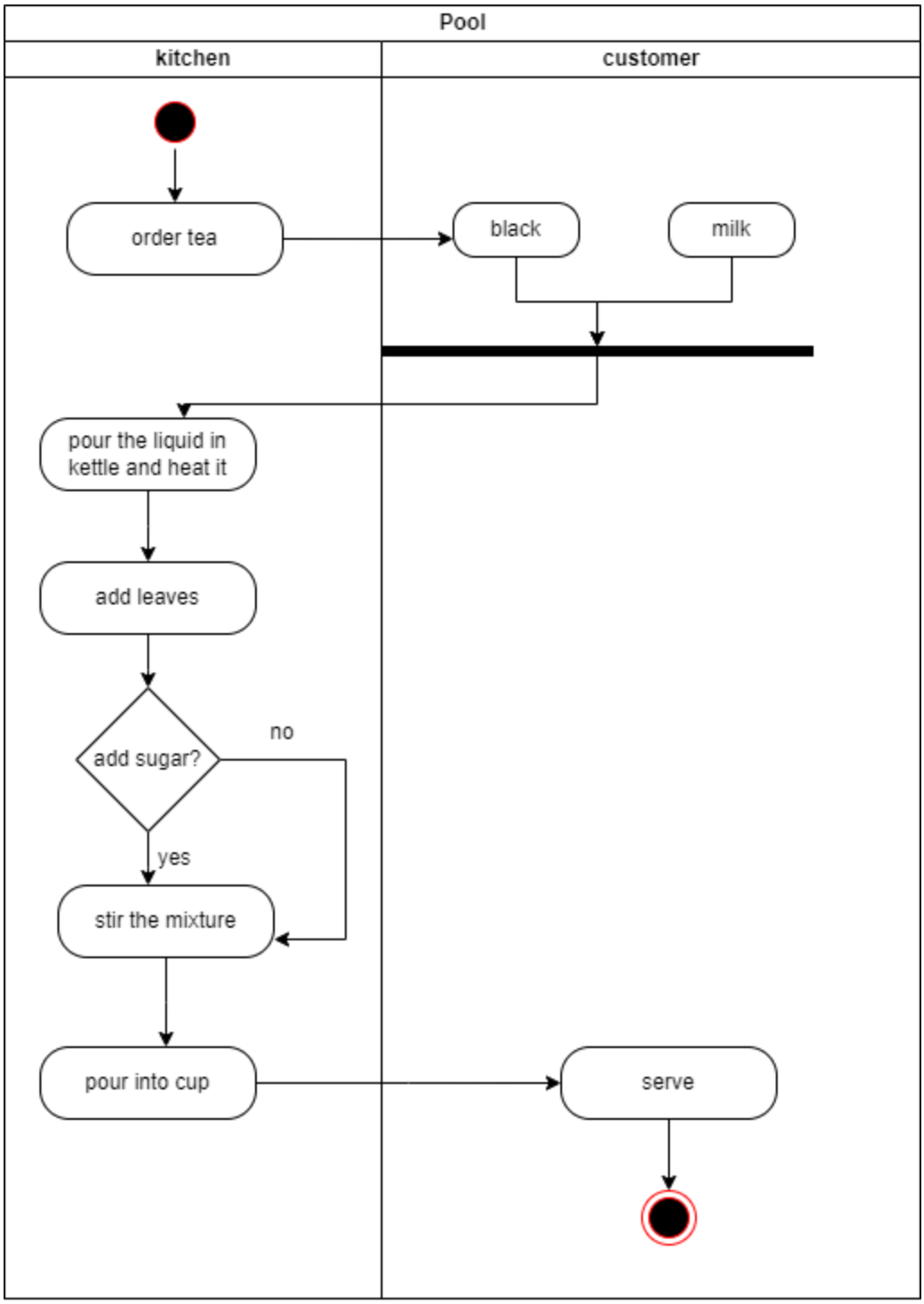
UML Activity Diagram

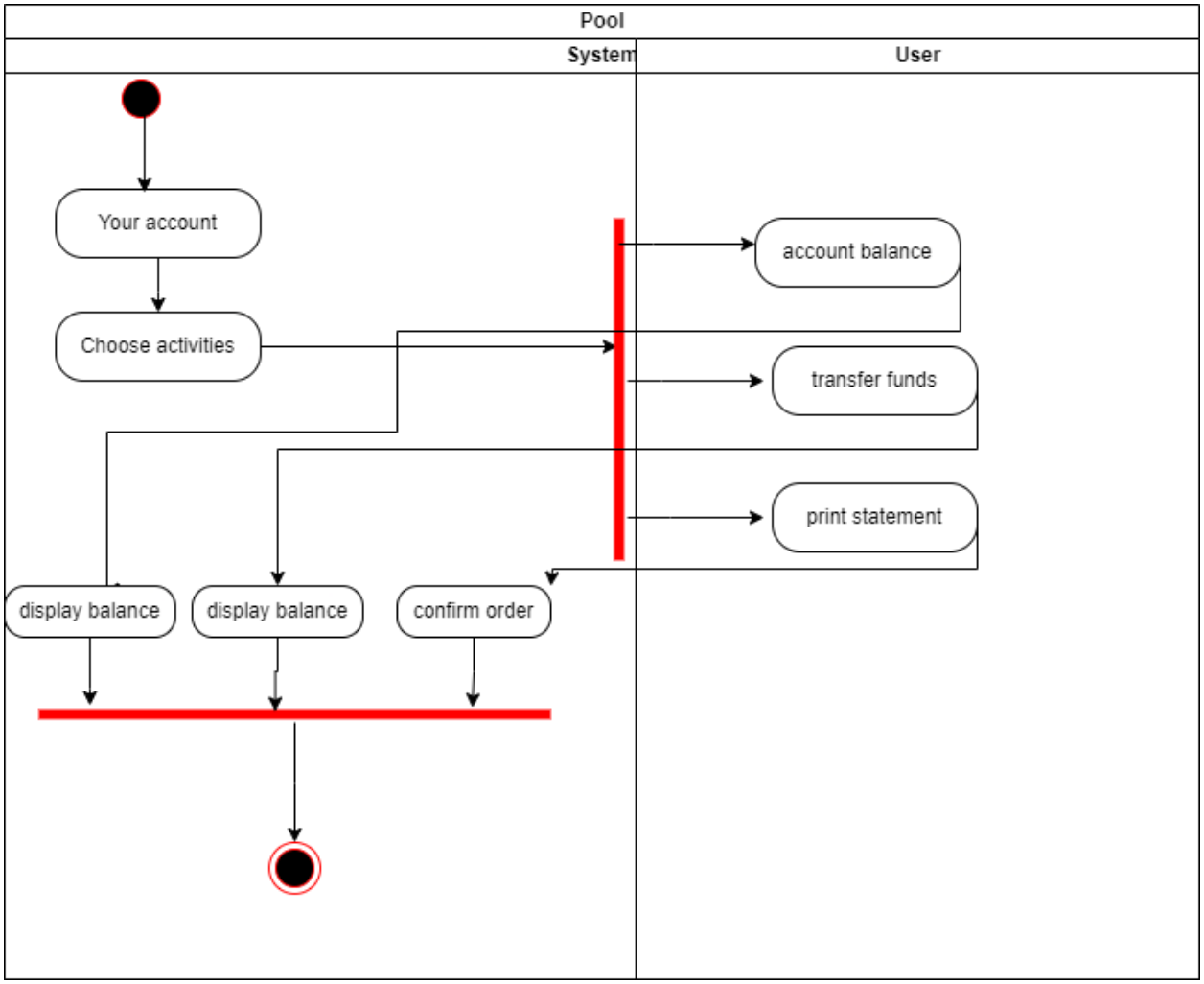
1. Create a UML activity diagram for the process of making a cup of tea, including the steps of boiling water, steeping tea leaves, adding milk and sugar (optional), and serving.



