

# Car rental system based on spring boot

Huaizhong Zhu, Shaolin Zheng, Yuguang Zhu, Yanni Zhao, Yurong Zhu\*  
School of Artificial Intelligence, Shanghai Normal University Tianhua College, Shanghai  
201815, China

## ABSTRACT

The Spring Boot based car rental system proposed in this paper is a B/S based system aimed at meeting the needs of the car rental industry. The development of a car rental system mainly uses Java technology, combined with the latest popular Spring Boot framework. The MySQL database and Eclipse development environment are used. The car rental system includes two roles: user and administrator. Its main functions include administrator: homepage, personal center, user management, vehicle brand management, vehicle information management, vehicle color management, rental order list management, return record management, administrator management, collection management, system management, user: homepage, personal center, vehicle information management, rental order list management, return record management; Functions such as vehicle information, system announcements, personal center, and backend management. Utilizing the Spring Boot framework to quickly build system infrastructure and integrating Spring Security MyBatis and other technologies enable user authentication and authorization, data access, and other functions. A user interface is built through a front-end framework. to achieve interaction between users and the system. The system supports multiple payment methods, including online payment and in store payment, to meet the different payment habits of users. At the same time, core business processes such as order management and vehicle scheduling have been implemented to ensure the integrity of system functions. The test results indicate that all functions of the system meet the design requirements and have high practical value and application prospects.

**Keywords:** Car rental, MySQL, Java, spring boot

## 1. INTRODUCTION

In recent years, with the continuous development of network technology, more and more people like to find various information they need on the network. Car rental system to users and administrators are of great help, through collaboration with database<sup>1</sup> management system software, to achieve a good operating platform between users and rental cars, based on it, a car rental system is designed.

## 2. SYSTEM DESIGN

The design of car rental system needs to consider user needs, system functions, data management, security and other aspects. First of all, the system should have user management functions, including user registration, login, personal information management, etc., while considering user rights management and security protection. Secondly, the vehicle management module is the core of the system, which needs to realize the entry, update, query and display of vehicle information, as well as the management of vehicle status, such as rentable status, maintenance status, etc., while considering the management of vehicle classification, brand, model and other information. Order management is one of the key functions of the system, which needs to realize the functions of user order, order cancellation, order inquiry, payment, etc., while considering the change of order status, reminder function and rent calculation. The payment module needs to integrate third-party payment interfaces to ensure safe and reliable payment. The rental calculation module needs to be flexibly calculated based on factors such as vehicle type and lease term<sup>2</sup>. The evaluation module allows users to evaluate and give feedback on the rental experience to help improve the quality of service. The design of the car rental system based on Java needs to fully consider user requirements, system functions, data management, security and other aspects to ensure that the system is stable and reliable, easy to operate, and has a good user experience<sup>3</sup>. The system design is shown in Figure 1.

\*1226350362@qq.com

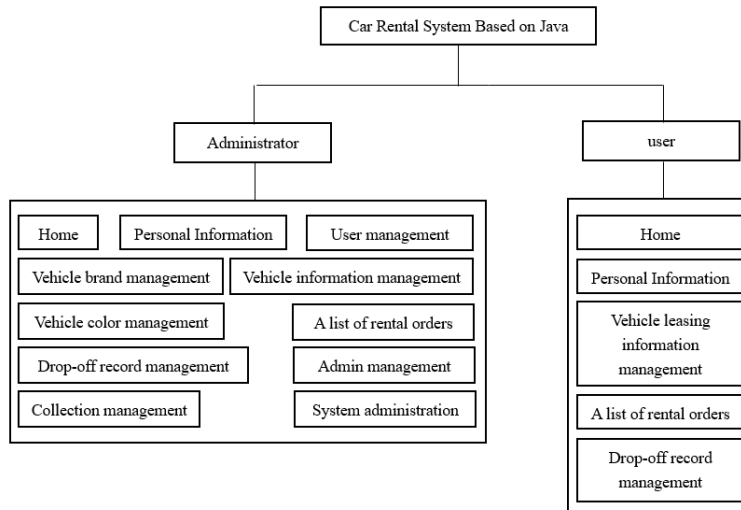


Figure 1. System design drawing.

