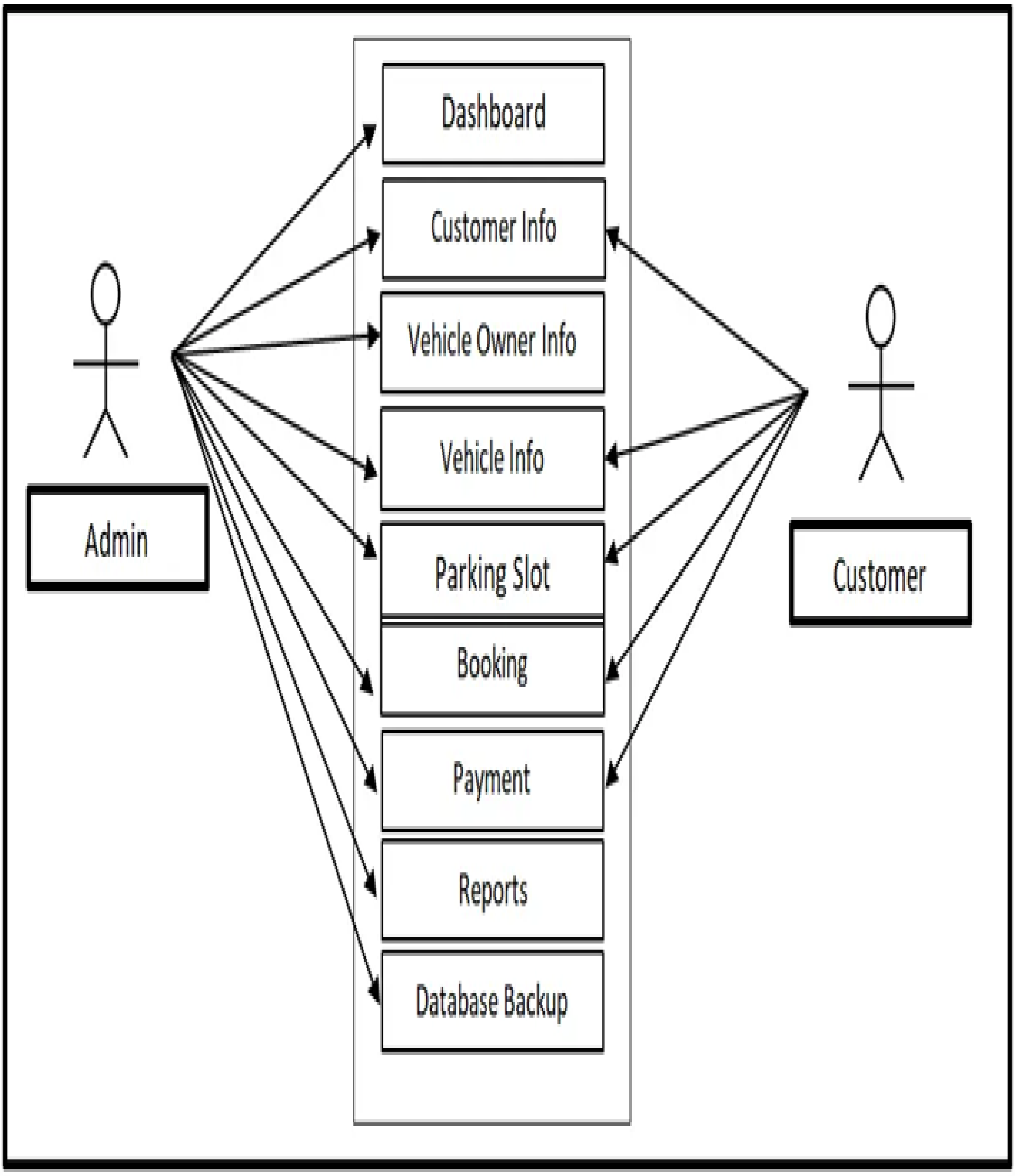


Fig 4.3 Class Diagram for Vehicle Parking Management System

4.3.1 Use Case Diagram

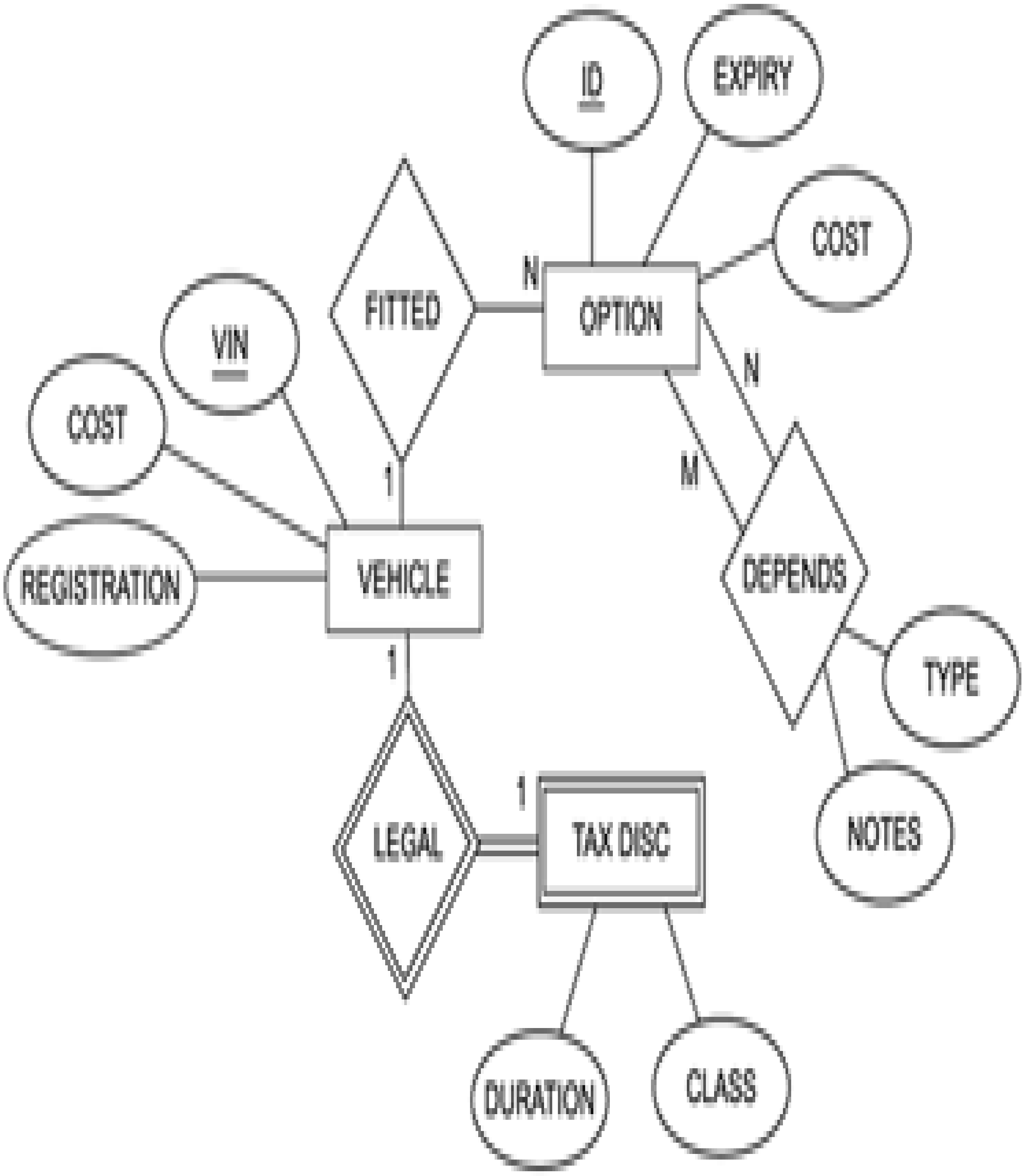
Use case diagram consists of actors, use cases and their relationships. These diagrams are especially important in organizing and modelling the behaviours of a system.

