

Car Rental Hub A Web-Based Vehicle Rental Management System

Hon Mayur Chandrakant¹ Prof. Vishwatej Pisal²

¹Student ²Assistant Professor

^{1,2}Master of Computer Applications

^{1,2}Anantrao Pawar College of Engineering & Research, Pune Affiliated to Savitribai Phule Pune University, India

Abstract — The rapid evolution and expansion of internet-based services have profoundly changed the manner in which people access transportation services. Conventional car rental companies, which traditionally rely on walk-in customers, telephone calls, and paper-based records, have become incapable of meeting the rapidly changing needs and expectations of modern customers, who demand speed, transparency, and instant availability of services. In this paper, the conceptual design, architectural planning, and the development blueprint of the proposed Car Rental Hub, a completely internet-based car rental and management system based on a Java Spring Boot backend, PostgreSQL relational database, and HTML/CSS/JavaScript frontend, is presented as a solution to the limitations of conventional car rental companies, which are unable to provide the required instant availability of services to customers, thereby creating scheduling conflicts and other challenges for the administrators, which in turn affects the quality of service provided to the end clients. The proposed project is presented within the broader context of digital transformation and web-based services.

Keywords: Car Rental System, Spring Boot, PostgreSQL, Fleet Management, Online Booking, REST API, Web Application, Vehicle Reservation, Digital Transformation, Transportation Technology;

I. INTRODUCTION

The transportation industry has experienced a major shift in its paradigm over the past decade or so. The advent of high-speed internet connectivity and the popularity of smartphones among the masses have led to a situation wherein people expect a seamless digital experience for all the services that they avail, including car rentals.

The core of a car rental business involves the management of three different but interrelated entities, namely, the cars, the customers, and the bookings. In a manual system, each of the above entities is maintained separately in the form of paper records or, at best, separate excel sheets. Such a system offers a fertile ground for conflicts, inconsistencies, and slower query resolution, especially as the number of cars and customers increases over a wide geographical area.

The challenges identified are being met directly by the Car Rental Hub System, which offers a single digital platform for the customers as well as the administrators. The customers can directly access a clean interface via a web browser, allowing them to search for available vehicles, compare specifications, choose dates for the rental, and make payments without the intervention of a human being. The administrators can access a control interface, which offers a wide range of tools to manage the fleet of vehicles, the current bookings, the users, the payments, and the analytical reports.

The paper is organized in a way that will give a holistic understanding of the system. Section 2 provides a review of the relevant literature and points out the drawbacks of the existing system. Section 3 states the problem statement and motivation. Section 4 explains the system architecture. Section 5 lists the functional and non-functional requirements. Section 6 explains the technology stack and its rationale. Section 7 explains the system's data flow and module details. Section 8 explains the methodology and phases of the project. Section 9 explains the outcomes and future scope. Section 10 concludes the paper.

II. LITERATURE SURVEY

A comprehensive review of the literature and existing systems will highlight some significant trends and gaps that necessitate the need for the Car Rental Hub. In the academic literature, the earliest studies of Vehicle Management Systems were largely focused on inventory control and logistics management. In other words, managing vehicles was seen as a special case of general Supply Chain Management. Such systems were adequate in the context of Enterprise Resource Planning but were not designed to support the concept of self-service. With the advent of web technologies in the early 2000s, basic web-based reservation systems were developed that allowed customers to make reservation requests using web-based forms. However, in most cases, a member of staff still had to manually review the reservation request and make further telephone calls to action.

Contemporary commercial systems, such as those represented within the global car rental industry, have evolved towards more automated pipelines. Online booking, multiple payment gateway support, and real-time availability are all features of contemporary commercial systems. However, research on such commercial systems has highlighted some of the key areas where such commercial systems are lacking. Real-time fleet tracking, where the customer can view the exact number of vehicles currently available, is not consistently implemented, which can create issues with the disparity between what is represented online and what is actually on the ground.

From an academic point of view, various researchers have proposed architectures for integrated transportation management systems. One common feature that has been identified in all such proposed architectures is the separation of concerns among the presentation layer, the business logic layer, and the data layer, which is exactly what Spring Boot is designed to do. Researchers in the domain of database management systems have already established that relational databases can handle the concurrent needs of a booking system, which requires atomic transactions to avoid double-booking.

