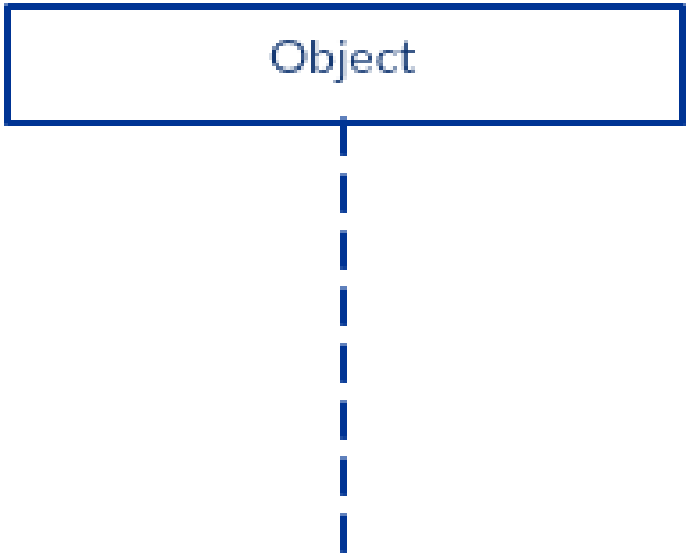
Sequence Diagram

- Sequence diagrams, commonly used by developers, model the interactions between objects in a single use case.
- They illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.

