

## **Project 3: Car Rental System**

This project is designed to test your php (functions, classes, user authentication, database manipulation, and sessions) and my ajax skills. The project is a simplified car rental system.

### **• The Data**

The E-R (Entity-Relationship) diagram of the system is shown in Figure 1. The details of each entity (table) are as follows:

#### **1. CarSpecs**

The CarSpecs entity is the specifications of a car (e.g. the car make, size, model, etc.). Many cars have the same specifications.

The attributes of CarSpecs are:

- **ID:** A unique value for each specs (primary key).
- **Make:** The car make (e.g. "Ford", "Volkswagen").
- **YearMade:** The year the car was made (e.g. 2002, 2003).
- **Model:** The model of the car (e.g. "Passat", "Class-A").
- **Size:** The size of the car (an enum type: the possible values are "Small", "Medium", and "Large").

#### **2. Car**

The Car entity is the car a customer can rent.

The attributes of a car are:

- **ID:** A unique value for each car. (it is specified as an auto-increment integer in the database).
- **Picture:** The picture of the car.
- **Picture\_type:** The type of the picture (e.g "image/jpeg").
- **Color:** The color of the car (e.g. "white", "black").
- **CarSpecsID:** The ID of the car specifications. This is a foreign key that links to a row in the CarSpecs table.
- **status:** The status of the car. When the value is "1", it means the car is available (customers can rent it). "2" means it is currently rented (it can't be rented and it won't show in search results).

#### **3. Rental**

This entity connects the customer with the car tables (meaning it makes it possible for a customer to rent many cars, and a car to be rented many times by a customer).

The attributes of a rental is:

- **ID:** A unique value for each rental (it is specified as an auto-increment integer in the database).
- **rentDate:** The date when the car got rented.
- **returnDate:** The date when the rental car got returned.

- **status:** The status of the rental. When the value is “1”, it means the car is currently rented. “2” means it has been returned.
- **CustomerID:** The ID of the customer. This is a foreign key that links to a row in the Customer table.
- **carID:** The ID of the rental car. This is a foreign key that links to a row in the Car table.

#### 4. Customer

The Customer entity is the customer that can rent cars.

The attributes of Customer are:

- **ID:** A unique ID for each customer.
- **Name:** The name of the customer.
- **Password:** The encoded password of the customer. (It is just the last name of the customer encoded using the md5 function)
- **Phone:** The phone of the customer.
- **Address:** The address of the customer.

#### • The System

**1. Login:** The customer logs into the system so that she can access the system. Figure 2 shows the login process. Note that the password of the customers are stored in encoded form. Hence, when you authenticate the user, you need to encode the password the user types in and match it with the password in the system. Just use the md5 function (it is a php built-in function).

**2. The System:** The system consists of three parts: find car , rented cars, and rental history. Below are the details.

**2. 1. Find Car:** This part allows the user to search for the cars she wants to rent (Figures 3 and 4). In the search box, the user can type in anything about the car they are looking for (e.g. the car model, make, color, size, and year). As a result, the system finds the cars with specifications that match any of the words the user typed in. For instance, Figure 3 shows all the “Passat” or the “white” cars.

Note that the cars that can be searched are the cars with status “1”. The cars with status “2” are currently rented, and shouldn’t show up in search results.

When the user clicks “Rent Car”, the car will be rented. This means the system creates a rental record. The rent date is simply the system current date, and the status of the rental is “1”. Moreover, the system will also update the status of this car to be “2” (meaning it’s not available now). Further, the system refreshes the rented cars (the list of the cars that the user has rented).

**2. 2. Rented Car:** This part shows the cars the user has rented (Figures 6 and 7). When the user hovers a car, the “Return Car” button shows up. When the user clicks it, the car gets “returned”. The return date will be the current date. As a result of clicking the “return car” button, the car should no longer show up in the rented cars list. Instead, it should show up in the rental history. Furthermore, the car should be available for rental (meaning the status of the car should be “1” now)

**2.3. Rental History:** This part shows the cars that have been returned (Figure 8). The list gets refreshed whenever the user returns a car (presses “Return Car”).

