Slide 5

UML INTERACTION DIAGRAMS

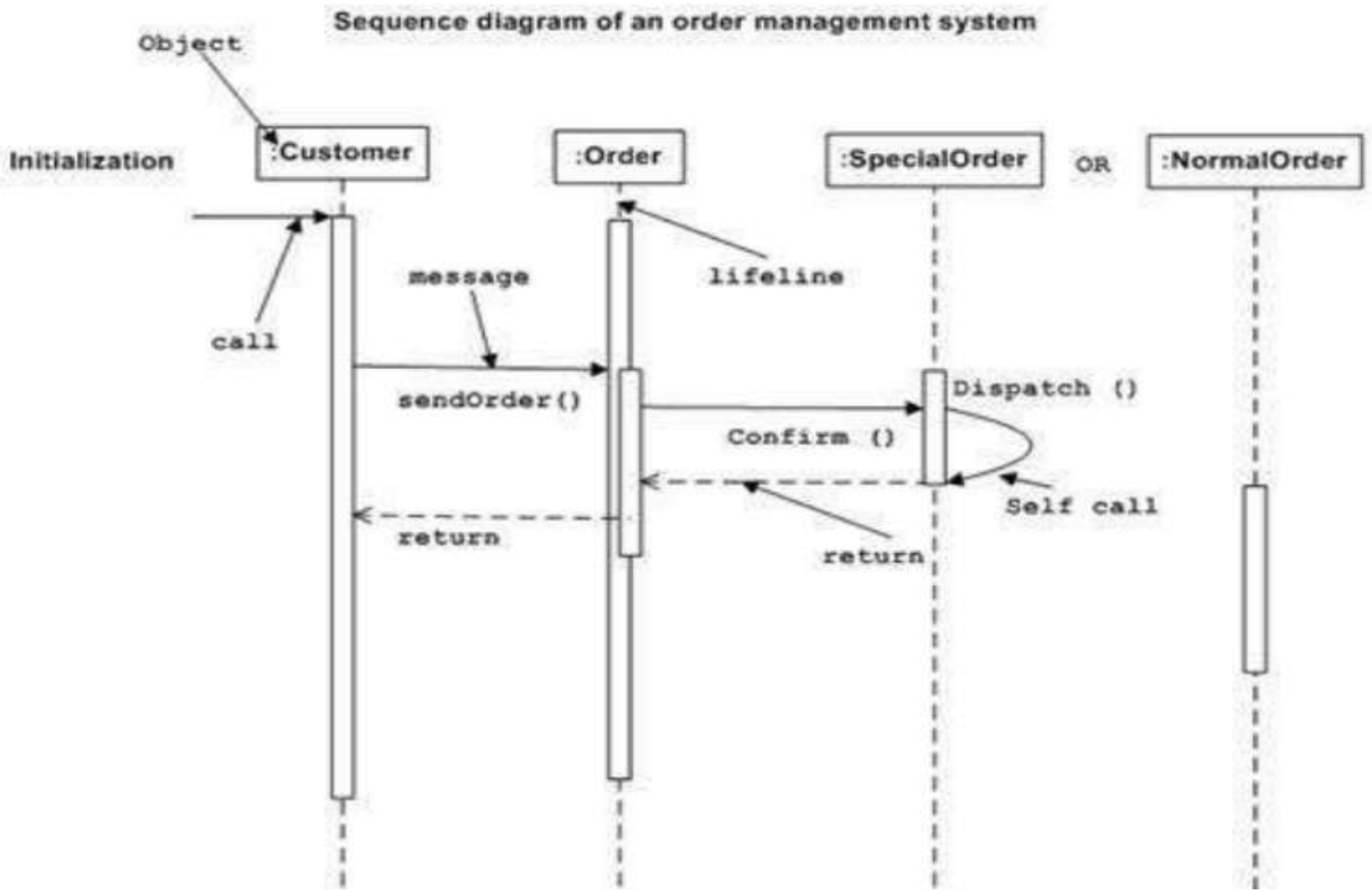
INTERACTION DIAGRAMS

- Interaction diagrams are diagrams that describe how groups of objects collaborate to get the job done.
- *Interaction diagrams capture the behavior of a single use case,* showing the pattern of interaction among objects.
- The diagram shows a number of example objects and the messages passed between those objects within the use case .
- There are two kinds of interaction models: sequence diagrams and collaboration diagrams.

Following things are to be identified clearly before drawing the interaction diagram

a) Objects taking part in the interaction.
b) Message flows among the objects.
c) The sequence in which the messages are flowing.
d) Object organization.

