Sequence Diagram for the Parking Lot

Create a sequence diagram for the card payment in the parking lot system and solve a challenge.

We'll cover the following

- Card payment
- Sequence challenge: Payment verification

A sequence diagram is 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 parking lot system. For the sake of this lesson, we will create sequence diagrams for the following two interactions:

- **Card payment:** This performs a payment using the card.
- **Sequence challenge:** This is for payment verification.

Card payment

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

- **Actor:** `Customer`
- **Object:** `CardReader`, `Payment`, and `ExitPanel`

Here are the steps in the card payment interaction:

1. The customer inserts the card into the card reader.
2. The card reader initiates a payment for the required parking fee.
3. The payment processes the payment and returns the payment status.
4. The card reader ejects the card.
5. If the payment is successful:

    I. The customer requests a receipt for the transaction

I. The customer requests a receipt for the transaction.

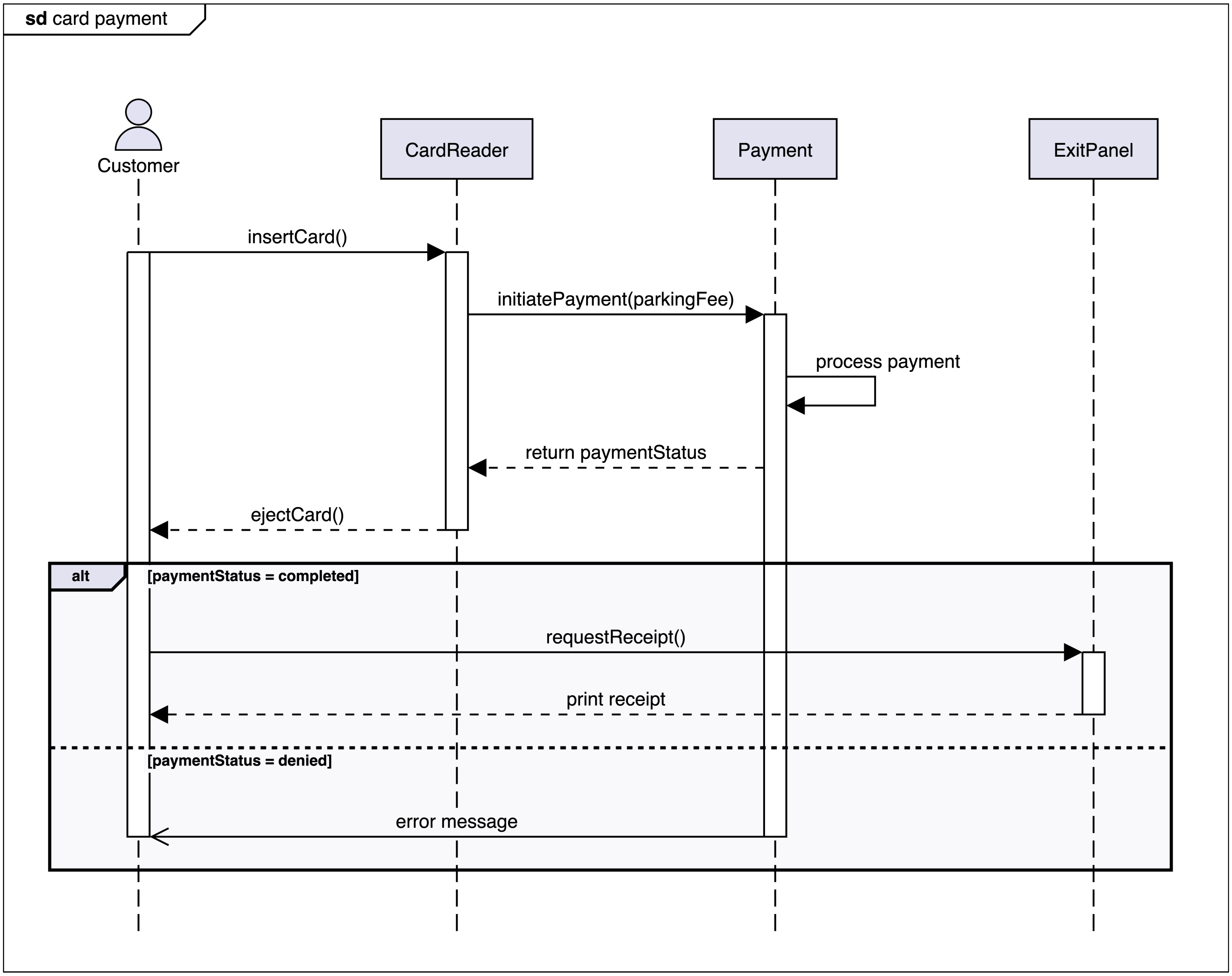
II. The exit panel prints a receipt for the customer.