User Diagram



4.3.1 Use Case Diagram for User

4.4 E-R Diagram



4.4 E-R Diagram for Vehicle Parking Management System

4.5 Data Flow Diagram

4.5.1 Zero Level DFD

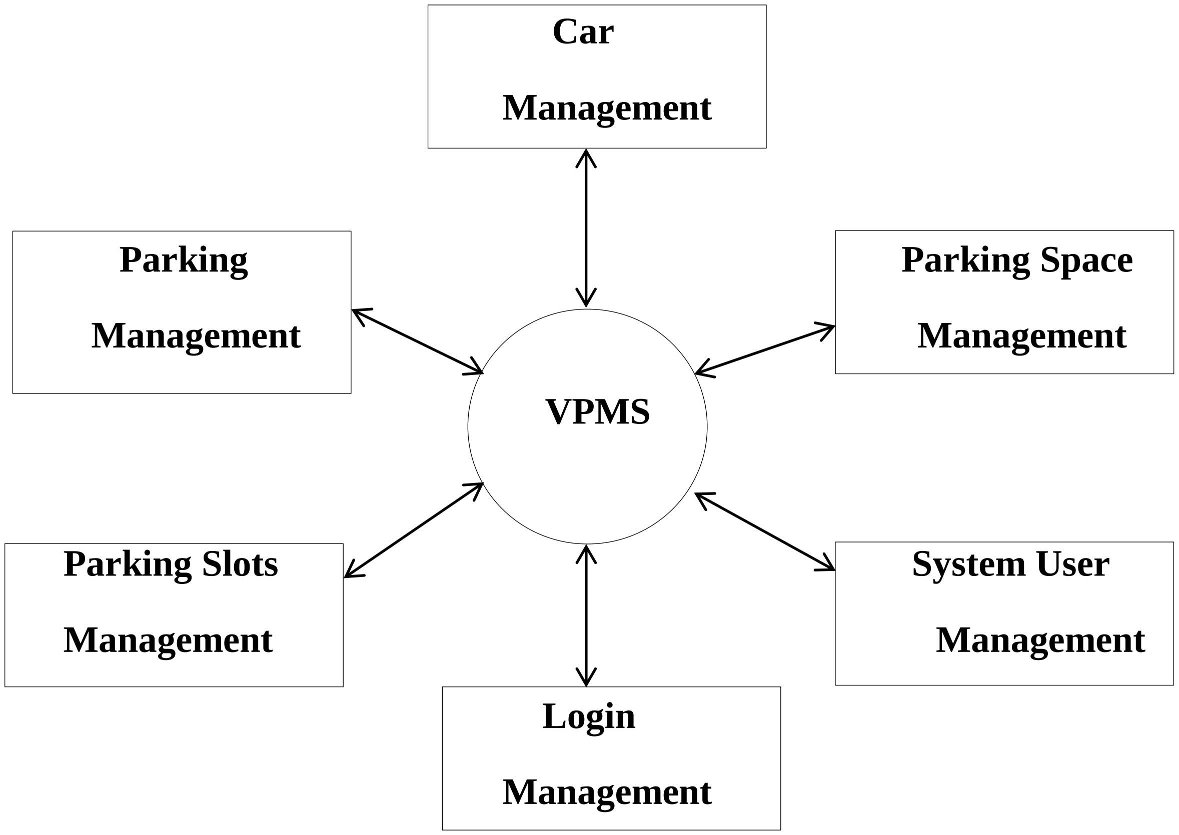


Fig 4.5.1 Zero Level DFD For Vehicle Parking Management System

4.5.2 First Level DFD

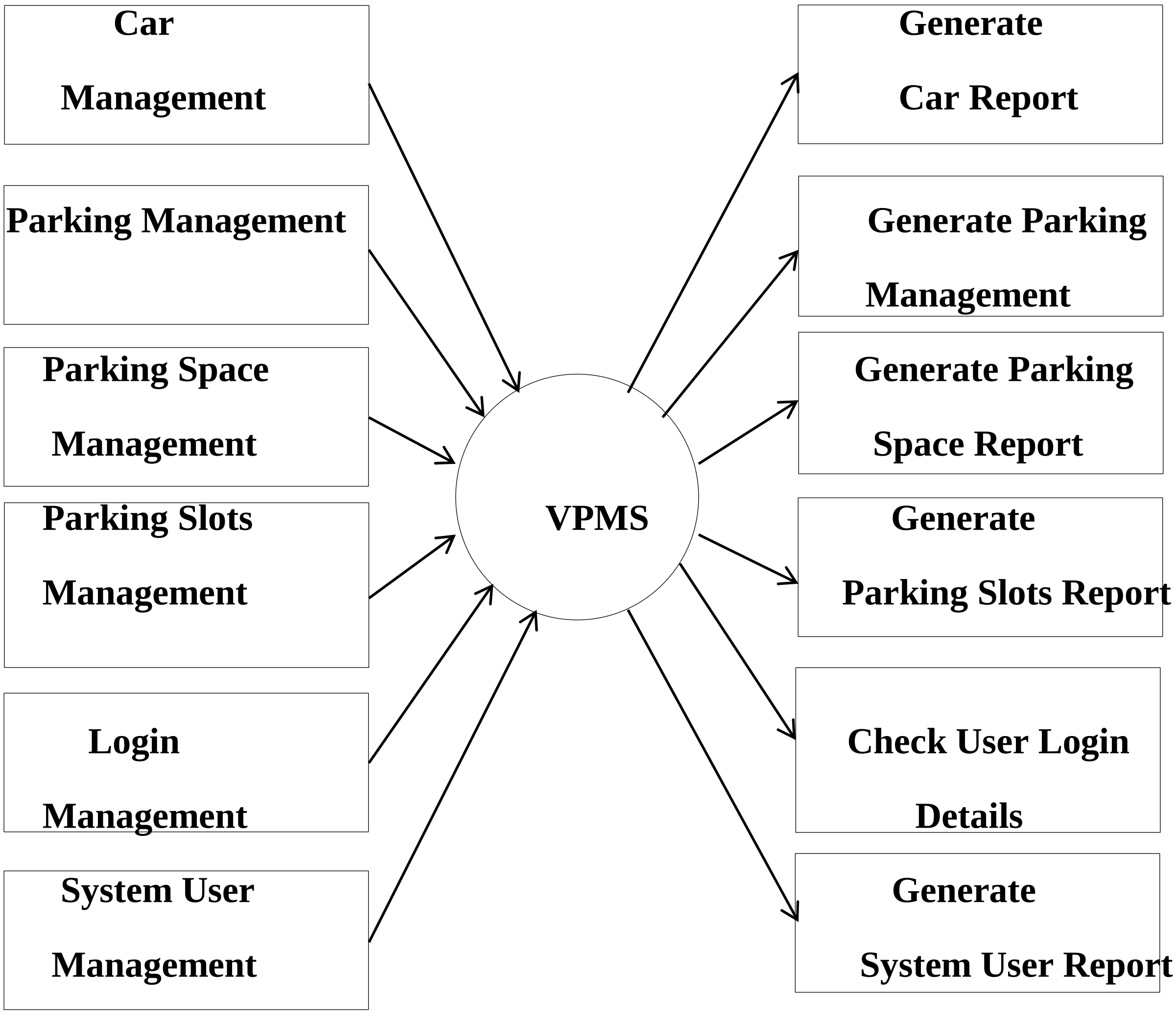


Fig 4.5.2 First Level DFD For Vehicle Parking Management System

CHAPTER 5

IMPLEMENTATION

Introduction

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.1 CODE

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['vpmsaid']==0)) {
  header('location:logout.php');
  } else{
  ?>
<!doctype html>
<html class="no-js" lang="">
```

