# 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](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Expectations-from-the-interviewee)
  + [Vehicle types](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Vehicle-types)
  + [Search interface](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Search-interface)
  + [Services](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Services)
  + [Reservation cancelation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Reservation-cancelation)
  + [Payment flexibility](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Payment-flexibility)
* [Design approach](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Design-approach)
* [Design pattern](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#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

**Requirements for the Car Rental System**

Let's have a look at the requirements for our car rental system.

**We'll cover the following**

* [Requirement collection](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Requirement-collection)

In this lesson, we’ll determine the requirements of our car rental system. This is a very crucial step as requirements define the scope of a problem, so getting them right from the interviewer and understanding them well will make the design of the rest of the system smooth and easy.

We’ll use the notational convention to identify each requirement with a unique label "Rn", where "R" is short for Requirement and "n" is a natural number.

**Requirement collection**

The set of requirements for the car rental system are listed below:

**R1:** There can be two types of users in the car rental system, i.e., customers and receptionists.

**R2:** The system should handle multiple types of vehicles. Initially, the system should cater to the following vehicles: cars, trucks, vans, and motorcycles.

**R3:** There can be multiple subtypes for vehicles. The car type can be economy, luxury, standard, and compact. The van type can be passenger or cargo. Moreover, the motorcycle type can be cruiser, touring, or sports.The truck type can be light, medium, or high-duty.

**R4:** The system should be able to keep a record of who reserved a particular vehicle and on which date the vehicle was issued.

**R5:** The system should be able to find out how many vehicles have been rented out by the specific customer.

**R6:** The customers should be able to cancel their reservations.

**R7:** To keep track of all events related to the vehicle, the system should maintain a vehicle log.

**R8:** The system should allow the users to add equipment to the reservations like a ski rack, child seat, and navigation.

**R9:** The system should allow the users to add services to the reservations like a driver, Wi-Fi, and roadside assistance.

**R10:** The system should send a notification to the customer and generate a fine if the vehicle is not returned within the due date.

**R11:** The system should allow the user to search the vehicles by type or model.

**R12:** A system should be able to manage the multiple branches of the car rental system.

**R13:** Every branch of the car rental system should have parking stalls to park the vehicles.

We’ve identified our requirements for the problem. In the next lesson, we’ll define different use cases of our car rental system.

# Use Case Diagram for the Car Rental System

Learn how to define use cases and create the corresponding use case diagram for the car rental system.

**We'll cover the following**

* [System](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#System)
* [Actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Actors)
  + [Primary actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Primary-actors)
  + [Secondary actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Secondary-actors)
* [Use cases](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Use-cases)
  + [Member](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Member)
  + [Receptionist](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Receptionist)
  + [System](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#System)
* [Relationships](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Relationships)
  + [Generalization](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Generalization)
  + [Associations](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Associations)
  + [Include](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Include)
  + [Extend](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Extend)
* [Use case diagram](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Use-case-diagram)

Let’s build the use case diagram for the car rental system and understand the relationship between its different components.

First, we’ll define the different elements of our system, followed by the complete use case diagram of the system.

## System

Our system is “rent a car.”

## Actors

Now, we’ll define the main actors of our car rental system.

### Primary actors

* **Member:**This is the primary actor of the system who can reserve a vehicle, make payment, and change or cancel the reservations of the vehicle.

### Secondary actors

* **Receptionist:**This actor acts as the system’s admin and can perform any task a “Member” can perform. This can also add, remove, or modify the vehicle or its reservations and update logs.
* **Worker:**This actor can update logs, return vehicles, and pay bills.
* **System:**This can send notifications related to reservation to members.

## Use cases

This section will define the use cases for "Rent a car." We have listed the use cases according to their respective interactions with a particular actor.

**Note:**You’ll see some use cases occurring multiple times because they are shared among different actors in the system.

### Member

* **Create a new account:**To create a new account on the system
* **Update/cancel account:**To update account information or to cancel an account
* **Login/Logout:**To log in and out of the car rental system
* **Search vehicle inventory:**To search for vehicles from the inventory
* **Make a reservation:** To make a reservation for a vehicle
* **Cancel reservation:**To cancel the reservation of the vehicle
* **Update reservation:**To update the reservation information of the vehicle
* **Return vehicle:**To return the vehicle to the car rental facility
* **Pickup vehicle:** To pick up the vehicle from the car rental facility
* **Pay bill:**To pay vehicle rent

### Receptionist

* **Create a new account:**To create a new account on the system
* **Update/cancel account:**To update account information or to cancel an account
* **Login/Logout:**To log in and out of the car rental system
* **Search vehicle inventory:**To search for vehicles from the inventory
* **Make a reservation:** To make a reservation for a vehicle
* **Cancel reservation:**To cancel the reservation of the vehicle
* **Update reservation:**To update the reservation information of the vehicle
* **Add vehicle:**To add a new vehicle to the car rental system
* **Remove vehicle:**To remove a vehicle from the car rental system
* **Modify vehicle:**To modify a vehicle status or information from the car rental system
* **Update log:**To update the vehicle log

### System

* **Send overdue notification:** To send a notification if the date and time for vehicle return are passed
* **Send reservation notification:** To send a notification of the reservation made
* **Send reservation canceled notification:** To send a notification of any canceled reservation

## Relationships

We describe the relationships between and among actors and their use cases in this section.

### Generalization

We’ll use the generalization relationship if we want to add, remove, or modify a vehicle. We also need to specify the vehicle type we want to add, remove or modify.

* “Add vehicle” has a generalization relationship with the “Add car,” “Add truck,” “Add van,” and “Add motorbike” use cases.
* “Remove vehicle” has a generalization relationship with the “Remove car,” “Remove truck,” “Remove van,” and “Remove motorbike” use cases.
* “Modify vehicle” has a generalization relationship with the “Modify car,” “Modify truck,” “Modify van,” and “Modify motorbike” use cases.

### Associations

The table below shows the association relationship between actors and their use cases.

|  |  |  |
| --- | --- | --- |
| **Member** | **Receptionist** | **System** |
| Create a new account | Create a new account | Reservation canceled notification |
| Update/cancel account | Update/cancel account | Send reservation notification |
| Login/Logout | Login/Logout | Overdue notification |
| Search vehicle inventory | Search vehicle inventory |  |
| Make a reservation | Make a reservation |
| Cancel reservation | Cancel reservation |
| Update reservation | Update reservation |
| Pickup vehicle | Add vehicle |
| Return vehicle | Remove vehicle |
| Pay bill | Modify vehicle |
|  | Update log |