Elements of a sequence diagram



The purpose of interaction diagram is –

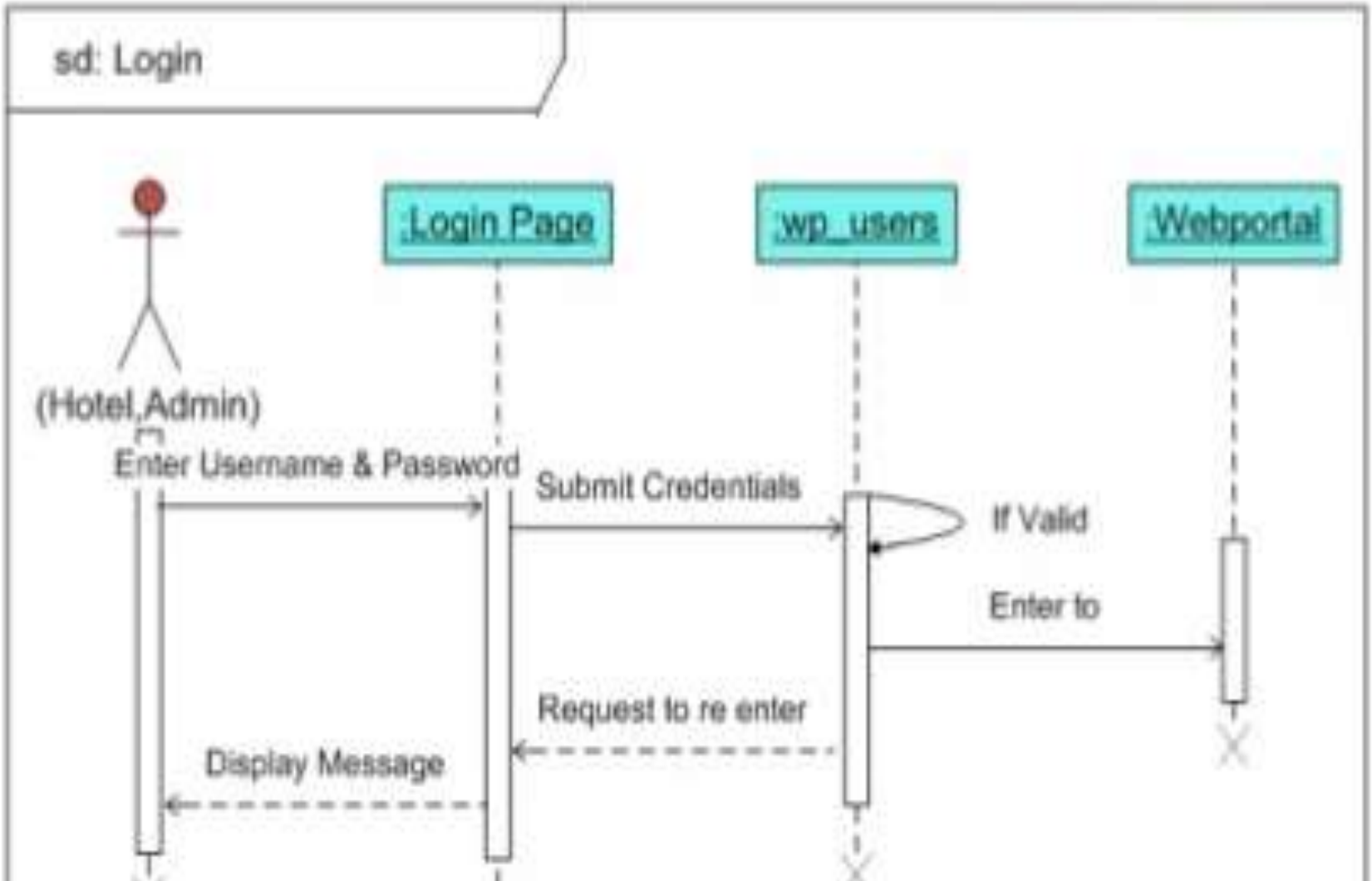
a) To capture the dynamic behaviour of a system.
b) To describe the message flow in the system.
c) To describe the structural organization of the objects.
d) To describe the interaction among objects.