```
<head>


   <title>VPMS - Search Vehicle</title>

 <link rel="apple-touch-icon" href="https://i.imgur.com/QRAUqs9.png">

   <link rel="shortcut icon" href="https://i.imgur.com/QRAUqs9.png">

   <linkrel="stylesheet"
href="https://cdn.jsdelivr.net/npm/normalize.css@8.0.0/normalize.min.css">

   <linkrel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/bootstrap.min.css">

   <linkrel="stylesheet"href="https://cdn.jsdelivr.net/npm/font-
awesome@4.7.0/css/font-awesome.min.css">

   <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/lykmapipo/themify-
icons@0.1.2/css/themify-icons.css">

   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/pixeden-stroke-7-
icon@1.2.3/pe-icon-7-stroke/dist/pe-icon-7-stroke.min.css">

   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/flag-icon-
css/3.2.0/css/flag-icon.min.css">

   <link rel="stylesheet" href="assets/css/cs-skin-elastic.css">

   <link rel="stylesheet" href="assets/css/style.css">

   <link
href='https://fonts.googleapis.com/css?family=Open+Sans:400,600,700,800'
rel='stylesheet' type='text/css'>

</head>
```

```
<body>
    <!-- Left Panel -->
  <?php include_once('includes/sidebar.php');?>
    <!-- Left Panel -->
    <!-- Right Panel -->
     <?php include_once('includes/header.php');?>
        <div class="breadcrumbs">
            <div class="breadcrumbs-inner">
                <div class="row m-0">
                    <div class="col-sm-4">
                        <div class="page-header float-left">
                            <div class="page-title">
                                <h1>Dashboard</h1>
                            </div>
                        </div>
                    </div>
                    <div class="col-sm-8">
                        <div class="page-header float-right">
                            <div class="page-title">
                                <ol class="breadcrumb text-right">
```

```
                                  <li>

    <a href="dashboard.php">Dashboard</a></li>

                                  <li>

    <a href="search-vehicle.php">Search Vehicle</a>

    </li>

                                  <li class="active">Search Vehicle</li>

                              </ol>

                          </div>

                      </div>

                  </div>

              </div>

          </div>

      </div>

      <div class="content">

          <div class="animated fadeIn">

              <div class="row">

               <div class="col-lg-12">

                  <div class="card">

                      <div class="card-header">

                          <strong class="card-title">Search Vehicle</strong>
```

```
                        </div>

                        <div class="card-body">

<form action="" method="post" enctype="multipart/form-data" class="form-
horizontal" name="search">


  <div class="row form-group">

 <div class="col col-md-3">

<label for="text-input" class=" form-control-label">Search By Parking
Number/ Registration No</label></div>


<div class="col-12 col-md-9"><input type="text" id="searchdata"
name="searchdata"class="form-control"required="required"
autofocus="autofocus" ></div>


 </div>

 <p style="text-align: center;"> <button type="submit" class="btn btn-primary
btn-sm" name="search" >Search</button></p>

                              </form>


 <?php

if(isset($_POST['search']))

{
```

```
$sdata=$_POST['searchdata'];

  ?>

  <h4 align="center">Result against "<?php echo $sdata;?>" keyword </h4>

                          <table class="table">

              <thead>

    <tr>

    <tr>

    <th>S.NO</th>




                  <th>Parking Number</th>

                  <th>Owner Name</th>

                  <th>Vehicle Reg. Number</th>

                 <th>Action</th>

              </tr>

              </tr>

             </thead>


             <?php
$ret=mysqli_query($con,"select *from    tblvehicle where ParkingNumber like
'$sdata%' || RegistrationNumber like '$sdata%'");
```

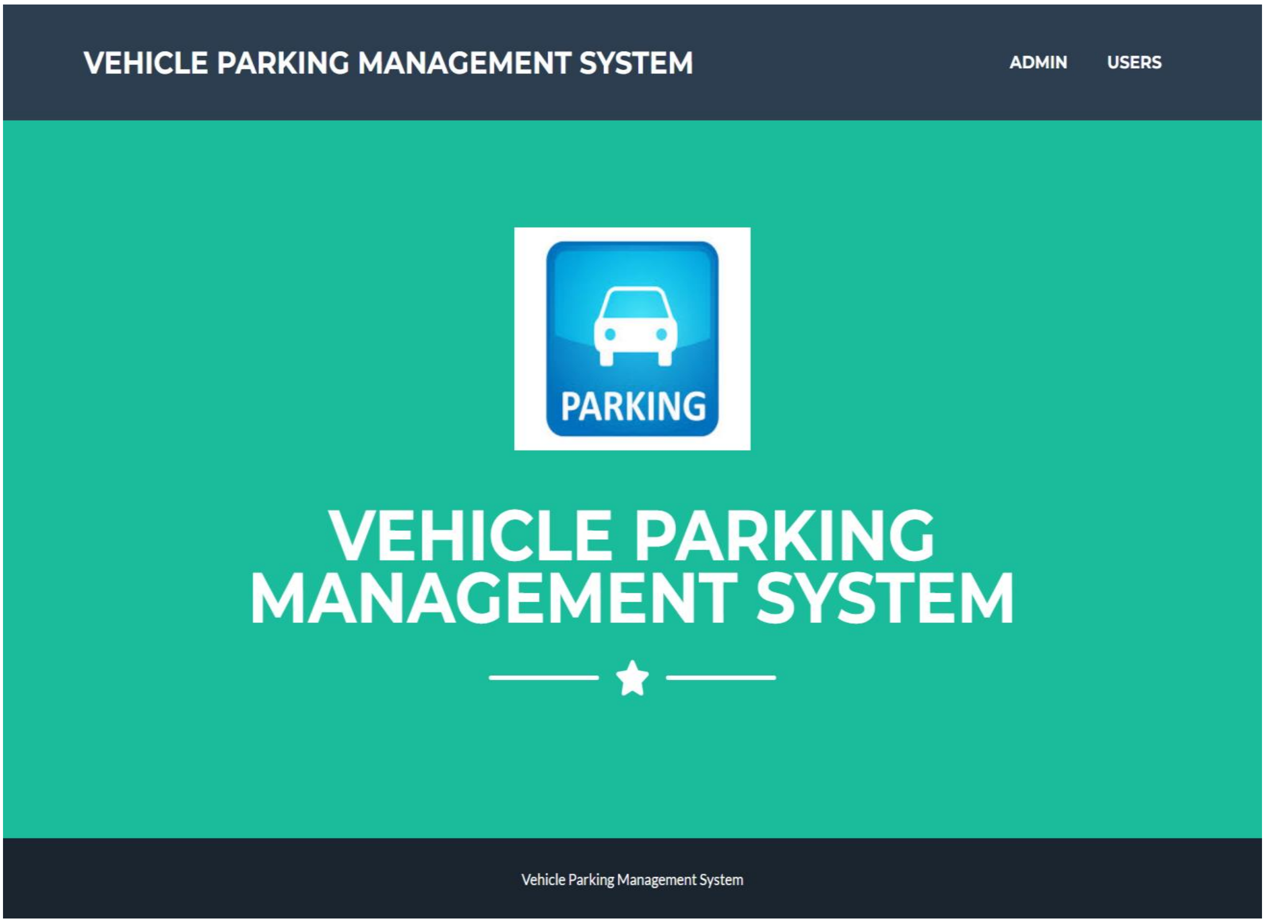
```
$num=mysqli_num_rows($ret);

