

Getting Ready: The Car Rental System

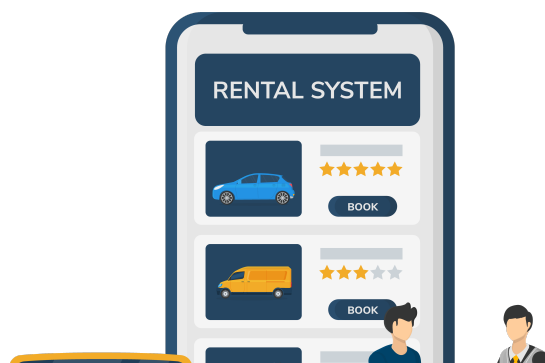
Understand the car rental system problem and learn the questions to further simplify this problem.

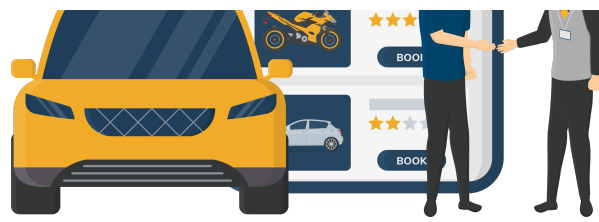
We'll cover the following



- Expectations from the interviewee
 - Vehicle types
 - Search interface
 - Services
 - Reservation cancelation
 - Payment flexibility
- Design approach
- Design pattern

A **car rental system** is an application that manages the renting of automobiles for a short period of time—a few hours to a few weeks. A car rental system has a number of offices in different locations within each city. It allows the users to reserve and return a vehicle from different locations and is primarily located near airports or city areas. The car rental system keeps a list of all its clients in a database. The database includes the name, address, and contact number of each new customer. A member can reserve a car for a certain number of days, hire a car, or return the car that was rented. A member makes a reservation by supplying the pickup and drop-off locations, the kind of vehicle, and the day and time of the reservation.





Application of the car rental system

Expectations from the interviewee

The car rental system consists of multiple components. Each component has its own functionality and constraints. This section provides an overview of some of the main expectations that the interviewer will want to hear you discuss in more detail during the interview.

Vehicle types

An interviewer would expect you to discuss the different vehicle types, and ask the following questions:

- What types of vehicles will that system support?
- How can we identify the specific vehicle?

Search interface

Members will use the application and add location and the reservation date. They will receive several options to select the vehicle. Therefore, an interviewer would expect you to ask questions listed below:

- Is it possible to search a vehicle using its name or type?
- Can we search for a vehicle by its model number?

Services

An interviewer would also expect you to discuss the services of the car rental system and may ask the following questions:

- Does a car rental system assign a driver to its customer?
- Does a car rental system provide roadside assistance to its customer?

Reservation cancelation

There will be many duplicate instances in our system. The interviewer expects you to ask questions listed below:

- Can the member be able to cancel a reservation?
- Which member is allowed to request a vehicle reservation cancelation and when?

Payment flexibility

One of the car rental system's most significant attributes is its customer payment structure. The payment depends on the vehicle type and time stamp. Therefore, an interviewer would expect you to ask questions listed below:

- How can customers pay at different branch locations and by different methods (cash, credit, or cheque)?
- If there are multiple branches of the car rental system, how will the system keep track of the customer having already paid at a particular branch?

Design approach

We'll design this car rental system using the bottom-up design approach. For this purpose, we will follow the steps below:

- Identify and design the smallest components, such as a vehicle, vehicle reservation, vehicle log, etc.
- Use these small components to design bigger components, such as building a car rental system that can be composed of multiple vehicles.
- Repeat the steps above until we design the whole system.

Design pattern

It is always a good practice to discuss the design patterns that a car rental system falls under, during the interview. Stating the design patterns will give the

interviewer a positive impression and shows that the interviewee is well-versed in the advanced concepts of object-oriented design.

The following design pattern can be used to design the car rental system:

- Decorator design pattern

Let's explore the requirements of the car rental system in the next lesson.

[← Back](#)

Code for the Movie Tic...

[Next →](#)

Requirements for the ...

☐

Mark as
Completed