### Include

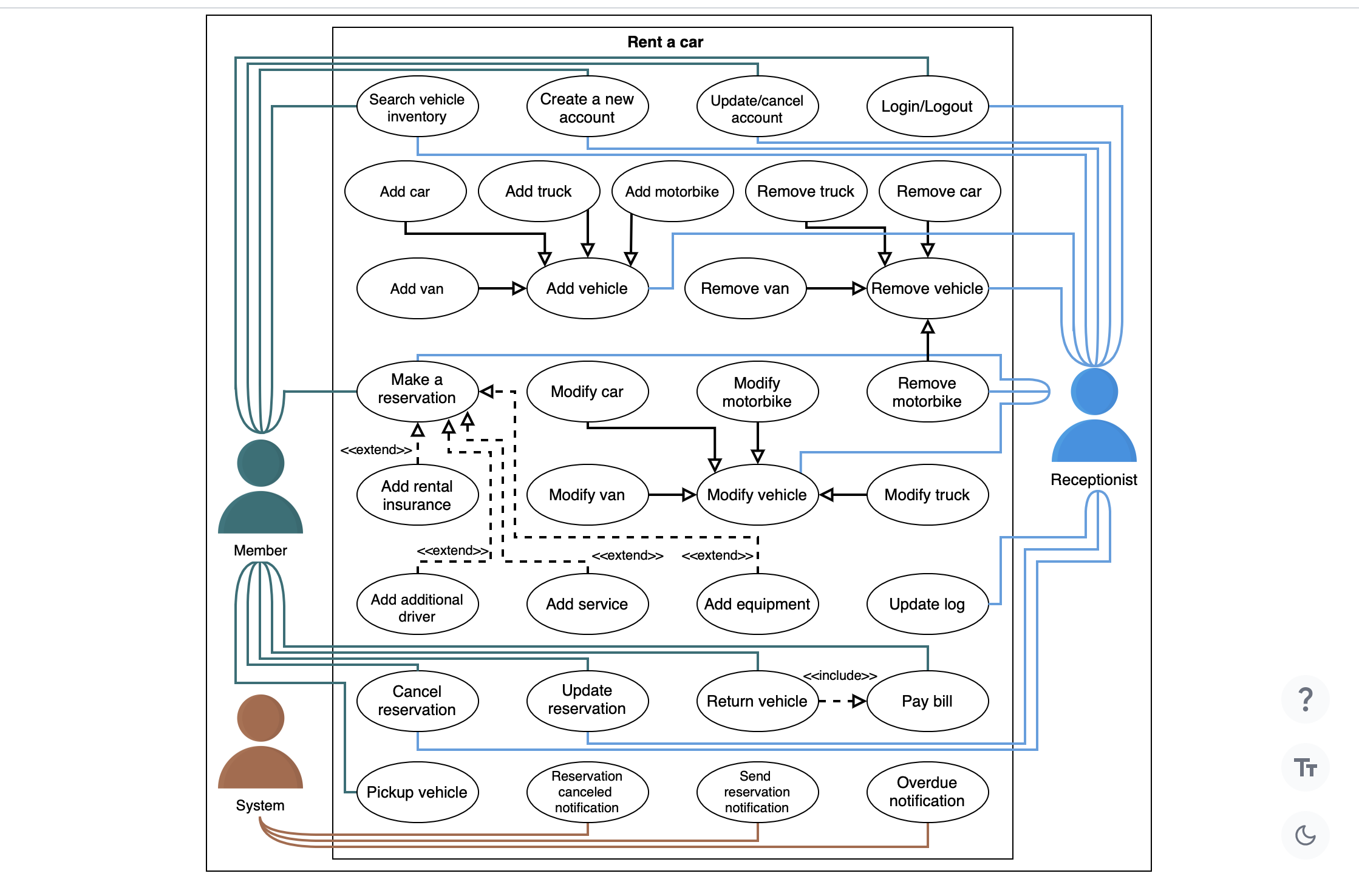
When a car is returned, the bill is paid. Therefore, “Return vehicle” has an include relationship with “Pay bill.”

### Extend

Whenever a reservation is made, we need to add its details like rental insurance, additional driver, service, and equipment. Therefore, the “Add rental insurance,” “Add additional driver,” “Add service,” and “Add equipment” use cases have an extend relationship with “Make a reservation.”

## Use case diagram

Here’s the use case diagram for the car rental system:



# Class Diagram for the Car Rental System

Learn to create a class diagram for the car rental system problem using the bottom-up approach.

**We'll cover the following**

* [Components of a car rental system](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Components-of-a-car-rental-system)
  + [Address and person](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Address-and-person)
  + [Account](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Account)
  + [Driver](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Driver)
  + [Vehicle](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Vehicle)
  + [Equipment](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Equipment)
  + [Service](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Service)
  + [Notification](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Notification)
  + [Parking stall](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Parking-stall)
  + [Vehicle log](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Vehicle-log)
  + [Vehicle reservation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Vehicle-reservation)
  + [Payment](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Payment)
  + [Fine](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Fine)
  + [Search interface and vehicle inventory class](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Search-interface-and-vehicle-inventory-class)
  + [Car rental system and branch](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Car-rental-system-and-branch)
  + [Enumerations](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Enumerations)
* [Relationship between the classes](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Relationship-between-the-classes)
  + [Association](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Association)
    - [One-way association](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#One-way-association)
    - [Two-way association](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Two-way-association)
  + [Composition](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Composition)
  + [Aggregation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Aggregation)
  + [Inheritance](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Inheritance)
* [Class diagram of the car rental system](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Class-diagram-of-the-car-rental-system)
* [Design pattern](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Design-pattern)
* [Additional requirements](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Additional-requirements)

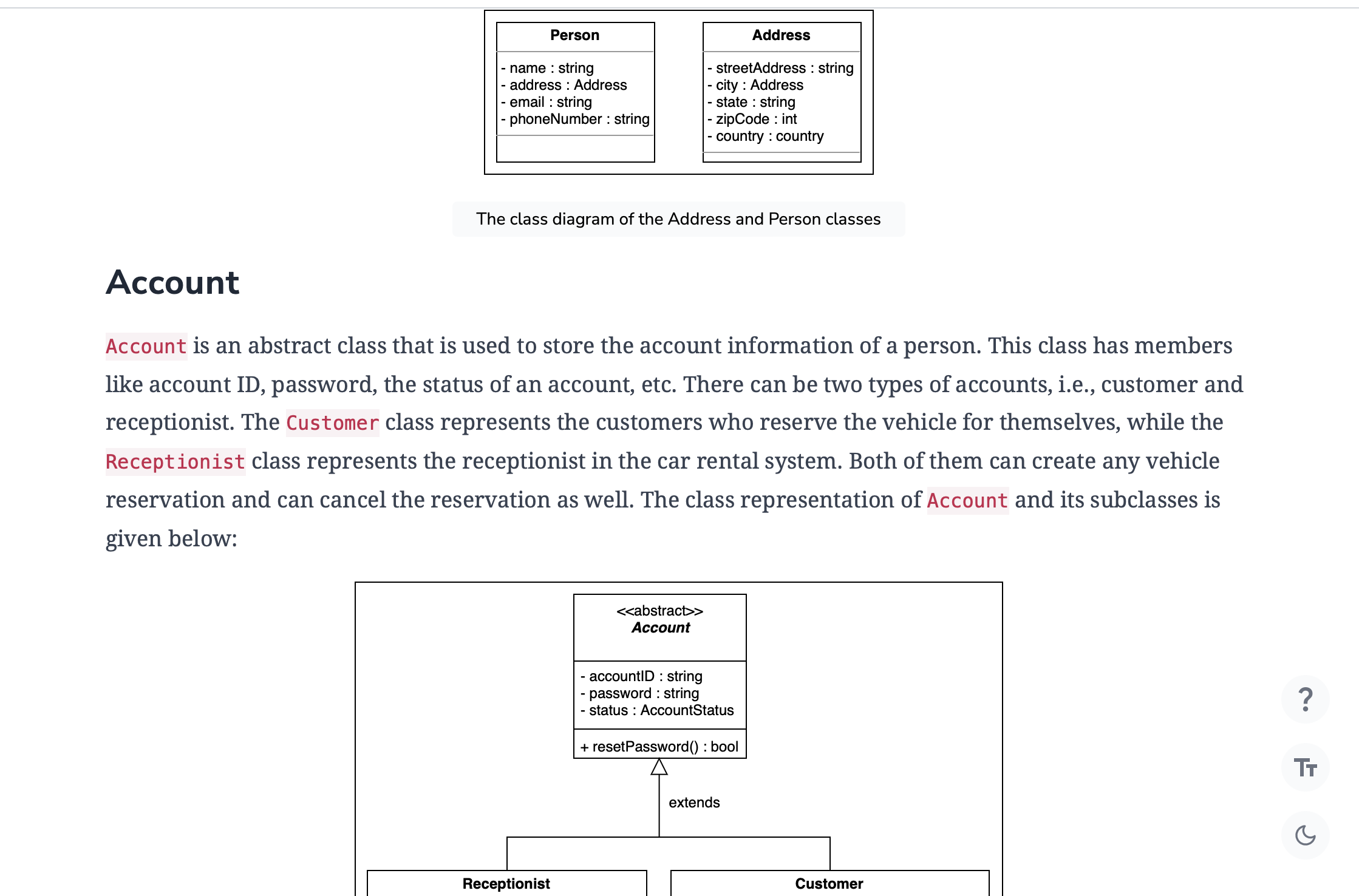
Now, we’ll create the class diagram for the car rental system on the basis of the given requirements. In the class diagram, we will first identify classes (concrete, abstract, or associated) and interfaces for the system. Then, we will determine the relationship between them, according to the requirements in the previous lesson.