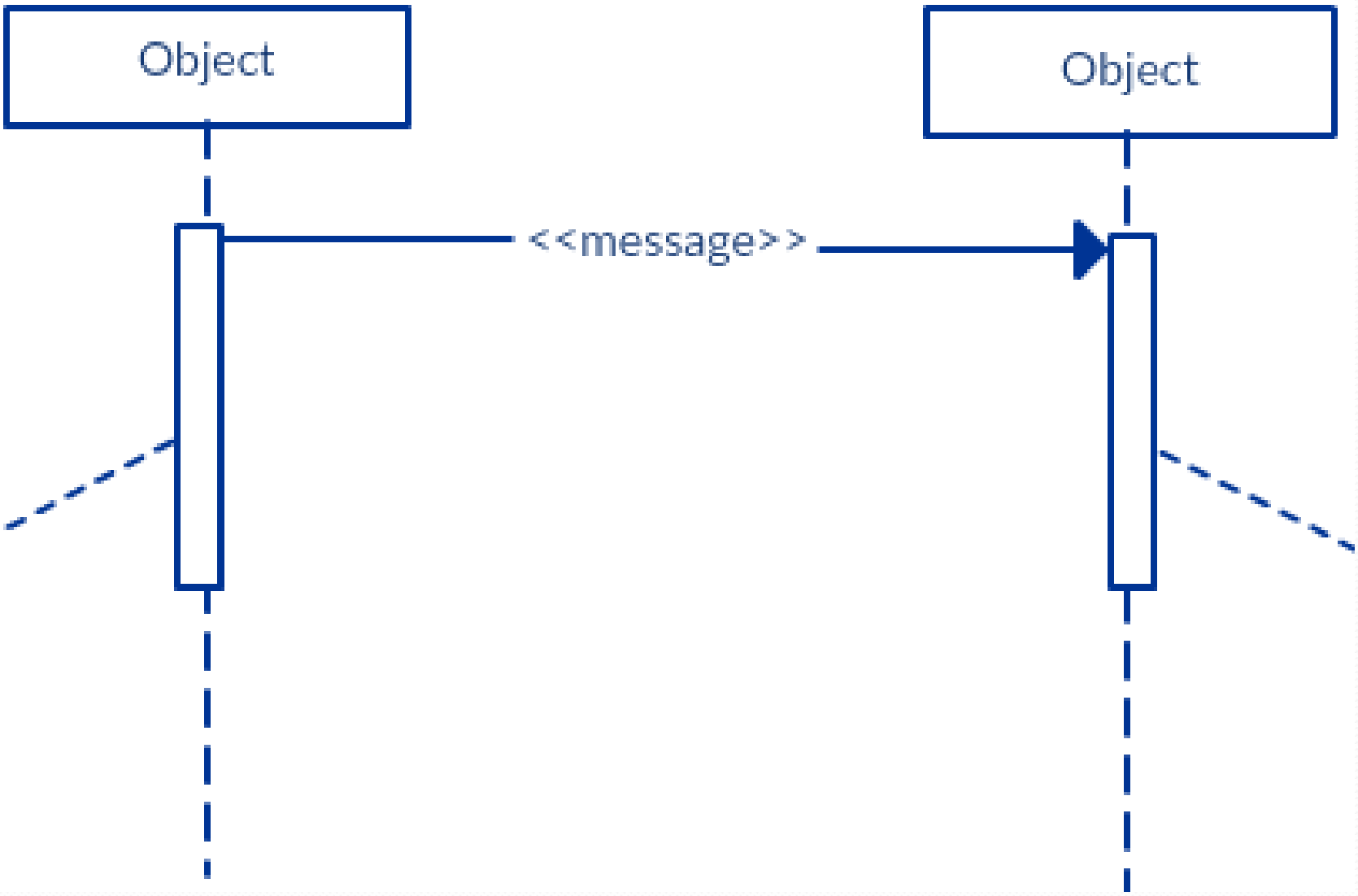
Lifeline Notation

- A sequence diagram is made up of several of these lifeline notations that should be arranged horizontally across the top of the diagram



Activation Bars

- Activation bar is the box placed on the lifeline. It is used to indicate that an object is active (or instantiated) during an interaction between two objects. The length of the rectangle indicates the duration of the objects staying active.
- In a sequence diagram, an interaction between two objects occurs when one object sends a message to another. The use of the activation bar on the lifelines of the Message Caller (the