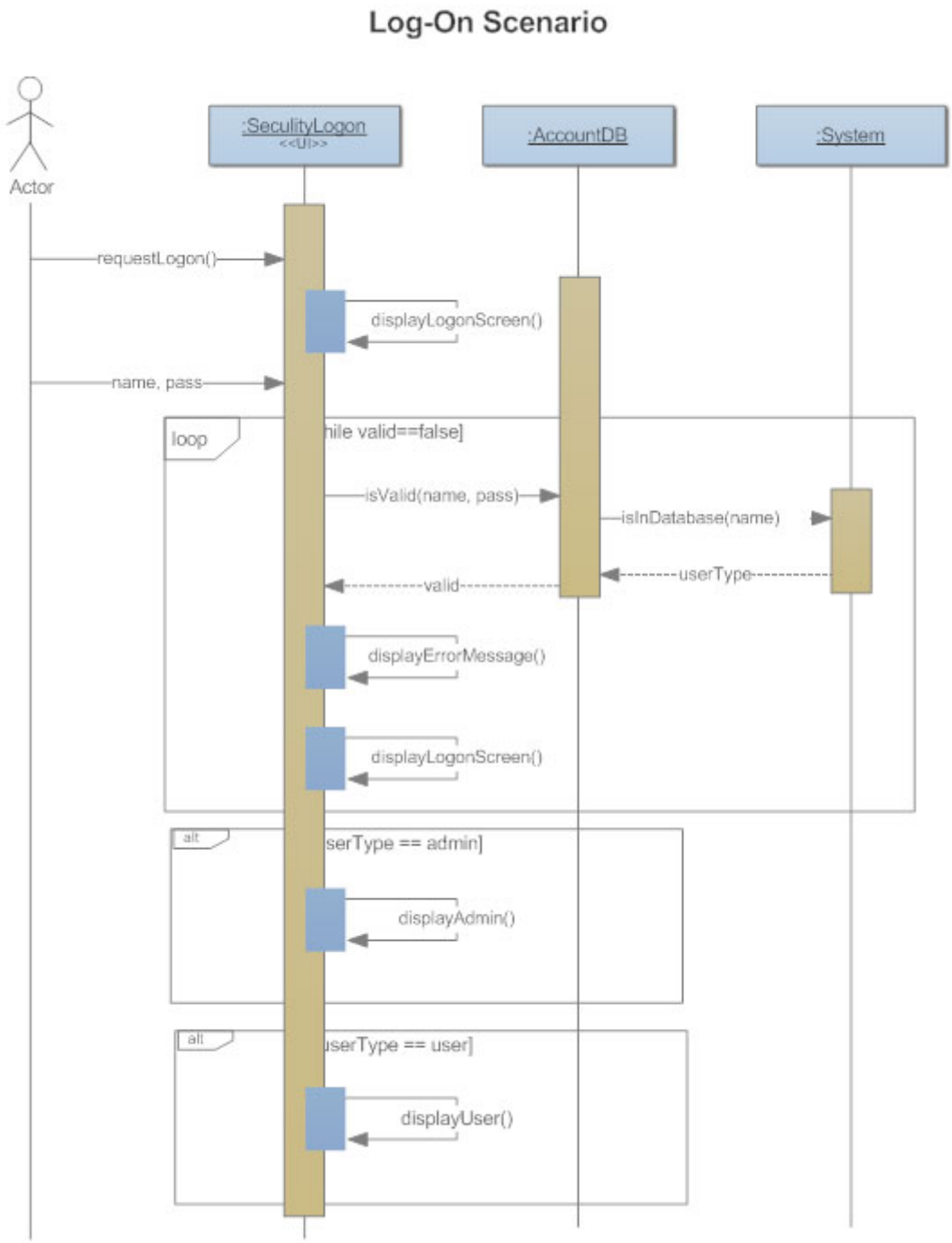
Sequence Diagram



What is a Sequence Diagram?