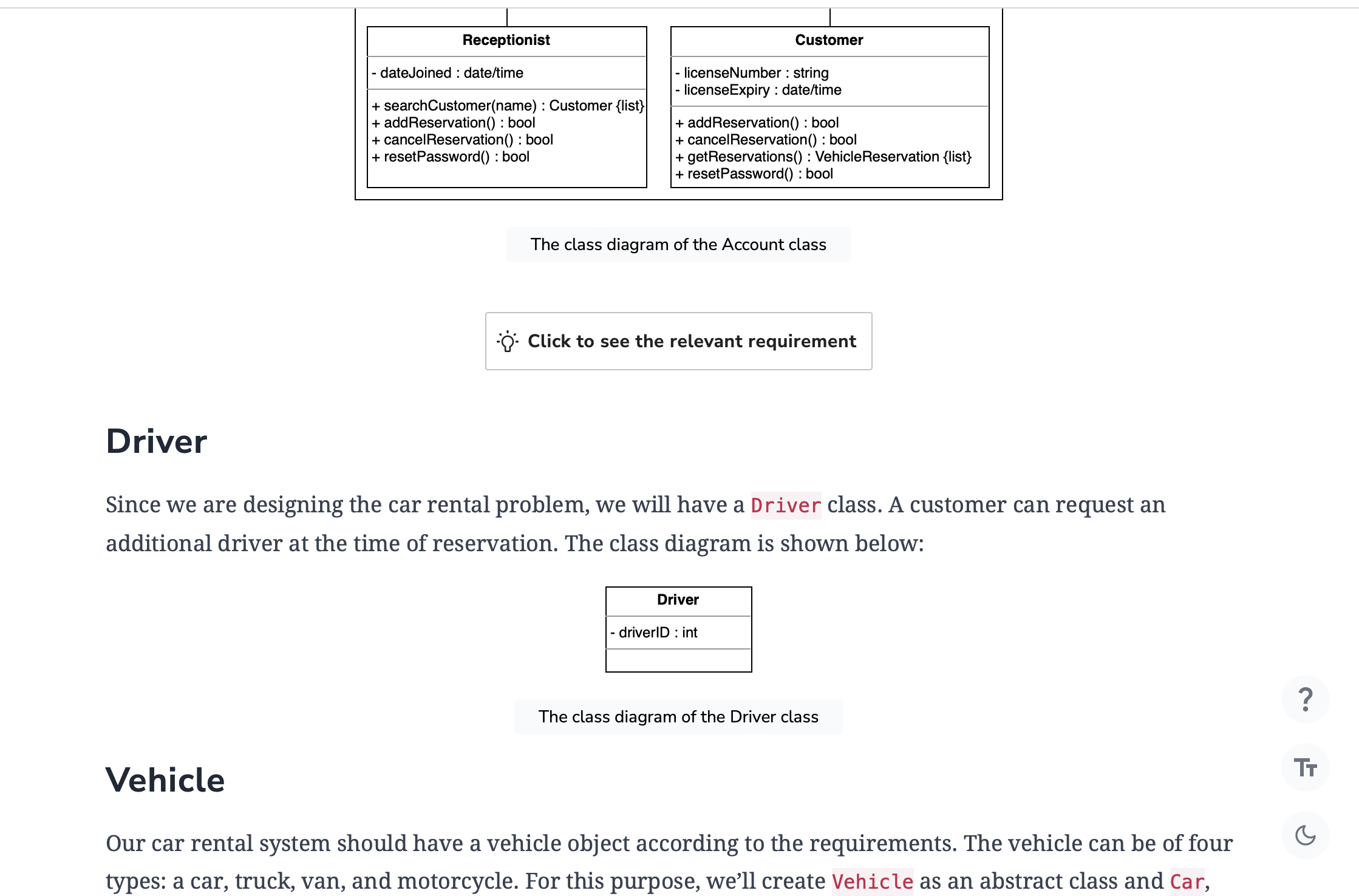
## Components of a car rental system

As mentioned earlier, we’ll design the car rental system using a bottom-up approach.

### Address and person

The Address is a custom data type that is required to store any address. The Address contains attributes like a street address, city, state, etc. In the car rental system, this class will be used to specify the address of any person or a car rental location or branch. The Person class stores information related to a person like a name, email, phone number, and address. In the Person class, there is an object of the Address type to specify the person’s address. The class representation of Address and Person is given below:



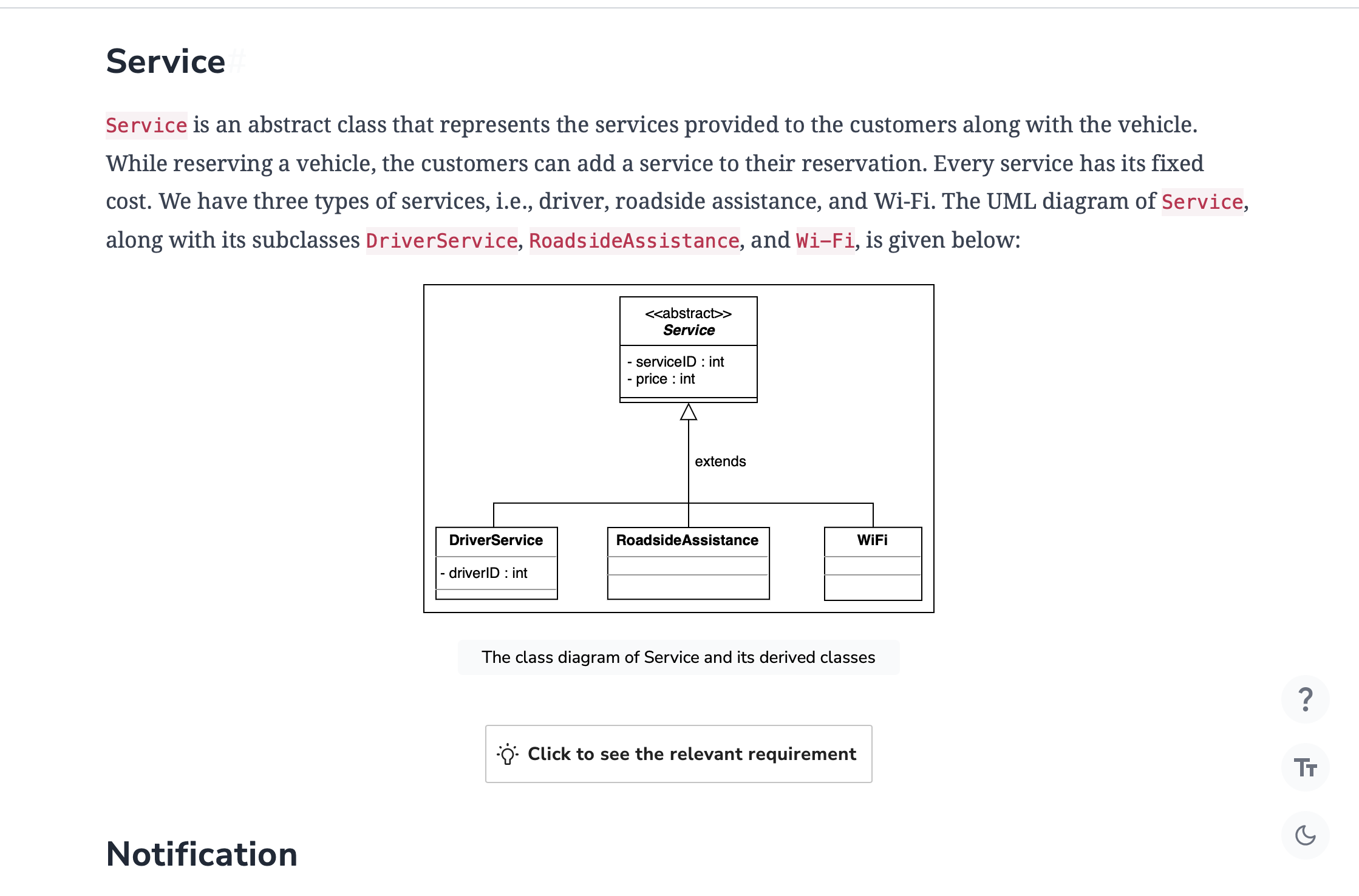


A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer screen

Description automatically generated

A screenshot of a parking garage

Description automatically generated

A screenshot of a car rental service

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

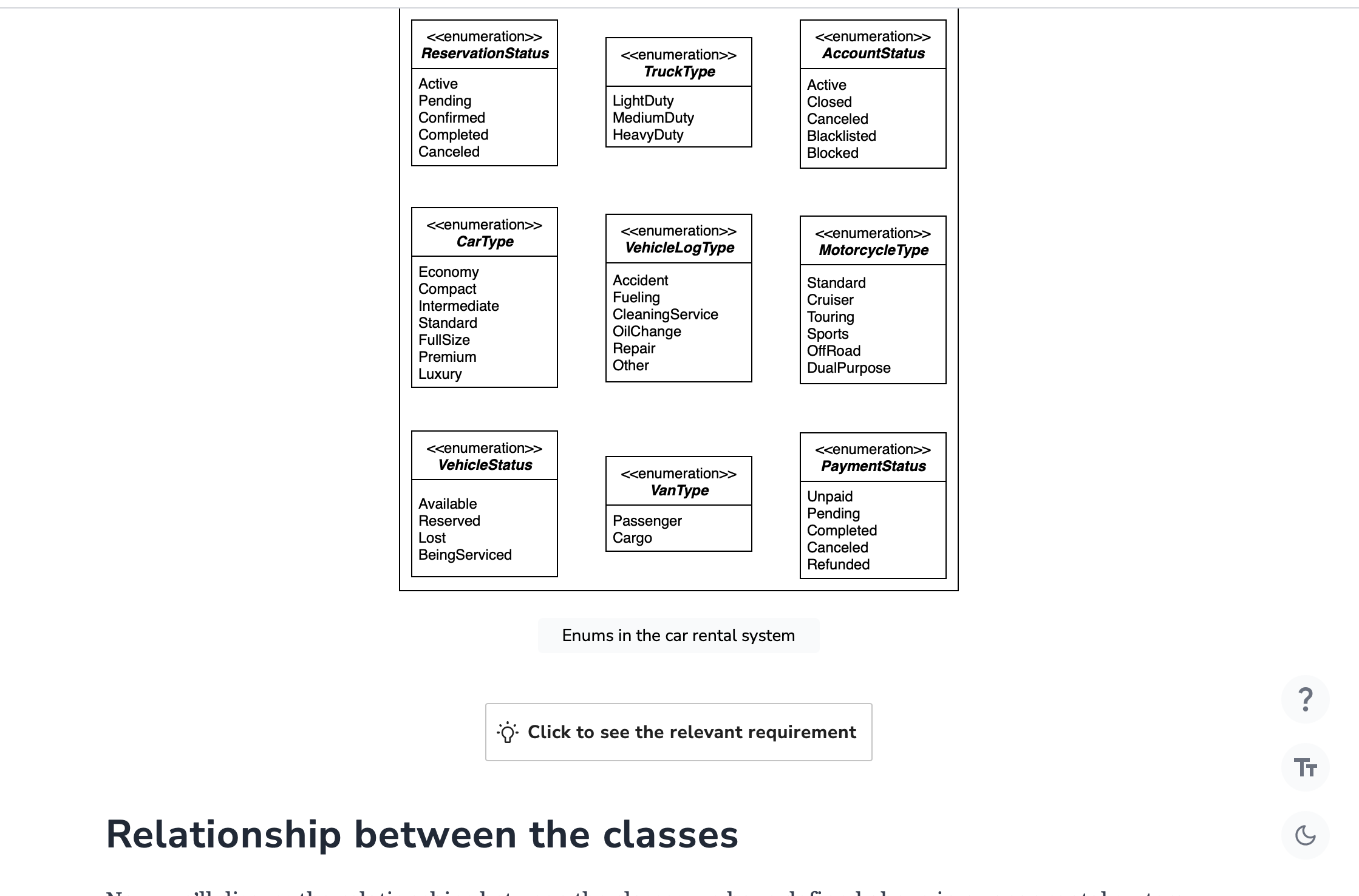
Description automatically generated

A screenshot of a computer screen

Description automatically generated

* CarType: The car type tells about the different types of cars, whether it is economy, compact, intermediate, standard, full size, premium, or luxury.
* MotorcycleType: Similar to the car type, the motorcycle type tells about the different types of motorcycles, whether it is standard, cruiser, touring, sports, off-road, or dual purpose.
* TruckType: The truck type specifies that the truck can be of three types, i.e, light-duty, medium-duty, or heavy-duty.
* VehicleLogType: The vehicle log type describes the type of a particular log of a vehicle, whether it is an accident, fueling, cleaning service, oil change, repair, or other.

These enumerations can be represented using the following class diagram:



## Relationship between the classes

Now, we’ll discuss the relationships between the classes we have defined above in our car rental system.

### Association

The class diagram has the following association relationships.

