

---

## Aim

Develop Sequence and Collaboration diagram for the project

## Theory

**UML**, Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

### **The primary goals in the design of the UML as follows:**

1. Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

UML 2.0 defines thirteen types of diagrams, divided into three categories:

### **Structure Diagrams include**

- o Class Diagram
- o Object Diagram
- o Component Diagram
- o Composite Structure Diagram
- o Package Diagram
- o Deployment Diagram.

### **Behaviour Diagrams include**

- o Use Case Diagram
- o Activity Diagram
- o State Machine Diagram.

### **Interaction Diagrams includes**

- o Sequence Diagram
- o Communication Diagram
- o Timing Diagram
- o Interaction Overview Diagram.

### Sequence Diagrams –

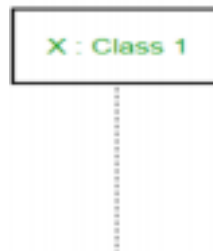
A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

- Actors – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram



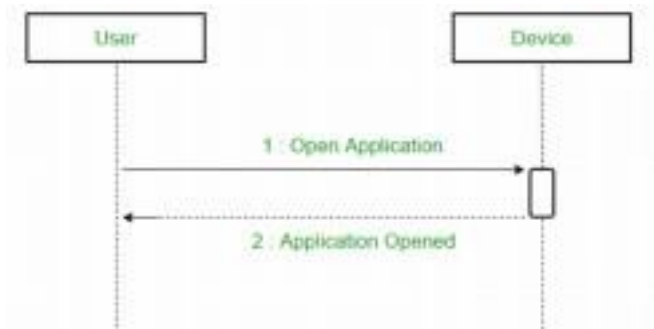
- Lifelines – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are



located at the top in a sequence diagram.

Lifeline

- Messages – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
  - o Synchronous messages – A synchronous message waits for a reply before the interaction can move forward. The sender waits until the receiver has completed the processing of the message. The caller continues only when it knows that the receiver has processed the previous message i.e. it receives a reply message. A large number of calls in object oriented programming are synchronous. We use a solid arrow head to represent a synchronous message.



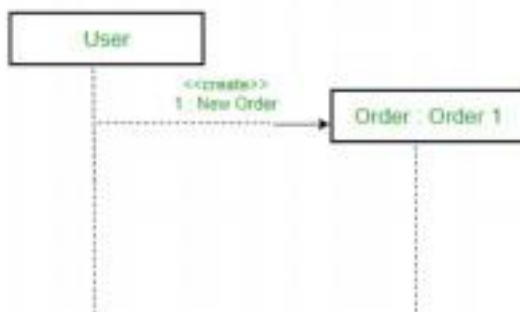
### Synchronous Message

o Asynchronous Messages – An asynchronous message does not wait for a reply from the receiver. The interaction moves forward irrespective of the receiver processing the previous message or not. We use a lined arrow head to represent an asynchronous message.



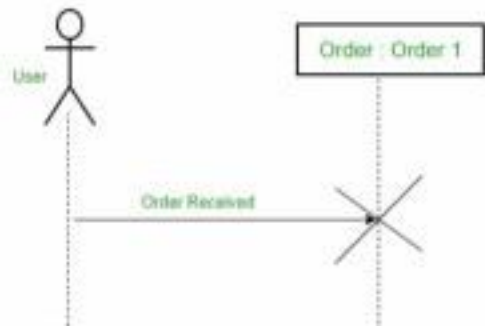
### Asynchronous Message

o Create message – We use a Create message to instantiate a new object in the sequence diagram. There are situations when a particular message call requires the creation of an object. It is represented with a dotted arrow and creates a word labelled on it to specify that it is the create Message symbol.



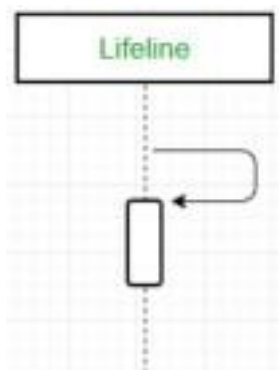
### Create Message

o Delete Message – We use a Delete Message to delete an object. When an object is deallocated memory or is destroyed within the system we use the Delete Message symbol. It destroys the occurrence of the object in the system. It is represented by an arrow terminating with a x.



#### Delete Message

o Self Message – Certain scenarios might arise where the object needs to send a message to itself. Such messages are called Self Messages and are represented with a U shaped arrow.



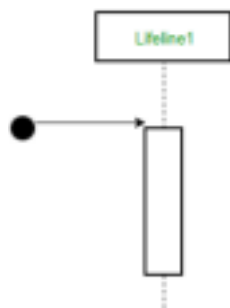
#### Self Message

o Reply Message – Reply messages are used to show the message being sent from the receiver to the sender. We represent a return/reply message using an open arrowhead with a dotted line. The interaction moves forward only when a reply message is sent by the receiver.



#### Reply Message

o Found Message – A Found message is used to represent a scenario where an unknown source sends the message. It is represented using an arrow directed towards a lifeline from an end point.



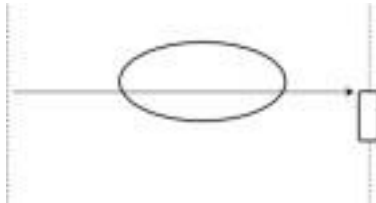
#### Found Message

o Lost Message – A Lost message is used to represent a scenario where the recipient is not known to the system. It is represented using an arrow directed towards an end point from a lifeline.