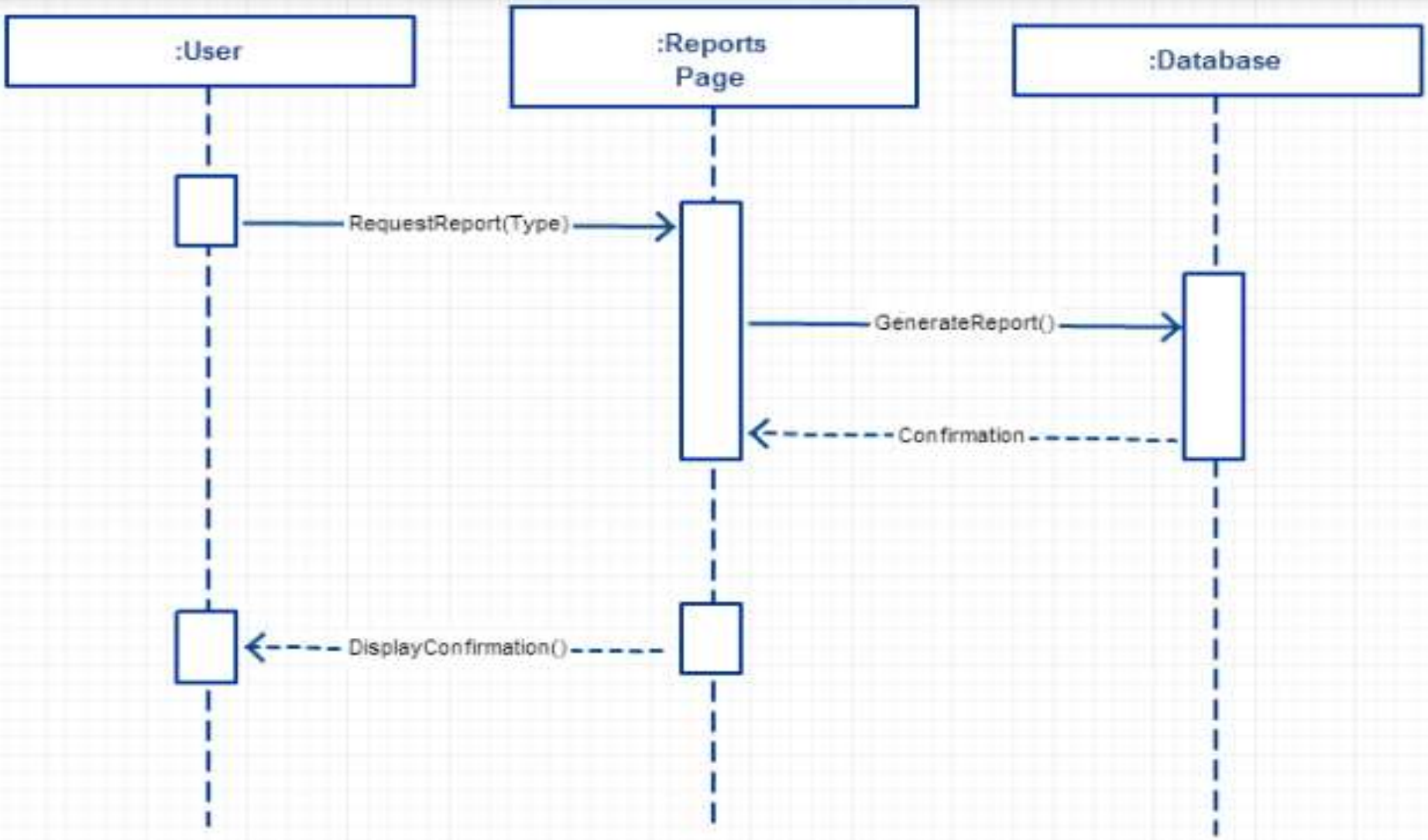
SEQUENCE DIAGRAM

- UML sequence diagram are used to show the communication between different participants in an interaction.
- Sequence diagram represents a process flow, workflow for any desired activity
- It is advisable to generate a sequence diagram for every basic flow of every use case.

- A sequence diagram shows an interaction arranged in a time sequence.
- It shows the objects participating in the interaction by their lifelines and the messages they exchange, arranged in a time sequence.
- The Horizontal axis contains the objects involved in an interaction
- The vertical axis contains the messages exchanged in a time sequence.

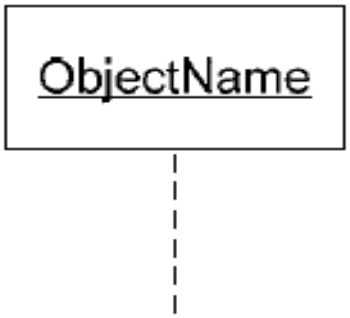
Sequence Diagram For Login




:User
:Reports Page
:Database
RequestReport(Type)
GenerateReport()
Confirmation
DisplayConfirmation()

SEQUENCE DIAGRAM NOTATIONS

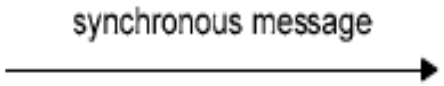
- Object lifeline: represents the role that an object plays as it interacts with other objects. If an object is created and destructed during the message sequence, then the lifeline represents the whole lifecycle of the object



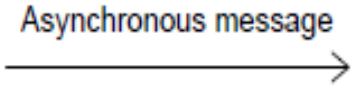
Activation bar: it is drawn at the bottom of the object lifeline to show the period during which the message is being processed. A message can only be sent or received from an activation bar.