if($num>0){

$cnt=1;

while ($row=mysqli_fetch_array($ret)) {

?>

<tr>  <td><?php echo $cnt;?></td>

        <td><?php  echo $row['ParkingNumber'];?></td>

        <td><?php  echo $row['OwnerName'];?></td>

        <td><?php  echo $row['RegistrationNumber'];?></td>


                <td><a  href="view-incomingvehicle-detail.php?viewid=<?php  echo
$row['ID'];?>" target="blank" class="btn btn-primary btn-sm">View</a></td>

              </tr>


       <?php

$cnt=$cnt+1;

} } else { ?>

    <tr>

   <td colspan="8"> No record found against this search</td>

  </tr>

 <?php } }?>
```

```
              </table>

                 </div>

               </div>

            </div>

         </div>


      </div><!-- .animated -->

    </div><!-- .content -->

    <div class="clearfix"></div>

    <?php include_once('includes/footer.php');?>

    </div><!-- /#right-panel -->


    <!-- Right Panel -->

    <!-- Scripts -->

    <script
    src="https://cdn.jsdelivr.net/npm/jquery@2.2.4/dist/jquery.min.js"></script>

    <script
    src="https://cdn.jsdelivr.net/npm/popper.js@1.14.4/dist/umd/popper.min.js"></
    script>
```

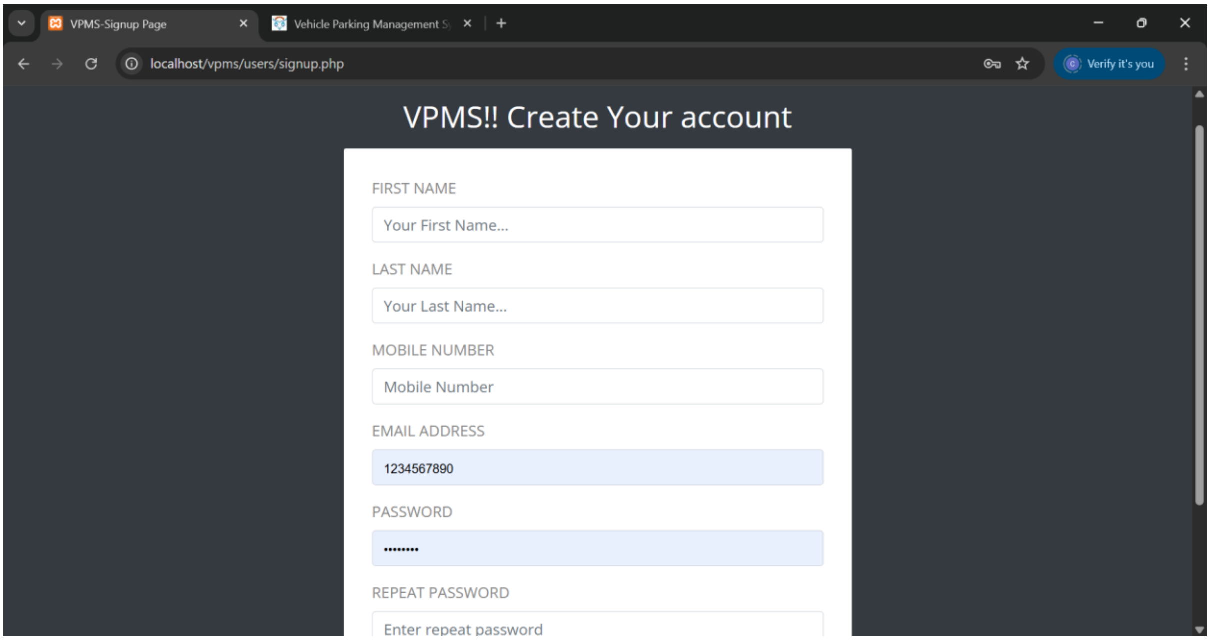
```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/jquery-match-height@0.7.2/dist/jquery.matchHeight.min.js"></script>