object that sends the message) and the Message Receiver (the object that receives the message) indicates that both are active/is instantiated during the exchange of the message.


Object
Object
<<message>>

Message Arrows

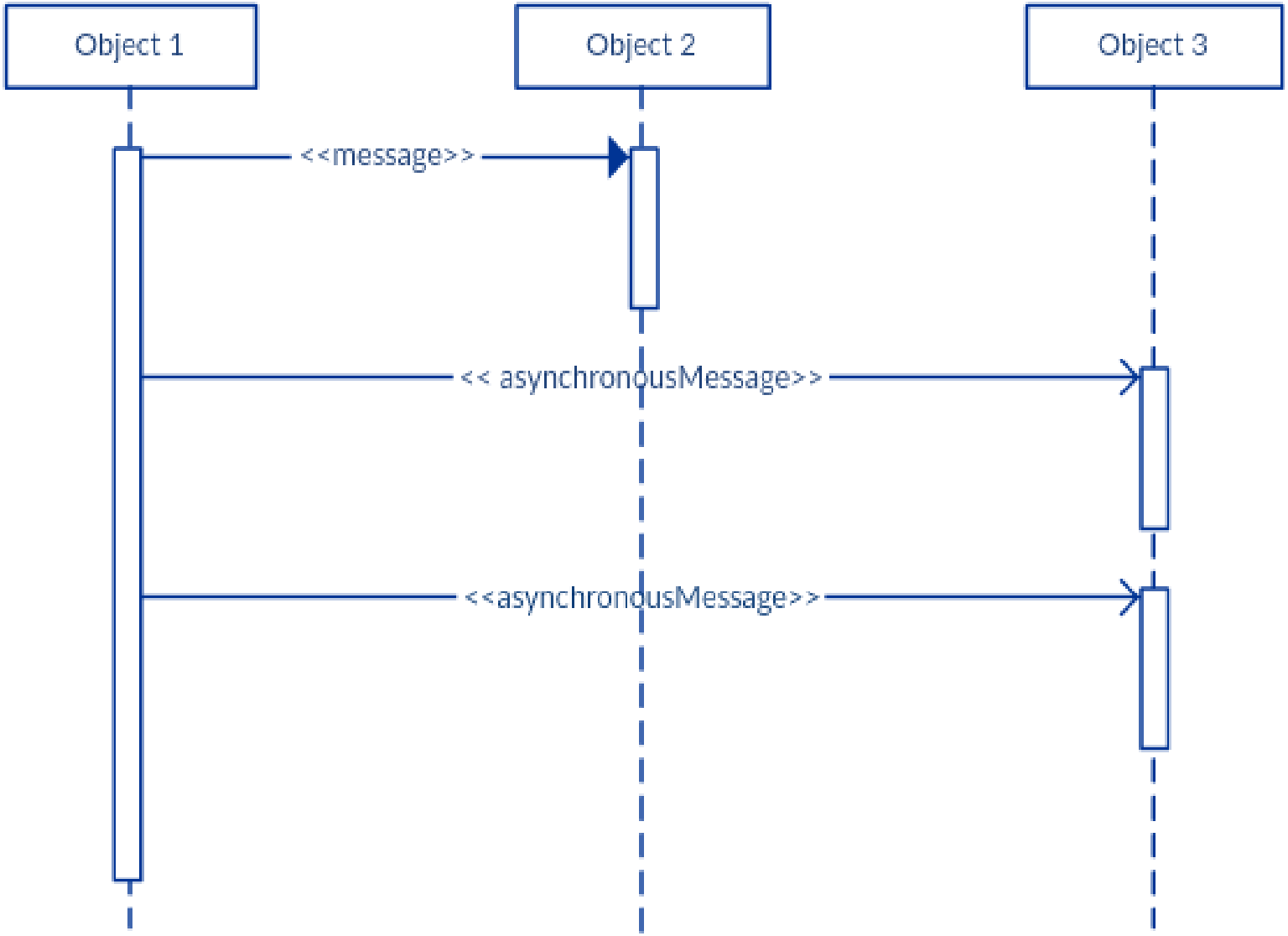
- An arrow from the Message Caller to the Message Receiver specifies a message in a sequence diagram. A message can flow in any direction; from left to right, right to left or back to the Message Caller itself. While you can describe the message being sent from one object to the other on the arrow, with different arrowheads you can indicate the type of message being sent or received.