#### One-way association

* Both the Account and VehicleReservation classes have a one-way association with the Vehicle class.
* The Fine class has a one-way association with Payment.

A screenshot of a computer

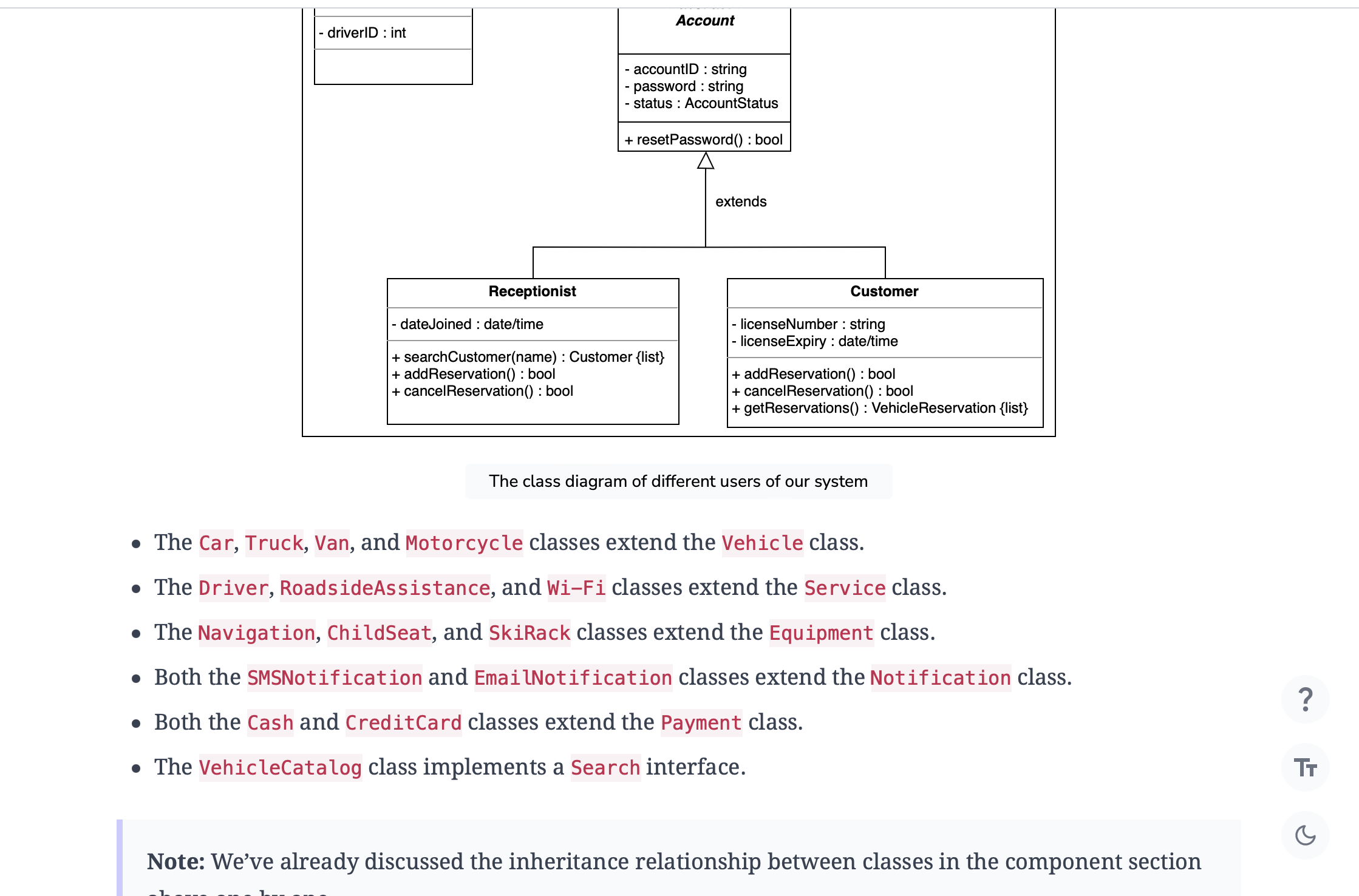
Description automatically generated