3. **Message:** is a description of some type of communication between objects [4]. A message is modelled as an arrow where the tail of the arrow identifies the sender and the head points to the receiver. There are three types of messages:

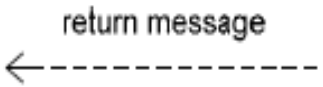
   a. **Synchronous message:** this is a message that must be processed on its own and a returned value or feedback is expected before another message is initiated since the feedback may influence the next message. The syntax is:



b. **Asynchronous message:** can be processed at the same time with other processes and does not expect a feedback neither does it influence other messages. The syntax is:



c. **Return Message:** this gives the feedback expected to the synchronous message issued. Mainly it reads and returns values without modifying them.



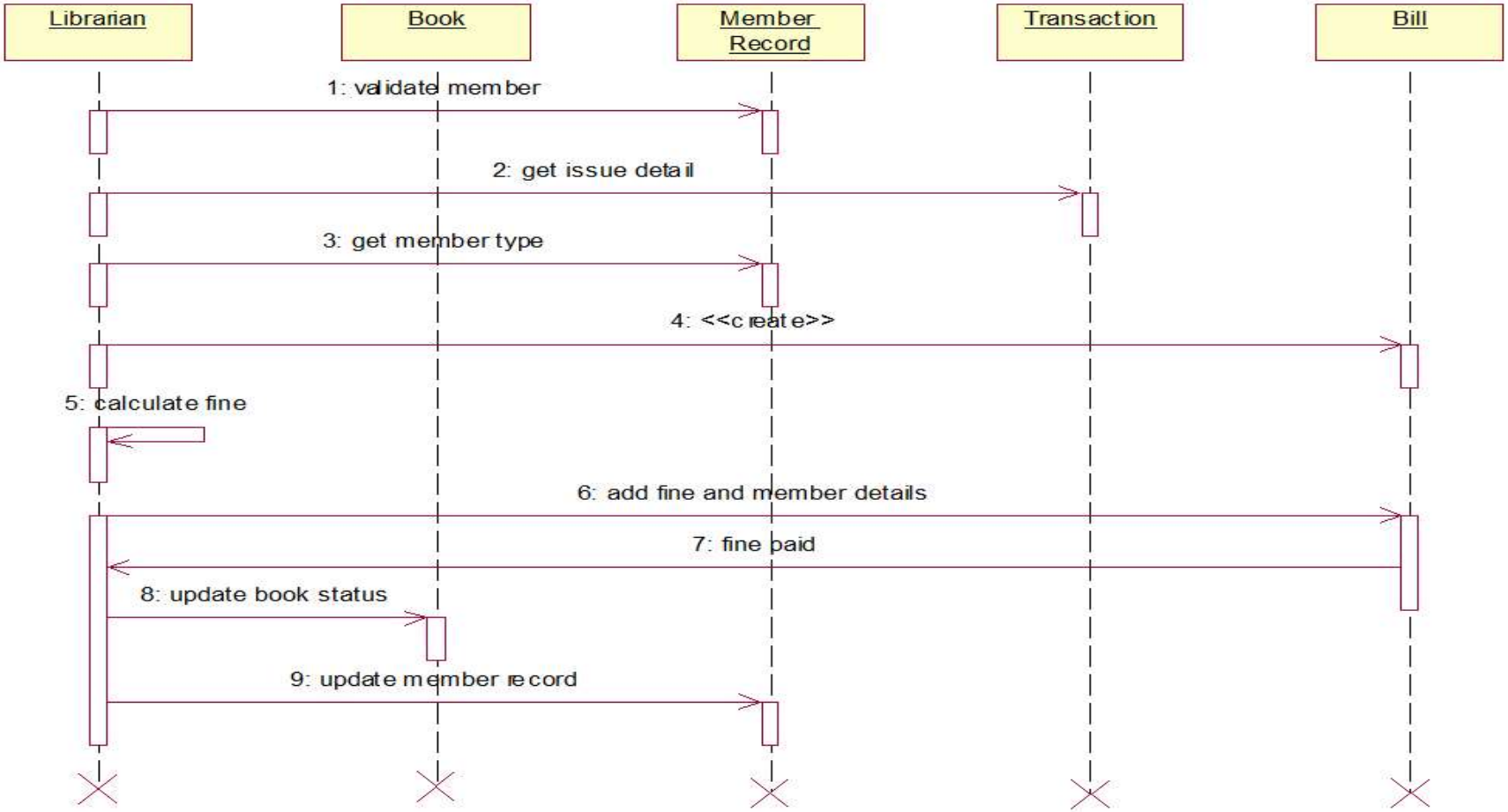
Class role represents the roles the object plays. The objects at the top of the diagram represent class roles.

