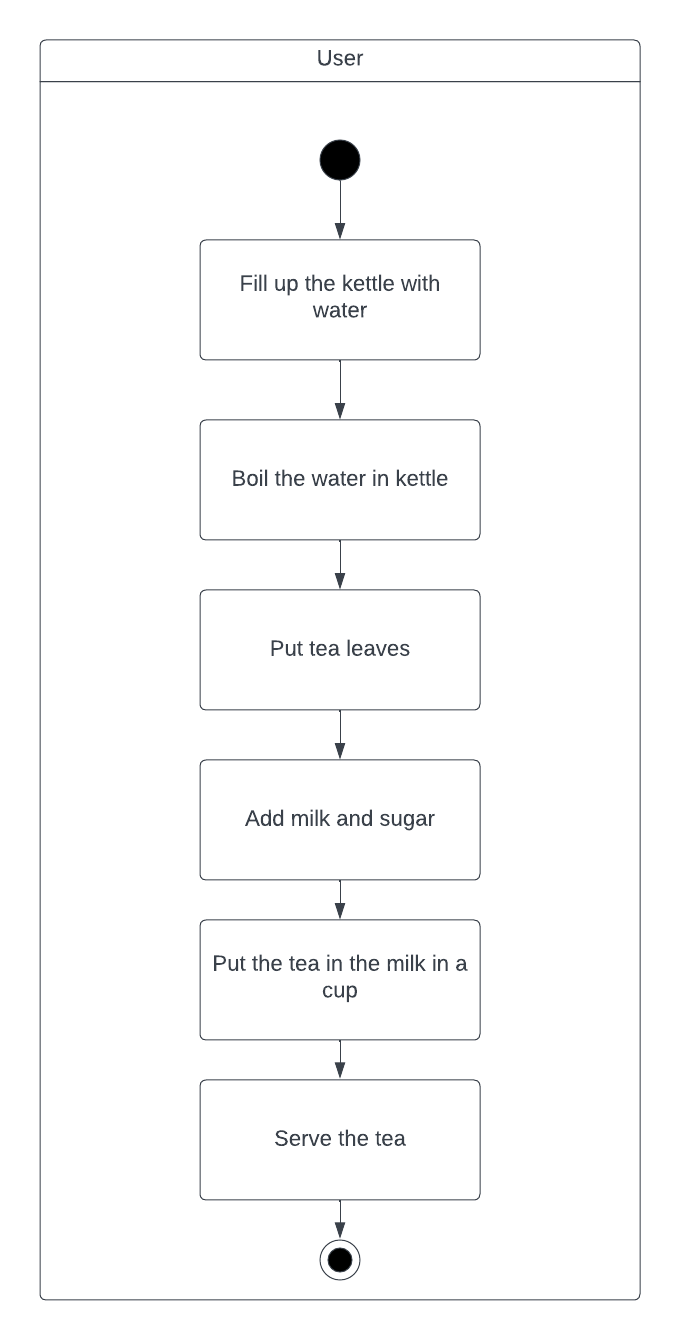