<script src="assets/js/main.js"></script>

</body>

</html>

<?php }  ?>
```

5.2 Screenshots

5.2.1 Home Page


VEHICLE PARKING MANAGEMENT SYSTEM
ADMIN
USERS
PARKING
VEHICLE PARKING MANAGEMENT SYSTEM
Vehicle Parking Management System

5.2.2 User Sign up



VPMS!! Create Your account

FIRST NAME

Your First Name...

LAST NAME

Your Last Name...

MOBILE NUMBER

Mobile Number

EMAIL ADDRESS

1234567890

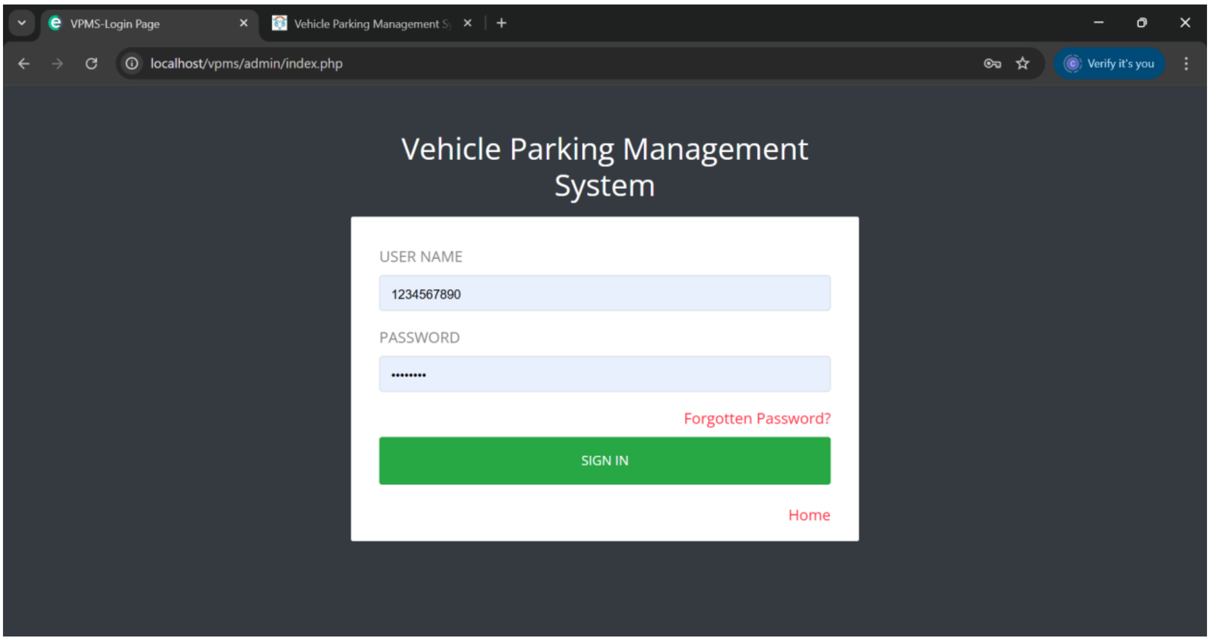
PASSWORD

••••••••

REPEAT PASSWORD

Enter repeat password

5.2.3 Admin Login Page



Vehicle Parking Management System

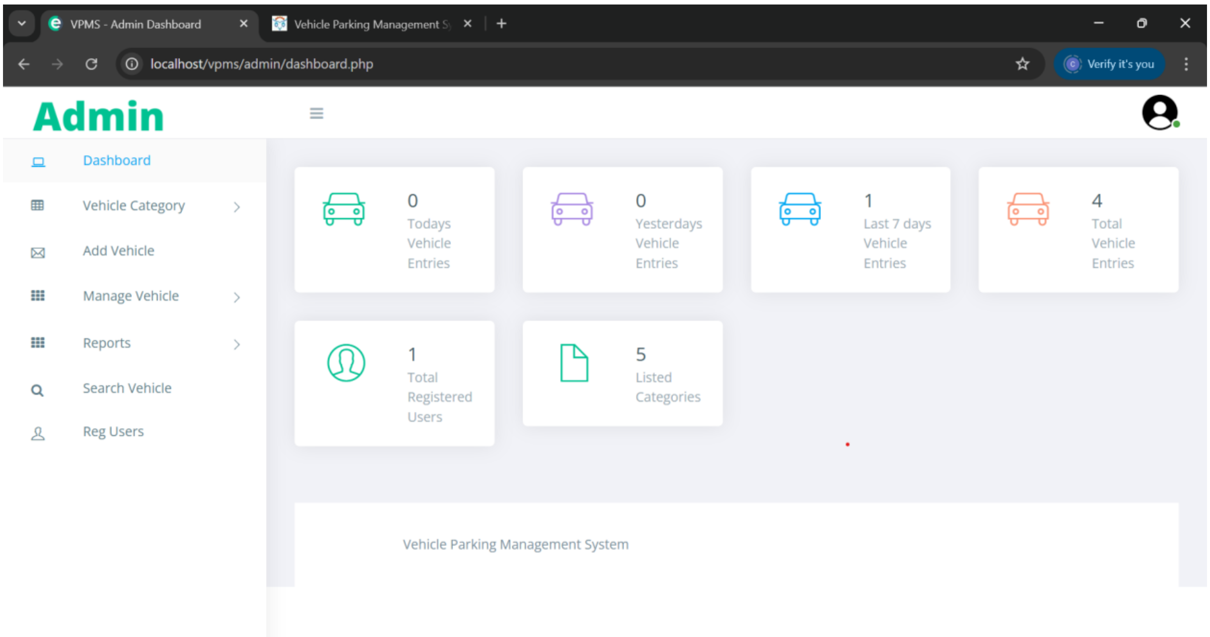
USER NAME

1234567890

PASSWORD

••••••••

5.2.4 Admin Dashboard



0 Todays Vehicle Entries

0 Yesterdays Vehicle Entries

1 Last 7 days Vehicle Entries

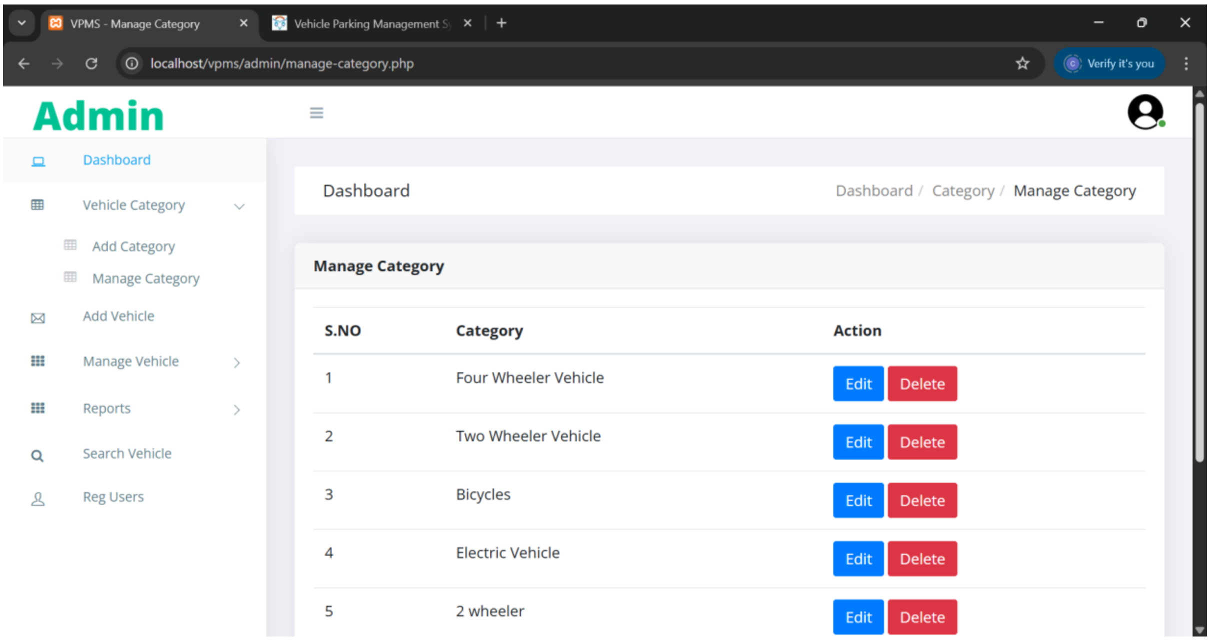
4 Total Vehicle Entries

1 Total Registered Users

5 Listed Categories

Vehicle Parking Management System

5.2.5 Manage Category

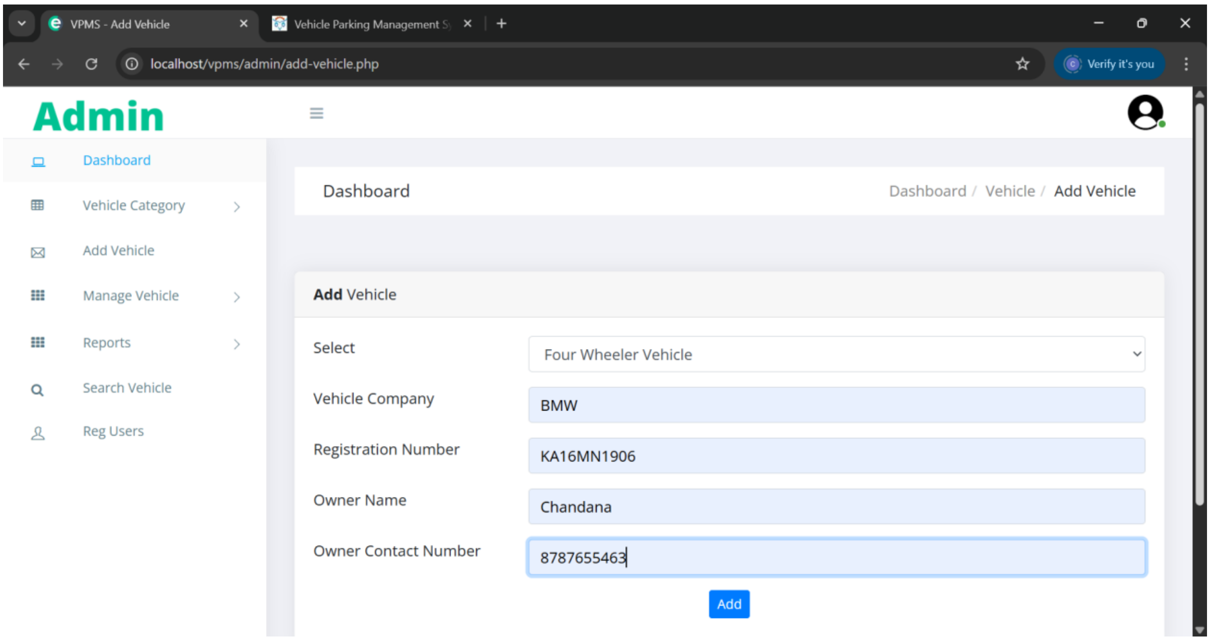


Dashboard — Dashboard / Category / Manage Category

Manage Category

S.NO	Category	Action
1	Four Wheeler Vehicle	Edit Delete
2	Two Wheeler Vehicle	Edit Delete
3	Bicycles	Edit Delete
4	Electric Vehicle	Edit Delete
5	2 wheeler	Edit Delete

5.2.6 Adding Vehicle

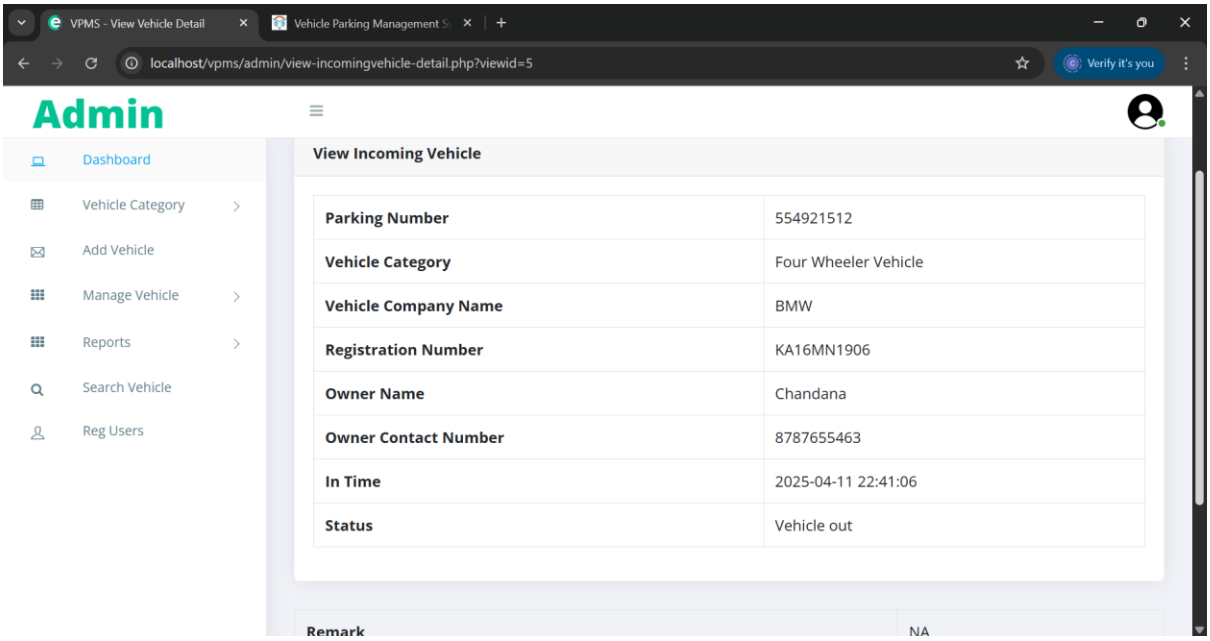


Dashboard

Add Vehicle

Select	Four Wheeler Vehicle
Vehicle Company	BMW
Registration Number	KA16MN1906
Owner Name	Chandana
Owner Contact Number	8787655463

5.2.7 Incoming Vehicle

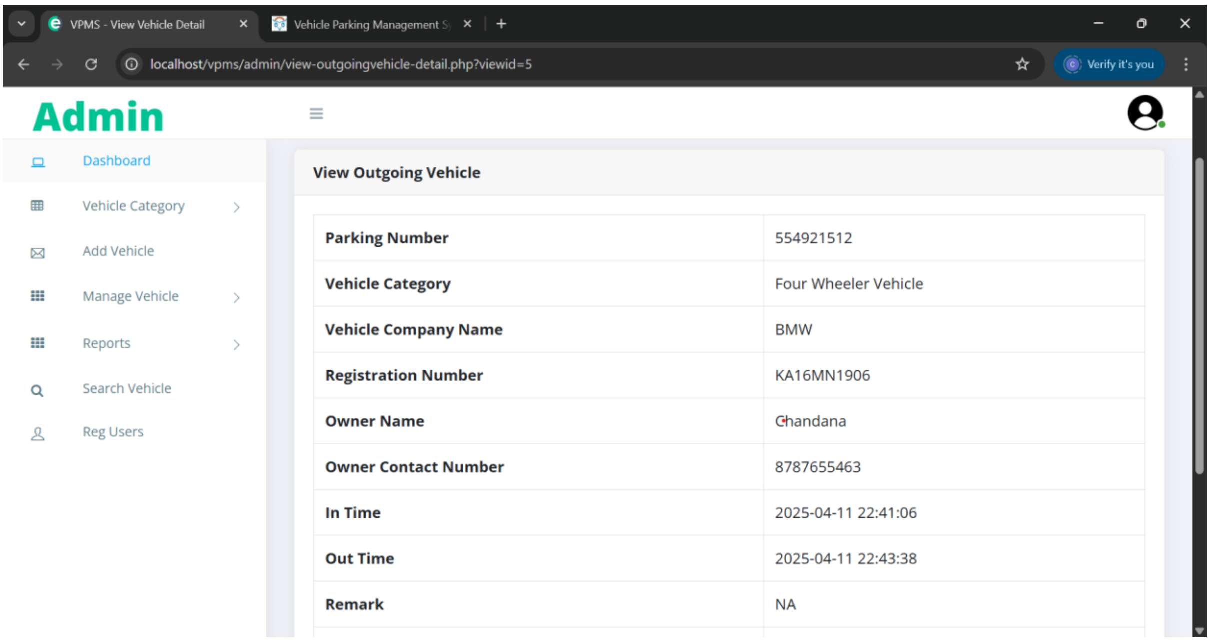

VPMS - View Vehicle Detail
Vehicle Parking Management S
localhost/vpms/admin/view-incomingvehicle-detail.php?viewid=5
Verify it's you
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Search Vehicle
Reg Users

View Incoming Vehicle

Parking Number	554921512
Vehicle Category	Four Wheeler Vehicle
Vehicle Company Name	BMW
Registration Number	KA16MN1906
Owner Name	Chandana
Owner Contact Number	8787655463
