Assignment No.9

Page No. 57
Date

Name - Kudharan Sumit Dattatraya
Roll No - 20U401
Class - TE4
Subject -

Problem Statement -

Consider the online shopping system in the assignment 2 and draw the sequence diagram using UML tool to show message exchanges.

Theory -

*Introduction of Sequence Diagram -

A sequence diagram or system sequence diagram(SSD) show object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario of and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are typically associated with use caserealizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them in order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

A system sequence diagram should specify & show the following.

- External actors
- Message (methods) invoked by these actors
- Return values (if any) associated with previous messages.

- Indication of any loops or iteration area

A sequence diagram is the most commonly used interaction diagram.

Interaction Diagram.

An interaction algorithm diagram is used to show the interactive behaviour of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features & aspects of interaction in a system.

* Purpose of Sequence Diagram.

- Model high-level interaction between active objects in a system.
- Model the interaction between obje instances within a collaboration that realizes a use case.
- Model the interaction between objects within a collaboration that realizes an operation.
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

* Sequence Diagram Elements-

There are four primary elements of a sequence diagram-

- Objects
- Lifelines
- Messages
- Focus of control

① Objects- objects that are involved in the sequence of events you are documenting should be placed at the top of the sequence diagram across its horizontal axis. As shown in figure 1, it's a good idea to place the actor that initiates a particular sequence at the upper left side of the diagram.

You can also place a "UI (user Interface) placeholder class on the diagram with which the actor interacts. This is an excellent tool for providing context for a use case. Next, you can place objects on the diagram that are instantiated by the UI or by other objects. You should place the most important objects to the left & ~~Subroutine~~ Subordinate objects to the right. It's best to place objects on the diagram in a way that minimizes lines that cross.

② Lifelines - The lifeline is the dotted line that extends down the vertical axis from the base of each object. The lifeline indicates the life span of an object over a period of time.

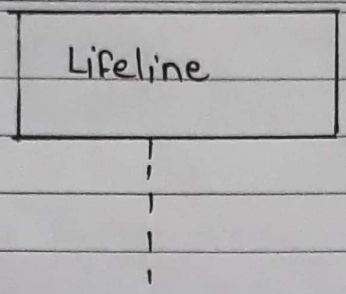
③ Messages - Messages are the most important elements of a sequence diagram. They indicate when one object calls an operation on another object (or itself). They are also used to indicate return values. Message flow begins at the top left object (which is usually an actor) & flows down the vertical axis from one object to another.

④ Focus of Control - Focus of control (FOC) is used in sequence diagrams to show the period of time during which an object performs an action. FOC is rendered as a thin, rectangular object that sits on top of object lifelines. The top of the FOC rectangle coincides with the receipt of a message. The bottom of the rectangle coincides with the completion of an action & can be marked with a return message. FOC rectangles can be stacked in layers on a sequence diagram to indicate focus of control nesting.

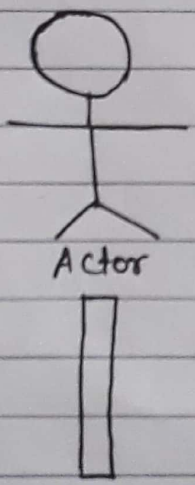
*Notations of a Sequence Diagram -

① Lifeline -

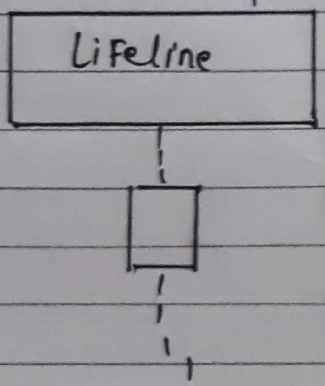
An individual participant in the sequence diagram is represented by a lifeline. It is positioned at top of of the diagram



② Actor- A role played by an entity that interacts with the subject is called an actor. It is out of the scope of the system. It represents the role, which involves human users & external hardware or subjects. An actor may or may not represent a physical entity, but it purely depicts the role of an entity. Several distinct roles can be played by an actor or vice versa.



