



Towards Efficient Fleet Handling: A Modular Vehicle Management System for Digital Platforms

Mukul Mahanwal and Mr. Yogesh Kumar

PG-IT Student, Department of Information Technology, Jagannath Community College, Rohini, New Delhi, India

Assistant Professor, Department of Information Technology, Jagan Institute of Management Studies, Rohini, New Delhi,

India

Abstract

The main objective of this project is developing and building a web-based Vehicle Management System that basically intends to streamline and digitize the procedure of vehicle management. While users need a common platform to search for vehicles, make bookings, and even chat with the service provider, the administrators on the other side require an interface for the management of bookings, vehicles, users, and drivers. The system has been developed in PHP and MySQL in order to keep data consistent as well as making it easily accessible through the frontend. Some of the major features that the platform offers include, among others, real-time update on the booking status, secured login module, and an administrative-level control panel. Implementation of this system will definitely improve operational efficiencies, reduce manual work, and provide better service delivery in the transport sector. This research hence dwells into system architecture, critical modules, and the overall impact on Vehicle service management.

Keywords:- Vehicle Management System, Web-Based Application, Fleet Management, Online Vehicle Booking, PHP and MySQL, Razorpay Integration, Driver Management System, Booking Automation, Administrative Dashboard, Session Management, Relational Database Design, Secure Login System, Role-Based Access Control, User Feedback System

Introduction

The growth in the demand for transport services is particularly noticed in today's world which is backed by technology. With urbanization on the rise, systematic vehicle allocation and organization is a more critical concern for fleet management companies. Administrative and record keeping tasks automated systems such as booking engines. contain a variety of functions which, when performed manually, can be tedious and time-consuming. Furthermore, these processes are not very responsive to change and user needs which is technology driven. It is clear that automated processes need to be responsive and adaptive to users' vehicle needs.

The system contains software modules for administrators, customers, and agents. It was developed in PHP and is backed by a structured, relational database. The system supports real time checks of the item or service availability and online payment through the razorpay payment system. In addition, it offers a built in chatbot which, among other things, answers users' most common questions. The integration of these advanced data management tools makes the system scalable: the use of ER diagrams and data dictionaries enable better data maintenance and organization.

The Vehicle Management System (VMS) project was developed as a comprehensive web-based system to streamline and improve vehicle service management. The system was designed to provide end-to-end support for both administrators and users. Users can easily view available cars, review the services and make reservations for the vehicles, and stay updated on status thanks to an intuitive interface. On the other hand, administrators can control a variety of tasks from a single interface, such as adding or deleting cars, registering drivers, accepting or rejecting reservations, and responding to client comments.

Literature Review

With the development of information technology, fleet management has been changed significantly. There are many studies and systems have been created over time to automate and improve the management of operations of the vehicles. This section examines current fleet administration, booking platform, and vehicle management research and systems to determine their value, constraints, and applicability to the project's advancement.

In many traditional vehicle management systems, manual record-keeping was done using spreadsheets and paper-based systems. These approaches are often ineffective, prone to human error, and lack the scalability required to manage extremely large volumes of data, despite their apparent simplicity. Companies that used manual systems often faced issues like duplicate bookings, lost records, and a lack of accountability, according to numerous studies.

Methodology

The development of the vehicle management system (VMS) followed a systematic function that aimed to provide a strong, user -friendly and scalable platform. The acting software developed for this project combines the principles of the life cycle (SDLC) with practical implementation strategies to ensure that the system meets both functional and non-functional requirements.

1. Requirement Analysis

The initial phase involves identifying the most important needs of the system from both the user and the administrative approach. This was done through a combination of domain research and real -world vehicle ordering and management work flight observation

2. System Design

When the requirements were defined, the next step system was the design of architecture and database scheme. The system was structured using a client-server model with three-level architecture:

- **Presentation layer:** User interface, HTML, CSS and JavaScript are created.
- **Logic layer:** PHP script to handle user requests, treatment and logic on the server side.
- **Data Layer:** MYSQL Database for User Information, Vehicle Data, Ordering Posts and Reaction Storage.