Lifeline represents the existence of an object over a period of time. A vertical dotted line extending from the object is used to denote a lifeline.

Activation represents the time during which an object is performing an operation. Thin rectangles placed on lifelines are used to denote activations.

Messages/ *Method invocation* **:** these are the arrows that represent the communication between the objects.

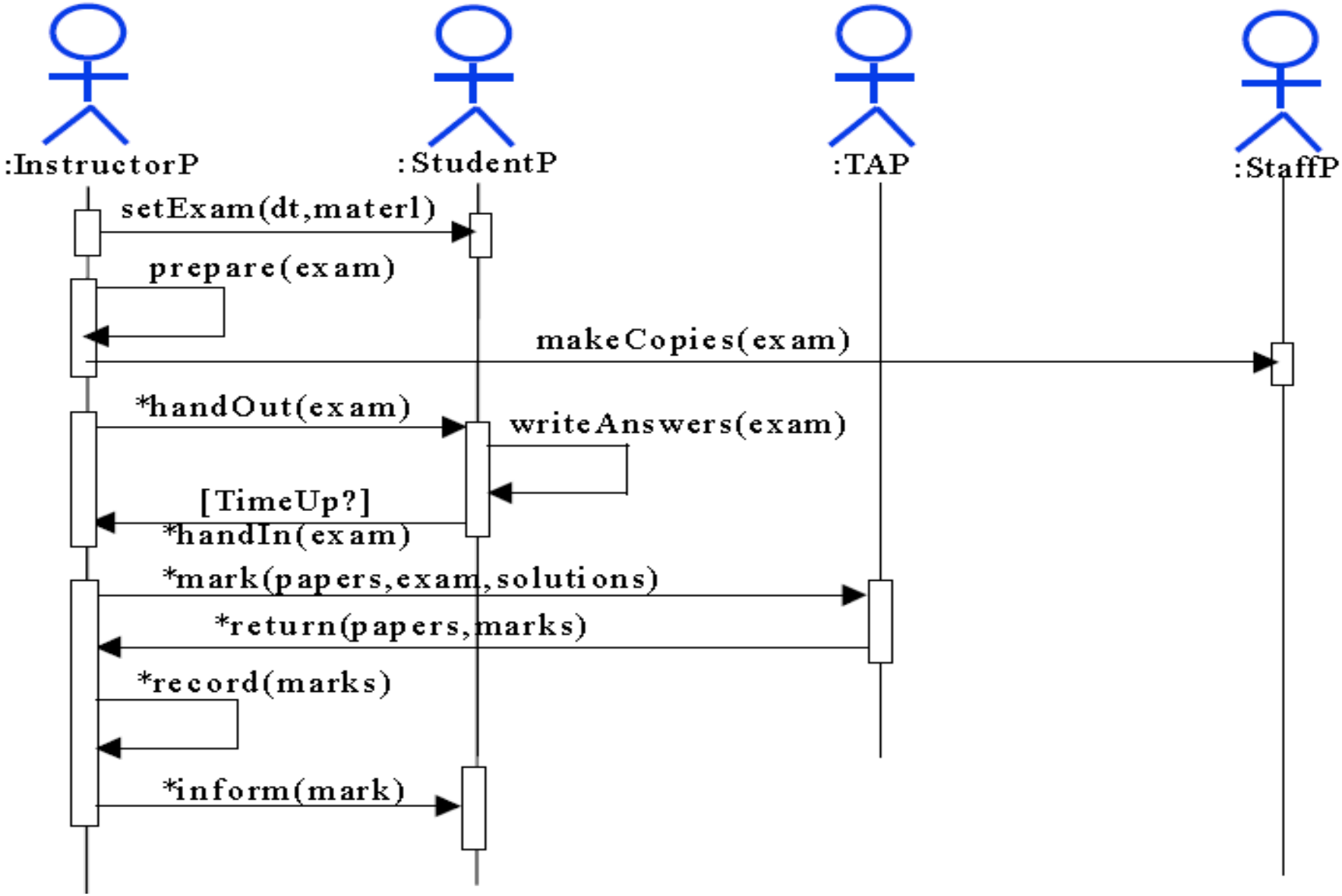
SAMPLE SEQUENCE DIAGRAM



- **Destroying Objects:** object can be terminated simply by showing x
- **Loops:** A repetition and a loop in the sequence diagram is shown by the rectangle box.
- **Synchronous Message:** It requires a response before the interaction can be continued.
- **Asynchronous Message:** It does not need a reply for the interaction to continue.
- **Reply or return message:** A reply message is shown with a dotted line with an arrowhead pointing back to the original lifeline.
- **self message:** a message object sends itself.
- **Create a message:** This is the message that creates a new object.