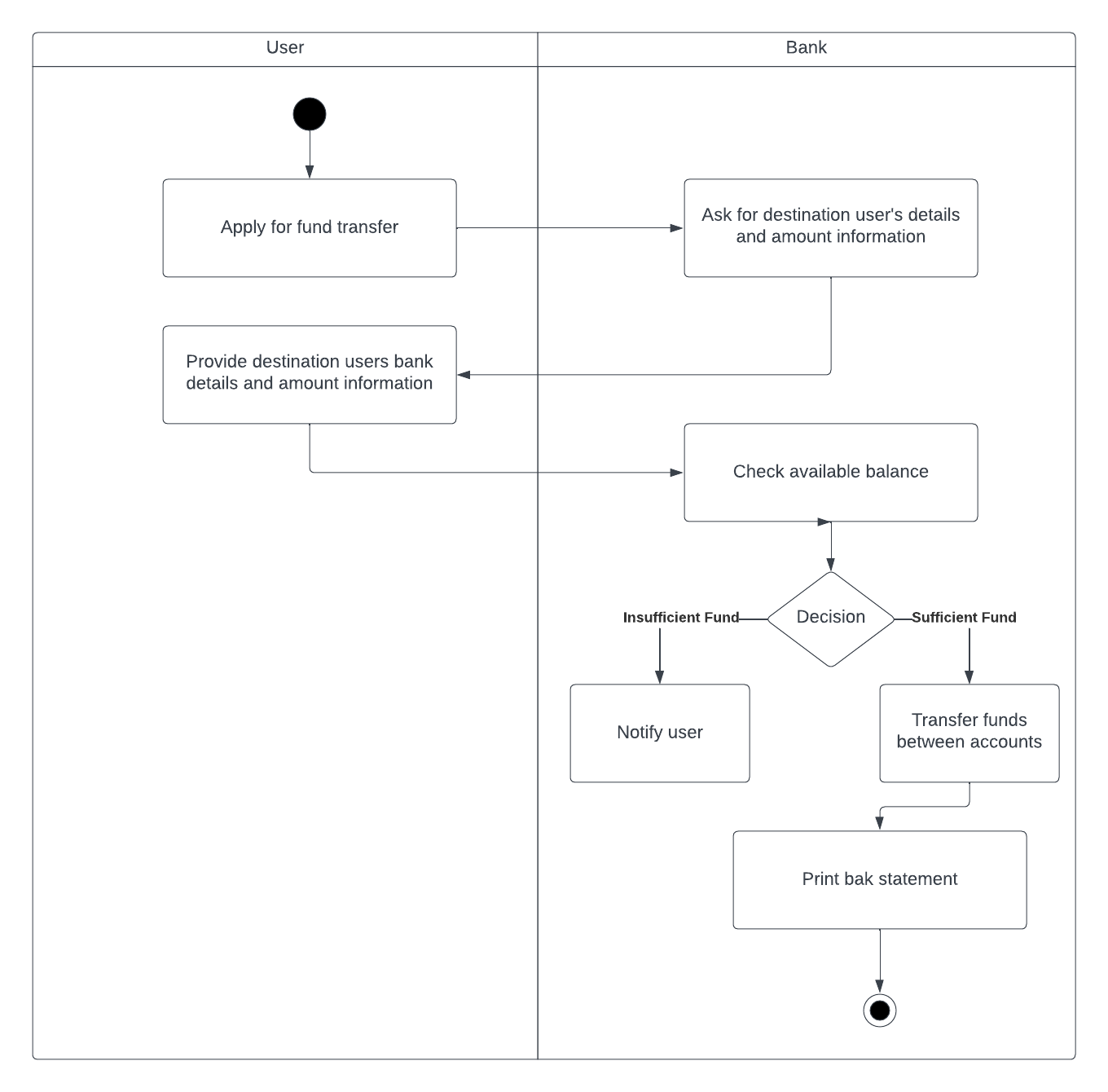
UML Activity Diagram

