

# Software Design

- **Use Case Diagram:** Shows the system's functionality from the perspective of the user. It focuses on what the system does by listing the use cases (features) and actors (users).

# USE CASE DIAGRAM

# Use Case Diagram



CUSTOMER



WAITER

# Use Case Diagram



CUSTOMER



WAITER

View Order menu

# Use Case Diagram



CUSTOMER



WAITER

View Order menu

Place Order

# Use Case Diagram



CUSTOMER



WAITER

View Order menu

Place Order

Cancel Order

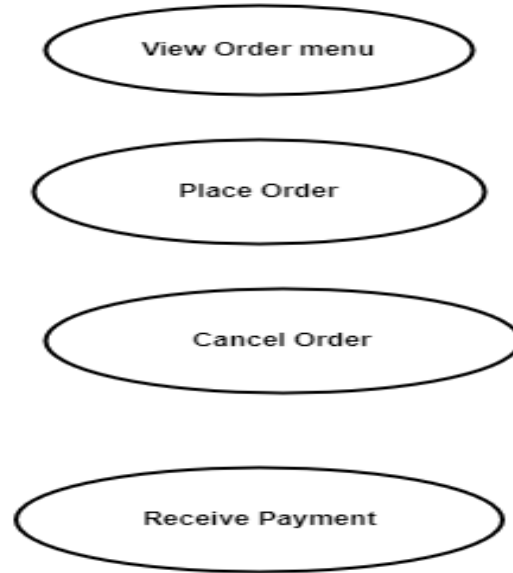
# Use Case Diagram



CUSTOMER



WAITER



# Use Case Diagram



CUSTOMER



WAITER

View Order menu

Place Order

Cancel Order

Receive Payment

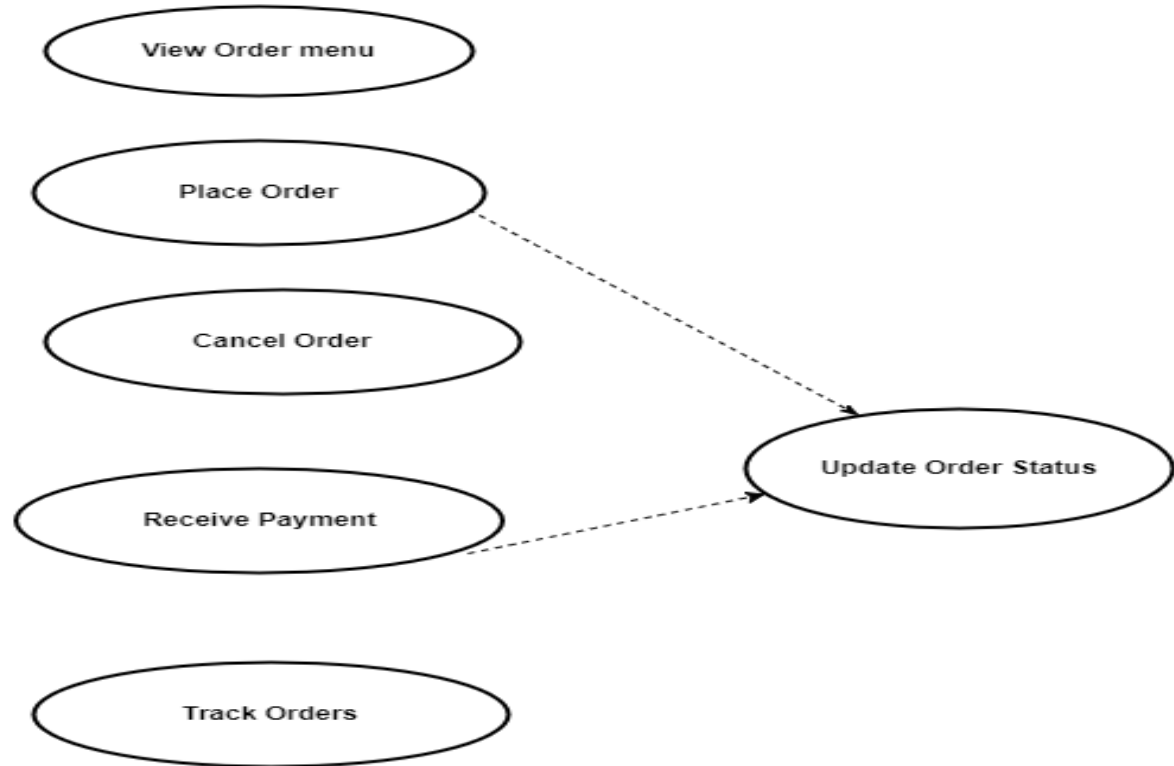
Track Orders



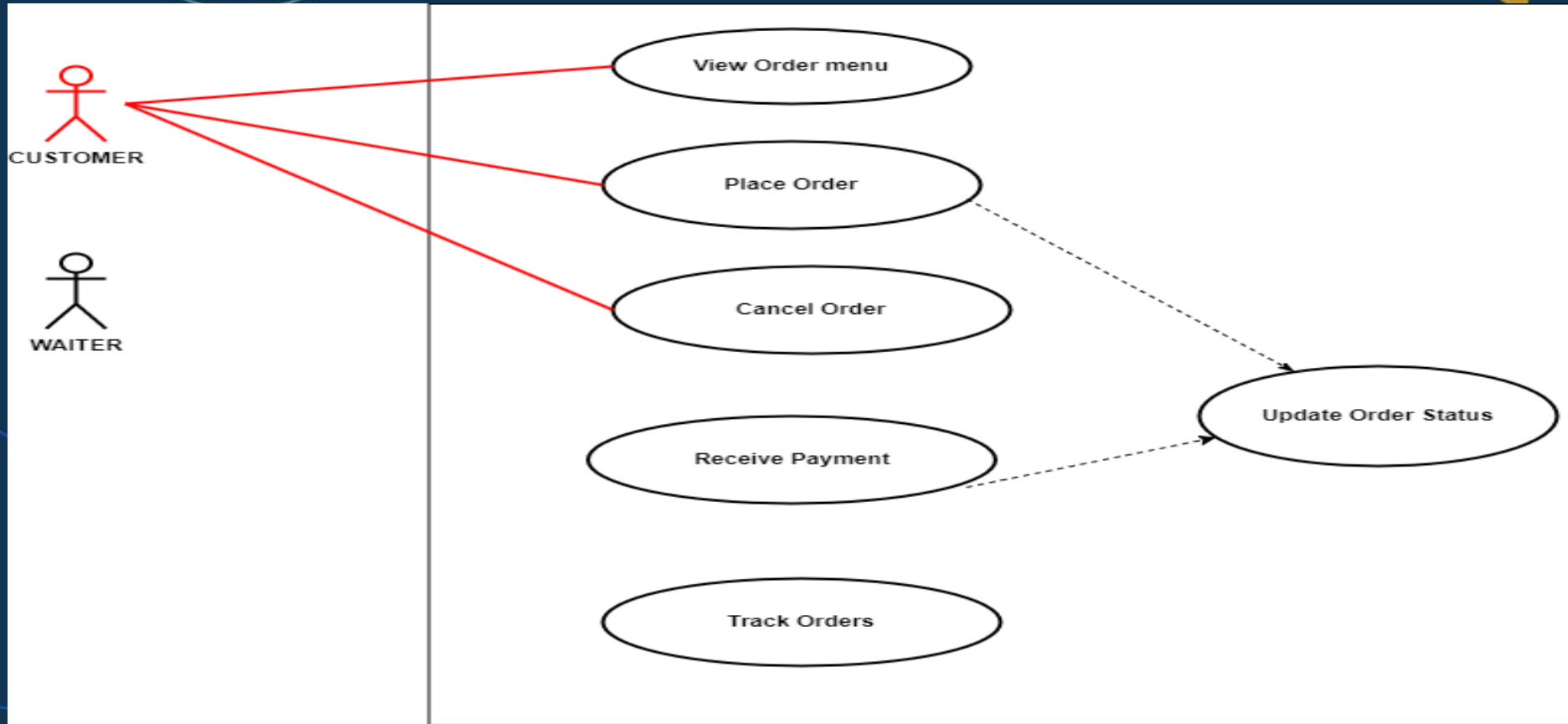
# Use Case Diagram

  
CUSTOMER

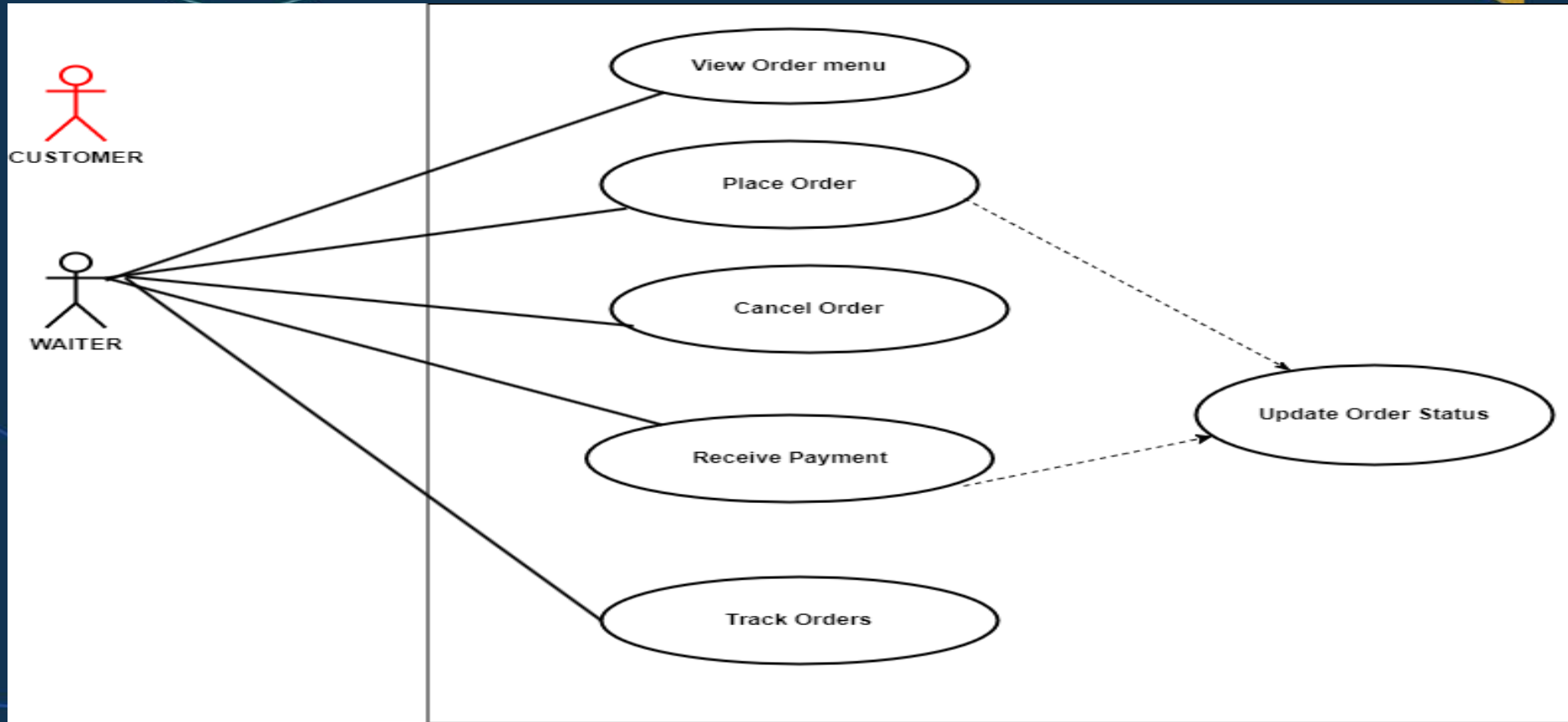
  