Synchronous message

- Synchronous message is used when the sender waits for the receiver to process the message and return before carrying on with another message. The arrow head used to indicate this type of message is a solid one, like the one below.

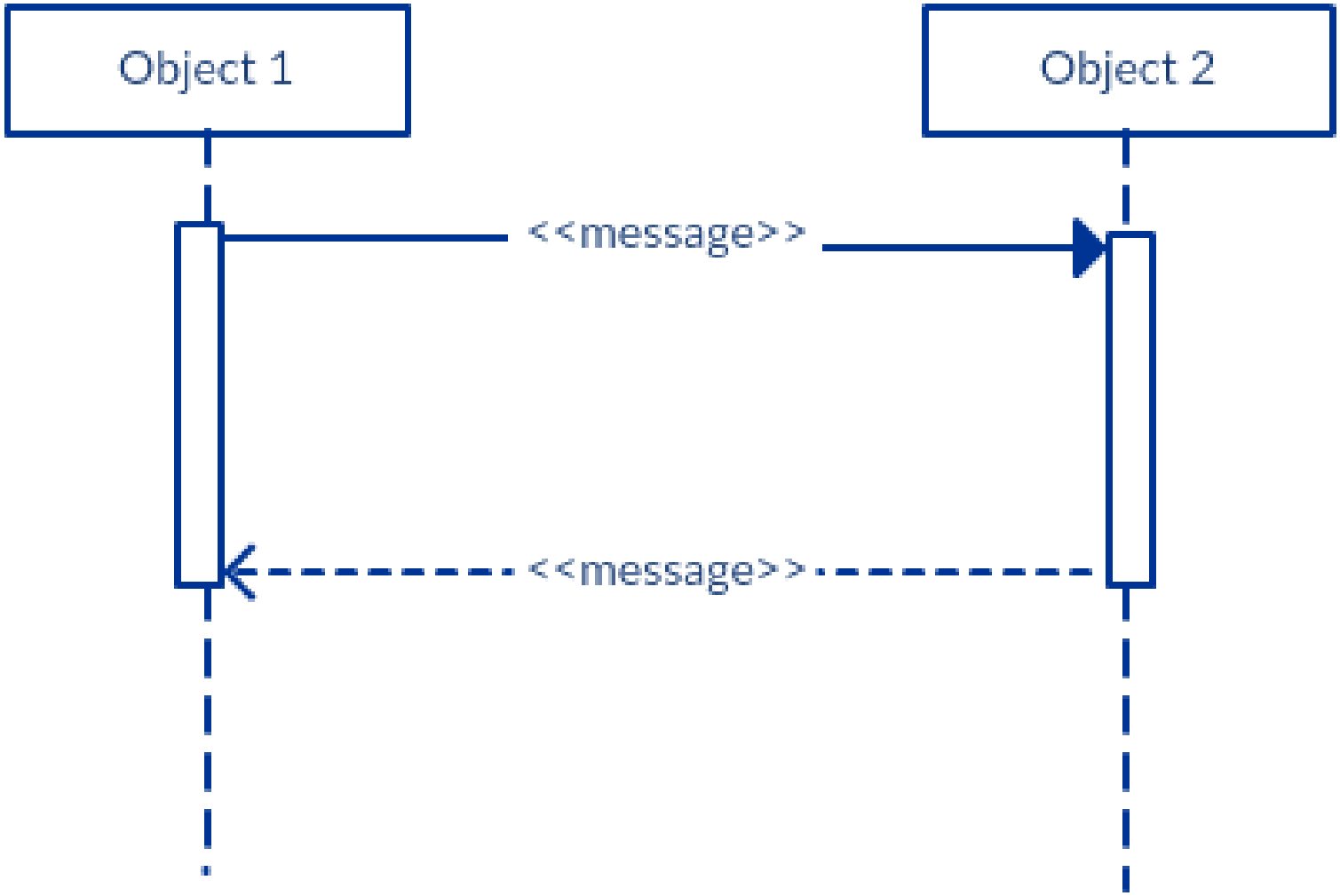
Asynchronous message

- An asynchronous message is used when the message caller does not wait for the receiver to process the message and return before sending other messages to other objects within the system. The arrow head used to show this type of message is a line arrow like shown in the example below.


Object 1
Object 2
Object 3
<<message>>
<< asynchronousMessage>>
<<asynchronousMessage>>

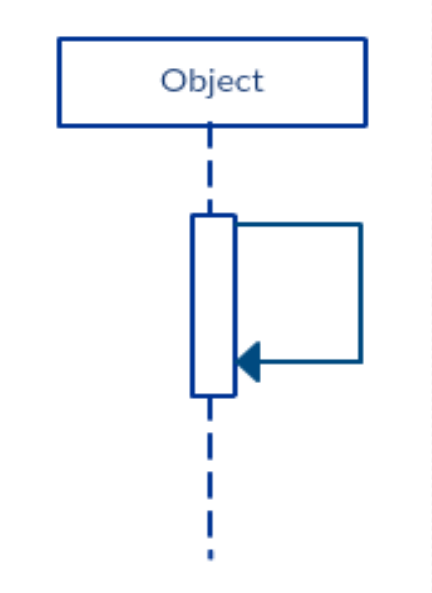
Return message

- A return message is used to indicate that the message receiver is done processing the message and is returning control over to the message caller. Return messages are optional notation pieces, for an activation bar that is triggered by a synchronous message always implies a return message.


Object 1
Object 2
<<message>>
<<message>>

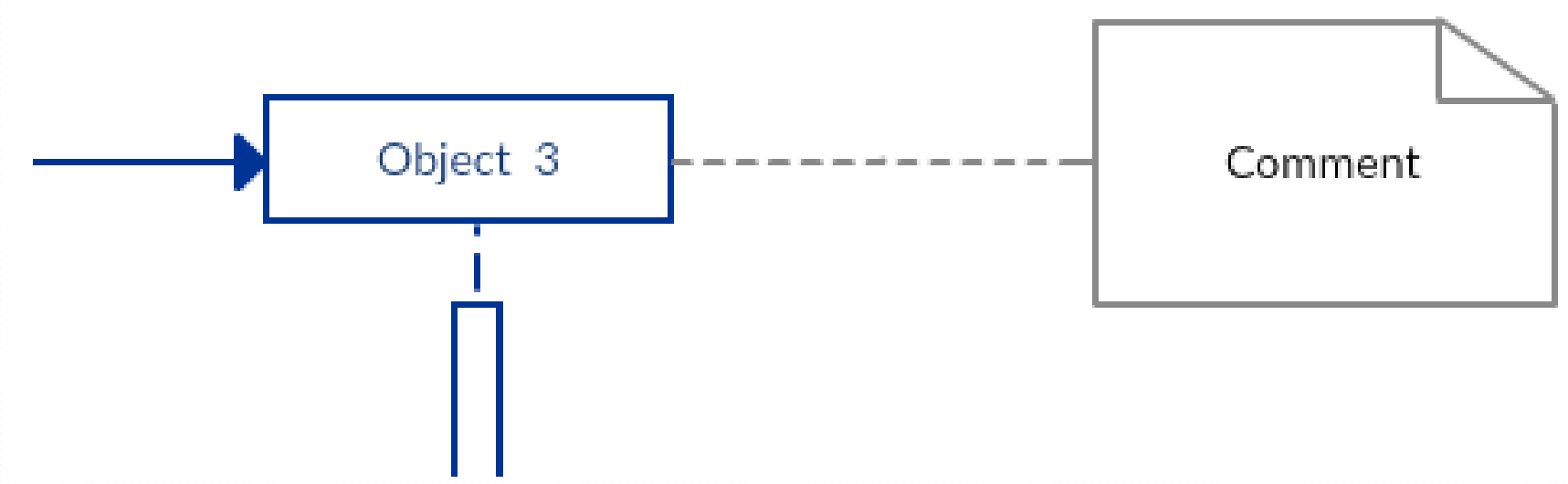
Reflexive message

- When an object sends a message to itself, it is called a reflexive message. It is indicated with a message arrow that starts and ends at the same lifeline like shown in the example below.