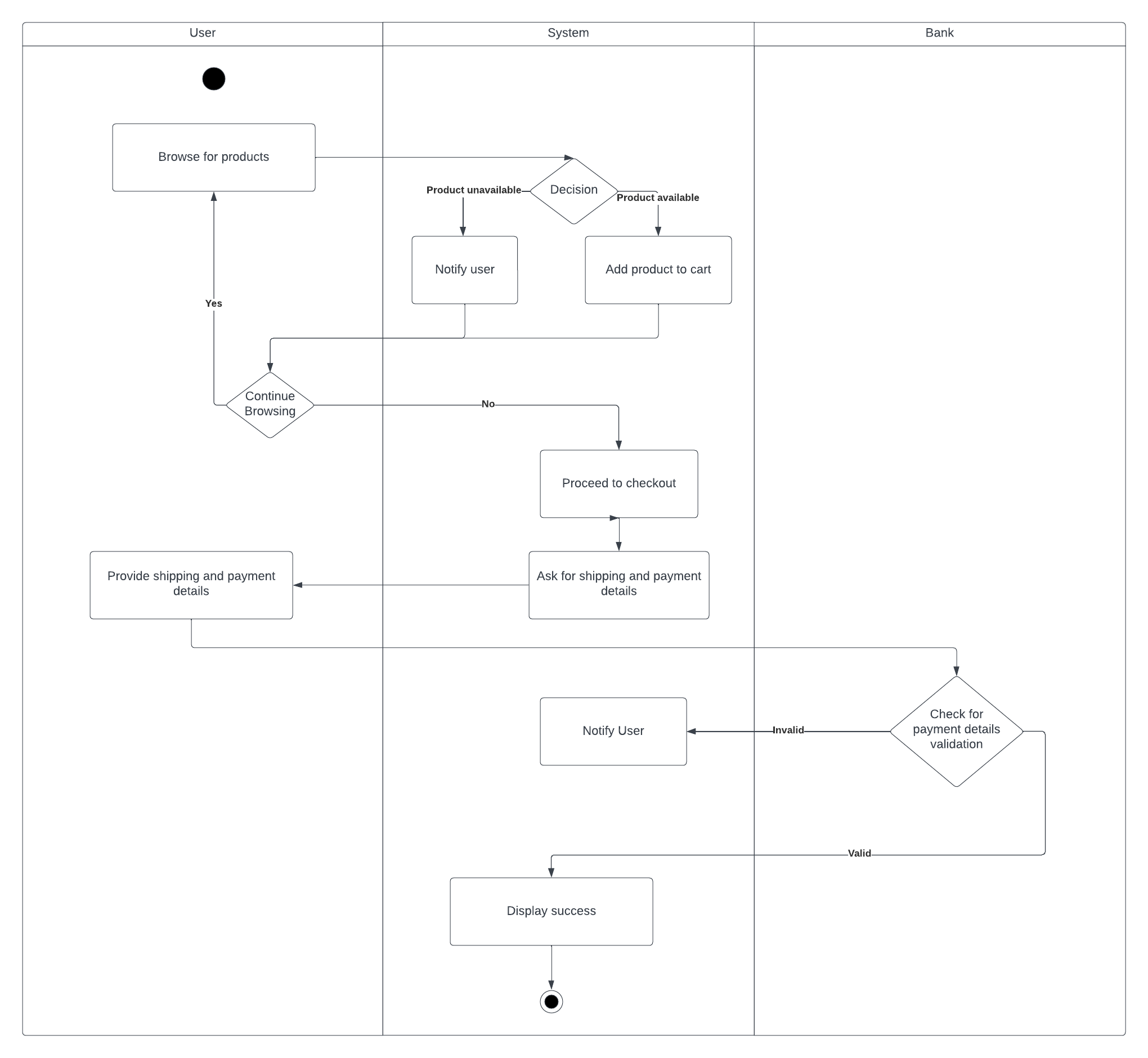
1. Create a UML activity diagram for the process of making a cup of tea, including the steps of boiling water, steeping tea leaves, adding milk and sugar (optional), and serving.