A screenshot of a computer

Description automatically generated

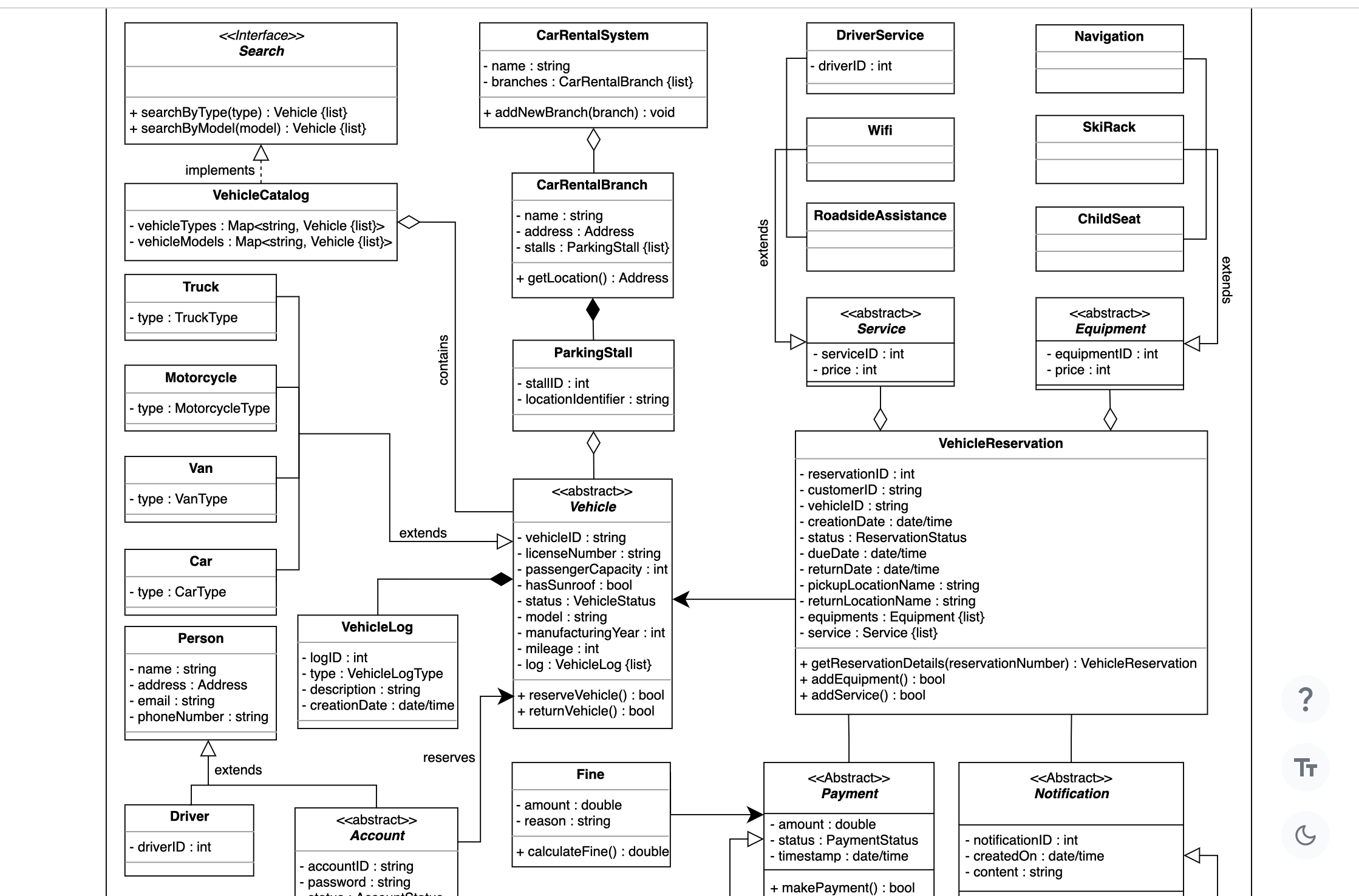
A screenshot of a computer

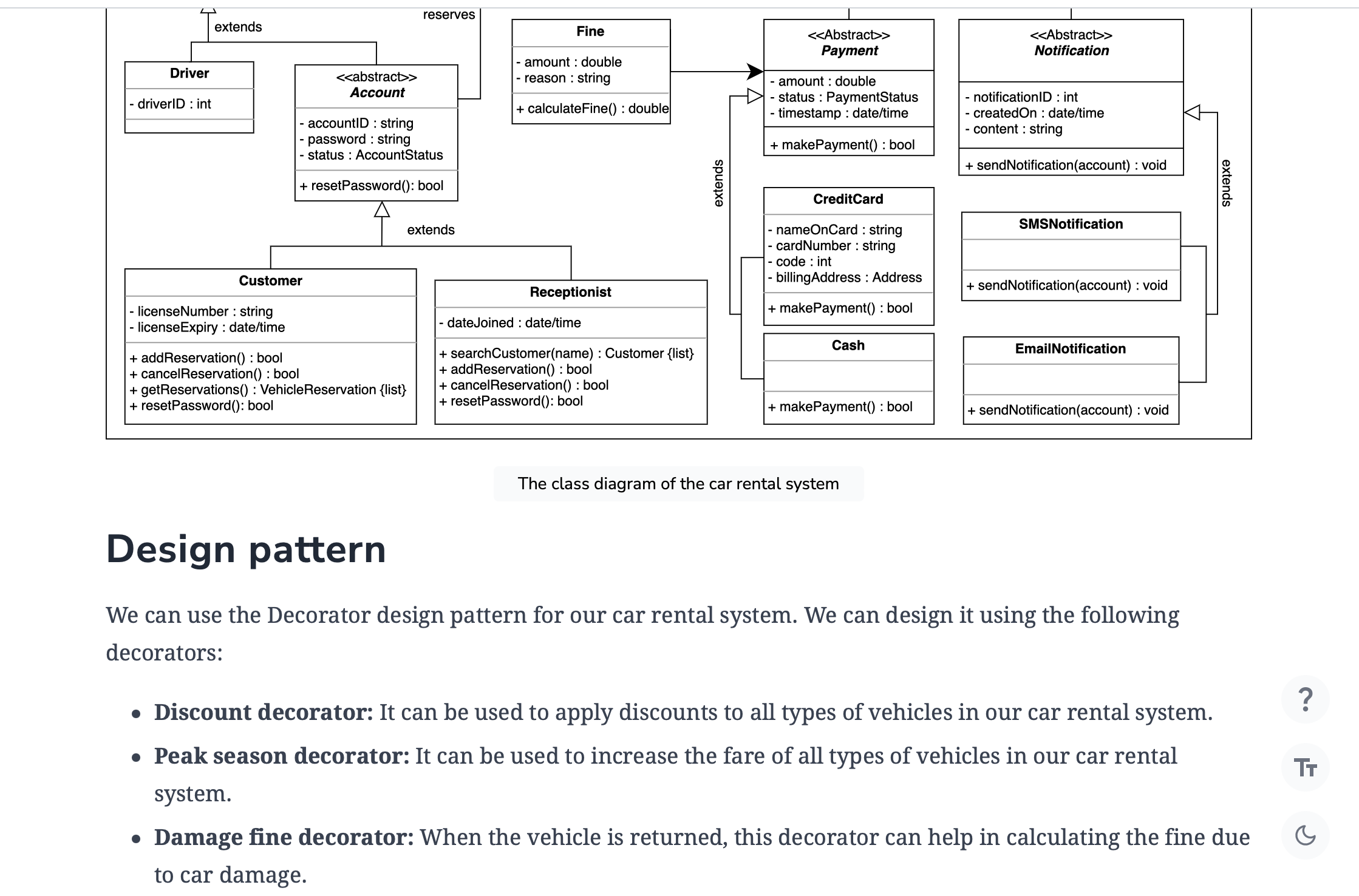
Description automatically generated



## Class diagram of the car rental system

Here’s the class diagram of the car rental system:



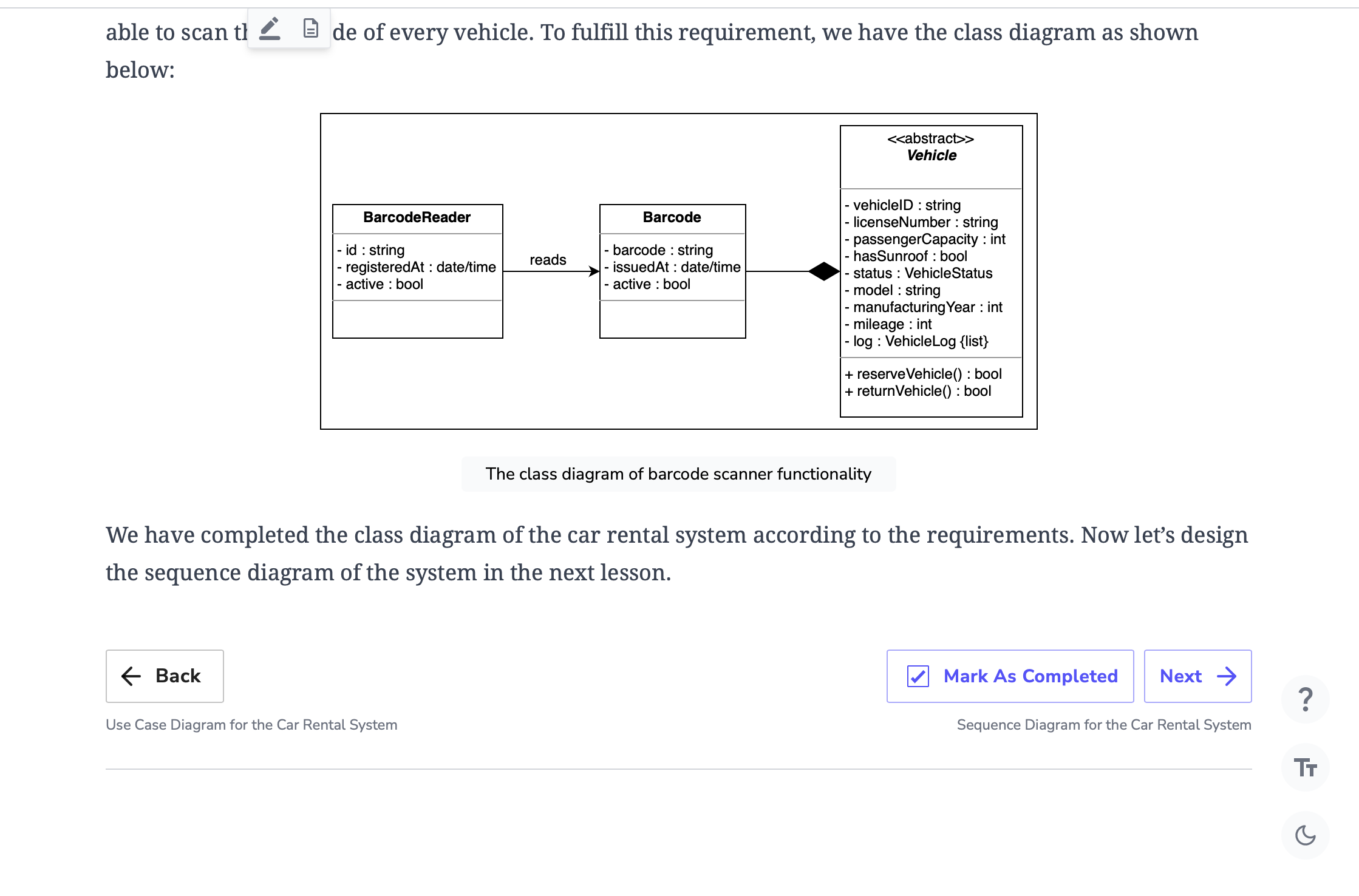


* **Partially filled fuel tank fine decorator:** When the vehicle is returned, this decorator can help calculate the fine due to the partially filled fuel tank.

Similarly, we can make several other decorators according to the system needs. These decorator fulfill the SRP and OCP design principles.

## Additional requirements

The interviewer can introduce some additional requirements in the car rental system, or they can ask some follow-up questions. Let's see an example of additional requirements:  
Barcode Scanner: Each vehicle should have a unique barcode associated with it, and the system should be able to scan the barcode of every vehicle. To fulfill this requirement, we have the class diagram as shown below:



**Sequence Diagram for the Car Rental System**

Create a sequence diagram for online vehicle reservation in the car rental system and solve a challenge.

**We'll cover the following**

* [Vehicle reservation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Vehicle-reservation)
* [Sequence challenge: Cancel reservation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Sequence-challenge:-Cancel-reservation)

Sequence diagrams are a great way to understand the interactions between different entities and objects in the system. There can be different sequence diagrams that we can create for our car rental system. For the sake of this lesson, we will create sequence diagrams for the following two interactions:

* **Vehicle reservation:** The member reserves a vehicle.
* **Sequence challenge:** The member cancels their reservation.

**Vehicle reservation**

The sequence diagram for vehicle reservation should have the following actors and objects that will interact with each other:

* **Actor:** Member
* **Objects:** Catalog, Reservation, and Payment
* System

Here are the steps in the reserve vehicle interaction:

1. The member searches for a vehicle.
2. If the vehicle is available:
   1. The catalog returns a list of available vehicles.
   2. The member selects a vehicle to reserve.
   3. The system creates a reservation against the vehicle.
   4. The reservation fee is sent to the system.
   5. The system requests payment from the member.
   6. The member initiates payment against the reservation fee.
   7. The payment is processed.
   8. If the payment is successful:
      1. The system is informed that the payment is completed.
      2. The system updates the reservation status to “confirmed”.
      3. The member is informed that the vehicle is reserved.
   9. Else if the payment is unsuccessful:
      1. The system is informed that the payment has failed.
      2. The system deleted the reservation.
      3. The system informs the member that the reservation is unsuccessful.
3. Else, if the vehicle is unavailable:
   1. Members are notified that no vehicles are available.

Based on the order above, the sequence diagram of a vehicle reservation in a car rental system is provided below:

A screenshot of a computer

Description automatically generated

## Sequence challenge: Cancel reservation

You’ll help us complete a sequence diagram for a reservation canceled by the member. A skeleton of the cancel reservation sequence diagram is given below:

A diagram of a diagram

Description automatically generated

Alternatively, you can also click the "Show complete diagram" button below to view the complete sequence diagram of the cancel reservation interaction.

Show complete diagram

Next, let’s look at the activity diagrams for the car rental system to understand the control flow of the system.

Back

# Activity Diagram for the Car Rental System

Create some activity diagrams for the car rental system problem

**We'll cover the following**

* [Vehicle pickup](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Vehicle-pickup)
  + [States](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#States)
  + [Actions](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Actions)
* [Activity challenge: Vehicle return](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Activity-challenge:-Vehicle-return)

Activities diagrams are a great way to visualize the flow of messages from one activity to the other in the system. There can be different activity diagrams that we can create for our car rental system. In this lesson, we will create activity diagrams for the following two activities:

* Vehicle pickup
* **Activity challenge:**Vehicle return

## Vehicle pickup

The states and actions that will be involved in this activity diagram are given below.

### States

**Initial state:**A member with a vehicle booking comes to the rent-a-car reception to pick up the vehicle.

**Final state:**The guest either successfully gets the vehicle or the system shows a reservation error.

### Actions

The member with a vehicle reservation arrives at the rent-a-car reception. The receptionist validates the reservation and updates the vehicle status.

Based on the order above, the activity diagram of the vehicle pickup from the car rental facility is given below.

A screenshot of a diagram

Description automatically generated

A diagram of a vehicle return

Description automatically generated

Notice that the actions in the diagram above are numbered from 1 to 9. The slots below represent the activities, and the arrows represent the flow from one activity to the other. Can you rearrange the slots below in the correct order they should appear in the activity diagram given above?

**Note:** If you get stuck, just click the “Show Solution” button to check the correct answer.

We've looked at some of the activity diagrams of our car rental system. In the next lesson, we will present the code for our designed classes in some of the most popular languages.

A screenshot of a diagram

Description automatically generated

# Code for the Car Rental System

Write object-oriented code to implement the design of the car rental system problem.

**We'll cover the following**

* [Car rental system classes](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Car-rental-system-classes)
  + [Enumerations](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Enumerations)
  + [Address, person, and driver](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Address,-person,-and-driver)
  + [Account](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Account)
  + [Vehicle](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Vehicle)
  + [Equipment](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Equipment)
  + [Service](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Service)
  + [Payment](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Payment)
  + [Vehicle log and Vehicle reservation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Vehicle-log-and-Vehicle-reservation)
  + [Notification](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Notification)
  + [Parking stall and fine](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Parking-stall-and-fine)
  + [Search interface and vehicle catalog](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Search-interface-and-vehicle-catalog)
  + [Car rental system and car rental branch](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Car-rental-system-and-car-rental-branch)
* [Wrapping up](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Wrapping-up)

We’ve reviewed different aspects of the car rental system and observed the attributes attached to the problem using various UML diagrams. Let’s explore the more practical side of things, where we will work on implementing the car rental system using multiple languages. This is usually the last step in an object-oriented design interview process.

We have chosen the following languages to write the skeleton code of the different classes present in the car rental system:

* Java
* C#
* Python
* C++
* JavaScript

## Car rental system classes

In this section, we will provide the skeleton code of the classes designed in the class diagram lesson.

**Note:**For simplicity, we are not defining getter and setter functions. The reader can assume that all class attributes are private and accessed through their respective public getter methods and modified only through their public methods function.

### Enumerations

First, we will define all the enumerations required in the car rental system. According to the class diagram, there are seven enumerations used in the system, i.e., VehicleStatus, AccountStatus, ReservationStatus, PaymentStatus, VanType, CarType, and VehicleLogType.The code to implement these enumerations is as follows:

**Note:** JavaScript does not support enumerations, so we will use the Object.freeze() method as an alternative that freezes an object and prevents further modifications.