1. The process starts.
2. The water is boiled.
3. The tea leaves are steeped in the boiled water.
4. The system checks if the user wants to add milk and sugar to the tea.

5. If the user chooses not to add milk and sugar, the tea is served.
6. If the user chooses to customize the tea, they can add milk and sugar.
7. After customization, the tea is served.
8. The process ends.

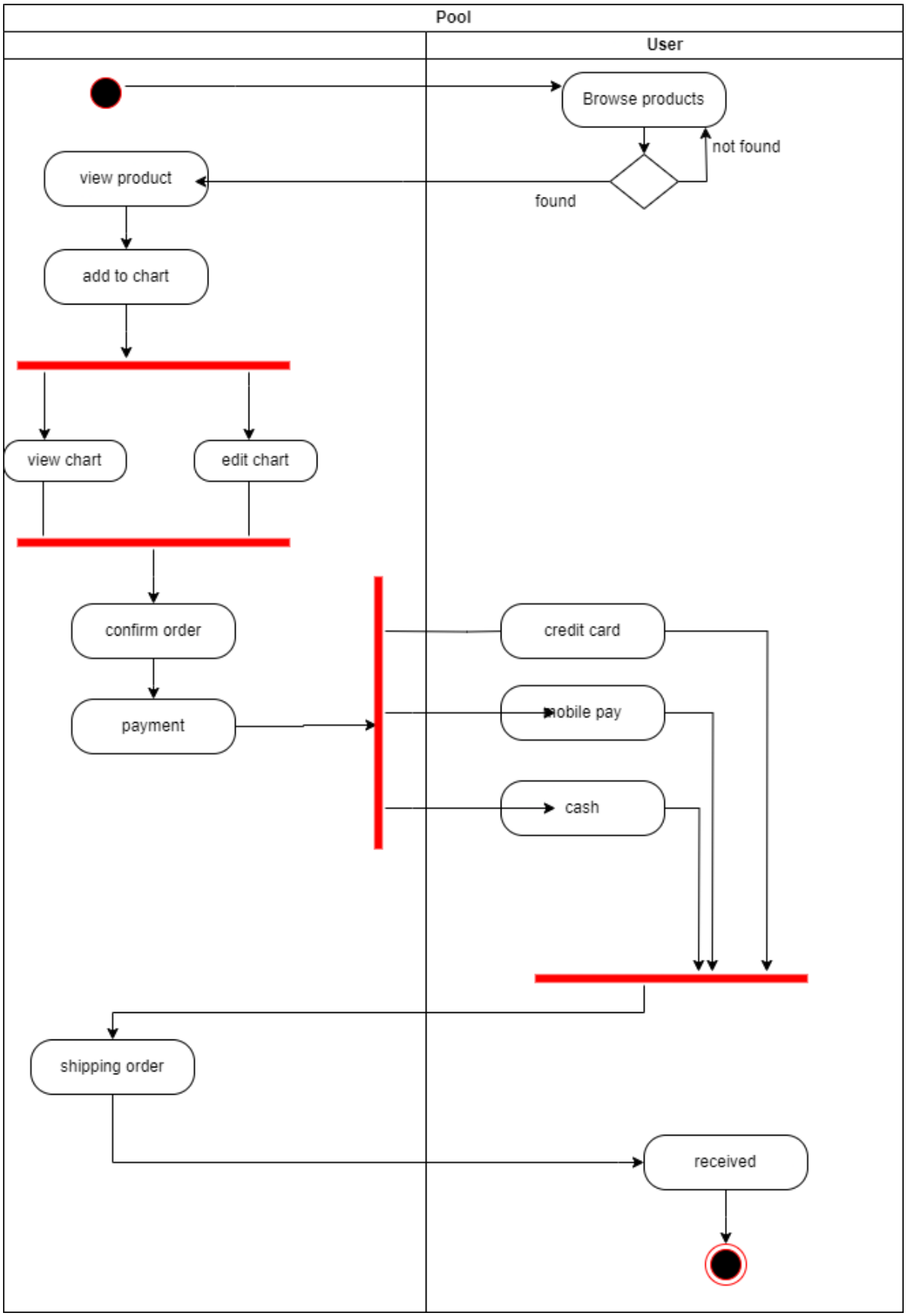
2. Design a UML activity diagram for a banking system, including activities such as checking an account balance, transferring funds between accounts, and printing a bank statement.