6. If the payment is unsuccessful:

   I. The customer sees an error message for an unsuccessful Payment.

> **Note:** The `Payment` object is created when a vehicle enters the parking lot.

Based on the order above, the sequence diagram of the card payment in a parking lot system is given below.

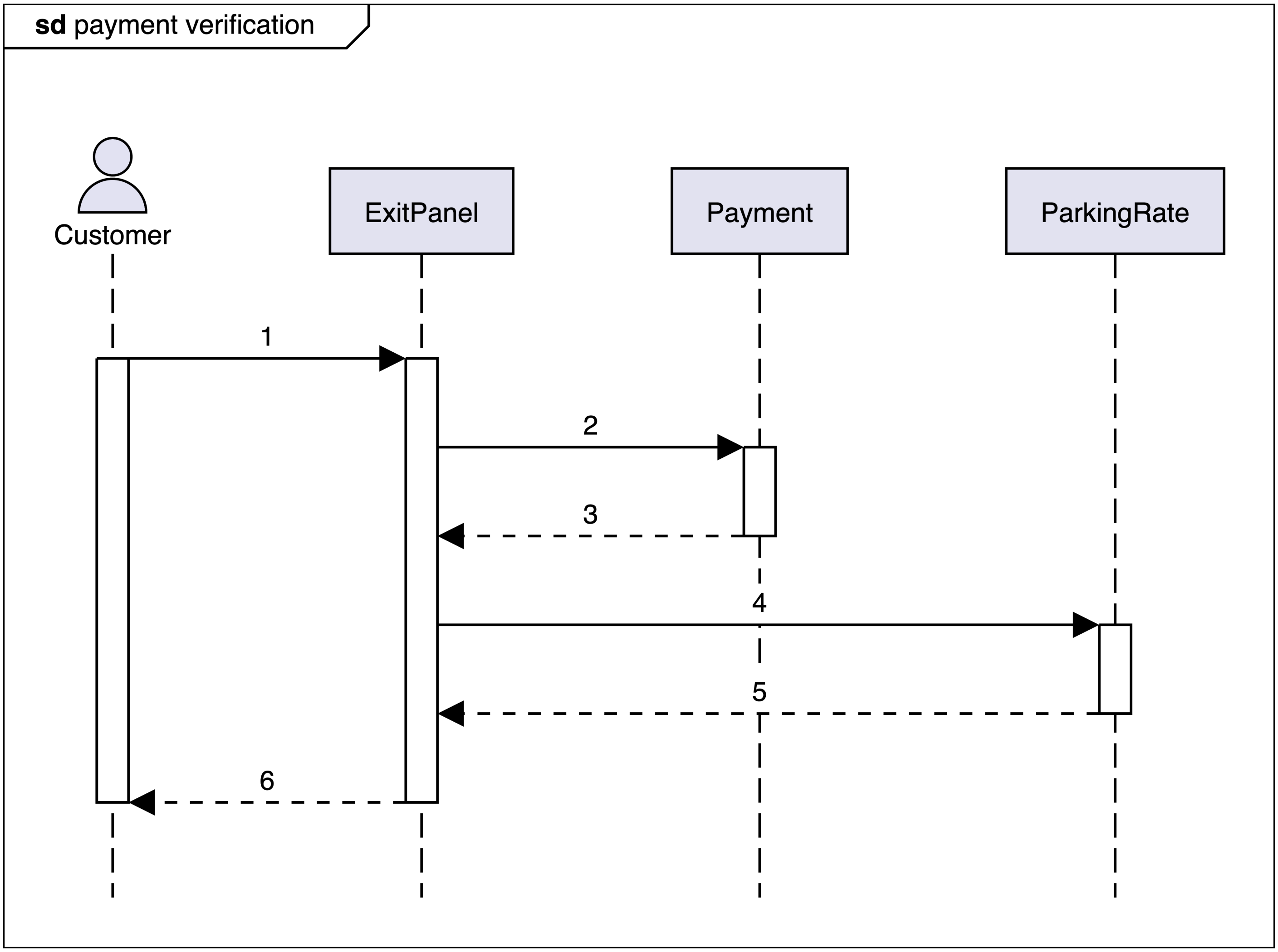


The sequence diagram for the card payment

Sequence challenge: Payment verification

In this section, you will help us in completing a sequence diagram for the payment verification of a customer at the exit panel.