Java

// definition of enumerations used in the car rental system

enum VehicleStatus {

AVAILABLE,

RESERVED,

LOST,

BEING\_SERVICED

}

enum AccountStatus {

ACTIVE,

CLOSED,

CANCELED,

BLACKLISTED,

BLOCKED

}

enum ReservationStatus {

ACTIVE,

PENDING,

CONFIRMED,

COMPLETED,

CANCELED

}

enum PaymentStatus {

UNPAID,

PENDING,

COMPLETED,

CANCELED,

REFUNDED

}

enum VanType {

PASSENGER,

CARGO

}

enum CarType {

ECONOMY,

COMPACT,

INTERMEDIATE,

STANDARD,

FULL\_SIZE,

PREMIUM,

LUXURY

}

enum MotorcycleType {

STANDARD,

CRUISER,

TOURING,

SPORTS,

OFF\_ROAD,

DUAL\_PURPOSE

}

enum TruckType {

LIGHT\_DUTY,

MEDIUM\_DUTY,

HEAVY\_DUTY

}

enum VehicleLogType {

ACCIDENT,

FUELING,

CLEANING\_SERVICE,

OIL\_CHANGE,

REPAIR,

OTHER

}

### Address, person, and driver

This section contains the Address, Person, and Driver classes, where the first two classes are used as a custom data type. The implementation of these classes is shown below:se classes can be found below:

Java

public class Address {

private String streetAddress;

private String city;

private String state;

private int zipCode;

private String country;

}

public class Person {

private String name;

private Address address;

private String email;

private String phoneNumber;

}

public class Driver extends Person {

private int driverId;

}

### Account

Account is an abstract class that represents the various people or actors that can interact with the system. There are two types of accounts: receptionist and customer. The implementation of Account and its subclasses is shown below:

public abstract class Account extends Person {

private String accountId;

private String password;

private AccountStatus status;

public abstract boolean resetPassword();

}

public class Receptionist extends Account {

private Date dateJoined;

public List<Customer> searchCustomer(String name);

public boolean addReservation();

public boolean cancelReservation();

public boolean resetPassword() {

// definition

}

}

public class Customer extends Account {

private String licenseNumber;

private Date licenseExpiry;

public boolean addReservation();

public boolean cancelReservation();

public List<VehicleReservation> getReservations();

public boolean resetPassword() {

// definition

}

}

### Vehicle

Vehicle will be another abstract class, which serves as a parent for four different types of vehicles: Car, Van, Truck, and MotorCycle. The definition of the Vehicle and its child classes is given below:

// Vehicle is an abstract class

public abstract class Vehicle {

private String vehicleId;

private String licenseNumber;

private int passengerCapacity;

private boolean hasSunroof;

private VehicleStatus status;

private String model;

private int manufacturingYear;

private int mileage;

private List<VehicleLog> log;

public boolean reserveVehicle();

public boolean returnVehicle();

}

public class Car extends Vehicle {

private CarType carType;

}

public class Van extends Vehicle {

private VanType vanType;

}

public class Truck extends Vehicle {

private TruckType truckType;

}

public class Motorcycle extends Vehicle {

private MotorcycleType motorcycleType;

}

### Equipment

Equipment is an abstract class, and this section represents different equipment: Navigation, ChildSeat, and SkiRack added in the reservation. The code to implement these classes is shown below:

// Equipment is an abstract class

public abstract class Equipment {

private int equipmentId;

private int price;

}

public class Navigation extends Equipment {

}

public class ChildSeat extends Equipment {

}

public class SkiRack extends Equipment {

}

### Service

Service is an abstract class, and this section represents different services: DriverService, RoadsideAssistance, and Wi-Fi  added to the reservation. The code to implement these classes is shown below:

// Service is an abstract class

public abstract class Service {

private int serviceId;

private int price;

}

public class DriverService extends Service {

private int driverId;

}

public class RoadsideAssistance extends Service {

}

public class WiFi extends Service {

}

### Payment

The Payment class is another abstract class, with the Cash and CreditCard classes as its child. This takes in the PaymentStatus enum to keep track of the payment status. The definition of this class is provided below:

// Payment is an abstract class

public abstract class Payment {

// Data members

private double amount;

// The Date datatype represents and deals with both date and time.

private Date timestamp;

private PaymentStatus status;

public abstract boolean makePayment();

}

public class Cash extends Payment {

public boolean makePayment() {

// functionality

}

}

public class CreditCard extends Payment {

// Data members

private String nameOnCard;

private String cardNumber;

private String billingAddress;

private int code;

public boolean makePayment() {

// functionality

}

}

### Vehicle log and Vehicle reservation

VehicleLog is a class responsible for keeping track of all the events related to a vehicle. VehicleReservation is a class responsible for managing the reservation of vehicles. The implementation of this class is given below:

public class VehicleLog {

private int logId;

private VehicleLogType logType;

private String description;

private Date creationDate;

}

public class VehicleReservation {

private int reservationId;

private String customerId;

private String vehicleId;

private Date creationDate;

private ReservationStatus status;

private Date dueDate;

private Date returnDate;

private String pickupLocation;

private String returnLocation;

private List<Equipment> equipments;

private List<Service> services;

public static VehicleReservation getReservationDetails();

public boolean addEquipment();

public boolean addService();

}

### Notification

The Notification class is another abstract class responsible for sending notifications, with the SMSNotification and EmailNotification classes as its child. The implementation of this class is shown below:

// Notification is an abstract class

public abstract class Notification {

private int notificationId;

// The Date data type represents and deals with both date and time.

private Date createdOn;

private String content;

public abstract void sendNotification(Account account);

}

class SmsNotification extends Notification {

public void sendNotification(Account account) {

// functionality

}

}

class EmailNotification extends Notification {

public void sendNotification(Account account) {

// functionality

}

}A screenshot of a computer

Description automatically generated

public interface Search {

public List<Vehicle> searchByType(String type);

public List<Vehicle> searchByModel(String model);

}

public class VehicleCatalog implements Search {

private HashMap<String, List<Vehicle>> vehicleTypes;

private HashMap<String, List<Vehicle>> vehicleModels;

// to return all vehicles of the given type.

public List<Vehicle> searchByType(String type) {

// functionality

}

// to return all vehicles of the given model.

public List<Vehicle> searchByModel(String model) {

// functionality

}

}

### Car rental system and car rental branch

The CarRentalSystem class is the base class of the system that is used to represent the whole car rental system (or the top-level classes of the system). CarRentalBranch represents the single branch of the system. The implementation of these classes is given below:

public class CarRentalBranch {

private String name;

private Address address;

private List<ParkingStall> stalls;

public Address getLocation();

}

public class CarRentalSystem {

private String name;

private List<CarRentalBranch> branches;

public void addNewBranch(CarRentalBranch branch);

// The CarRentalSystem is a singleton class that ensures it will have only one active instance at a time

private static CarRentalSystem system = null;

// Created a static method to access the singleton instance of CarRentalSystem class

public static CarRentalSystem getInstance() {

if (system == null) {

system = new CarRentalSystem();

}

return system;

}

}

**Wrapping up**

We've explored the complete design of a car rental system in this chapter. We've looked at how a basic car rental system can be visualized using various UML diagrams and designed using object-oriented principles and design patterns.