### 3. SYSTEM IMPLEMENTATION

The car rental system takes advantage of SpringBoot's rapid development features to simplify the configuration process. The project adopts the modular SpringMVC architecture and implements the RESTfulAPI<sup>4</sup>, which makes the separation of front end and back end more obvious. At the database level, create database tables: vehicle information table, user table, rental order list, rental record table, administrator table. The user management module includes user registration, login verification, and permission control; Vehicle management module includes vehicle information input, inquiry and lease status update; The Orders module handles the booking, cancellation, and payment processes. The front end uses modern frameworks such as Angular or Vue<sup>5</sup> to provide a dynamically interactive interface. The whole system follows the concept of microservices and continuous integration<sup>6</sup>, which ensures the scalability and stability of the code.

#### 3.1 User function module

Users can enter the system to view the home page, personal center, vehicle information management, rental order list management, return records management and other operations. The user use case diagram is shown in Figure 2.

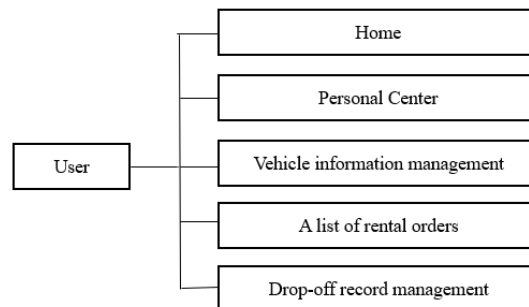


Figure 2. User use case diagram.

The user function module part code is as follows:

```
// User registration
public boolean register(String username, String password) {
if (users.containsKey(username)) {
System.out.println("The username already exists, please re-enter it!");
```

```

return false;}
User newUser = new User(username, password);
users.put(username, newUser);
System.out.println("Registration is successful!");
return true;
}
// User login
public boolean login(String username, String password) {
if (users.containsKey(username)) {
User user = users.get(username);
if (user.getPassword().equals(password)) {
System.out.println("Login successful!");
return true;
} else {System.out.println("Wrong password!");
}
} else {System.out.println("The username does not exist!");
}
return false;
}
// View all vehicle information
public void viewCars() {
if (cars.isEmpty()) {
System.out.println("There are currently no vehicles available.");
} else {
for (Car car: cars.values()) {
System.out.println("Vehicle ID:" + car.getCarId());
System.out.println("Models:" + car.getCarModel());
System.out.println("Daily rent:" + car.getCarPrice());
}
}
}
}
}

```

### 3.2 Administrator function module

After the administrator logs in, the main functional modules include home page, personal center, user management, vehicle<sup>7</sup> brand management, vehicle information management, vehicle color management, rental order list management, return record management, administrator management, collection management, system management and other functions. The administrator use case diagram is shown in Figure 3.

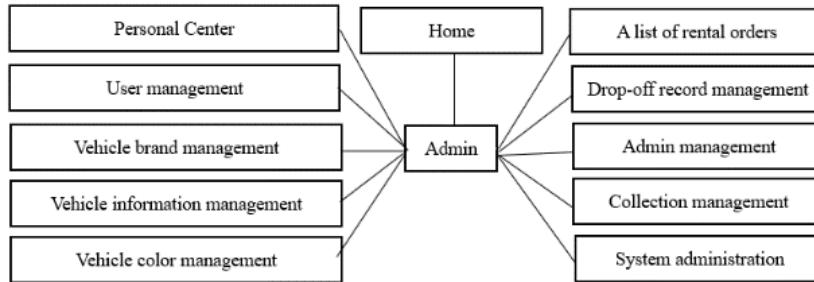


Figure 3. Administrator use case diagram.

The code of the administrator function module is as follows:

```

// Add car
public void addCar(String carId, String carModel, double carPrice) {
    Car newCar = new Car(carId, carModel, carPrice);
    cars.put(carId, newCar);
    System.out.println("The car was added successfully!");
}

// Manage user
public void manageUser(String username, String action) {
    if (action.equals("add")) {System.out.println("The user was added successfully!");
    } else if (action.equals("remove")) {
        System.out.println("The deletion of the user is successful!");
    } else {
        System.out.println("The operation is invalid!");
    }
}

// View Rental order
public void viewOrders() {for (Order order : orders.values()) {
    System.out.println("Order ID:" + order.getOrderid());
    System.out.println("Username:" + order.getUsername());
    System.out.println("Car ID:" + order.getCarId());
    System.out.println("Lease date:" + order.getRentalDate());
    System.out.println("Return Date:" + order.getReturnDate());
}
}
  
```

## 4. SYSTEM TEST

### 4.1 Administrator module test

Test point: Administrator module

Test objective: Verify that the input information is available.