The web application security literature points to the special susceptibility of booking systems to security breaches

that include unauthorized access to booking records, payment data interception, and session hijacking. The recommended solutions in the literature that have been incorporated into the Car Rental Hub system include stateless authentication using token-based authentication schemes, encrypted communication, and parameterized database queries to prevent injection attacks.

A survey of mobile-first web design literature points to the need for responsive designs that can scale to different screen sizes. Car rentals customers are often in transit and need to make changes to their reservation while away from their homes.

To briefly summarize, the literature and the assessment of existing systems have drawn a common conclusion that there is a need for a purpose-built integrated car rental management system that can bring together the front-end and back-end aspects of car rental management. This is precisely the purpose and intent behind the Car Rental Hub.

III. PROBLEM STATEMENT AND MOTIVATION

Despite the technology that is currently within reach, many small to medium-scale car rental enterprises continue to run on manual and semi-automated systems. Compounding problems plague such enterprises as they scale up. The fundamental issue can be broken down into five different dimensions:

- **Availability Conflicts:** Without a real-time system in place, there is every likelihood that one car can be allocated to multiple individuals, thereby creating conflicts that erode consumer trust and necessitate costly resolution efforts.
- **Administrative Inefficiency:** Employees waste valuable time on basic administrative activities such as answering queries on availability, generating invoices, and maintaining records that can be automated and divert valuable human resources away from activities that can create more value.
- **Data Fragmentation:** Customer data, vehicles, and booking data are not necessarily in the same location, and there are no automated systems in place to connect the data. This makes cross-referencing difficult.
- **Payment Handling Risks:** The process of manual payment collection, either in the form of cash or through bank transfer, is not transparent and makes it difficult to account for payments. The lack of automated billing makes the service less professional.
- **Customer Experience Deficit:** The current customer base has been socialized by other digital services in other domains, like food delivery, hotel booking, and taxi hailing. They have come to expect the same level of convenience in car rental services. A service that requires phone calls and business hours to make bookings or check availability is not aligned with the current customer experience.

The Car Rental Hub is driven by the desire to solve all of these aspects of the business in one cohesive software package that is both technologically viable and operationally and user-friendly.

IV. SYSTEM ARCHITECTURE

The architectural pattern of the Car Rental Hub is a three-tier system, which is the standard for creating a scalable and maintainable web application. The three layers of the three-tier system are the presentation layer, the application logic layer, and the data layer.

A. Presentation Layer

The presentation layer of the system is the part of the system with which the end user interacts directly. The presentation layer of the system is implemented using the latest version of the HyperText Markup Language, i.e., HTML5, for semantic purposes, the Cascading Style Sheets, i.e., CSS3, for presentation, and JavaScript for client-side scripting, which can run in any of the available browsers without the need for installing any plugins, ensuring the highest level of portability for the system. The presentation layer of the system can adapt smoothly to different screen sizes of desktop monitors, laptop screens, and mobile devices due to the flexible grid-based layout and the responsive design of the CSS3 specification. The interface components of the system include the vehicle search panel, the booking calendar, the customer registration panel, the customer login panel, the booking confirmation panel, the booking history panel for returning customers, and the control panel for the administrator.

B. Application Logic Layer

The application logic layer is implemented in Java using the Spring Boot framework. This was chosen for its convention-over-configuration approach, which greatly reduces the amount of code needed to configure a web service for deployment. The application provides a set of RESTful API endpoints that can be consumed by the front-end using asynchronous HTTP requests. This provides a clean separation of front-end and business logic. The main tasks of the application logic layer include handling booking requests and checking availability of the vehicles, enforcing business rules such as minimum period for renting and age limits, managing payment processing and bill generation, managing user authentication and session security, and managing data read and write operations using the database access layer. The Spring Boot application uses an embedded Apache Tomcat server to serve the web service. This greatly simplifies deployment as there is no need to configure a separate application server.