- **delete a message:** This is the message that destroys an object.
- **found message:** a message sent from an unknown recipient.
- **lost message:** A message is sent to an unknown recipient.

SEQUENCE DIAGRAM EXAMPLE

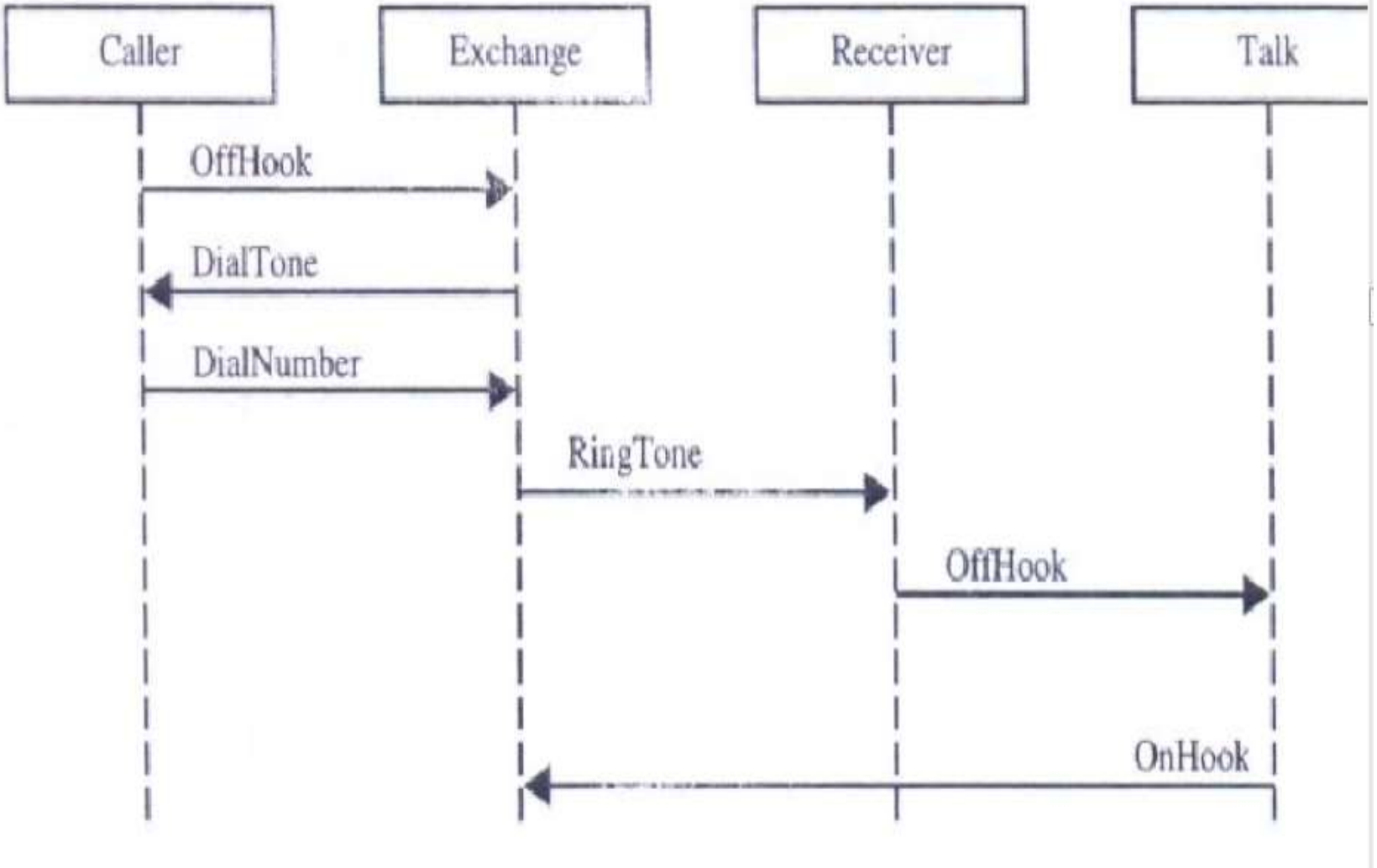
To give an exam, an instructor first notifies the students of the exam
date and the material to be covered. She then prepares the exam paper (with sample solutions), gets it copied to produce enough copies for the class, and hands it out to students on the designated time and location. The students write their answers to exam questions and hand in their papers to the instructor. The instructor then gives the exam papers to the TAs, along with sample solutions to each question, and gets them to mark it. She then records all marks and returns the papers to the students. Draw a sequence diagram that represents this process. Make sure to show when is each actor participating in the process. Also, show the operation that is carried out during each interaction, and what its arguments are.