WAITER



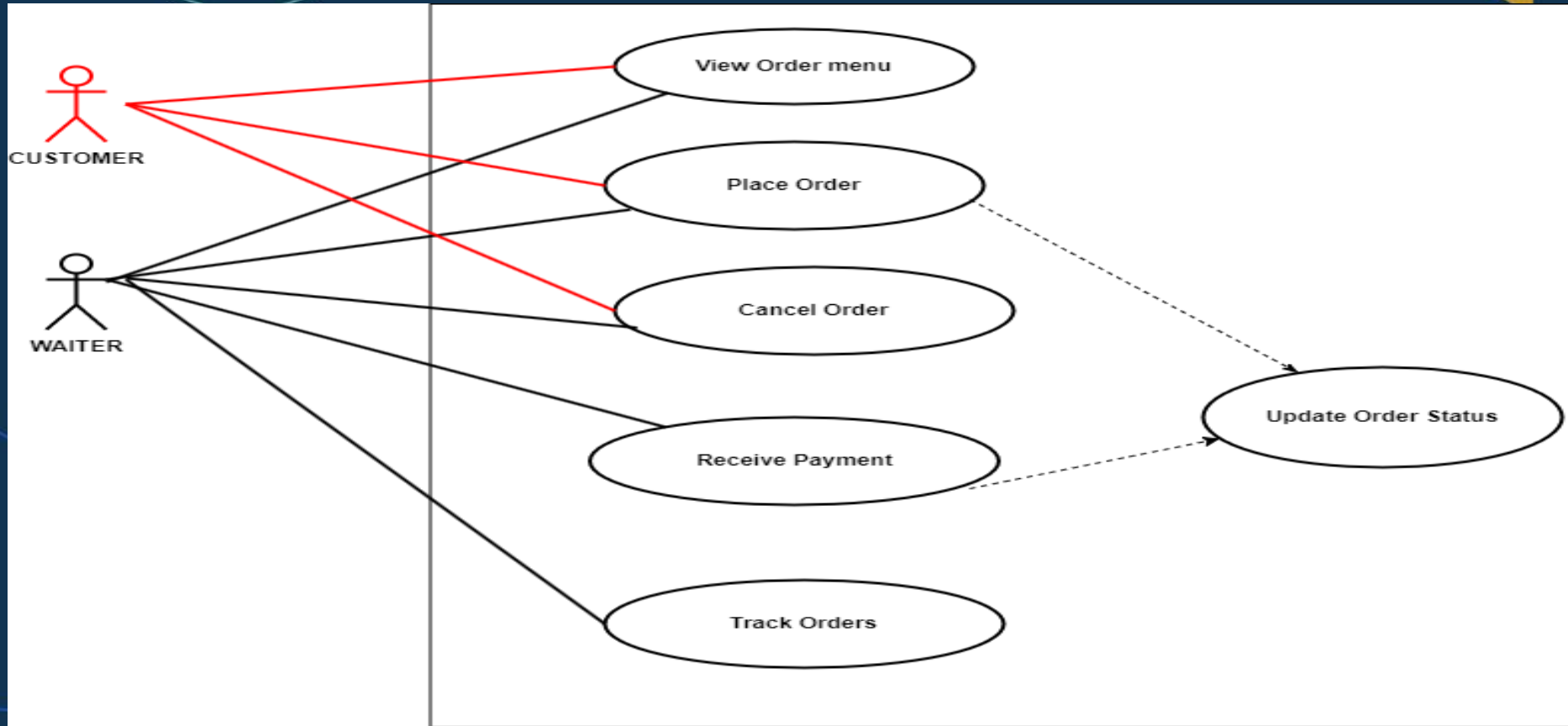
# Use Case Diagram



# Use Case Diagram



# Use Case Diagram

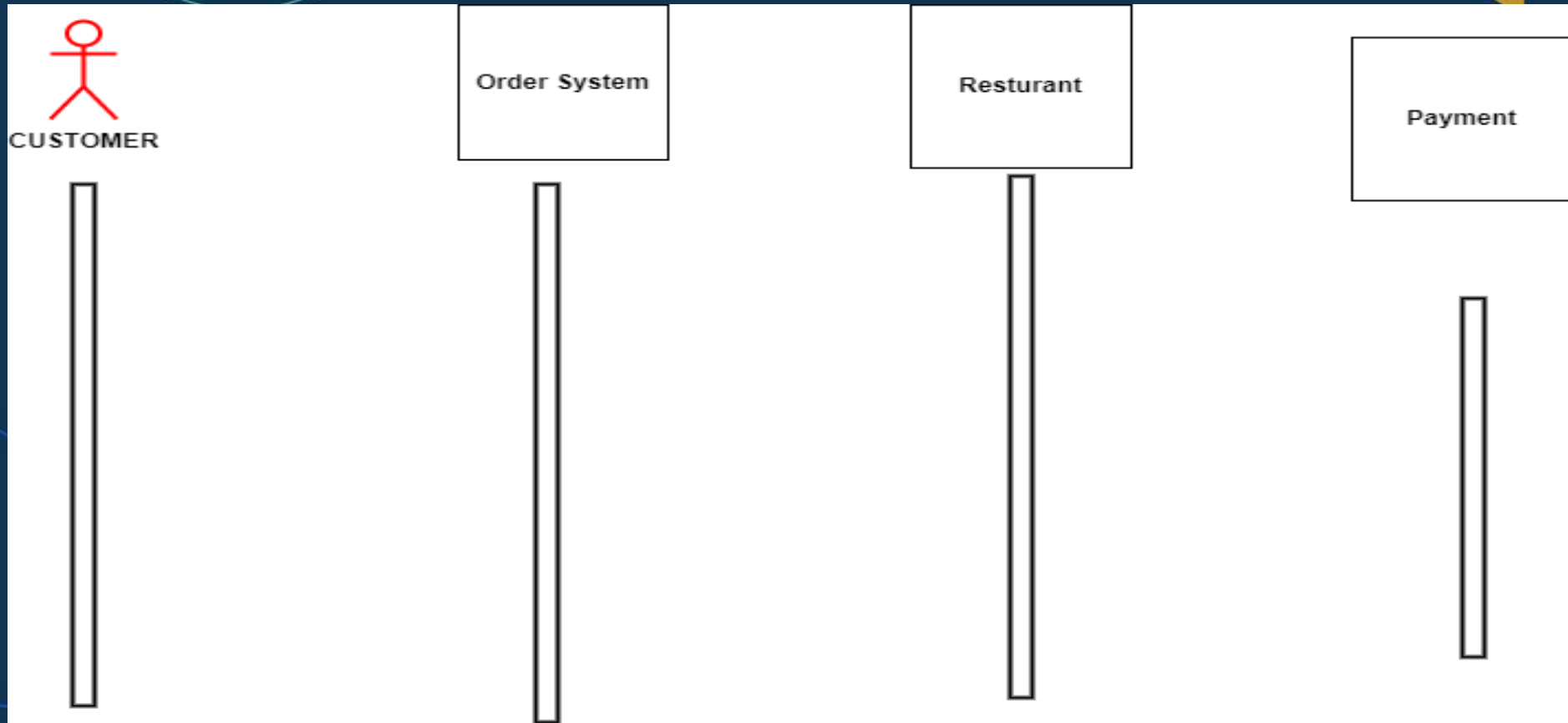


# SEQUENCE DIAGRAM

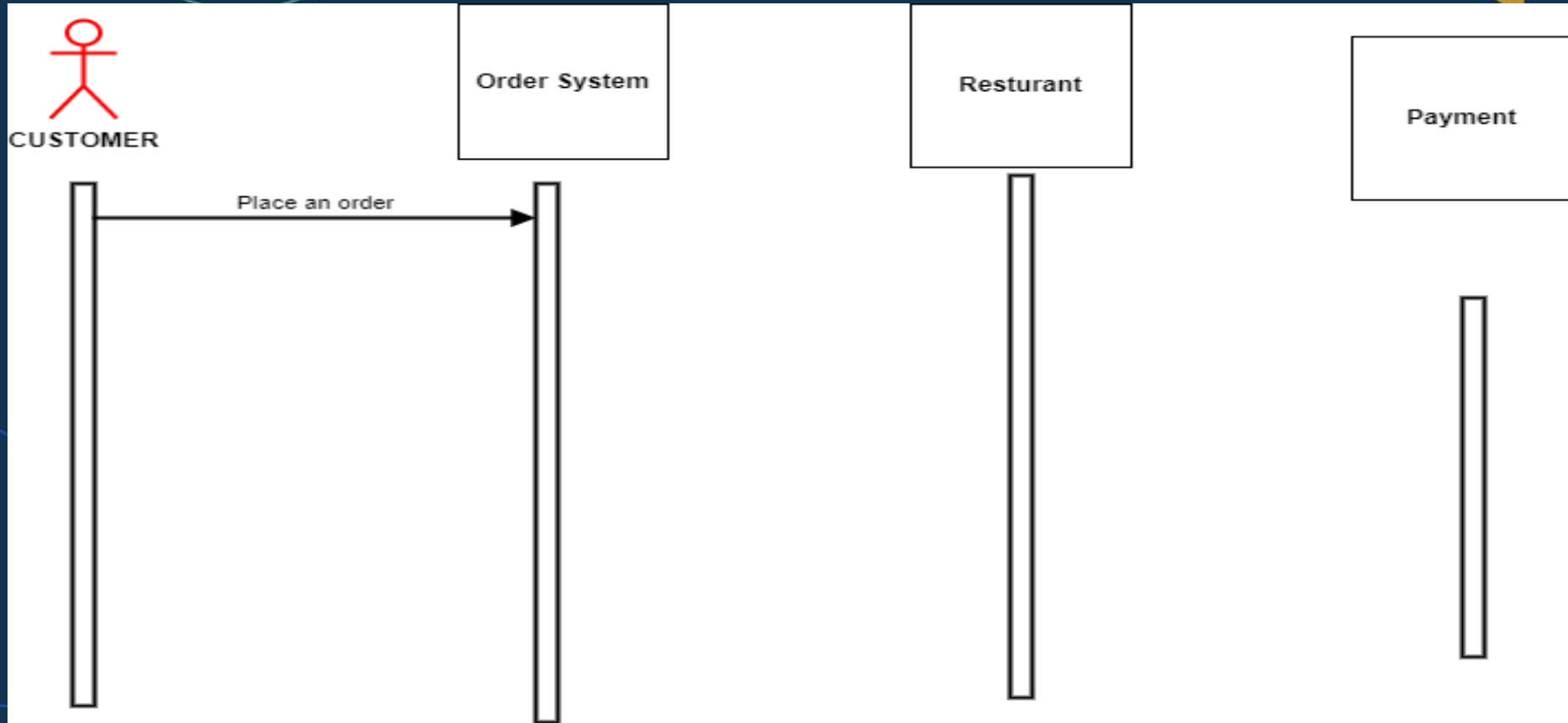
# Software Design

- **Sequence Diagram:** Shows the sequence of messages exchanged between objects to perform a specific operation. Focuses on how the system behaves over time.