3. System development

The system was developed as a backend programming language using PHP, which is due to its compatibility with Webservers and easy integration with MySQL. Development was a modular, such as focused on individual characteristics: for example:

- **User module:** Registration, login, vehicle netting, ordering and feedback.
- **Administrator modules:** Dashboards, vehicles and driver management, ordering and user management.

4. Tests

To ensure that the system was flawless and several levels were tested to work expected:

- **Unit test:** Each module and function was personally tested to confirm the expected output.
- **Integration test:** The module was tested at the same time to ensure uninterrupted communication and data flow.
- **User acceptance testing (UAT):** The simulated user sessions were performed from the end user's point of view to evaluate the correct and performance of the system.

5. Deployment

After a hit testing, the gadget became deployed on a neighborhood server the usage of XAMPP. This allowed for a managed environment where final evaluations could be made. The deployment segment also covered getting ready person documentation and admin manuals to support effective system utilization.

6. Documentation

Detailed technical documentation changed into prepared, along with a statistics dictionary and ER version, to assist further improvement, renovation, and scalability of the machine.

System Design

The layout of the Vehicle Management System (VMS) plays a crucial position in figuring out the performance, scalability, and maintainability of the software. This section presents complete assessment of the system architecture, components, database design, and user interface design. The purpose of this design is to provide a streamlined and structured platform for coping with automobile bookings, users, vehicles, drivers, and administrative capabilities.

1. Architecture Overview

The machine follows a three -level structure:

- Presentation layer: Interface with a consumer via a browser (HTML, CSS, JavaScript).
- Application team: Commercial business handles general knowledge (PHP)
- Data Layer: Database manages operations (MySQL).

The gadget is divided into later core modules:

a. User module

- User registration and login.
- Browse car listing with information and photo.
- Fully based book motors for accessibility.
- Submit the response to services.

b. Admin -Module

- Administrator login and dashboard.
- Add, replace or delete car information.
- Managing Personal and Driving Power Register.
- Approve or reject reserving requests.
- Monitor the response presented through customers.

c. Booking Module

- Order form with car and individual details.
- Check the availability before confirming a reserve.
- See all future, running or finished ordering.
- Administrative approval work flow for each reservation.

d. Driver administration module

- Register new drivers with personal information and license information.
- Assign drivers for specific cars.
- Monitor the driver's reputation and general performance.

e. Driver Module

- Register a new driver and manage driver details
- Assign drivers to the Vehicles
- View Drivers

f. Feedback module

- Submit feedback given by the user
- Manage and view feedback (admin)

2. Database Design

The database is structured using relationship tables with primary and foreign major obstacles to maintaining the stability of data. Large tables include:

- Users (User_id, Name, E -Post, Password, Role)
- Vehicles (vehicle_ID, type, model, registration_o, status)
- Booking (booking_id, user_id, vehicle_id, date, status)
- Driver_id, name, license_no, phone, vehicle_id)
- Feedback (feedback_id, user_id, message, date)

Each table is generalized for at least third general form (3NF) to reduce profits and improve the query performance.

4. User Interface Design

The user interface is designed to be simple, responsible and comfortable. The pages include:

- Website: System shows links to observation, services and other pages.
- Gallery page: Lists available vehicles with pictures and specifications.
- Order page: To select a vehicle, user user to provide information and request ordering.

- Administrator Dashboard: Sidefield -based navigation with access to all management functions.