1. The process starts.
2. The account balance is checked.

3. The system performs the activity of checking the account balance.
4. After checking the account balance, the system proceeds to the activity of transferring funds between accounts.
5. The system performs the activity of transferring funds.
6. Following the fund transfer, the system moves to the activity of printing a bank statement.
7. The system performs the activity of printing the bank statement.
8. The process ends.

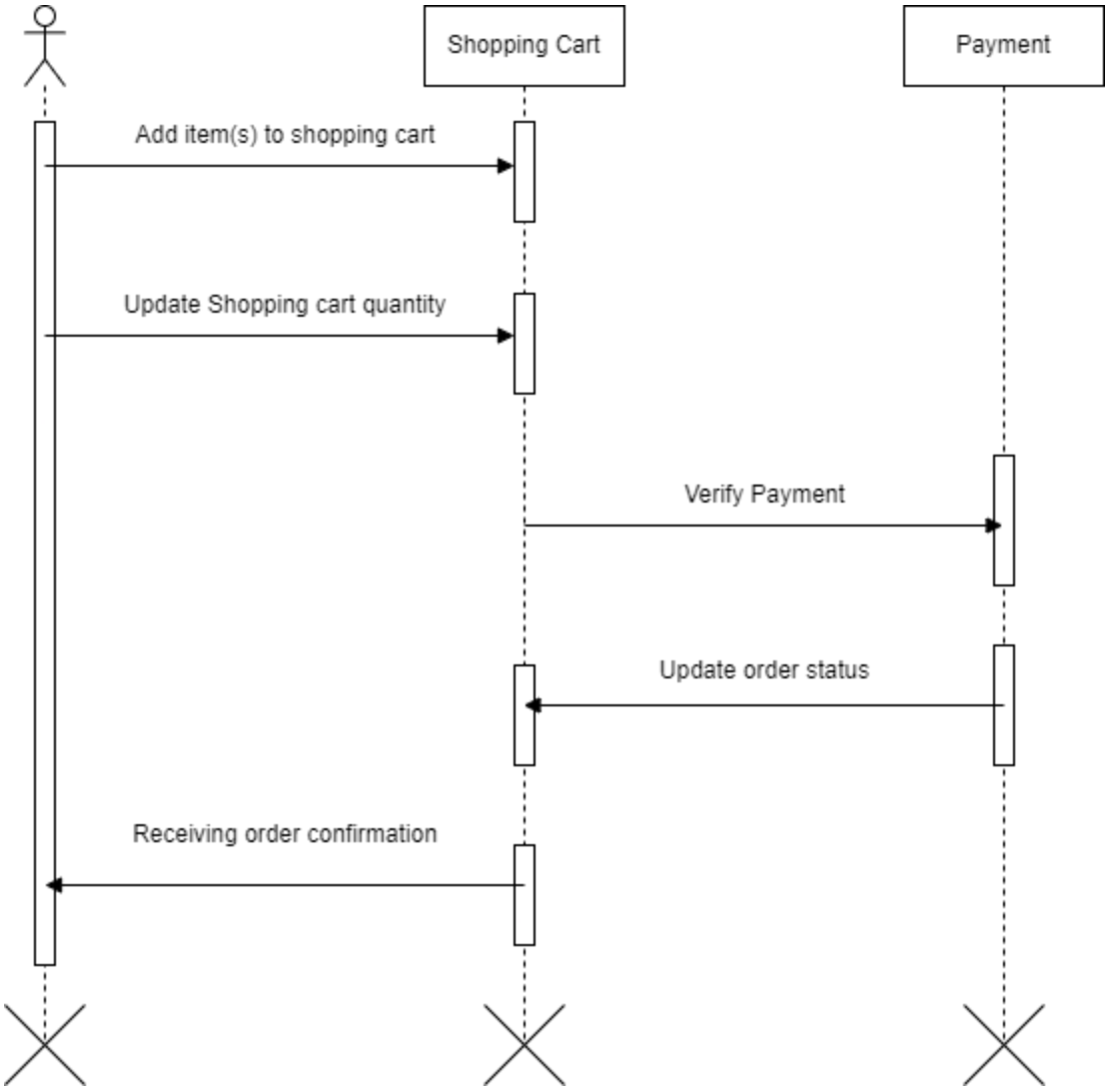
3. Create a UML activity diagram for an online shopping system. Include activities such as browsing products, adding items to the cart, proceeding to checkout, entering shipping and payment details, and completing the purchase. Consider incorporating loops for product selection and decision points for payment methods.