Test environment: Windows10 and Google Chrome.

Enter data: name, description, picture, etc.

Steps:

- (a) The browser is opened and the administrator page is gone to.
- (b) The Add button on the left side of the page is clicked, the Add page appears, and the necessary information is entered to add.
- (c) Home page, personal center, user management, vehicle brand management, vehicle information management, vehicle color management, rental order list management, vehicle return record management, administrator management, collection management, system management and other functions are mainly tested.

#### **4.2 User module testing**

Test point: User module

Test goal: After entering the account password and verification code, the system will automatically verify whether it is correct; Whether it is possible to rent a car and whether the total amount is calculated correctly.

Environment used: Windows10 and Google Chrome.

Input information: User name and password.

Steps:

- (a) The browser is opened and the login page of the system is gone to.
- (b) After the login section of the page is entered, the user's name, password and verification code can be tested; The car rental interface is opened, the rental and other operations are tested, as well as whether the amount of information is correct.

#### **4.3 Car rental module testing**

Test point: Car rental module

Test objective: users can successfully rent vehicles<sup>8</sup>; The system correctly calculates and displays the lease fee; Order information is accurately recorded in the system and inventory is updated.

Environment used: Windows10 and Google Chrome.

Steps:

- (a) The user browses the list of vehicles and selects a vehicle.
- (b) The lease date is selected; The rental fee is displayed.
- (c) The user confirms the lease information and submits the order.
- (d) Whether the information displayed is correct is checked.

## **5. CONCLUSION**

In this paper, the design and implementation of the car rental system based on Java is discussed in detail. Through the requirement analysis, system design, function realization and test verification of the system, a stable, reliable, easy to expand and maintain car rental platform is built. After a series of tests, the functions of each module have been effectively verified; The interaction and data transfer between the modules of the system are normal; The system test verifies the overall function and performance of the system. The comprehensive test results show that the system is stable and reliable in all aspects, the function runs normally, and meets the needs of users. Compared with other car rental systems, this system makes full use of the object-oriented, cross-platform and security features of Java language providing an efficient solution for car rental business. Here, this paper is expected to provide some reference value for researchers and developers in related fields. At the same time, the car rental system still has a lot of room for improvement.

## ACKNOWLEDGMENTS

This research was supported by Shanghai Jiading Vocational Education Group's 2023-2025 Vocational Education Reform and Construction Project (JDZJJT-2023-007).

## REFERENCES

- [1] Le Gouil S., Bachy, E., et al., "First results of DLBCL patients treated with car-t cells and enrolled in DESCAR-t registry, a French real-life database for car-t cells in hematologic malignancies," *Hematological Oncology*, 39(S2), (2021).
- [2] Wang, Q. and Wang, J., "Sharing cars will influence factors and development mode study," *Journal of Inner Mongolia Highway and Transport* 3, 58-62 (2022).
- [3] Nie, H., Liu, Y. and Liu, Z., "Design and implementation of accompanying broadcast car," *China Cable Television* 9, 30-33 (2023).
- [4] Luo, X., "Analysis on the construction of safe and efficient intelligent cloud platform for learning and education," *China Media Technology* 12, 127-130 (2022).
- [5] Yu, T., Yuan, Y., Deng, W., et al., "Design of front-end architecture based on revenue settlement management platform," *Electronic Technology* 52(12), 36-38 (2019).
- [6] Sun, Y., "Research on the application of AI cloud platform in smart home," *Software* 45(01), 41-43 (2024). (in Chinese)
- [7] Śmieszek, M., Mateichyk V., Tsiumann M., et al., "Information system for remote monitoring the vehicle operational efficiency," *IOP Conference Series: Materials Science and Engineering*, 1199(1), (2021).
- [8] Ankur, S., Vikas, G. and Vikram, S., "Does consideration for future consequences matter in consumer decision to rent electric vehicles?" *Energy Policy*, 181, (2023).