5. Security Design

To ensure security and privacy, the following measures are used:

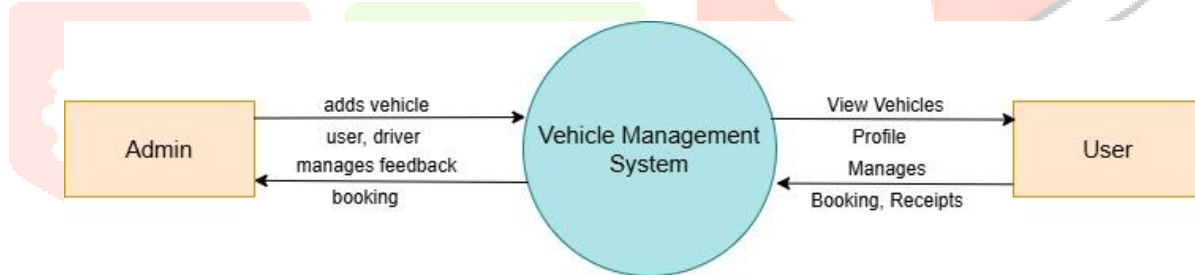
- Password for user approval user.
- Increased control to limit unauthorized access.
- Verification of the servers side to prevent SQL injection and script across the site (XSS).
- Roll -based access control (user/administrator) for data separation.

6. Scalability and Extensibility

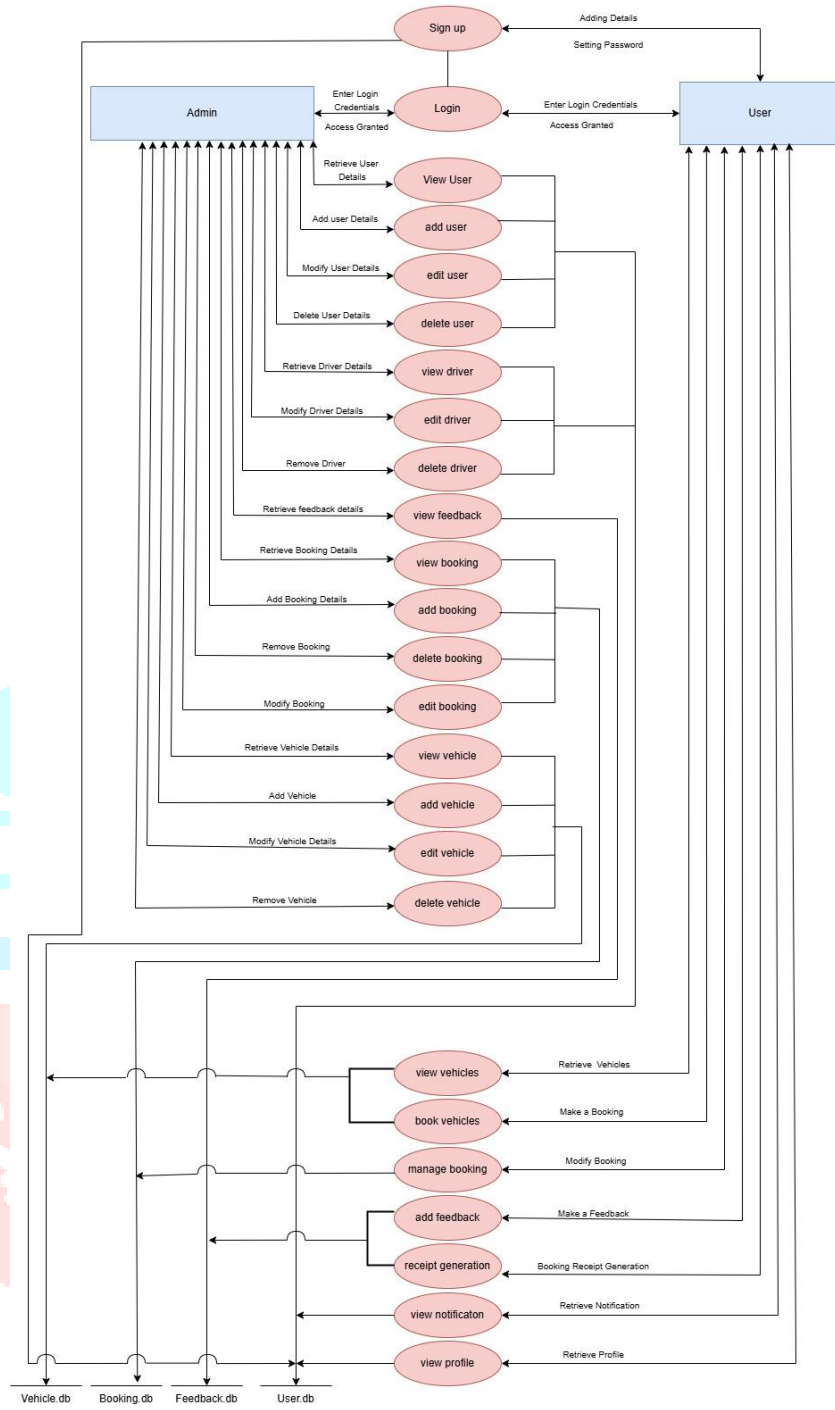
The system is designed with modularity in the mind, which provides easy integration of several facilities in the future. Possible marketing includes: Integration of payment gateways.