- **CSS:**

The project folder has premade CSS. However, ***It is okay to use your own CSS as long as the system functions as described above.***

- **Templates:**

You can find the HTML templates that you might want to use to generate dynamic HTML for the rented cars, rental history, etc.

<https://umkc.box.com/s/vli7zpphagyaou0jy8i5qir9e42mo29y>

- **Database:**

You can find the database here (it contains statements that create and populate the database of the car rental system):

<https://umkc.box.com/s/69xlkixsxttrdk76yjex20rlszqktoob7>

- **Project:**

You can find the project here (HTML, CSS, JS, images):

<https://umkc.box.com/s/2mwycru8ruvmykvpe9xkee094kakh56c>

- **Requirements:**

**1. (Weight: 20%) Log in/out:** Write php and ajax code that allows the customer to log in to the system. The login process is described in the system section above. The authentication should be secure (use the md5 function that is built in php). Further, start a new session when the user logs in. When the user logs out, the session should be destroyed, and the system shouldn't be accessible.

**2. (Weight: 35%) Find Cars:** Write php and ajax code that allows the customer to find cars. The customer can type anything about the car (e.g. color, year, model) and the system will look for available cars that match any of the words that the user types. Then system then displays the cars as shown in figure 4. When the user hovers a search item, the rent car button shows up. When the user clicks it, the car gets rented (a message box shows up telling the user that the car has been rented successfully). As a result, the search results and rented cars are refreshed (through ajax requests).

**Note: Message boxes can be defined with a Javascript-based alert function.**

**3. (Weight: 25%) Rented Cars:** Write php and ajax code that shows a list of the currently rented cars by the customer. When the user hovers an item, the return car button shows up. When the user clicks it, the car gets returned. As a result, the rented cars and rental history are refreshed (through ajax requests).

**4. (Weight: 20%) Rental History:** Write php and ajax code that shows a list of the cars that were returned by the customer.

- **Submission**

Submit the entire Netbeans project (including the php, CSS, JS, images, etc.). Don't submit the database.