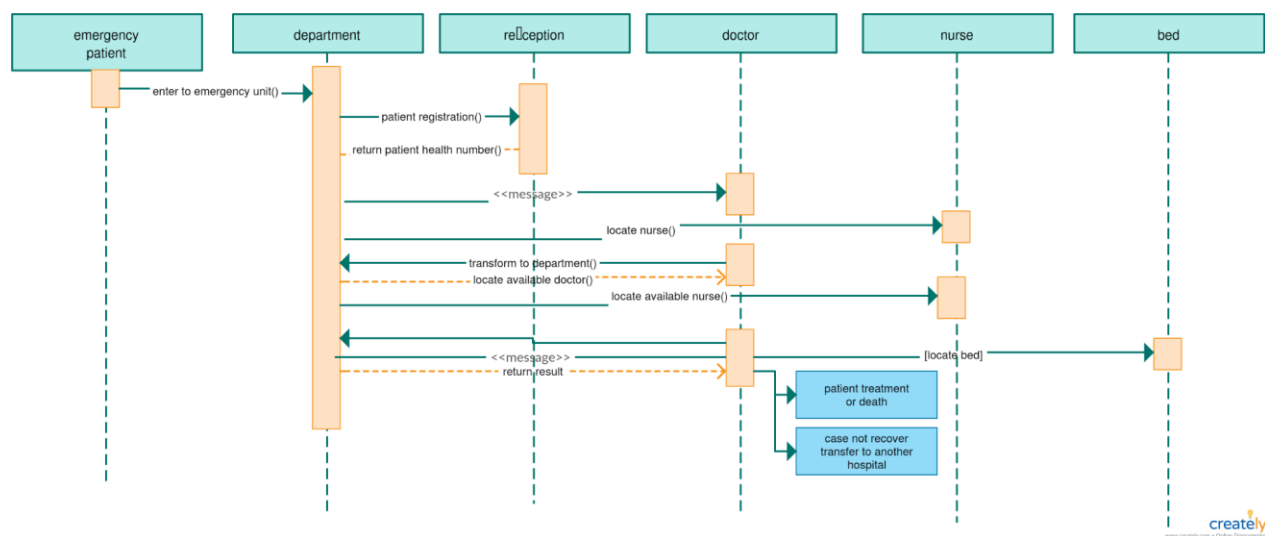


### Lost Message

- Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.



## For hospital management system



### Uses of sequence diagrams –

1. Used to model and visualise the logic behind a sophisticated function, operation or procedure.
2. They are also used to show details of UML use case diagrams.
3. Used to understand the detailed functionality of current or future systems.
4. Visualise how messages and tasks move between objects or components in a system

### UML Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the

system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

### Notations of a Collaboration Diagram

The representation of an object is done by an object symbol with its name and class underlined, separated by a colon.

- o The object is represented by specifying their name and class.
- o It is not mandatory for every class to appear.
- o A class may constitute more than one object.
- o In the collaboration diagram, firstly, the object is created, and then its class is specified.
- o To differentiate one object from another object, it is necessary to name them.

### Actors

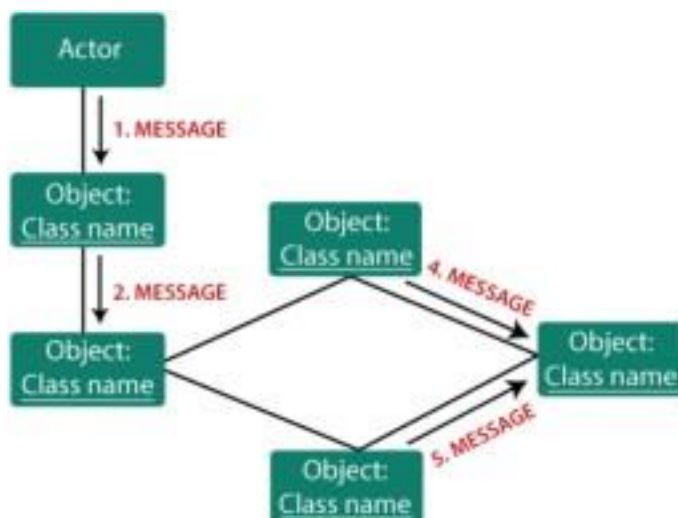
In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.

### Links

The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.

### Messages

It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.



### Benefits of a Collaboration Diagram

1. The collaboration diagram is also known as Communication Diagram. 2. It mainly puts emphasis on the structural aspect of an interaction diagram, i.e., how lifelines are connected.

3. The syntax of a collaboration diagram is similar to the sequence diagram; just the difference is that the lifeline does not consist of tails.
4. The messages transmitted over sequencing are represented by numbering each individual message.
5. The collaboration diagram is semantically weak in comparison to the sequence diagram.
6. The special case of a collaboration diagram is the object diagram.
7. It focuses on the elements and not the message flow, like sequence diagrams.
8. Since the collaboration diagrams are not that expensive, the sequence diagram can be directly converted to the collaboration diagram.

### **Drawback of a Collaboration Diagram**

1. Multiple objects residing in the system can make a complex collaboration diagram, as it becomes quite hard to explore the objects.
2. It is a time-consuming diagram.
3. After the program terminates, the object is destroyed.
4. As the object state changes momentarily, it becomes difficult to keep an eye on every single that has occurred inside the object of a system.

## **Conclusion**

In this experiment we learnt to Develop Sequence and Collaboration diagrams. The sequence diagram represents the UML, which is used to visualize the sequence of calls in a system that is used to perform a specific functionality where time sequence is the main focus whereas the collaboration diagram is used to visualize the organization of the objects and their interaction where the object organization is the main focus. The sequence diagrams are better suited for analysis activities and the collaboration diagrams are better suited for depicting simpler interactions of the smaller number of objects. Thus, we created a Sequence and Collaboration diagram for Hospital Management System and hence we successfully completed the experiment.