- SMS/E Postal alerts for confirmation of order.
- Mobile app version for wider access.

Diagram

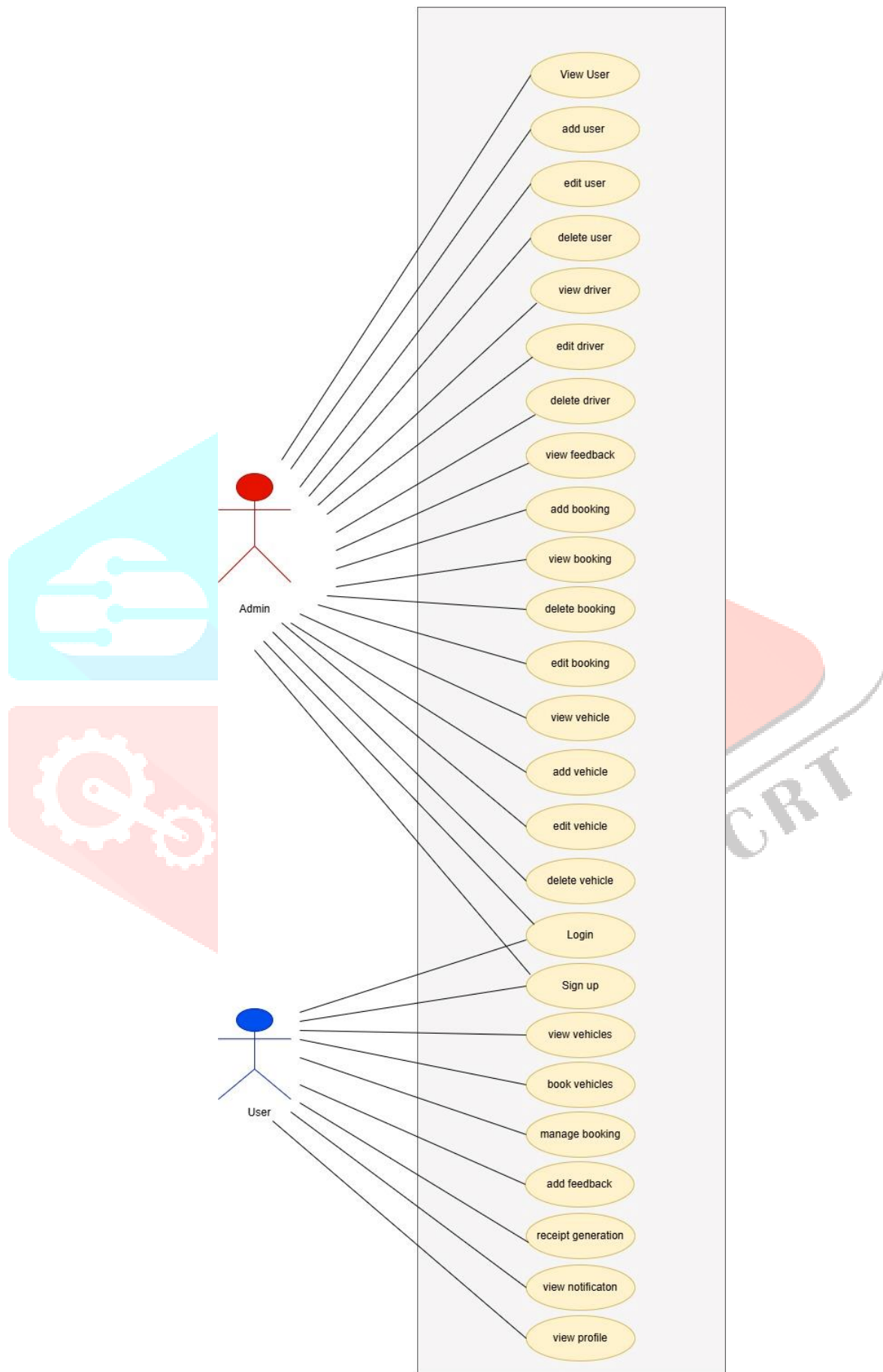


DFD Level 1



DFD Level 2

USE CASE DIAGRAM



Results and Discussion

• User registration and login

Users were able to register, safely log in and reach personal ordering and reaction facilities. Session Management worked as expected to prevent unauthorized access.

Surfing and ordering vehicles: Users can view all available vehicles, check their specifications and submit the order request. The order was recorded exactly in the database, and the administrators were available for the review.

• Administrative dashboard

Administrators had full control over the platform. Facilities such as vehicle joints, driver registration, ordering and response moderation were tested and found that it worked firmly.

• Database integrity

Use of a ratio allowed for even data processing. Raw operations (creating, reading, updating, deleting) were continuously performed without loss of data or profits.

• Feedback system

Users can present reviews or suggestions, which were stored and set by administrators for service form analysis.

Conclusion

The vehicle management system successfully addresses the challenges related to manual ordering of vehicles and fleet management by offering a reliable, user -friendly and skilled digital solution. Through its modular design and role -based access control, the operation for both users and administrators simplifies. Developed with PHP and MySQL, the system indicates how web technologies can effective transport -related services. While the project achieves the main objectives, it also opens up for future promotional opportunities such as online payment, GPS tracking and mobile application integration to expand its functionality and user access to integration.