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

Sequence Diagram Tutorial

Start with one of SmartDraw's included sequence diagram templates. You'll notice that all the notations and symbols you need are docked to the left of your drawing area. Simply stamp them to your page and connect the symbols.

- Model and document how your system will behave in various scenarios
- Validate the logic of complex operations and functions

Learn how to draw UML diagrams of all kinds with SmartDraw.

Basic Sequence Diagram Notations

Class Roles or Participants

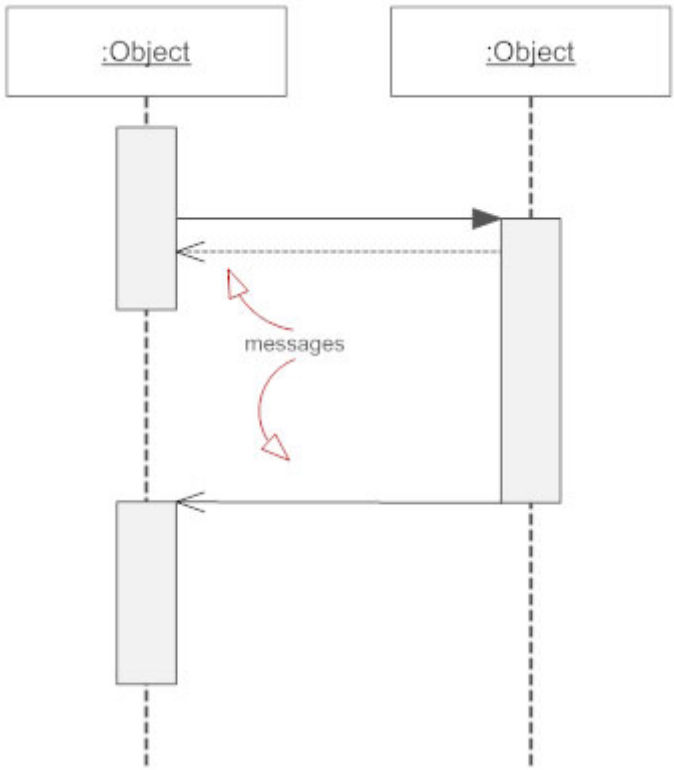
Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

Activation or Execution Occurrence

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.

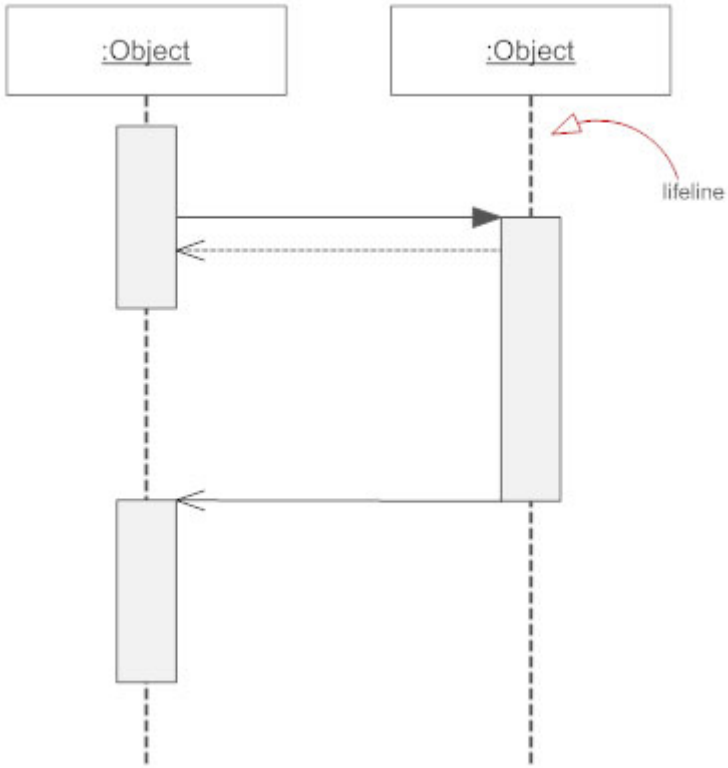
Messages

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks. For message types, see below.