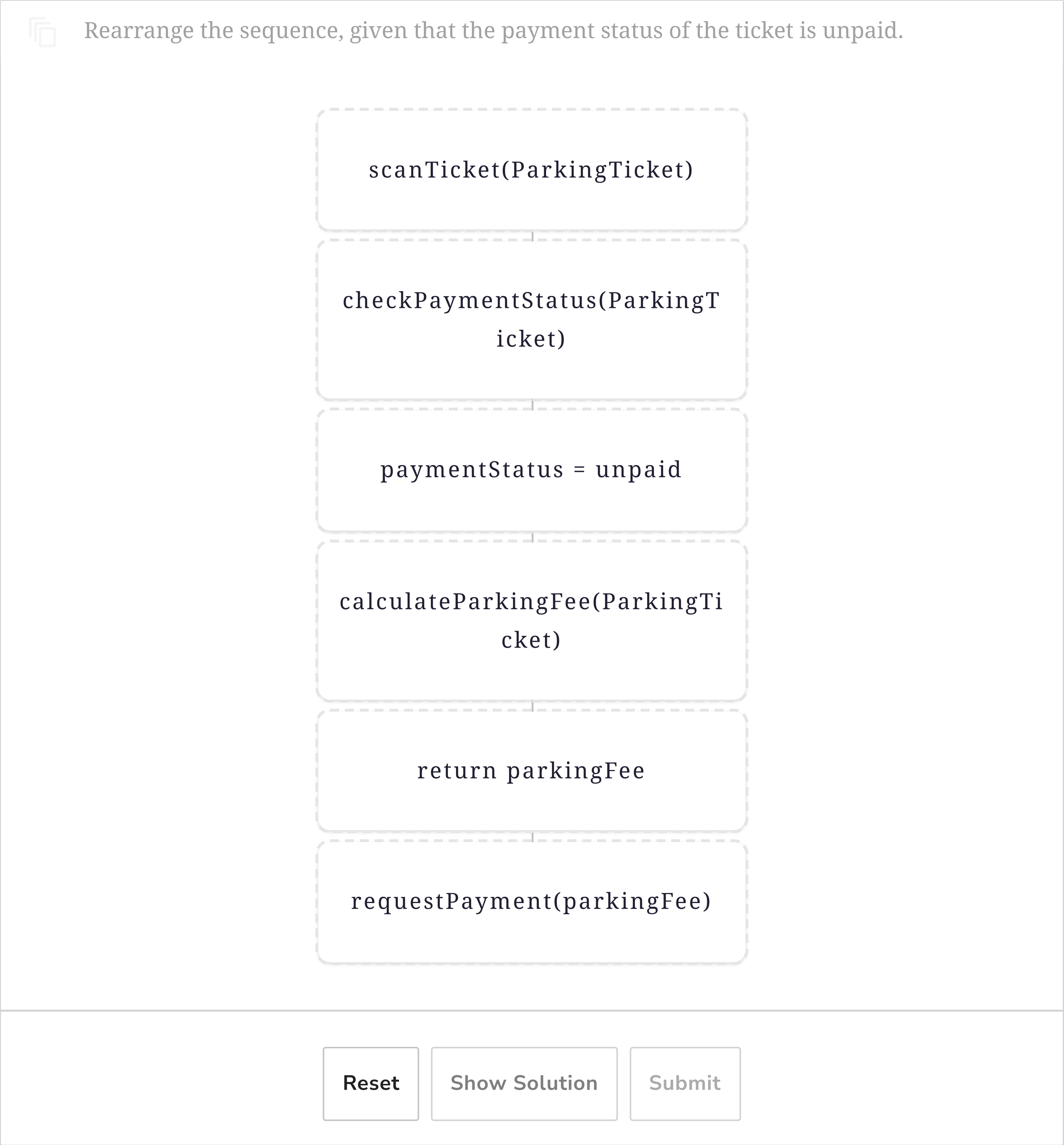
The skeleton below represents the payment verification sequence diagram. Here the payment status of the ticket is currently unpaid and the parking fee has to be calculated.



The sequence diagram for payment verification

Notice that the arrows in the diagram above are numbered from 1 to 6. The message boxes shown below are the messages to be exchanged between the actor(s) and object(s). Can you rearrange the messages below in the correct sequence of the order they should appear in the skeleton of the sequence diagram given above?