References

1. Haider, S., et al. (2018). "Web-based Campus Management System using PHP and MySQL." *International Journal of Computer Applications*, 180(42), 30–34.
Focus: System design and modular development using PHP-MySQL stack.
2. Mahmud, R., et al. (2020). "Secure Web-Based Information System Using PHP and MySQL." *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(4), 234–240.
Focus: Authentication, session handling, and SQL injection prevention in PHP-MySQL web apps.
3. Munir, M., & Ali, S. (2017). "A Security Framework for Web-based Applications Using Role-Based Access Control." In *2017 International Conference on Open Source Systems and Technologies (ICOSST)*, IEEE.

Focus: Role-based access controls and session management for secure systems.

4. Haritha, T. A., & Thomas, M. (2021). "Secure Online Payment System Using Razorpay Gateway." *International Journal of Engineering Research & Technology (IJERT)*, 10(4), 22–27.
Focus: Integrating Razorpay API for secure transaction handling.
5. Chandrasekaran, A., & Vinoth, S. (2022). "A Smart Vehicle Booking System Based on Web Application." *International Journal of Engineering Research & Technology (IJERT)*, 11(2), 167–171.
Focus: Online booking interfaces and vehicle allocation systems using web technologies.
6. Agrawal, A., et al. (2021). "Vehicle Fleet Management System using Web and Android Platform." *International Research Journal of Engineering and Technology (IRJET)*, 8(6), 4583–4587.
Focus: Multi-platform fleet system using web.
7. Kumar, A., & Yadav, R. (2020). "Design and Implementation of a Web-Based Transport Management System." *International Journal of Advanced Research in Computer and Communication Engineering*, 9(4), 55–60.
Focus: Web-based platform for admin, vehicle, and booking management.