1. 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.

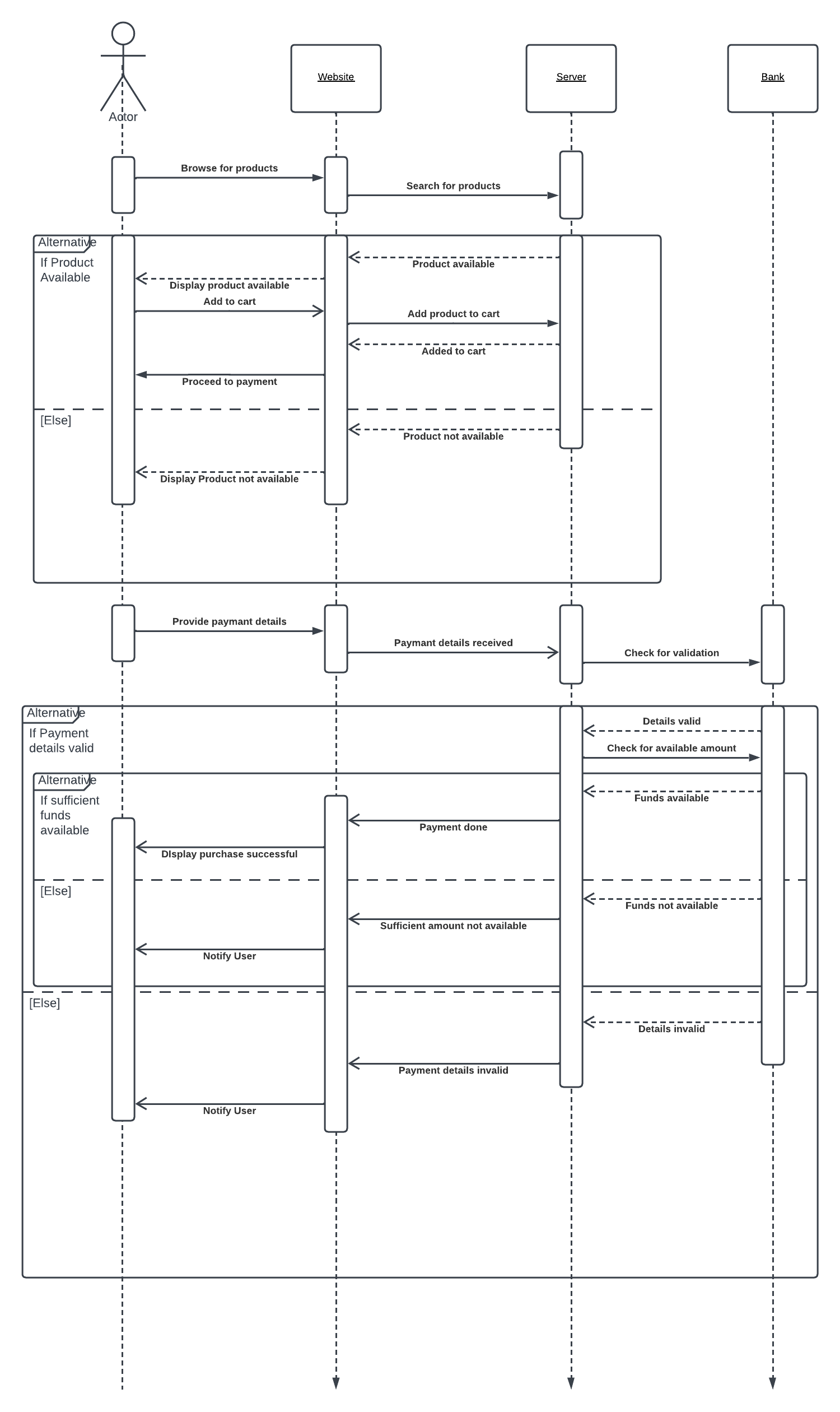


1. Create a UML activity diagram for an online shopping system. Include activities such as browsing products, adding items to the cart, proceeding to checkout, entering shipping and payment details, and completing the purchase. Consider incorporating loops for product selection and decision points for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