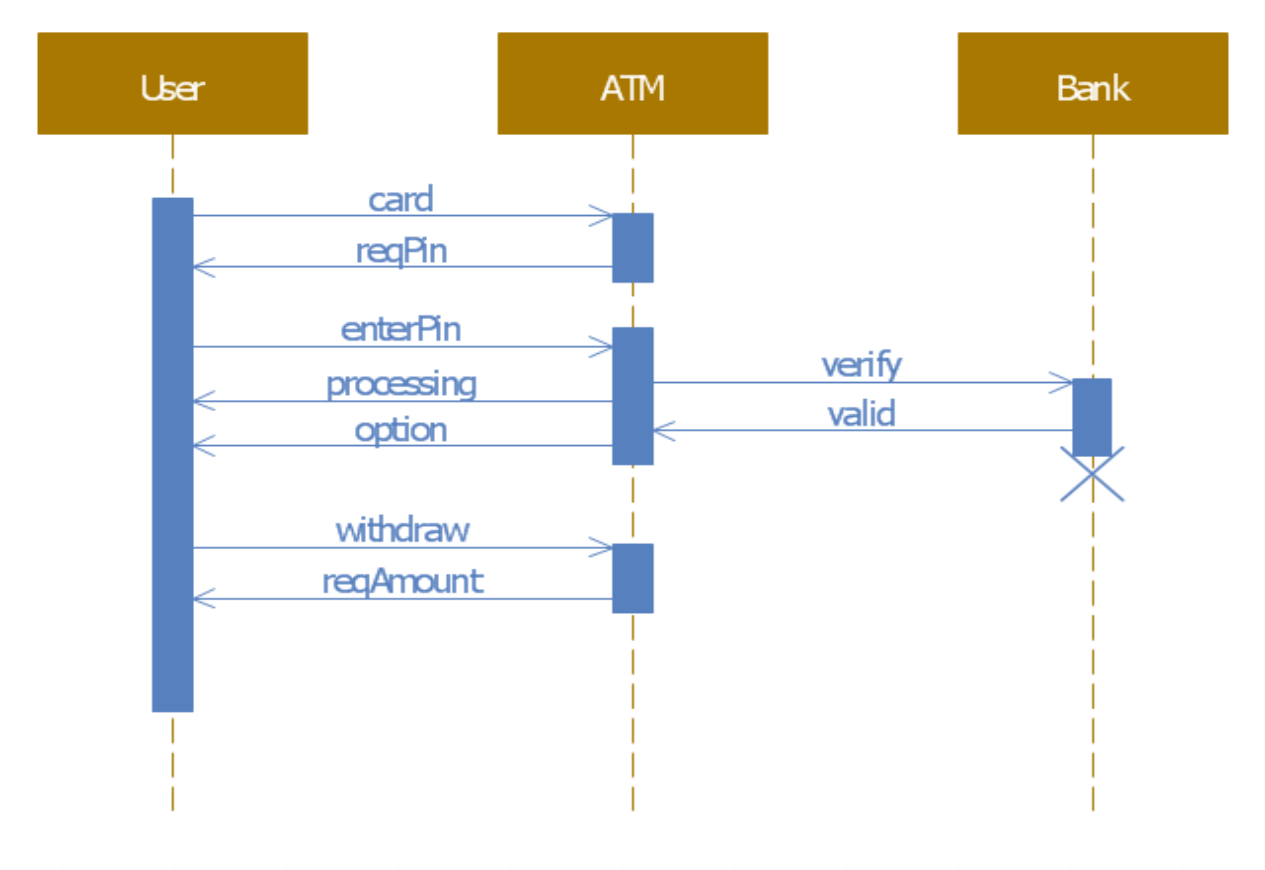


Comment

- The comment object is a rectangle with a folded-over corner as shown below. The comment can be linked to the related object with a dashed line.



Example


User
ATM
Bank
card
reqPin
enterPin
processing
option
verify
valid
withdraw
reqAmount