- E-R Diagram

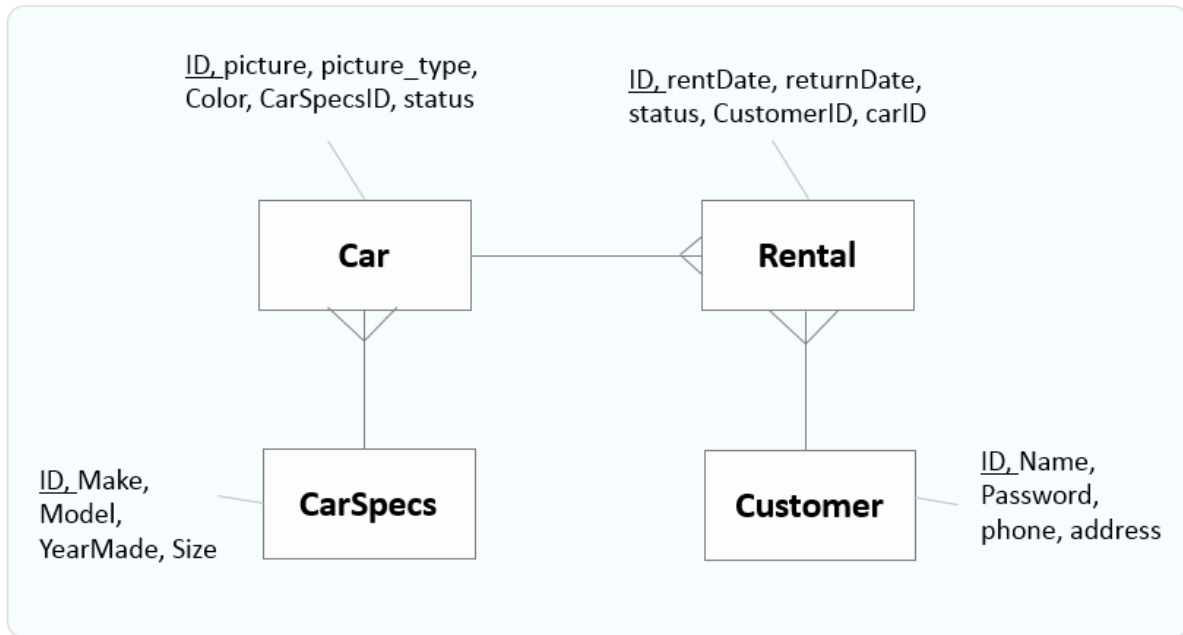


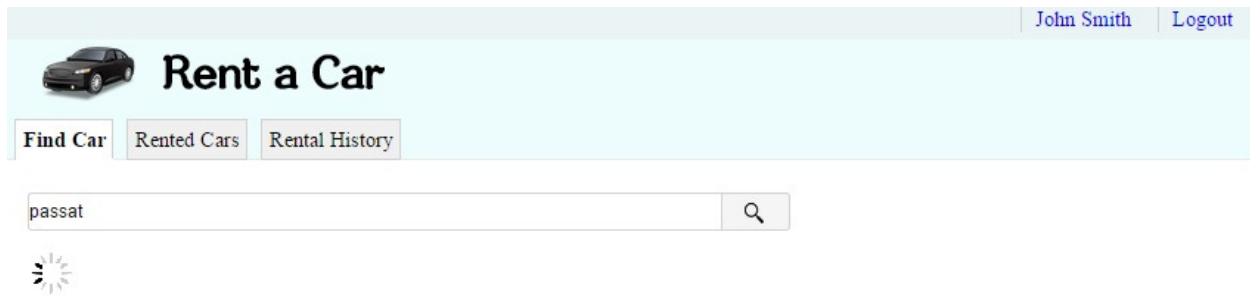
Figure 1: The E-R diagram of the car rental system.

- **Figures**

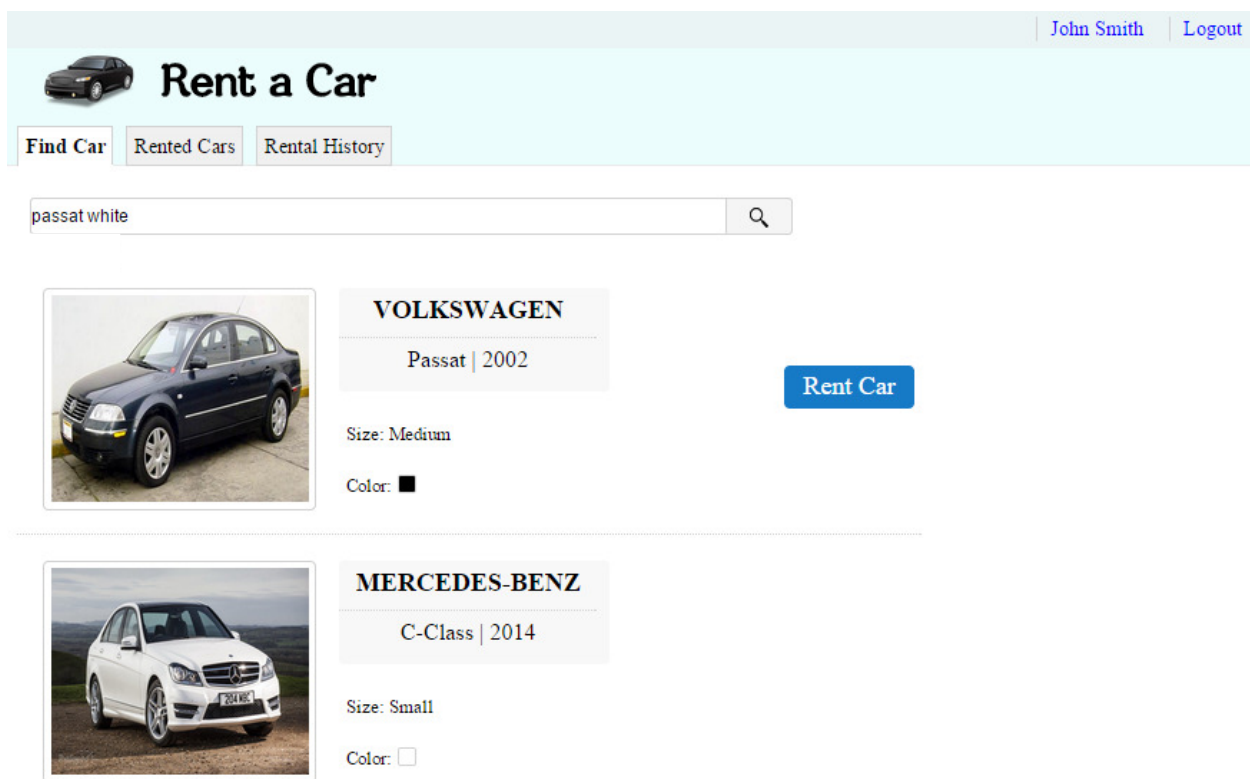
The figure consists of four numbered screenshots illustrating the login process:

- 1**: A login form with fields for "Username:" and "Password:", and a blue "Login" button.
- 2**: The same login form, but the "Username:" field contains "j.smith" and the "Password:" field contains two asterisks. A loading spinner is visible below the password field, and the "Login" button is still present.
- 3**: The login form is shown with an error message "Invalid username or password" displayed below the password field. The "Login" button is still present.
- 4**: A screenshot of the "Rent a Car" system page after successful login. The page has a light blue header with a car icon and the text "Rent a Car". In the top right corner, the user's name "John Smith" and a "Logout" link are displayed. Below the header, there are three tabs: "Find Car", "Rented Cars", and "Rental History". The "Find Car" tab is active, showing a search input field with the placeholder text "Type car make, model, year, color, etc." and a search button.

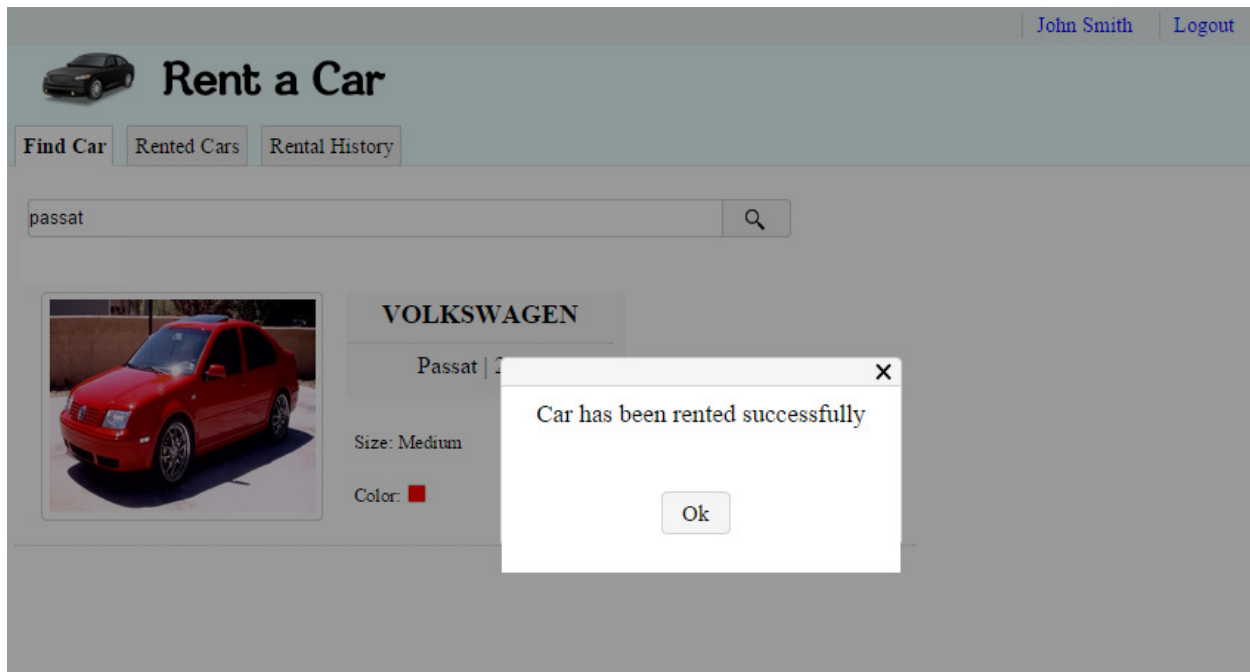
**Figure 2:** The login process: (1) Before the user tries to log in. (2) When the user presses “enter” in the password field or when she clicks the login button. (3) An error message is displayed if the user provides the wrong username or password. (4) If the login succeeds, the system takes the user to the car rental system page. Notice that the customer’s name is shown at the top left corner.