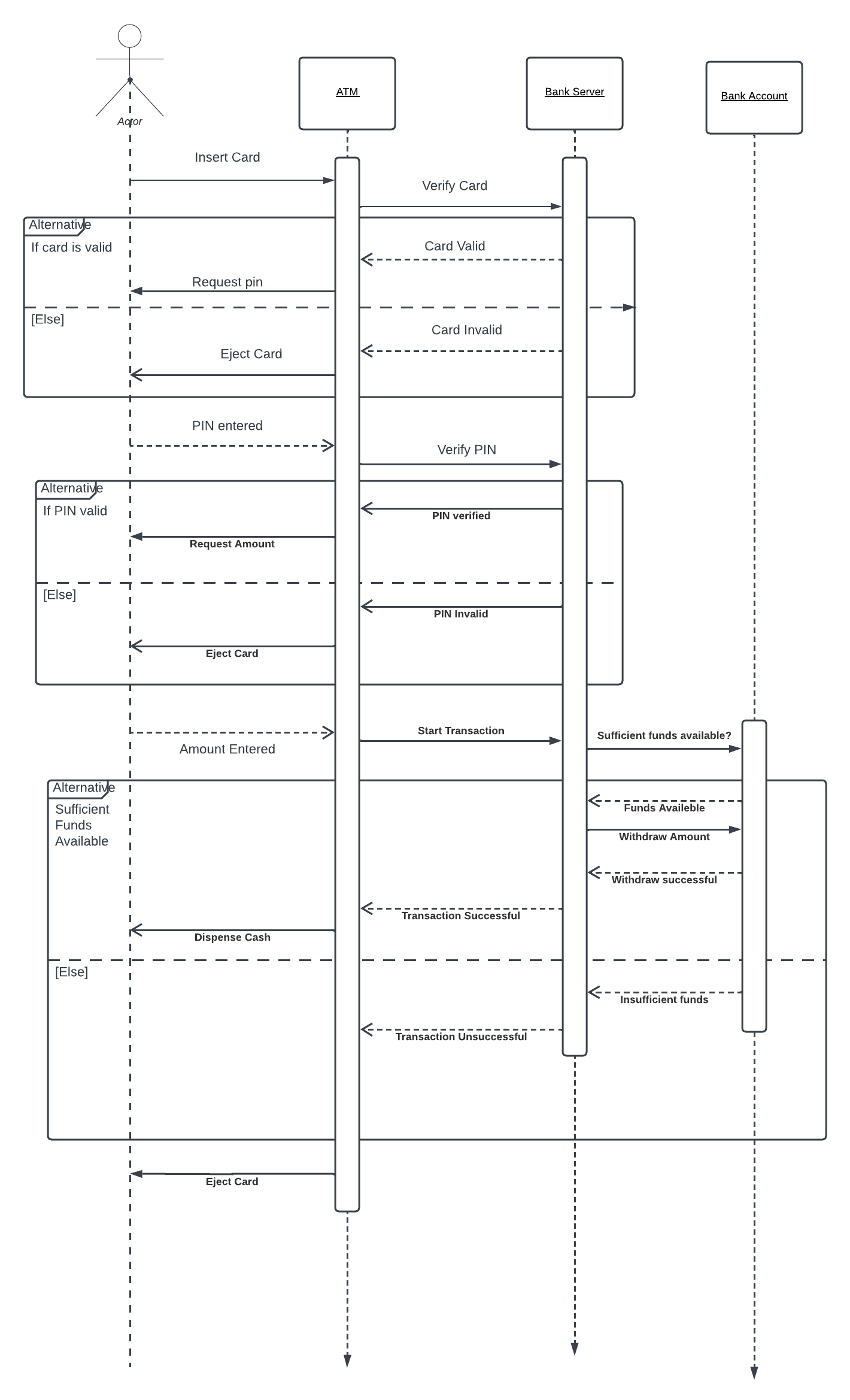


UML Sequence Diagram

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.



1. 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.



UML Deployment Diagram

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

**Answer:** A UML deployment diagram is a type of structural diagram in the Unified Modeling Language (UML) that depicts the physical components of a system and the relationships between them. It shows the hardware and software components of a system, as well as the connections between them. Deployment diagrams are used to:

Visualize the physical structure of a system. This can be helpful for understanding how the system will be deployed and for identifying potential bottlenecks or areas of vulnerability.

Communicate the system architecture to stakeholders. Deployment diagrams can be used to show stakeholders, such as developers, managers, and customers, how the system is physically structured. This can help to ensure that everyone is on the same page and that the system can be developed and deployed successfully.

Document the system architecture. Deployment diagrams can be used to document the system architecture for future reference. This can be helpful for troubleshooting problems, making changes to the system, or porting the system to new hardware or software.

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

**Answer:**

The main components or elements in a UML deployment diagram are:

Nodes: Nodes represent physical components of a system, such as computers, servers, and network devices.

Artifacts: Artifacts represent software components that are deployed on nodes, such as executables, libraries, and data files.

Connections: Connections represent the communication links between nodes.

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

**Answer:**

Communication between nodes in a UML deployment diagram is represented using a communication association. A communication association is a line that connects two nodes and shows the path of communication between them. The communication association can be labeled with the name of the protocol or network that is used for communication.

Communication associations can be used to show the communication links between any two nodes in a deployment diagram. This can be helpful for visualizing the flow of data and messages between different components of a system.

1. 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.