C. Data Layer

For the data layer, PostgreSQL relational database management system is used because of its reliability, ACID compliance, and strong support for concurrent transaction processing. All persistent application data is stored in this database, including user profiles, vehicle data, booking records, payment transactions, and logs. The database design is based on third normal form to prevent data redundancy and ensure referential integrity. Spring Data JPA is used to implement object-relational mapping, enabling the application logic layer to use intuitive Java objects and repository interfaces to access and manipulate data in the database instead of writing native SQL queries, although native queries can be used for performance optimization.

V. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

A. Functional Requirements

The system should meet the following functional requirements, which are categorized according to the roles of the users of the system:

1) Customer-Facing Functions:

- Registration of customer accounts with email verification and storage of passwords securely
- Login and authentication of customers, including session management
- Browse available vehicles with search functions by vehicle type, capacity, fuel type, & price range
- View the specifications of the available vehicles, including images, rental prices, and rental terms
- Choose the rental dates and submit a request for a booking
- Obtain real-time notifications of the availability of the booked vehicle
- Make a payment for the rental service using the integrated payment system of the system
- Obtain a confirmation of the booking request and a digital receipt from the system
- View the history of personal bookings, including the status of the current booking
- Submit ratings and write feedback after the completion of the rental service
- Modify or cancel the booking within the allowed policy period.

B. Administrator Functions:

- Admin login system with role-based access control
- Add/Edit/Delete vehicles in the fleet with full specification details
- View the availability status of each and every vehicle in real time
- View and accept/reject pending customer bookings
- View comprehensive details of all bookings with filtering and exporting options
- Modify the status of registered users and review user accounts
- Generate reports based on specific date ranges for financial and operational purposes
- View and respond to customer feedback submitted through the system

C. Non-Functional Requirements

- **Performance:** The most important pages must be loaded within three seconds on average under normal network conditions. The processing of booking transactions must be completed within five seconds.
- **Security:** Data transmission must be encrypted. Passwords for users must be hashed with a strong encryption algorithm. The system must be secure against major web application attacks such as SQL injection and cross-site scripting.
- **Scalability:** The system architecture must be horizontally scalable to meet the needs of increasing numbers of users without any architectural changes.

- **Reliability:** The system must be highly available during business hours with a process to recover from any interruption in service and data loss.
- **Usability:** The interface for customers must be easy to use for people without any technical knowledge. The completion of tasks must be possible in four steps or less for any major workflow from any entry point.
- **Compatibility:** The system must be compatible with all major browsers such as Chrome, Firefox, Edge, and Safari.

VI. TECHNOLOGY STACK

The selection of the technology stack for the Car Rental Hub was based on the maturity of the technologies, support for the technologies, characteristics of the technologies, and compatibility of the technologies with the expertise of the development team. The table below represents the technology stack:

Layer	Technology	Role in the System
Frontend	HTML5, CSS3, JavaScript	User interface rendering, client-side validation, and dynamic interactions
Backend	Java + Spring Boot	Business logic implementation, REST API endpoint definitions, and request processing
Database	PostgreSQL	Data storage with transactional support
ORM	Spring Data JPA	Object-relational mapping and repository abstractions
Server	Apache Tomcat (embedded)	Hosting environment for HTTP server running the Spring Boot application
Dev Tools	VS Code / Spring Tool Suite	Authoring, debugging, and management of code
OS Support	Windows / Linux	Development and deployment environments

Java is one of the most popular programming languages for backend programming for web and web-based applications. It provides the advantages of platform independence by using the Java Virtual Machine and has a vast collection of libraries. Spring Boot is based on the power of the Spring Framework in terms of its dependency injection and aspect-oriented programming features while providing much simpler configuration compared to the traditional Spring Framework.

The full power of SQL is combined with additional features like JSON columns, window functions, full-text search, and support for concurrent read-write operations. All these features will be useful in the future as the Car Rental Hub will need more sophisticated features in terms of analytics and report generation. In addition to that, the open-source nature of PostgreSQL also provides a cost-effective solution for small and medium-sized applications.

VII. SYSTEM MODULES AND DATA FLOW

The overall architecture of the Car Rental Hub has six main functional modules, each of which represents a well-defined

domain of the application's functionality. The modules are tightly coupled through well-defined service interfaces, making the application easy to maintain.