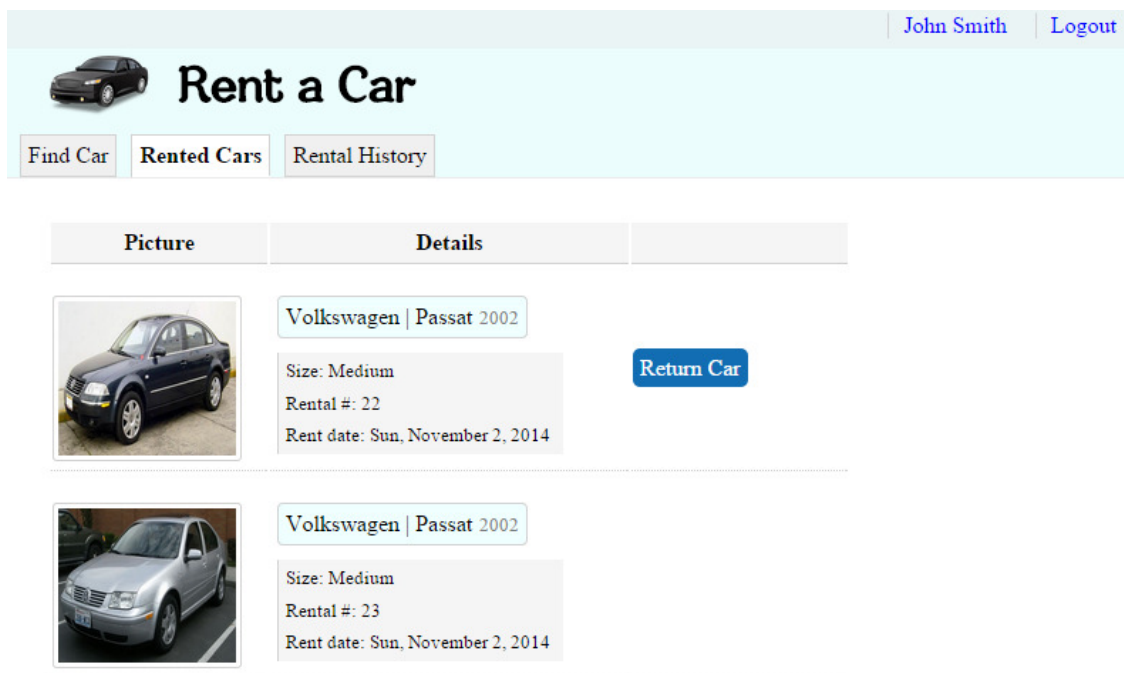
**Figure 3:** When the user types in something about the car, and presses enter or clicks the magnifying glass, the system looks for a car that matches what the user typed in, but meanwhile it displays the loading icon.



**Figure 4:** The car rental system (the search tab): The user can type in anything about the car they are looking for (e.g. the car model, make, color, size, year). The system will find the cars with specifications that match any of the words the user typed in. For instance, the system is showing all the Passat or the white cars. When the user hovers a search result item, the “Rent Car” button shows up.

