

SOFTWARE ENGINEERING LAB

EXERCISE – 3

TOPIC – 4

UML DIAGRAMS – SEQUENCE

Sequence Diagram

- A sequence diagram is a type of UML diagram that illustrates how objects in a system interact with each other over time.
- It shows the order in which messages are sent between objects, helping visualize the sequence of interactions in a system.
- This is particularly useful when trying to understand the flow of actions or events between different components of a system.

2. Components of a Sequence Diagram

1. Actors:

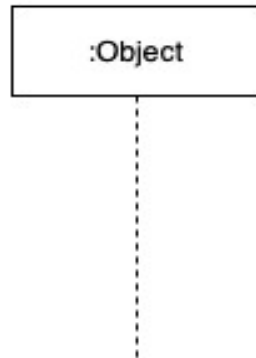
- Actors represent external entities interacting with the system.
- These could be people (like users) or other systems (like a database or another software system).
- In a sequence diagram for an ATM system, for example, the Customer would be an actor.



2. Objects:

- Objects are parts of the system that perform specific actions.
- Each object represents a class or an entity that plays a role in the interaction.

- For example, in an ATM system, objects could include:
 - The ATM machine itself.
 - The Bank Database that stores the customer's account information.



3. Lifelines:

- A Lifeline is a dotted vertical line that extends below each object, representing the object's existence over time.
- The activation box on the lifeline shows when the object is actively performing an action (e.g., processing a transaction).
- For example, if a customer initiates a withdrawal, the ATM machine's lifeline shows the duration during which it processes the request.



4. Messages:

- Messages are arrows between objects indicating the communication between them.
- Each arrow represents a specific action or message being passed from one object to another.



- The arrow shows:
 - Direction of the message (who sends and who receives it).
 - Order of the messages.
- In the ATM system example:
 - The Customer might send a message to the ATM: insertCard().
 - The ATM then sends a message to the Bank Database: authenticateUser().

3. Example: Sequence Diagram for an ATM System

Let's break down how a sequence diagram might look for an ATM system, considering the interactions for a cash withdrawal process.

Actors:

- Customer (the person using the ATM).

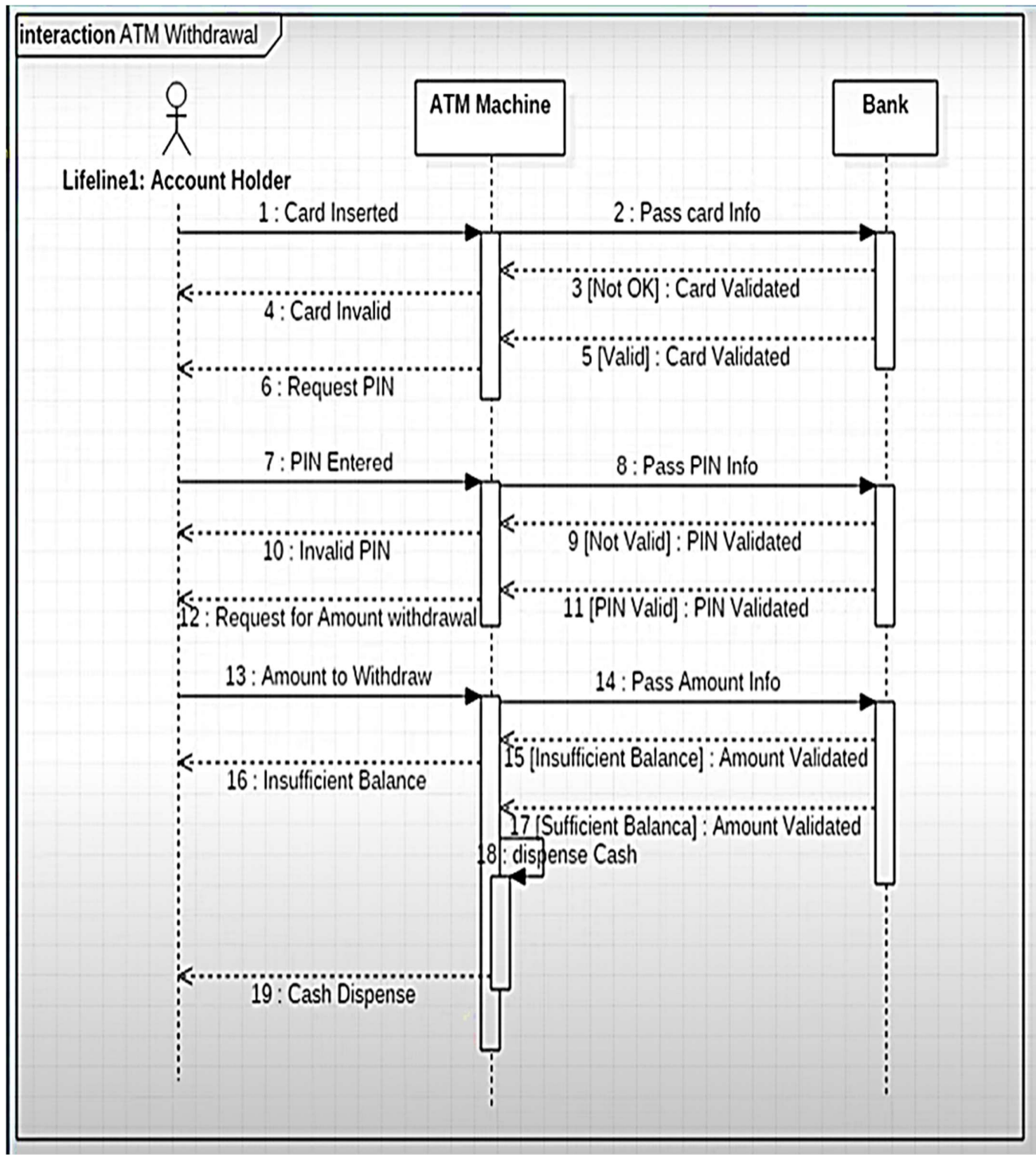
Objects:

- ATM (the machine the customer is interacting with).
- Bank Database (the system that holds the customer's account details).

Messages:

1. The Customer inserts the card into the ATM: insertCard().
2. The ATM requests the PIN from the customer: requestPIN().
3. The Customer enters the PIN: enterPIN().
4. The ATM sends the entered PIN to the Bank Database for verification: authenticateUser().
5. The Bank Database verifies the PIN and sends a response: authenticationResult().
6. If the authentication is successful, the ATM prompts the Customer to enter the withdrawal amount: requestAmount().
7. The Customer enters the withdrawal amount: enterAmount().

8. The ATM sends a request to the Bank Database to withdraw money: `withdrawMoney()`.
9. The Bank Database confirms the transaction and updates the balance: `transactionConfirmation()`.
10. The ATM dispenses the cash and ends the transaction: `dispenseCash()`.



Purpose of Sequence Diagrams

- Sequence diagrams are used to model the dynamic behavior of a system. They help in understanding:
 - How different parts of the system work together to complete a process.
 - The order in which actions or events happen in the system.
 - The flow of messages between different objects over time.

In an ATM system, for example, the sequence diagram helps visualize the step-by-step interaction between the customer, the ATM machine, and the bank's database during a transaction like withdrawing cash. It highlights how the ATM system interacts with the external user and bank database to complete the transaction process smoothly.