# Sequence Diagram

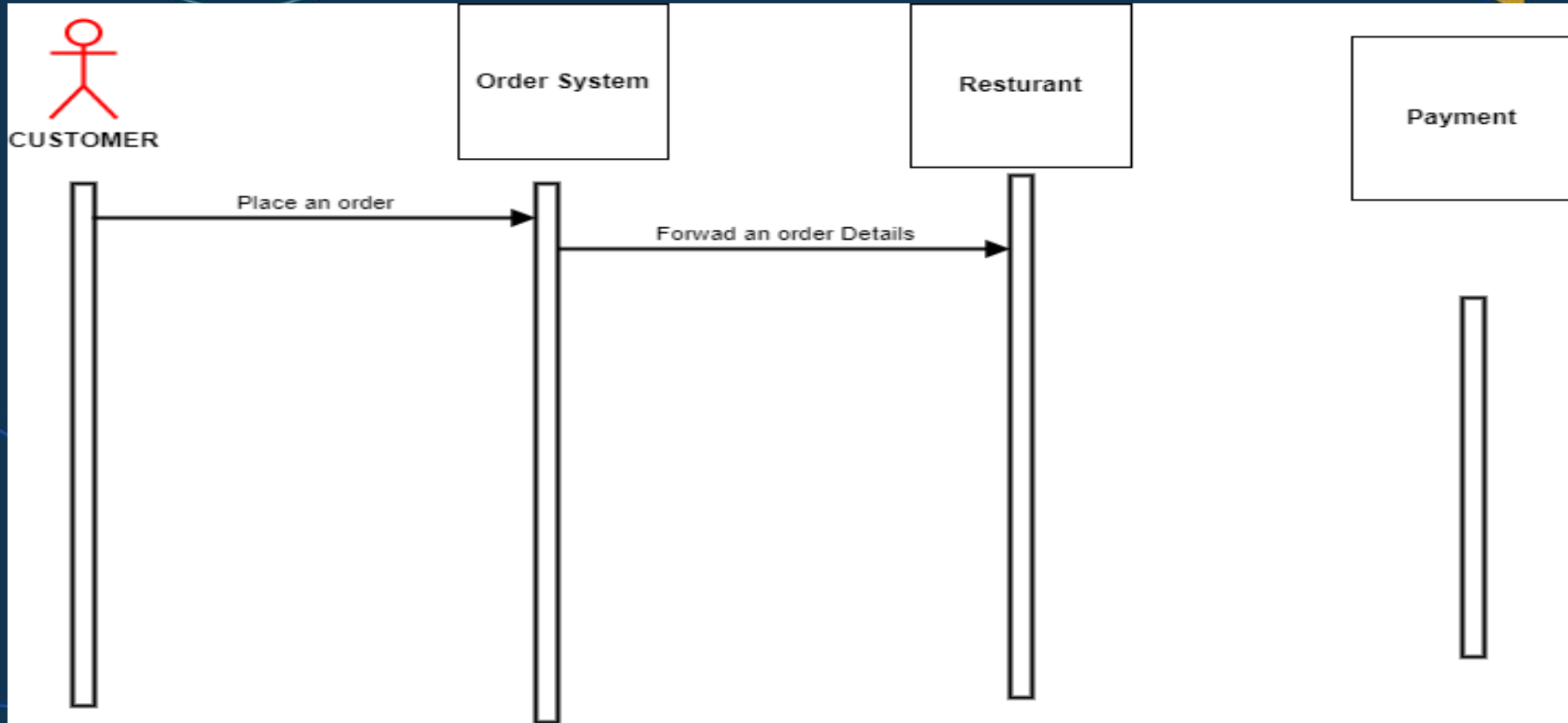


# Sequence Diagram

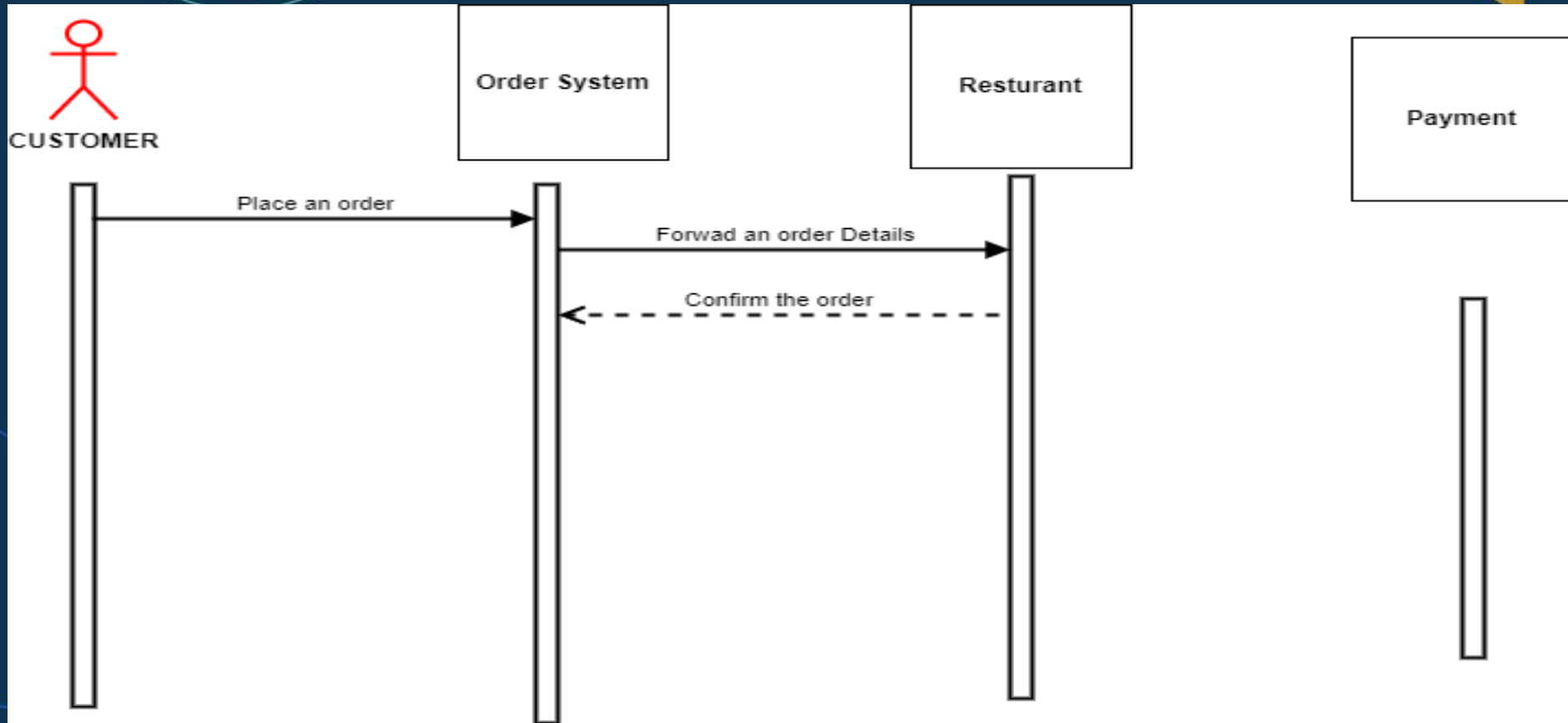




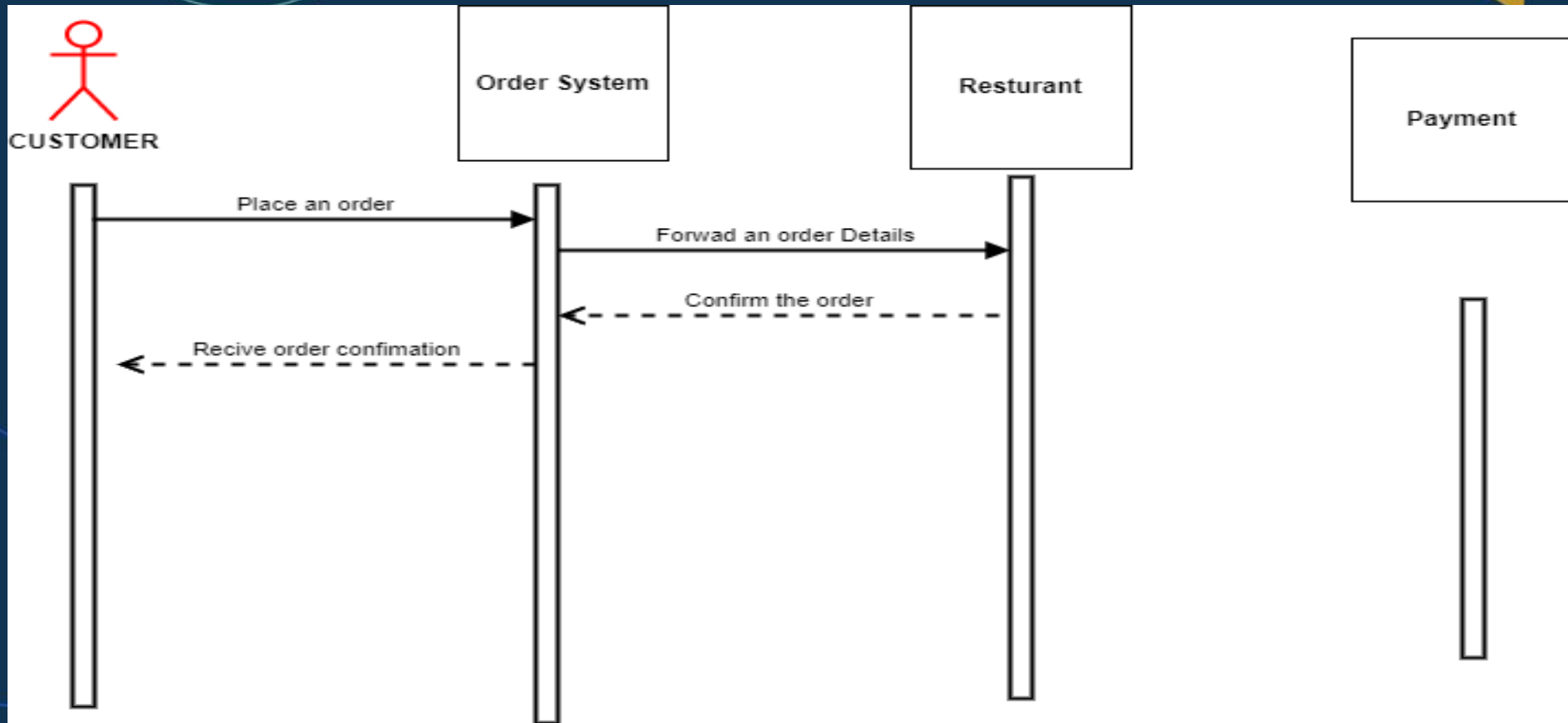
# Sequence Diagram



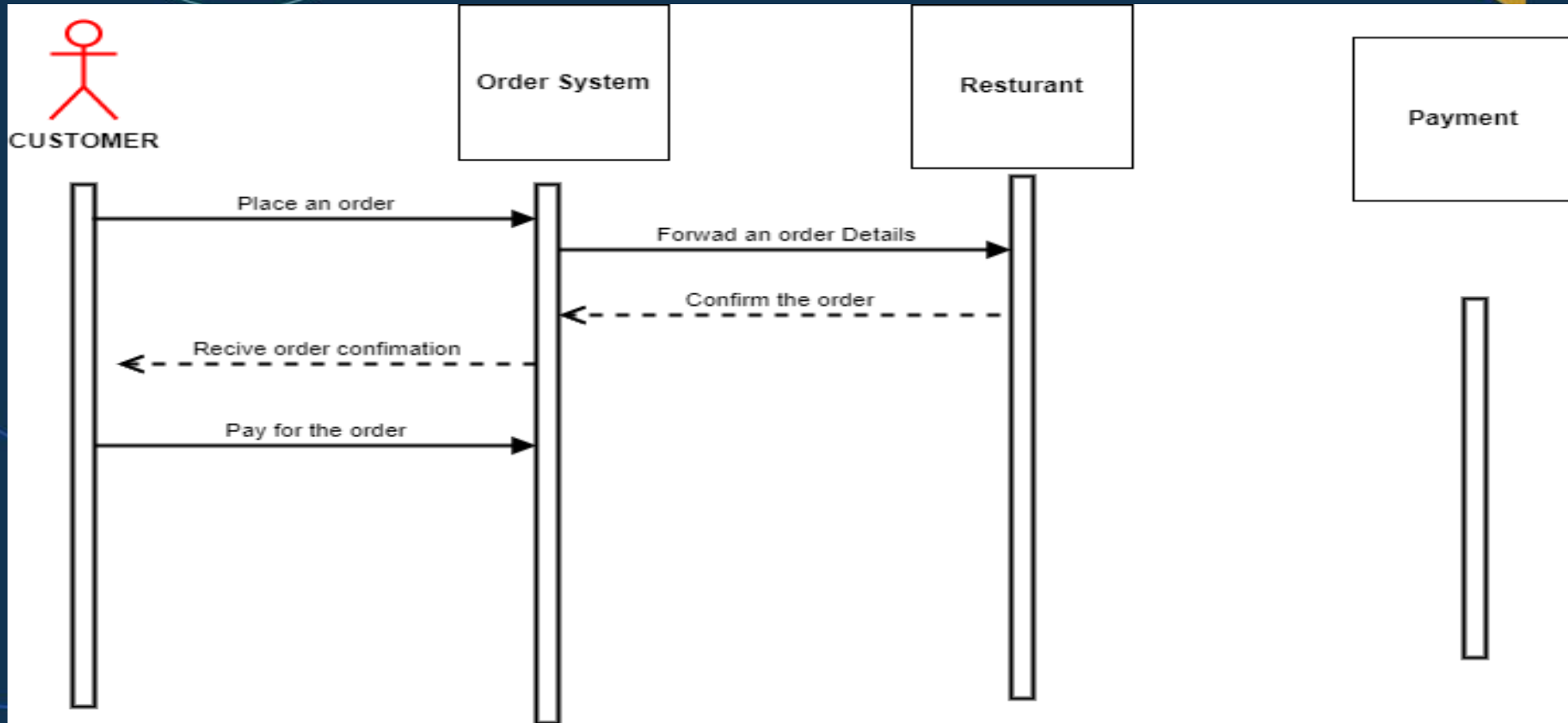
# Sequence Diagram



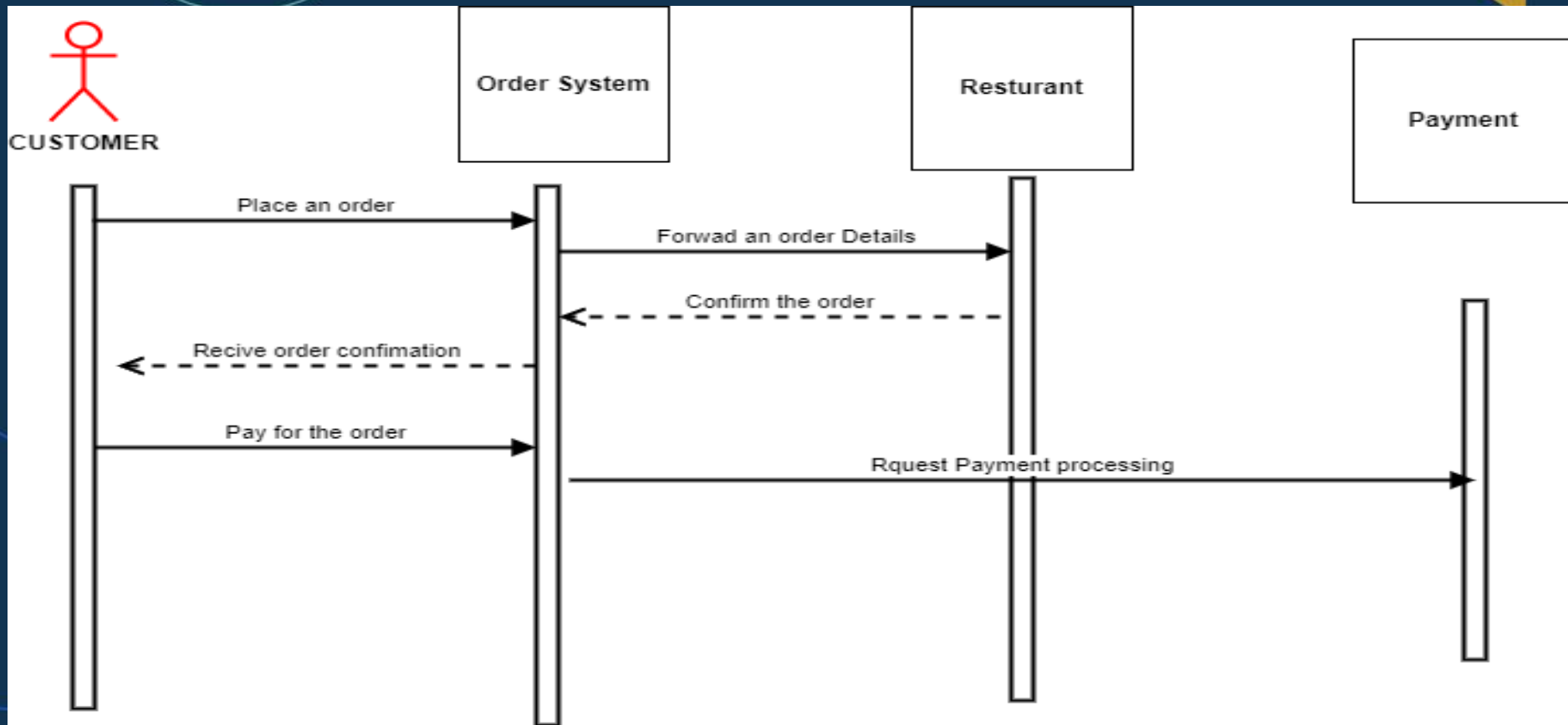