In Time	2025-04-11 22:41:06
Status	Vehicle out

Remark	NA

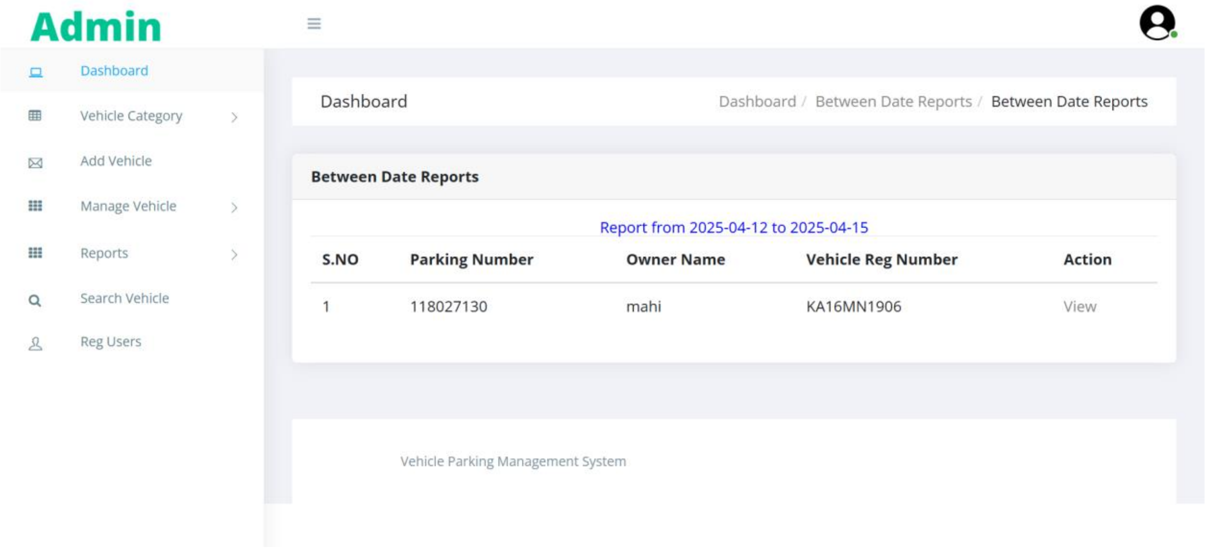
5.2.8 Outgoing Vehicle


VPMS - View Vehicle Detail
Vehicle Parking Management S
localhost/vpms/admin/view-outgoingvehicle-detail.php?viewid=5
Verify it's you
Admin
Dashboard
Vehicle Category
Add Vehicle
Manage Vehicle
Reports
Search Vehicle
Reg Users

View Outgoing Vehicle

Parking Number	554921512
Vehicle Category	Four Wheeler Vehicle
Vehicle Company Name	BMW
Registration Number	KA16MN1906
Owner Name	Chandana
Owner Contact Number	8787655463
In Time	2025-04-11 22:41:06
Out Time	2025-04-11 22:43:38
Remark	NA

5.2.9 Reports



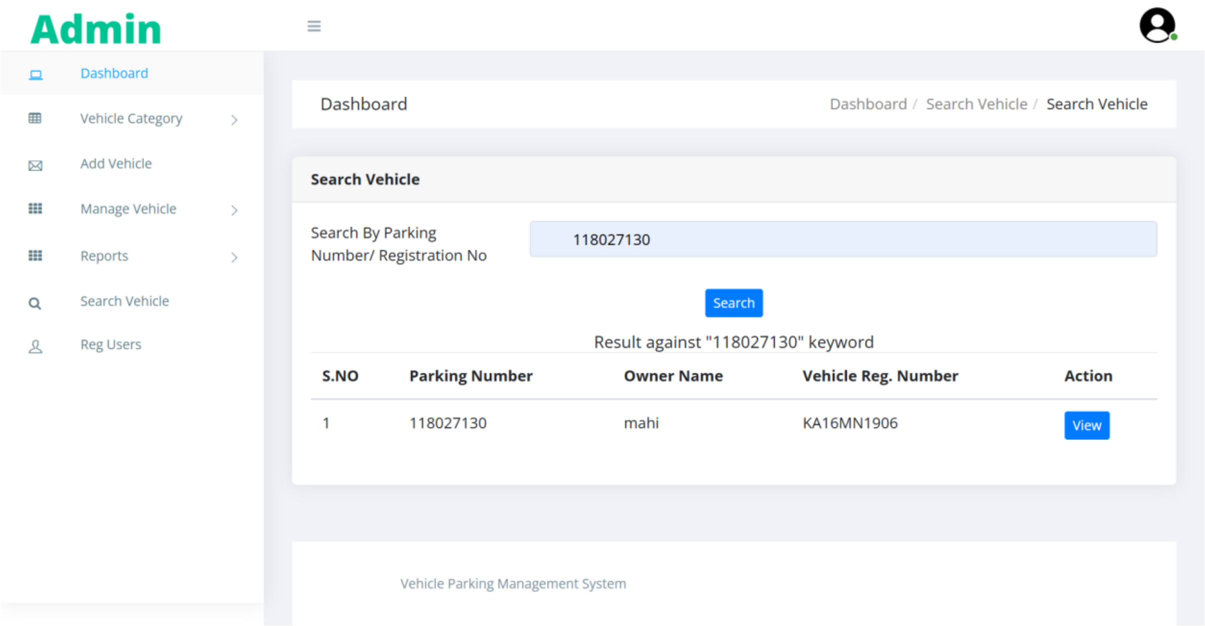
Between Date Reports

Report from 2025-04-12 to 2025-04-15

S.NO	Parking Number	Owner Name	Vehicle Reg Number	Action
1	118027130	mahi	KA16MN1906	View

Vehicle Parking Management System

5.3 Searching Vehicle



Search Vehicle

Search By Parking Number/ Registration No: 118027130

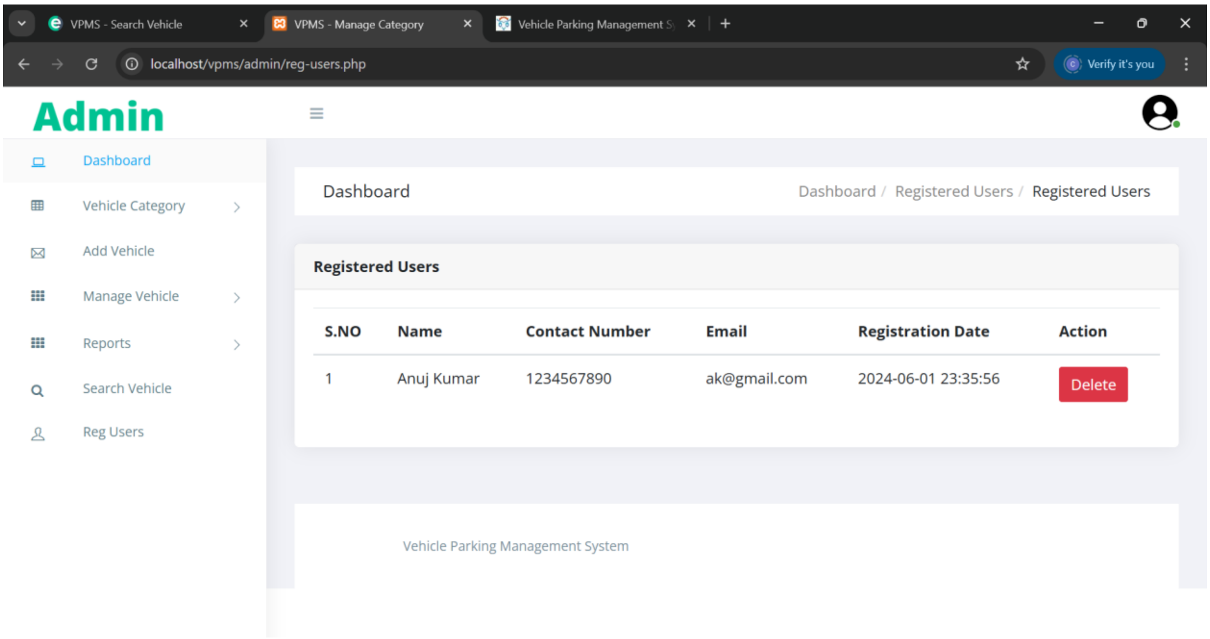
Search

Result against "118027130" keyword

S.NO	Parking Number	Owner Name	Vehicle Reg. Number	Action
1	118027130	mahi	KA16MN1906	View

Vehicle Parking Management System

5.3.1 User Registration

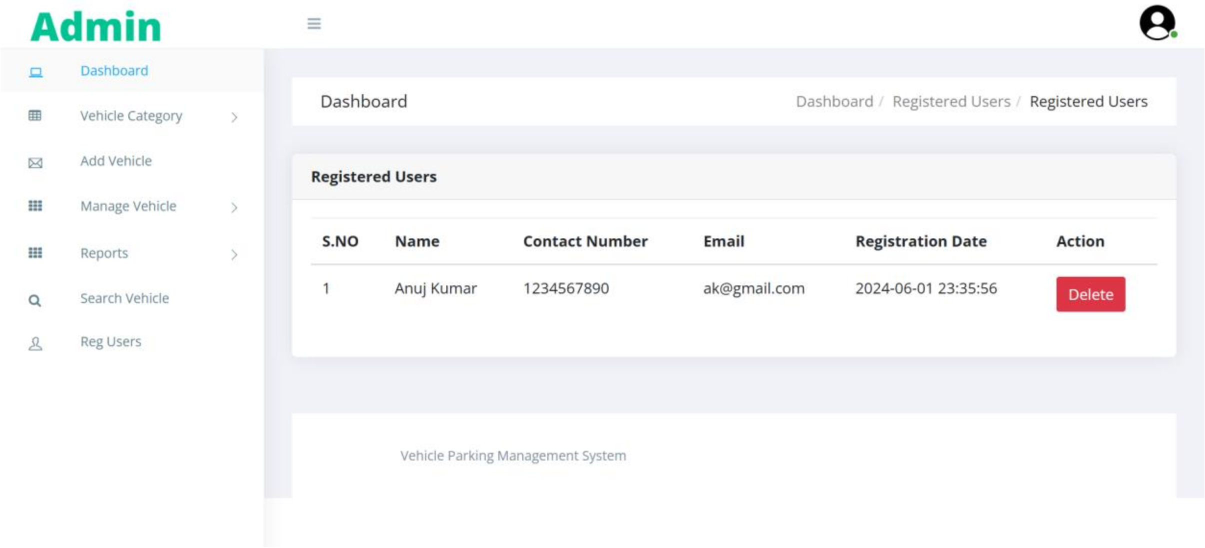


Registered Users

S.NO	Name	Contact Number	Email	Registration Date	Action
1	Anuj Kumar	1234567890	ak@gmail.com	2024-06-01 23:35:56	Delete

Vehicle Parking Management System

5.3.2 Registered Users



Registered Users

S.NO	Name	Contact Number	Email	Registration Date	Action
1	Anuj Kumar	1234567890	ak@gmail.com	2024-06-01 23:35:56	Delete

5.4 TESTING

Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionalities of components, sub assemblies, and/or a finished product it is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

Types of Testing

5.4.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing, we have is white box oriented and some modules the steps are conducted in parallel.