Lifelines

Lifelines are vertical dashed lines that indicate the object's presence over time.



Destroying Objects

Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

Loops

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [ ].

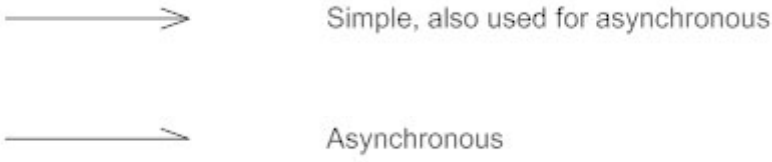
Types of Messages in Sequence Diagrams

Synchronous Message

A synchronous message requires a response before the interaction can continue. It's usually drawn using a line with a solid arrowhead pointing from one object to another.

Asynchronous Message

Asynchronous messages don't need a reply for interaction to continue. Like synchronous messages, they are drawn with an arrow connecting two lifelines; however, the arrowhead is usually open and there's no return message depicted.

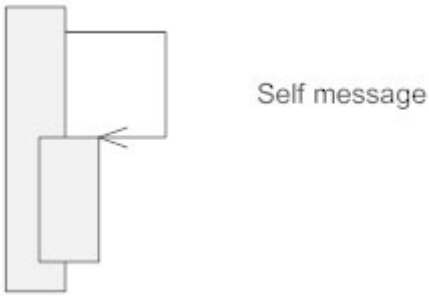


Reply or Return Message

A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline.

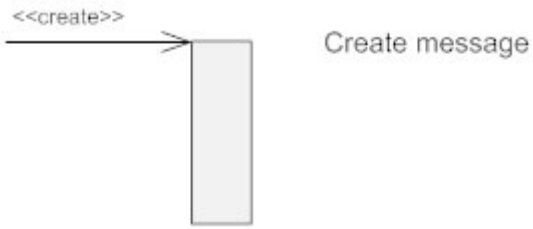
Self Message

A message an object sends to itself, usually shown as a U shaped arrow pointing back to itself.



Create Message

This is a message that creates a new object. Similar to a return message, it's depicted with a dashed line and an open arrowhead that points to the rectangle representing the object created.



Delete Message

This is a message that destroys an object. It can be shown by an arrow with an x at the end.

Found Message

A message sent from an unknown recipient, shown by an arrow from an endpoint to a lifeline.

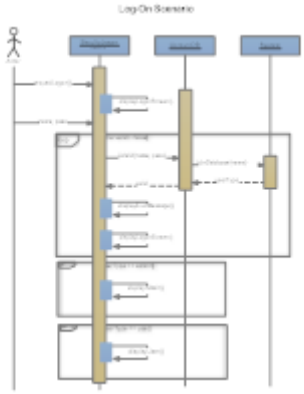
Lost Message

A message sent to an unknown recipient. It's shown by an arrow going from a lifeline to an endpoint, a filled circle or an x.

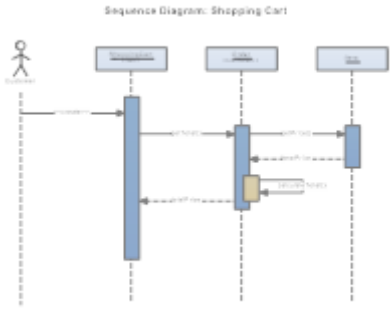
Sequence Diagram Examples

The best way to understand sequence diagrams is to look at some examples of sequence diagrams.

Click on any of these sequence diagrams included in SmartDraw and edit them:



Sequence Diagram - Log On Scenario



Sequence Diagram - Shopping Cart

Browse SmartDraw's entire collection of sequence diagram examples and templates

More info

- UML diagram tool
- Software design diagram templates
- Data flow diagram software

Other Diagrams →