:InstructorP
:StudentP
:TAP
:StaffP
setExam(dt,materl)
prepare(exam)
makeCopies(exam)
*handOut(exam)
writeAnswers(exam)
[TimeUp?]
*handIn(exam)
*mark(papers,exam,solutions)
*return(papers,marks)
*record(marks)
*inform(mark)

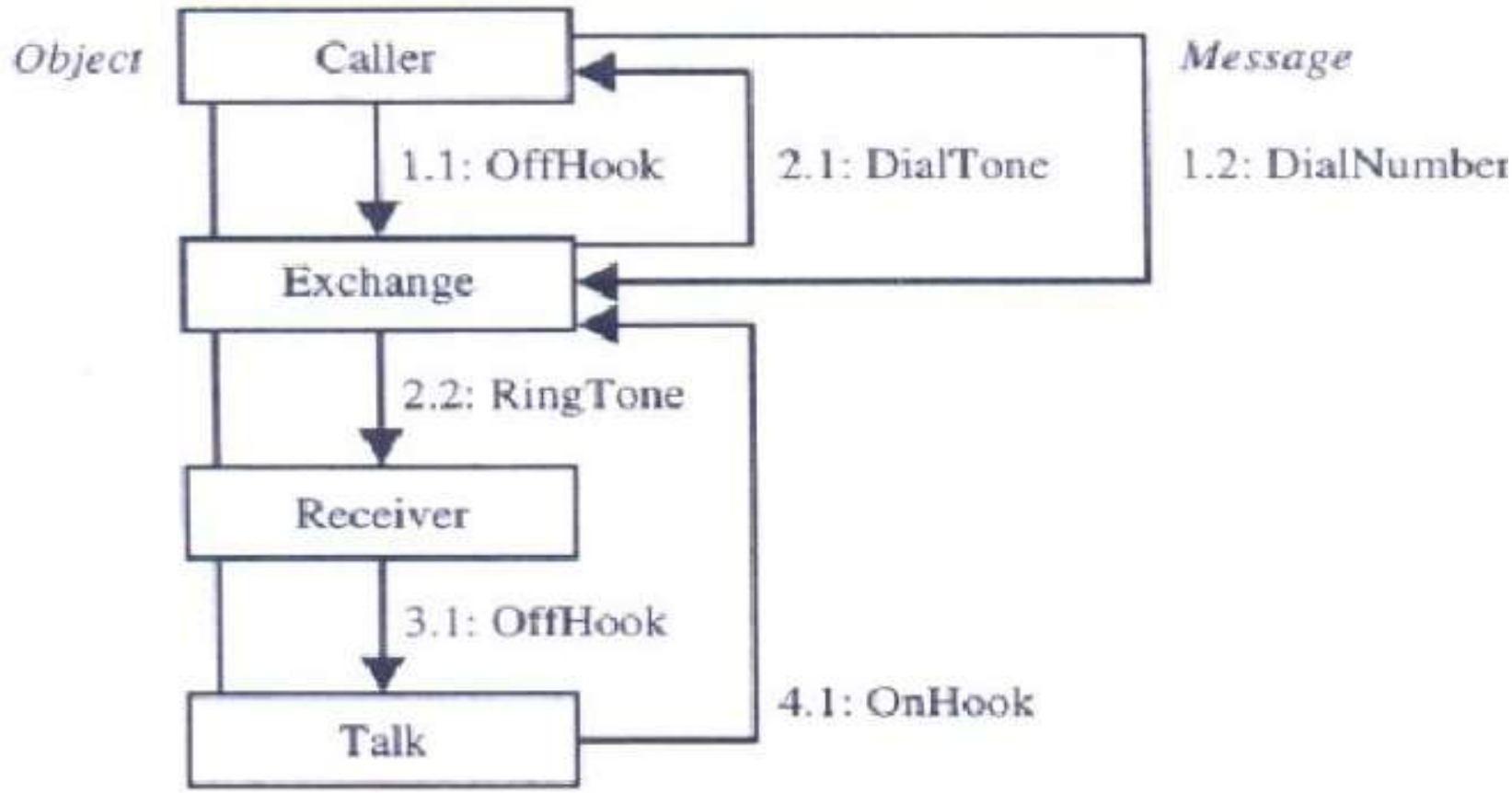
COLLABORATION DIAGRAMS

- *A collaboration diagram represents a collaboration, denoted by a set* of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.
- Closely related to sequence diagram. In a collaboration diagram, the sequence is indicated by numbering the messages.