5.4.2 Integration Testing

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus the system testing is a confirmation that all is correct and an opportunity to show the user that the system works. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items.

5.4.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.4.4 White Box Testing

This type of testing ensures that All independent paths have been exercised at least once All logical decisions have been exercised on their true and false sides All loops are executed at their boundaries and within their operational bounds All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

CHAPTER 7

FUTURE ENHANCEMENT

Future enhancements for a Vehicle Parking Management System can significantly improve efficiency, user convenience, and sustainability. One of the key advancements would be the integration of smart technologies such as IoT-based sensors and AI-powered cameras for real-time detection of parking space availability. This can be coupled with automatic number plate recognition (ANPR) to streamline vehicle entry and exit without the need for manual intervention. Mobile app integration can further enhance user experience by offering features like real-time updates, pre-booking of parking spots, digital payments, and even in-app navigation to guide users directly to their designated space.

Additionally, the system can support dynamic pricing models that adjust rates based on demand, helping to manage peak-hour congestion and maximize revenue. Integration with electric vehicle (EV) charging stations, along with support for contactless and digital payments, would cater to the growing EV user base. From an administrative perspective, the use of data analytics and machine learning can provide valuable insights into usage patterns, enabling predictive planning and efficient space utilization. Moreover, sustainable practices such as smart lighting and carbon footprint monitoring can be incorporated to promote eco-friendly operations. Overall, these enhancements can transform traditional parking systems into intelligent, user-friendly, and environmentally conscious solutions.