# Sequence Diagram



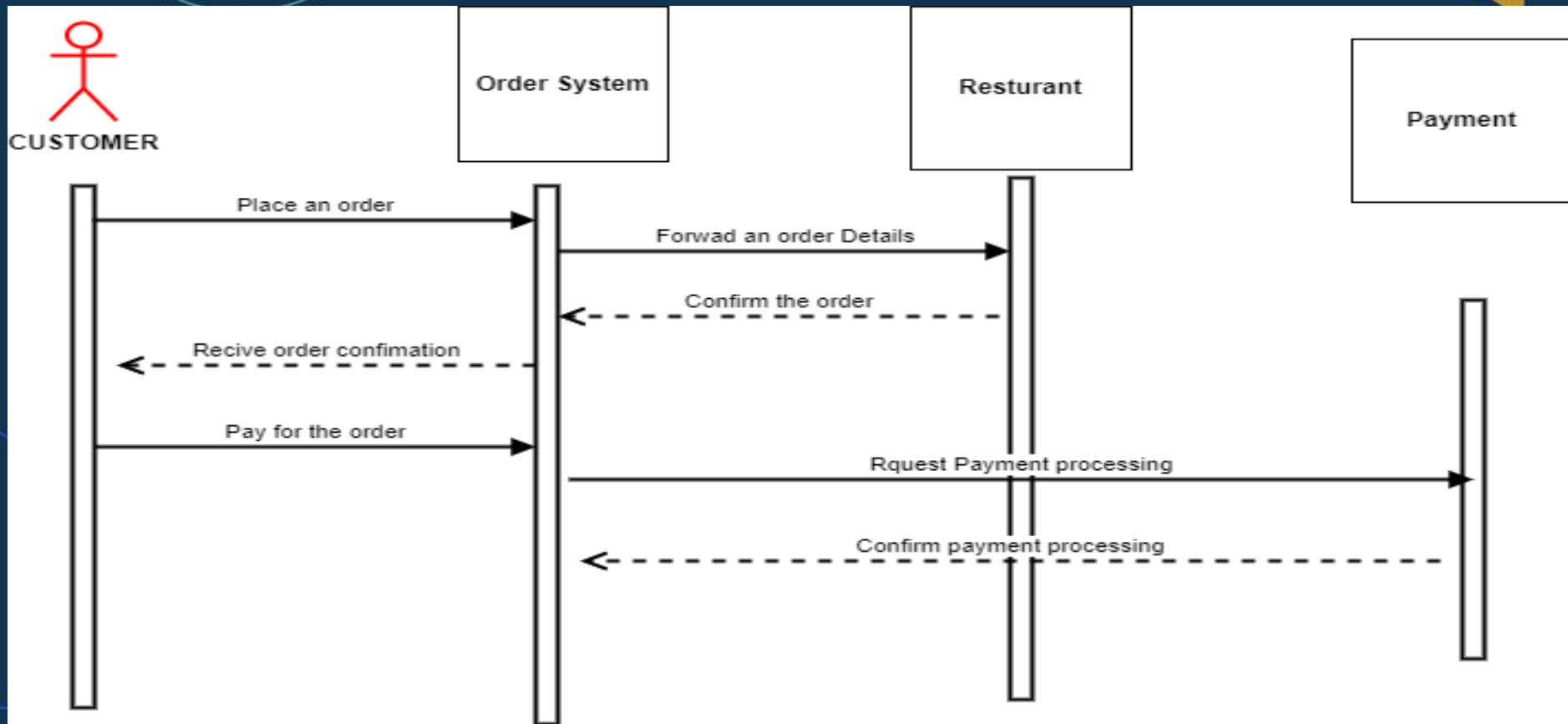
# Sequence Diagram



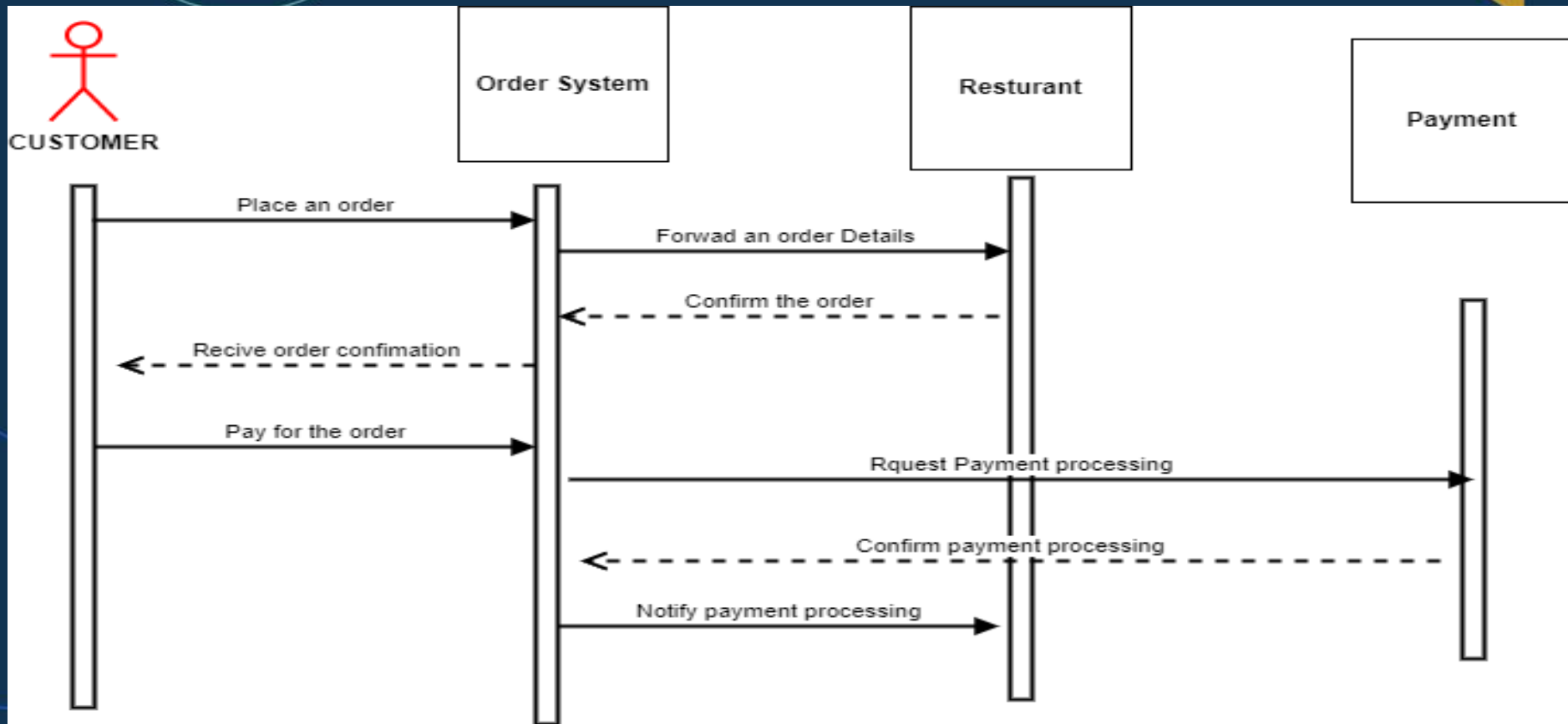
# Sequence Diagram



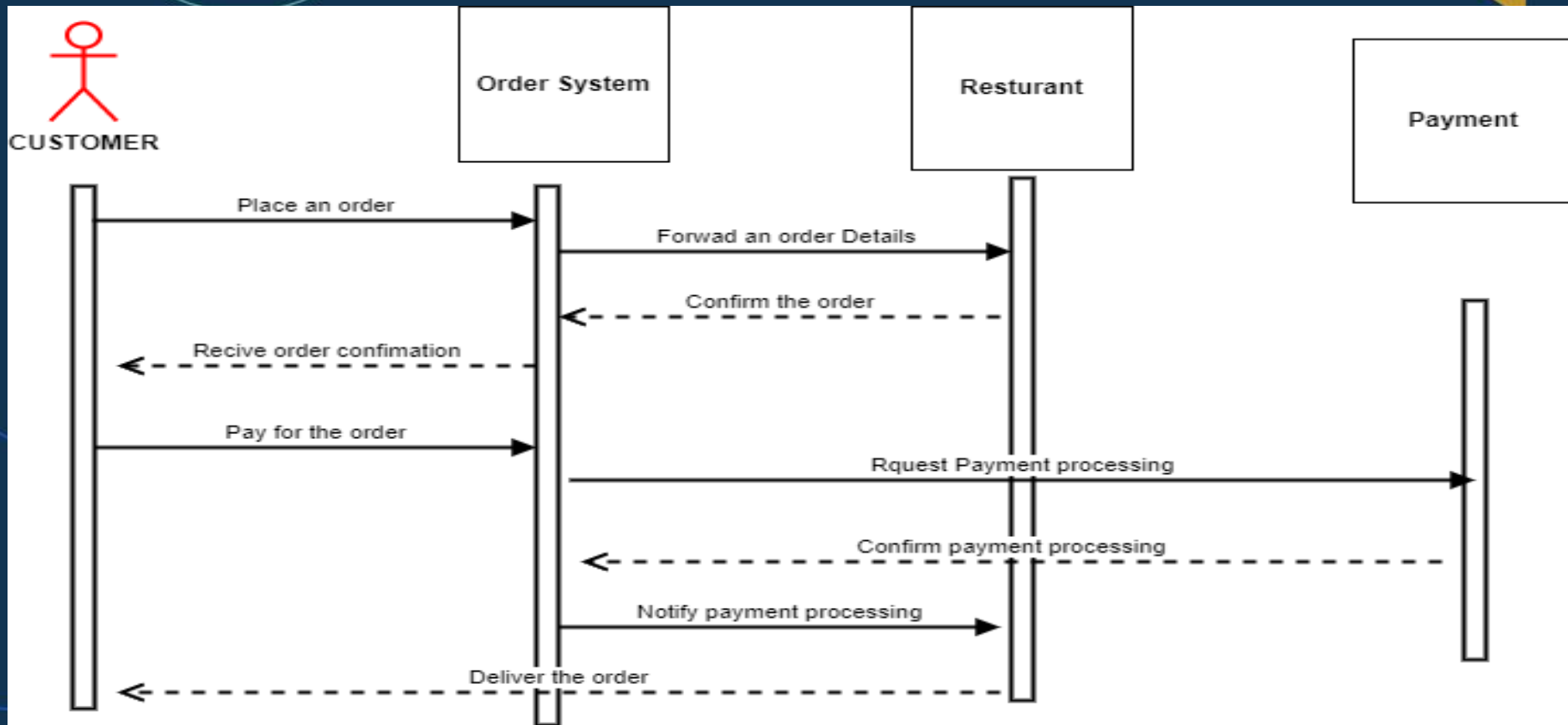
# Sequence Diagram



# Sequence Diagram



# Sequence