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



Rearrange the sequence, given that the payment status of the ticket is unpaid.

scanTicket(ParkingTicket)

checkPaymentStatus(ParkingTicket)

paymentStatus = unpaid

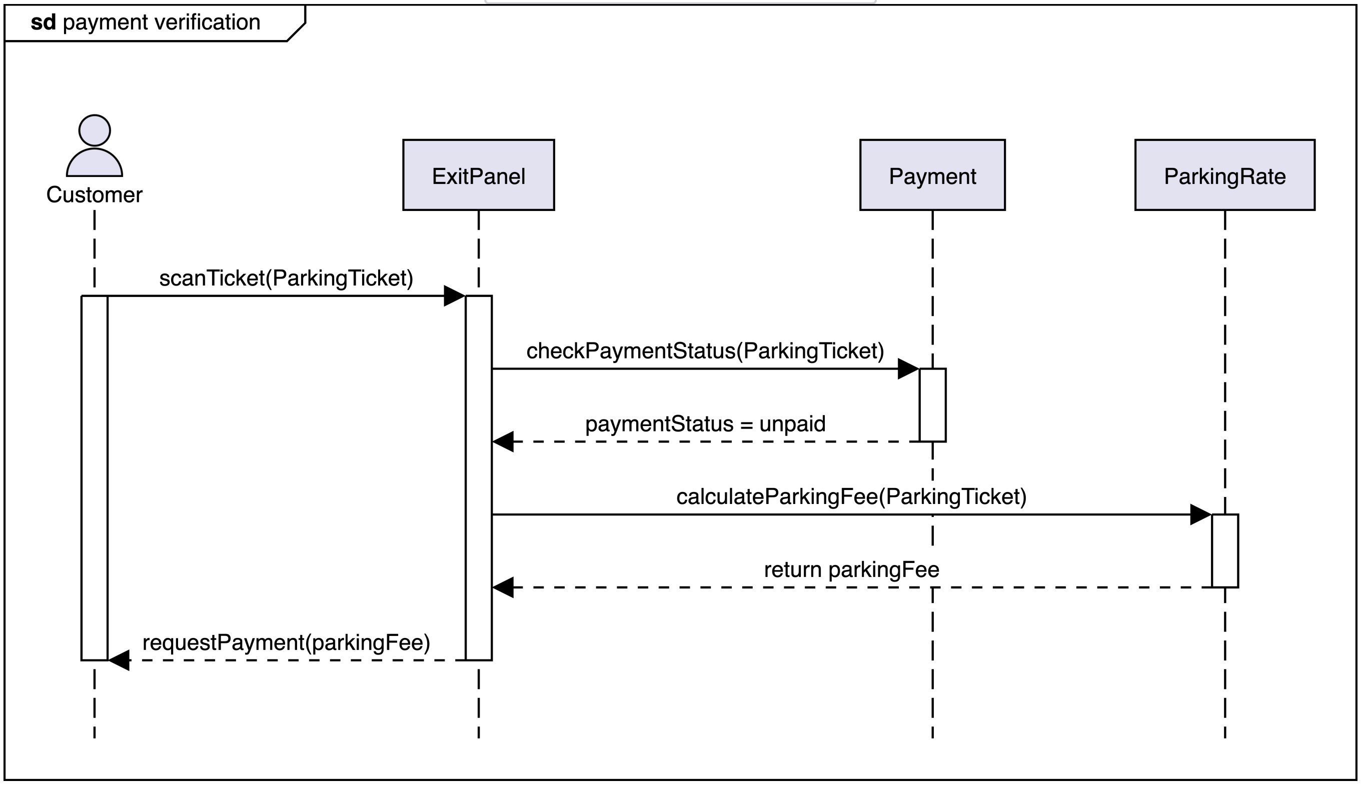
calculateParkingFee(ParkingTicket)

return parkingFee

requestPayment(parkingFee)

Alternatively, you can also click the "Show complete diagram" button below to see the complete sequence diagram.

Hide complete diagram



sd payment verification
Customer
ExitPanel
Payment
ParkingRate
scanTicket(ParkingTicket)
checkPaymentStatus(ParkingTicket)
paymentStatus = unpaid
calculateParkingFee(ParkingTicket)
return parkingFee
requestPayment(parkingFee)

In the next lesson, we will create activity diagrams for the parking lot design problem.

← Back

Class Diagram for the ...

Next →

Activity Diagram for th...

□ Mark as Completed