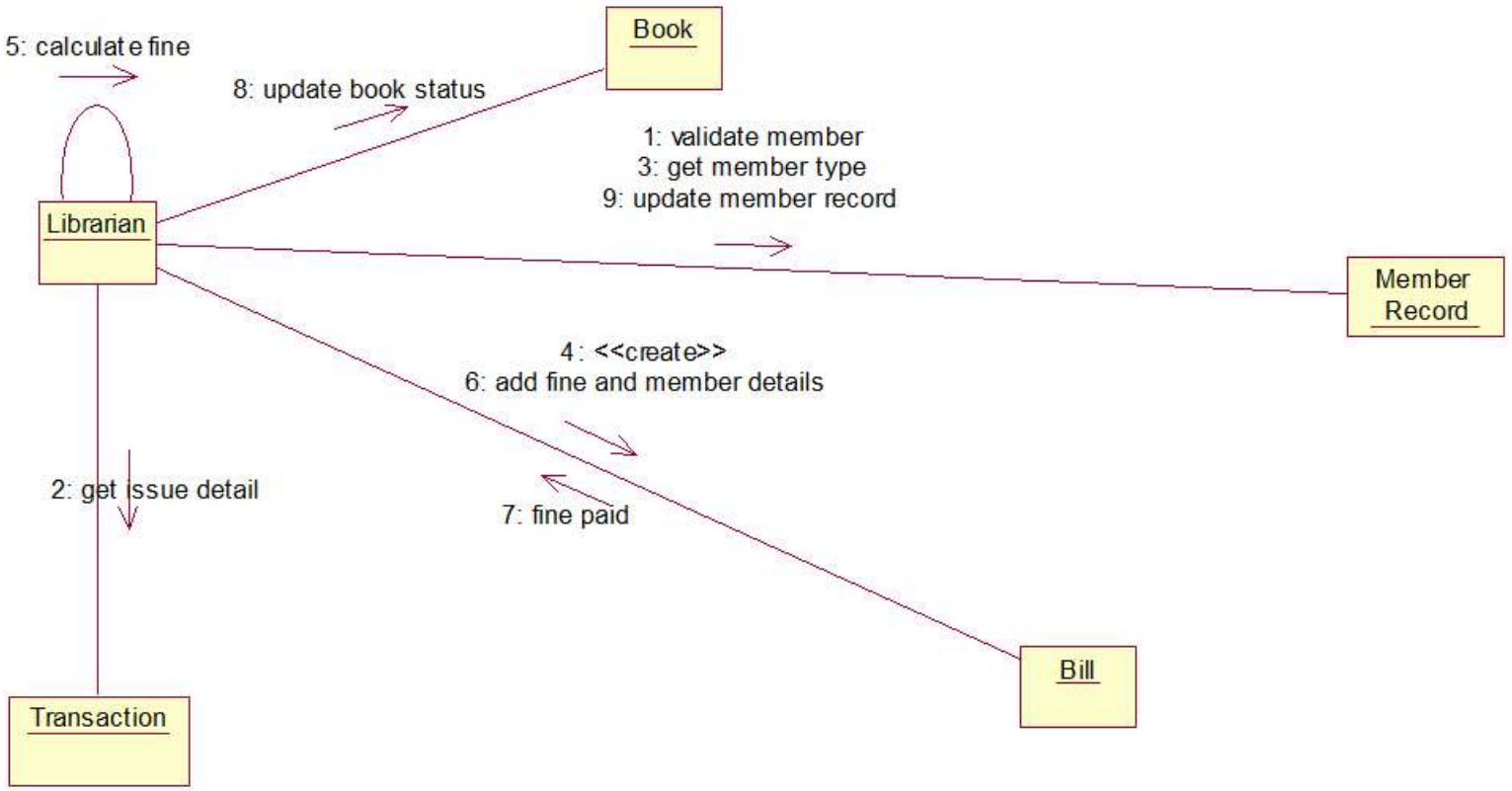
Telephone Call



Telephone Call



SAMPLE COMMUNICATION DIAGRAM



END.
Wairagu G.R