CHAPTER 8

CONCLUSION

In conclusion, the **Vehicle Parking Management System** project successfully addresses the growing challenges of managing parking spaces efficiently in urban environments. By automating the process of vehicle entry, exit, and space allocation, the system reduces human error, saves time, and improves overall convenience for both administrators and users. The system offers features like real-time availability tracking, secure data storage, automated billing, and user-friendly interfaces, making it a scalable solution for malls, offices, and public parking areas.

Goals Achieved

- Automation of Parking Processes
- Real-time Parking Space Monitoring
- User-friendly Interface
- Database Management
- Improved Efficiency & Time Management

CHAPTER 9

BIBLIOGRAPHY

- www.w3schools.com
- **php**.net
- en.wikipedia.org/wiki/**PHP**
- www.hotscripts.com/category/**php**/
- www.**apache**.org
- www.**mysql**.com/click.php?e=35050

1. Proposed a fully automated vehicle parking management system based on sensors and cameras. [1]
2. Developed an automated parking system using an intelligent parking assistant.[2]
3. Designed a smart parking system utilizing IoT sensors for vehicle detection and cloud computing to store and process parking data. [3]
4. Introduced a mobile-based parking system that allows users to book parking spots in advance. [4]
5. Focused on implementing a mobile application integrated with a smart parking system that notifies users of space availability in real time. [5]
6. Proposed a machine learning-based approach for predicting parking demand based on historical parking data. [6]