Pool
User
Browse products
not found
found
view product
add to chart
view chart
edit chart
confirm order
payment
credit card
mobile pay
cash
shipping order
received

This activity diagram showcases the flow of activities in an online shopping system, incorporating browsing products, adding items to the cart, proceeding to checkout, entering shipping and payment details, and completing the purchase. It includes loops for product selection and decision points for payment methods.

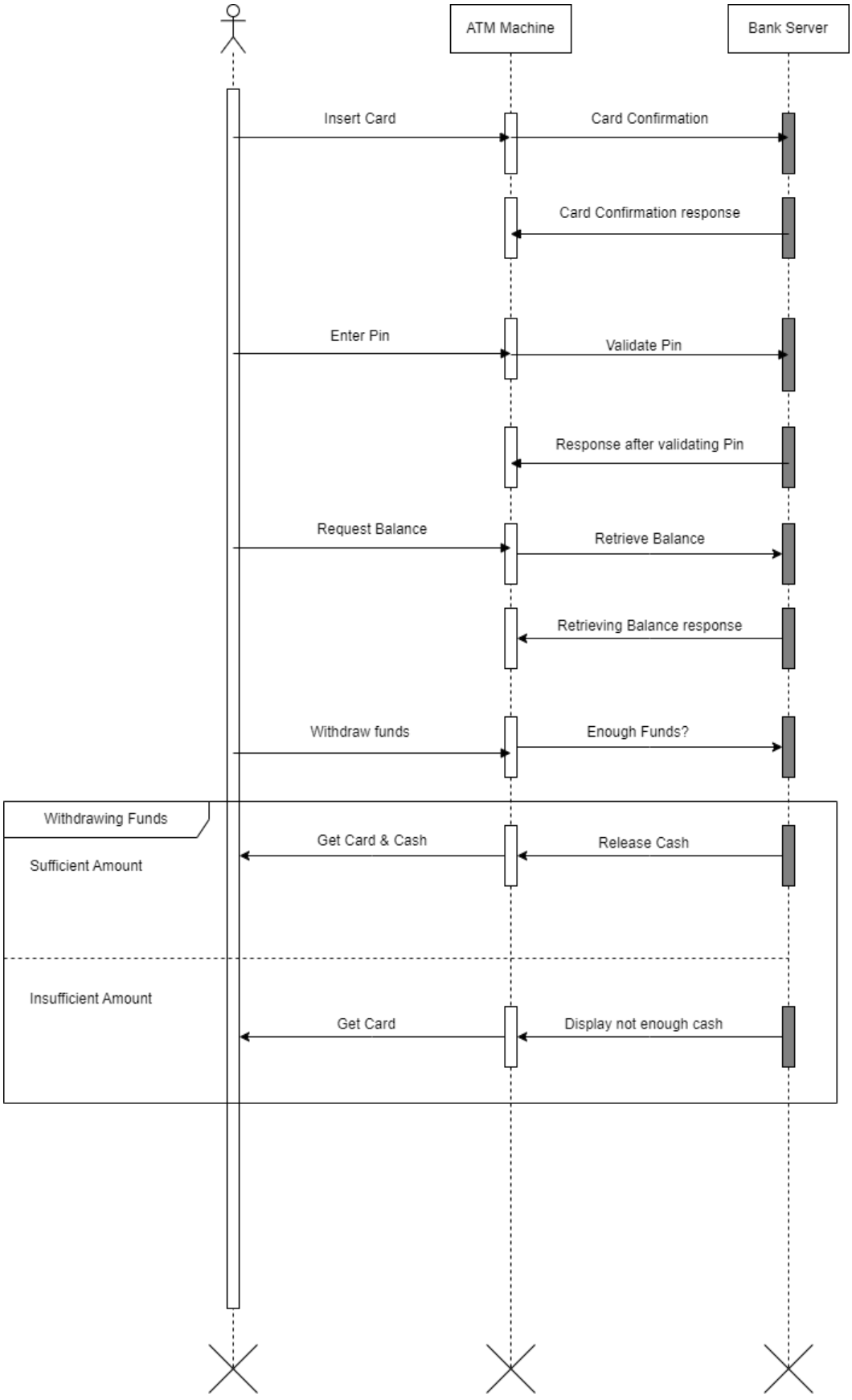
UML Sequence Diagram

1. Create a UML sequence diagram for a simple online shopping process, including interactions between a customer, a shopping cart, and a payment gateway. Include messages for adding items to the cart, updating quantities, processing payment, and receiving order confirmation.



This sequence diagram provides a visual representation of the sequential interactions between the customer, shopping cart, payment gateway, and payment processor during a simple online shopping process, including adding items to the cart, updating quantities, processing payment, and receiving order confirmation.

2. Design a UML sequence diagram for a banking system, including interactions between a customer, an ATM machine, and a bank server. Include messages for entering a PIN, requesting a balance, withdrawing funds, and receiving transaction confirmation.


ATM Machine
Bank Server
Insert Card
Card Confirmation
Card Confirmation response
Enter Pin
Validate Pin
Response after validating Pin
Request Balance
Retrieve Balance
Retrieving Balance response
Withdraw funds
Enough Funds?
Withdrawing Funds
Sufficient Amount
Get Card & Cash
Release Cash
Insufficient Amount
Get Card
Display not enough cash

This sequence diagram provides a visual representation of the sequential interactions between the customer, ATM machine, and bank server in a banking system. It includes entering a PIN, requesting a balance, withdrawing funds, and receiving transaction confirmation.

UML Deployment Diagram

1. What is a UML deployment diagram, and what is its purpose?

Ans: A UML deployment diagram is a type of diagram that represents the physical deployment of software artifacts to hardware nodes. The purpose of the diagram is to describe how software is deployed into the hardware system and how software interacts with the hardware to execute the complete functionality

