What is sequence diagram

To understand what a sequence diagram is, it's important to know the role of the unified modeling language, better known as UML. UML is a modeling toolkit that guides the creation and notation of many types of diagrams, including behavior diagrams, interaction diagrams, and structure diagrams.

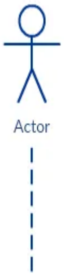
It shows the dynamic collaboration between multiple objects by describing the temporal order in which messages are sent between them. Commonly used by developers, model the interactions between objects in a single use case. Sequence diagrams are commonly used in software development to illustrate the behavior of a system or to help developers design and understand complex systems.

The Objects of Sequence Diagram

- See how objects and components interact with each other to complete a process.
- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- Plan and understand the detailed functionality of an existing or future scenario.
- Compared with other UML diagrams, a temporal sequence diagram places more emphasis on the chronological order of interaction behavior.
- It can visually describe the process of concurrency.

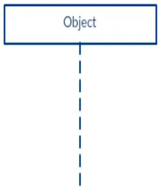
Basic symbols and components

Actor



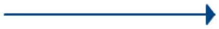
System actors, which can be people, machines, other systems, subsystems; used to represent in the temporal sequence diagram.

Lifeline



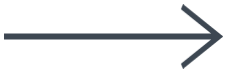
A dashed line extending down from the object icon in the timing diagram, indicating how long the object has existed.

Synchronous message



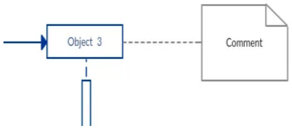
synchronous message is used when the sender waits for the receiver to process the message and return before carrying on with another message. The arrowhead used to indicate this type of message is a solid one.

Asynchronous message



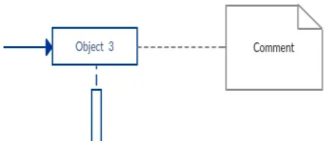
An asynchronous message is used when the message caller does not wait for the receiver to process the message and return before sending other messages to other objects within the system. The arrowhead used to show this type of message is a line arrow

Replay message



Represented by a dashed line with a lined arrowhead, these messages are replies to calls.

Notes and comments



Type additional information into a *Comment* shape from the UML 2.5 shape

Frame labels for sequence fragment

Operator	**Meaning**
alt	**Alternative multiple fragments: only the one whose condition is true will execute.**
opt	**Optional**: the fragment executes only if the supplied condition is true. Equivalent to an alt only with one trace.
par	**Parallel**: each fragment is run in parallel.
loop	**Loop**: the fragment may execute multiple times, and the guard indicates the basis of iteration.
critical	**Critical region**: the fragment can have only one thread executing it at once.
neg	**Negative**: the fragment shows an invalid interaction.
ref	**Reference**: refers to an interaction defined on another diagram. The frame is drawn to cover the lifelines involved in the interaction. You can define parameters and a return value.
sd	**Sequence diagram**: used to surround an entire sequence diagram.

How to read a UML sequence diagram

Read a sequence diagram from the top down. The further you progress down a sequence diagram; the more time has elapsed, and the more events have occurred. The processes are represented vertically and interactions are shown as arrows. Each system/object instance and actor is placed on a lifeline - a vertical dotted line - going across the top of the sequence diagram. The messages that pass between the lifelines are connectors - solid for an initial message or outgoing call ,and dotted for a return value (optional).

Drawing sequence diagrams

- Before you start drawing the sequence diagram or decide what interactions should be included in it, you need to draw the use case diagram and ready a comprehensive description of what the particular use case does.
- You can draw UML diagrams online or offline.
- In the UML, an object in a sequence diagram is drawn as a rectangle containing the name of the object, underlined. An object can be named in one of three ways: the object name, the object name and its class, or just the class name (anonymous object).