Diagram



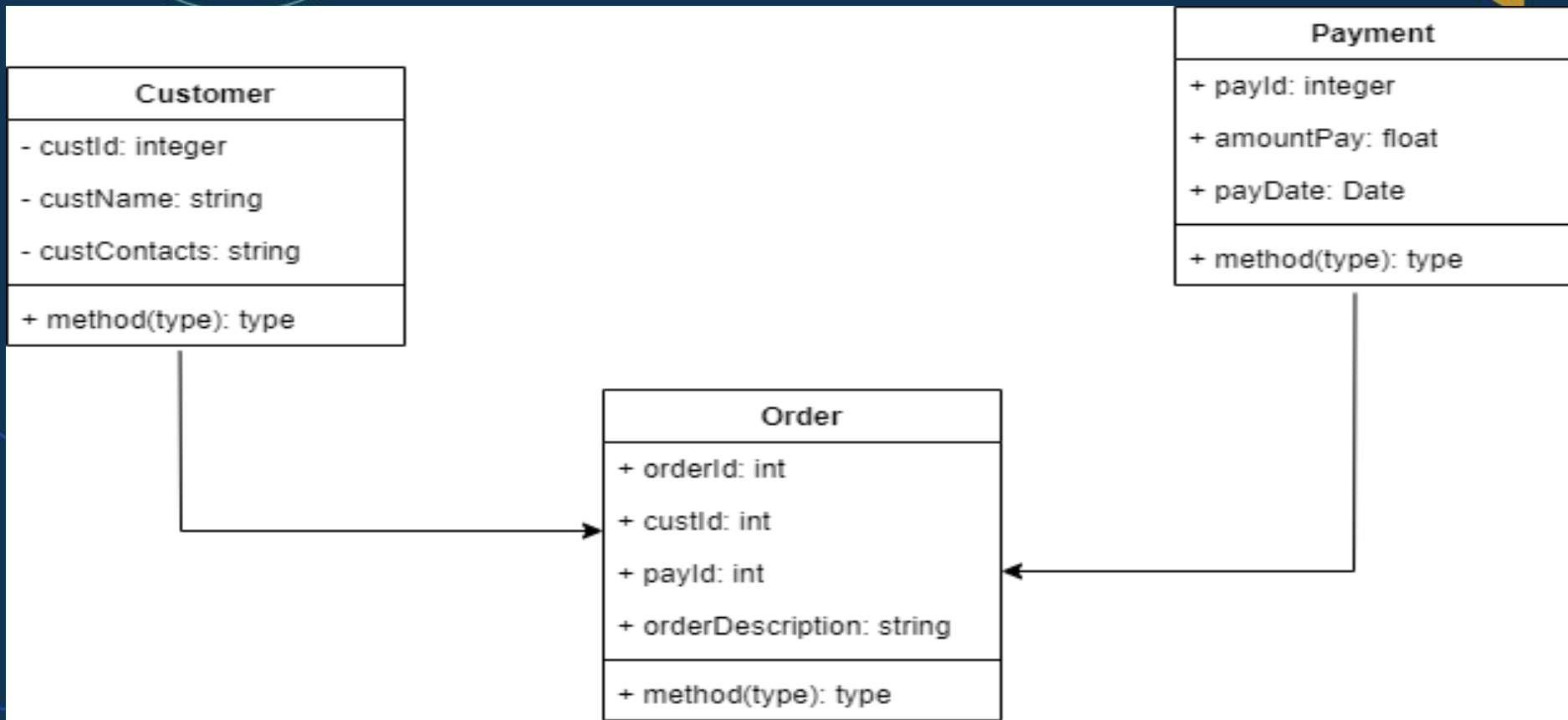


# CLASS DIAGRAM

# Software Design

- **Class Diagram:** Shows the static structure of the system, focusing on the classes, their attributes, methods, and relationships.

# Class Diagram



# Class Diagram

Customer