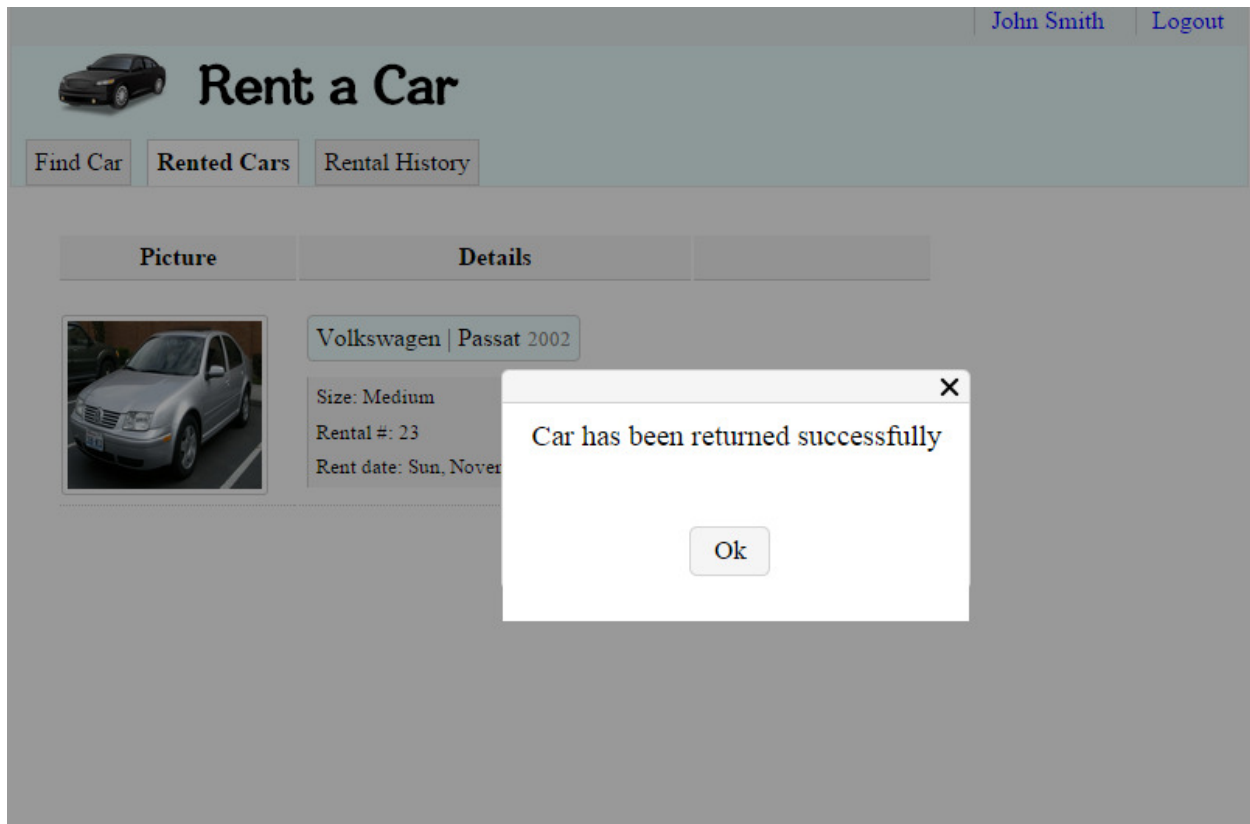


**Figure 5:** When the user clicks a rent car button, the system makes a record in the rental table for the car and the customer. The system displays a message that tells the user whether the rent was completed successfully or not.






**Figure 6:** A list of the rented cars. This list gets refreshed whenever the user rents a new car. It also gets refreshed when the user returns a car. This happens when the user clicks "Return Car". The "Return Car" button shows up when the user hovers a rented a car.





**Figure 7:** When the user clicks a “Return Car” button, the system updates the status of the rental to “2” (meaning the rental has been returned) and the status of the car to be “1” (meaning the car is now available). The list of rented cars and rental history will be updated as a result of returning a car.

Picture	Details
	<div>Volkswagen   Passat 2002</div> <div>Size: Medium Rental #: 1 Return date: Sat, November 1, 2014</div>
	<div>Kia   Cadenza 2015</div> <div>Size: Large Rental #: 10 Return date: Sun, November 2, 2014</div>
	<div>Volkswagen   Passat 2002</div> <div>Size: Medium Rental #: 11 Return date: Sun, November 2, 2014</div>

**Figure 8:** The list of returned cars. It gets refreshed whenever a car is returned.