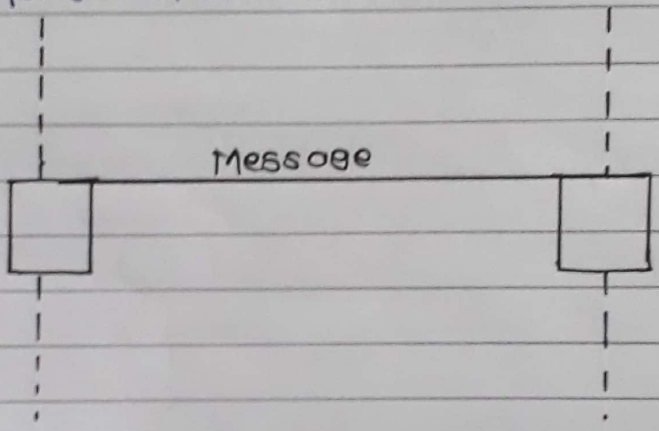
③ Activation - It is represented by a thin rectangle on the lifeline. It describes that time period in which an operation is performed by an element, such that the top & the bottom of the rectangle is associated with the initiation & the completion time, each respectively.



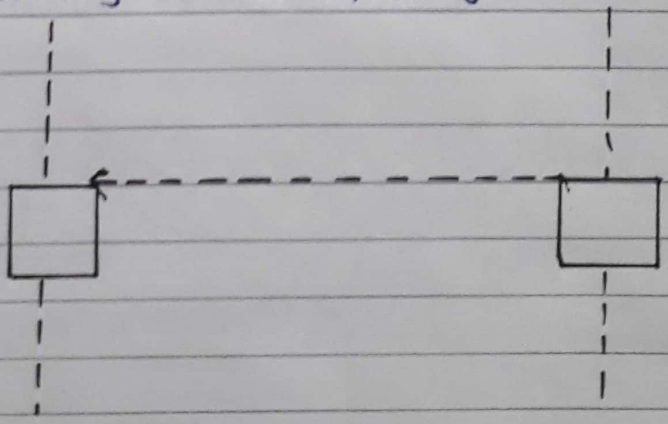
(5) Messages - The messages depict the interaction between the objects & are represented by arrows. They are in the sequential order on the lifeline. The core of the sequence diagram is formed by messages and lifelines.

Following are types of messages enlisted below:

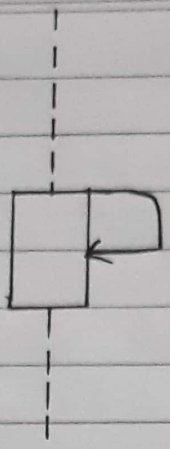
(i) call Message - It defines a particular communication between the lifelines of an interaction, which represents that the target lifeline has invoked an operation.



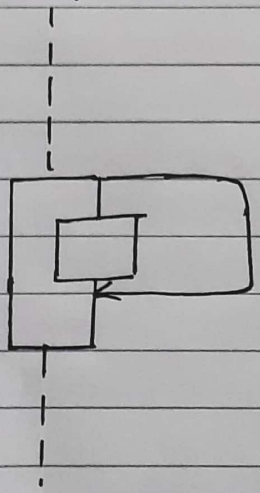
(ii) Return Message - It defines a particular communication between the lifelines of interaction that represent the flow of information from the receiver of the corresponding caller Message.



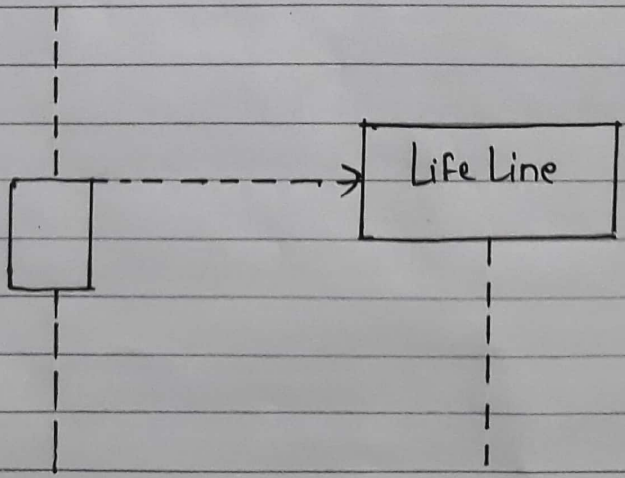
(iii) Self Message - It describes a communication particularly between the lifelines of an interaction that represents a message of the same lifeline has been invoked.