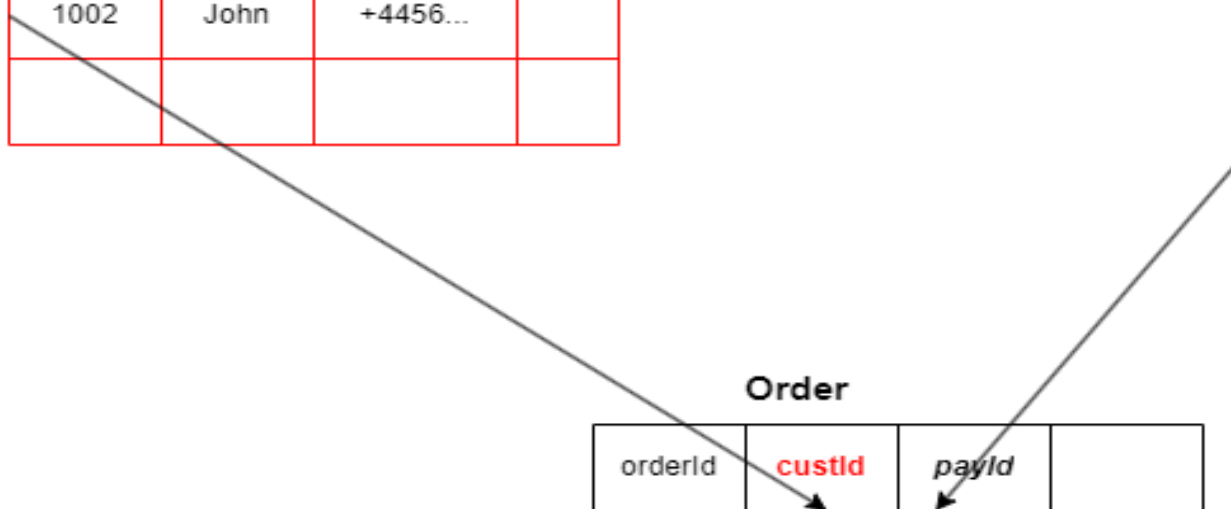
<b>custId</b>	custName	custContacts	
1002	John	+4456...	

Payment

<i>payId</i>	amountPay		
130	156		
106	250		

Order

orderId	<b>custId</b>	<i>payId</i>	
34	1002	106	



# Software Design

- MVC Components:
  - Model: Represents the data and logic
  - View: Displays the user interface
  - Controller: Handles user input and updates the model and view
- CRUD Matrices:
  - Purpose: Identifying the operations (Create, Read, Update, Delete) needed for each entity

# Questions and Answers