2. What are the main components or elements in a UML deployment diagram?

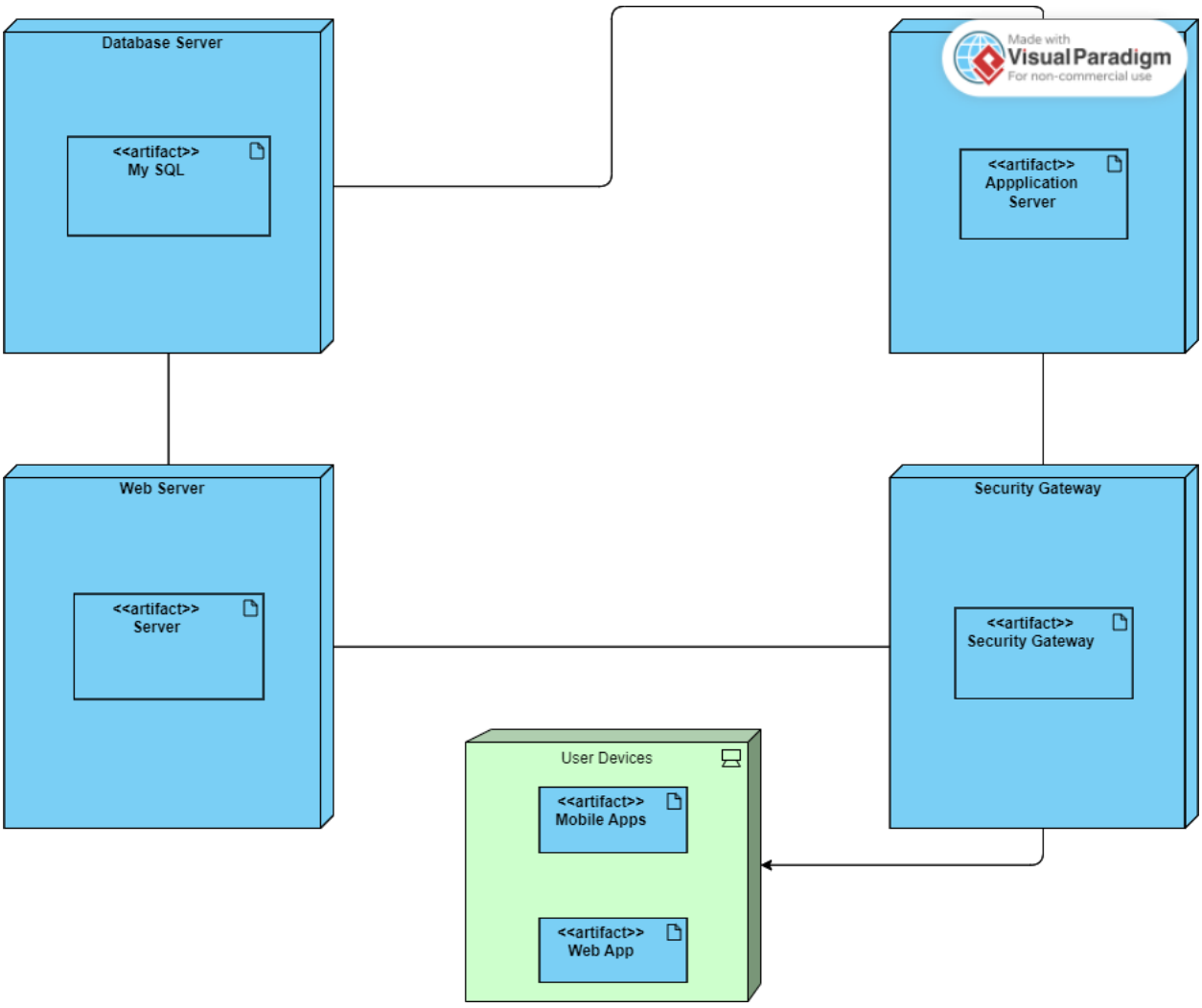
Ans: The main components or elements are: Nodes, Components, Artifacts and Associations.

3. How do you represent communication between nodes in a UML deployment diagram?

   Ans: In a UML deployment diagram, communication between nodes is represented using a communication path or association relationship.

4. You are working on a UML deployment diagram for an online banking system. The system consists of multiple components, including web servers, application servers, a database server, a security gateway, and user devices. The web servers handle user requests and communicate with the application servers for processing transactions. The database server stores user account information and

transaction data. The security gateway ensures secure communication between the components and performs authentication and encryption. User devices include desktop computers, smartphones, and tablets, each running a different banking application. Design a comprehensive UML deployment diagram that showcases the deployment of these components, considering load balancing, high availability, and data privacy.



In the online banking system, the web servers act as the initial point of contact for users, handling their requests and serving them with the web-based interface. They establish communication with the application servers using the HTTP/HTTPS protocols, allowing the flow of user requests and responses. The application servers, responsible for

processing transactions and implementing the core logic of the system, interact with the database server to access and update user account information and transaction data. This interaction takes place using the JDBC protocol, enabling seamless communication between the application servers and the database. To ensure secure communication and protect sensitive information, a security gateway is in place. The security gateway acts as a protective barrier, enabling secure HTTPS communication between the web servers and application servers, ensuring data privacy and establishing a secure channel for the system's smooth functioning. Ultimately, this deployment architecture facilitates efficient communication and collaboration between the different components, prioritizing security and data integrity in the online banking system.