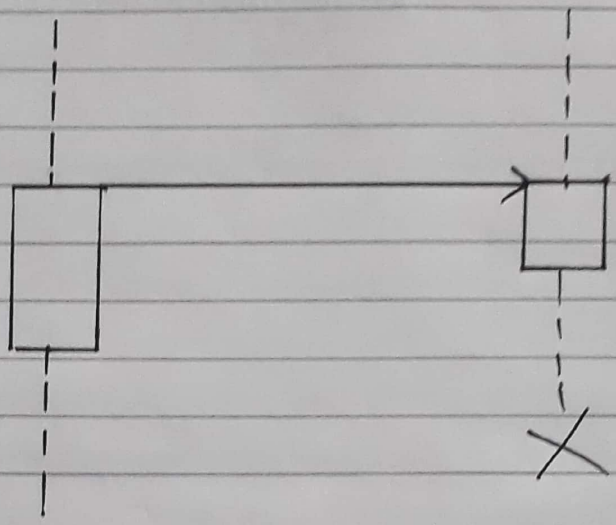
(iv) Recursive Message - A self message sent for recursive purpose is called a recursive message. In other words, it can be said that the recursive message is a special case of the self message as it represents the recursive calls.



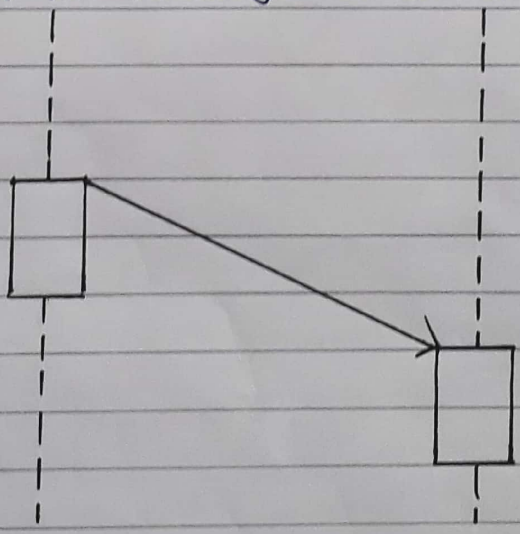
(v) Create Message - It describes a communication particularly between the lifelines of an interaction describing that the target (lifeline) has been instantiated.



(vi) Destroy Message - It describes a communication, particularly between the lifelines of an interaction that depicts a request to destroy the lifecycle of the target.

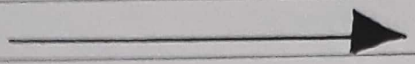


(vii) Duration Message - It describes a communication particularly between the lifelines of an interaction which portrays the time passage of the message while modelling a system.

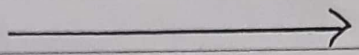


(viii) Common message symbols - Use the following arrows & message symbols to show how information is transmitted between objects. These symbols may reflect the start & execution of an operation or the sending the reception of a signal.

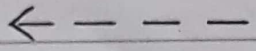
(ix) Synchronous message Symbol - Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call & the reply.



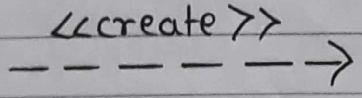
(x) Asynchronous Message Symbol - Represented by a solid line with a lined arrowhead. Asynchronous message don't require a response before the sender continues. only the call should be included in the diagram.



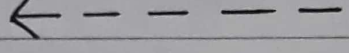
(xi) Asynchronous return message Symbol - Represented by a dashed line with a lined arrowhead.



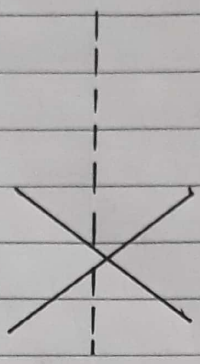
(xii) Asynchronous Create Message Symbol - Represented by a dashed line with a lined arrowhead. This message creates a new object.



(xiii) Reply message symbol - Represented by a dashed line with a lined arrowhead these message are replies to calls.



(xiv) Delete message symbol - Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object.



*Advantages of Sequence Diagram -

- It explores the real time application.
- It depicts the message flow between the different objects.
- It has easy maintaince.
- It is easy to generate.
- Implement both forward & reverse engineering.
- It can easily update as per the new change in the system.

*Disadvantages of Sequence Diagram -

- In the case of too many lifelines the sequence diagram can get more complex.
- The incorrect result may be produced, if the order of the flow of messages changes.
- Since each sequence needs distinct notations for its representation it may make the diagram more complex.
- The type of sequence is decided by the type of message.

Conclusion -

Sequence diagrams are an excellent tool for modelling the dynamic aspects of your system. They help you see the big picture of message flow between objects that carry out the logic of use cases. With a little pratice, sequence diagrams can help you build more streamlined & bug-free interactions between objects in your software application