A. User Management Module

The User Management Module takes care of all the identity management-related activities within the application. When a new customer attempts to access the application, they are presented with the registration workflow, where they are asked to provide their details. The application also checks the customer's email by sending them a confirmation link. The password is not stored in plain text; instead, it goes through a cryptographic hashing function before being persisted. The application checks the credentials entered by the customer when they attempt to log in. If the credentials are correct, the application generates an authenticated session token, which must accompany all the requests made by the customer. The application has separate profiles for customer and admin roles, ensuring that the admin functionality is not exposed to the customer.

B. Vehicle Management Module

The vehicle management module is essentially a computerized representation of the car rental firm's fleet of vehicles. Each record of a vehicle in the system is characterized by its registration number, make, model, manufacturing year, type of fuel used, type of transmission used, capacity, daily rental charge, insurance details, and status of operation. The status of a vehicle is defined by its availability, booked status, under maintenance status, and retired status. This ensures that only available vehicles are displayed to the customer when he/she initiates a search query. The administrator can add a new vehicle to the system, modify the details of a vehicle as it goes for maintenance or upgrades, and retire a vehicle as its service life comes to an end.

C. Booking and Reservation Module

This is the most operationally critical module in the system. When a customer makes a booking request, the module checks the availability of the vehicle against the existing reservation calendar of the vehicle for the requested dates. This operation is performed as a transaction to the database to prevent race conditions in case there are simultaneous requests by different customers to book the same vehicle. If the vehicle is available, a provisional booking is made, valid for a short period until the customer completes the payment transaction. Once the payment transaction is successful, the provisional booking is converted into a confirmed booking, and the availability calendar of the vehicle is updated accordingly. The module also processes amendments or cancellations to existing bookings, applying the relevant rules in the process.

D. Payment and Billing Module

The payment and billing system is responsible for managing all financial transactions that are involved in renting a vehicle. Once a booking is confirmed by a customer, this system will determine how much the customer has to pay for renting a vehicle by using information like the price of a vehicle per day, the total days of renting a vehicle, and any extra services that a customer wants to add to their booking.

Once this information is obtained, a bill will be presented to the customer for approval before making a payment. All information is stored in a database and will be recorded with a timestamp and information regarding the payment made by a customer. The system will be able to generate a bill in a digital format that can be accessed by a customer through their booking history.

E. Administrative Dashboard Module

The administrative dashboard is used by the rental company staff to monitor all aspects of the system. This includes summary information, such as total active bookings, fleet availability percentage, revenue for the current period, and recent activity by customers. They can drill down on any booking to see all history, contact associated customers, or update status. There is also a section for monitoring customer feedback, where staff can keep tabs on how well their service is performing, including areas that may require a change in operations.

F. Reporting and Analytics Module

The reporting module converts the transactional information stored across the system into business intelligence. Standardized report templates include vehicle utilization rates, revenue over specific time periods, customer demographic information, most popular vehicle types, and lead times for bookings. Reports can be sorted based on date ranges, vehicle types, or customer groups. They can also be exported in standard formats. This module helps managers of rental companies make informed decisions about their fleets, pricing models, advertising campaigns, and workforce management.

VIII. DEVELOPMENT METHODOLOGY AND PROJECT PLANNING

The Car Rental Hub project is based on an iterative development methodology in phases. The iterative development methodology is based on the waterfall milestone discipline. The project is appropriate for an academic project setting where the requirements are generally known in advance, but some changes in the details of implementation may be necessary.

Phase	Name	Key Activities
Phase 1	Requirements & Design	Requirement gathering, conducting interviews with stakeholders, designing data models, specifying system architecture, and designing interface wireframes
Phase 2	Frontend Development	Creating HTML pages, applying CSS for styling and responsive design, adding JavaScript for validation and interactivity, and cross-browser compatibility checking
Phase 3	Backend & Database	Setting up a Spring Boot project, developing a REST API, creating a PostgreSQL schema, JPA mapping, service, and repository implementation

Phase 4	Integration & Testing	Integrating frontend and backend, unit testing, integration testing, user acceptance, performance, and security testing, and resolving bugs
Phase 5	Deployment & Review	Configuring servers, deploying, monitoring, completing documentation, and project review and future scope planning

The hardware requirements for development and deployment are deliberately low; they are an Intel i3 processor or better, at least 2 GB RAM (4 GB RAM recommended), and 256 GB storage. This low-level accessibility ensures that the system can be developed and tested on widely available computing equipment in institutions.

IX. EXPECTED OUTCOMES AND FUTURE ENHANCEMENTS

A. Expected Outcomes

After the successful development and deployment of the Car Rental Hub, it is expected to achieve the following results:

- **Gains in Efficiency:** The automation in booking, billing, and record maintenance is likely to reduce the time taken for these processes by a significant margin.
- **Elimination of Booking Conflicts:** The real-time availability engine, along with database operations, will ensure that the issue of double bookings is completely eradicated.
- **Improved Customer Satisfaction:** The availability of self-service for bookings at any time and instant confirmations with clear billing will ensure a level of service that meets modern standards for service through digital means.
- **Improved Data Integrity:** By centralizing all data within a normalized relational database, where integrity constraints are enforced, data inconsistencies will be minimized, and the system's data will be authoritative.
- **Revenue Visibility and Control:** The reporting and billing modules will offer management visibility into revenue streams, thus enabling more responsive pricing and promotional strategies.
- **Scalability Foundation:** By its layered design, this system is well-suited for scalability, whether it is in terms of additional vehicle types, geographical locations, or additional service channels such as a dedicated mobile application.

B. Future Enhancements

The current design provides a solid foundation for the system's operations, but several areas for enhancement are identified for consideration in the future:

- **Mobile Application Development:** A mobile application for both iOS and Android platforms could be developed to expand the system's customer base by allowing mobile-based interactions.
- **GPS Vehicle Telematics Integration:** Integration with GPS devices installed in vehicles could provide real-time

vehicle tracking, mileage tracking, and geofencing services.

- **AI-based Recommendation System:** The system could be integrated with machine learning algorithms using booking history data to provide recommendations to loyal customers.
- **Third Party Calendar and Travel Platform Integration:** Integration with third-party platforms could increase the system's exposure by allowing it to be integrated into the broader travel landscape.
- **Multi-Language and Multi-Currency Support:** The system could be designed to provide multi-language and multi-currency support to cater to different markets.
- **Loyalty and Reward System:** A loyalty program could be integrated into the system to provide incentives to loyal customers.

X. CONCLUSION

The Car Rental Hub is a purposeful and well-thought-out solution to address the issues that car rental companies face in their operations if they have not adopted a digital transformation strategy. By integrating vehicle management, customer booking and payment systems, and management control into a cohesive online solution, this system has removed all the inefficiencies and shortcomings that exist in traditional manual systems.

The technology stack that has been adopted for this system is a well-thought-out solution that has aligned itself with established technologies that are stable and have a good future prospect. With a Java Spring Boot backend and a PostgreSQL database, combined with a responsive HTML/CSS/JavaScript frontend, this system is a well-designed solution that has adopted a modular design that allows each of these domains to be developed independently of each other. Similarly, a phased development approach has been adopted that is a realistic approach to getting this system from concept to delivery.

In addition to its direct functional benefit, the Car Rental Hub also represents a model for the effective application of modern web technologies to transform service-based businesses. The design principles represented here, including the importance of real-time data consistency, human-centered interface design, transactional integrity, security, and analytics-based management, are relevant to the service industry as a whole. They provide a model that can be adapted to other service domains where manual management has persisted beyond its usefulness.

As the service industry in transportation continues to develop toward greater digital integration, the type of system represented here will not only be a competitive advantage but a required piece of infrastructure. The work represented in this paper has established a strong foundation for this development, providing multiple avenues for productive work in the future regarding intelligent service management in the transportation industry.

REFERENCES

- [1] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann Publishers (Elsevier). ISBN: 978-0123814791.

- [2] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). Database System Concepts (6th ed.). McGraw-Hill Education. ISBN: 978-0073523323.
- [3] MDN Web Docs — Mozilla Foundation. (2026). HTML, CSS, and JavaScript reference documentation. Retrieved from <https://developer.mozilla.org>.
- [4] VMware (Broadcom). (2026). Spring Boot Reference Documentation. Retrieved from <https://spring.io/projects/spring-boot>.

