SOOAD

UNIT 5

OBJECT ORIENTED ANALYSIS & DESIGN

UNIT -5 Sequence, Collaboration, Activity & State Chart Diagram

- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- State Chart Diagram

SEQUENCE DIAGRAM

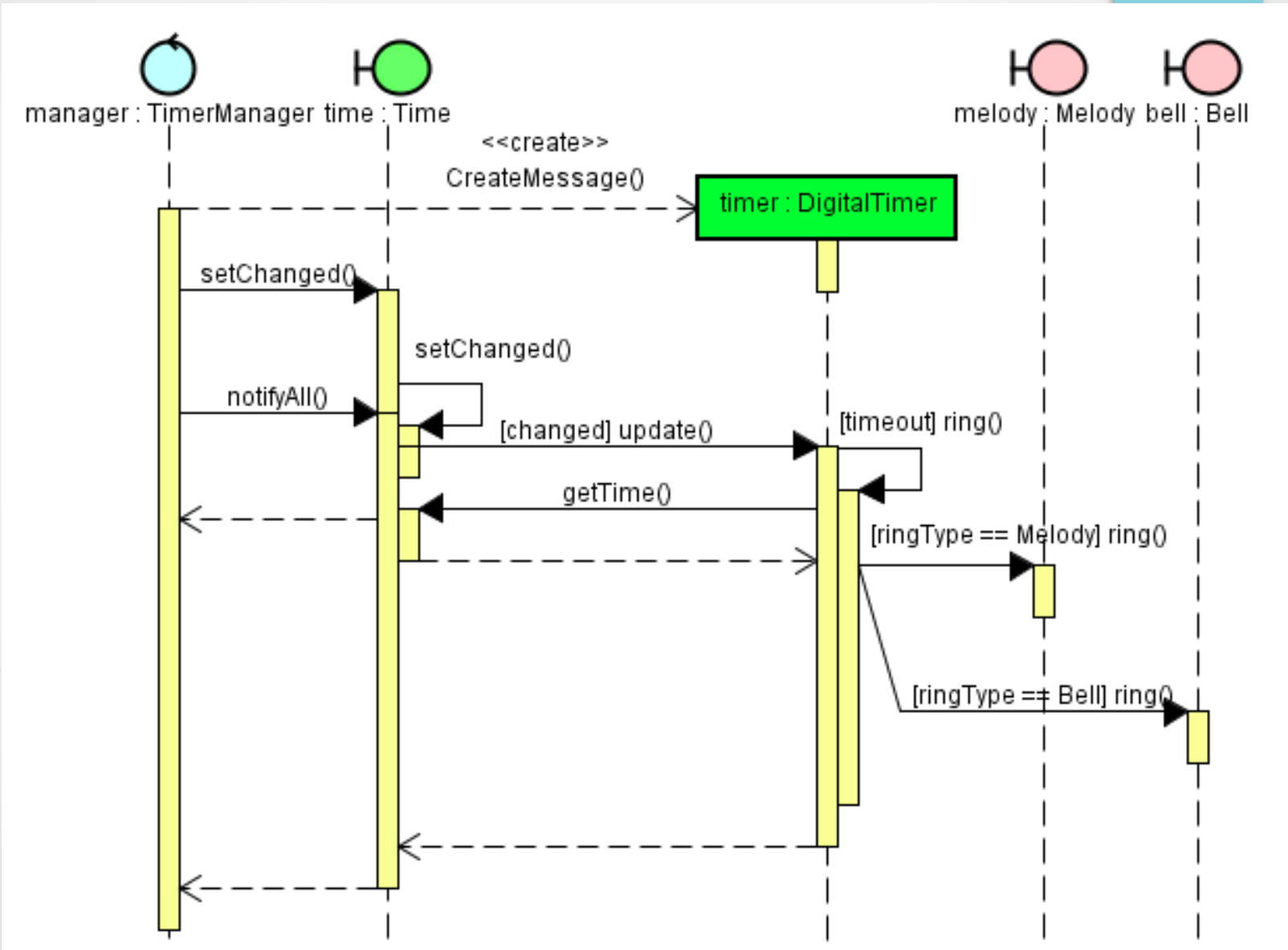
UNIT -5 Sequence Diagram

- Sequence Diagram
  - Introduction
  - Elements of Sequence Diagram
  - Guidelines for designing of Sequence Diagram
  - Case Study

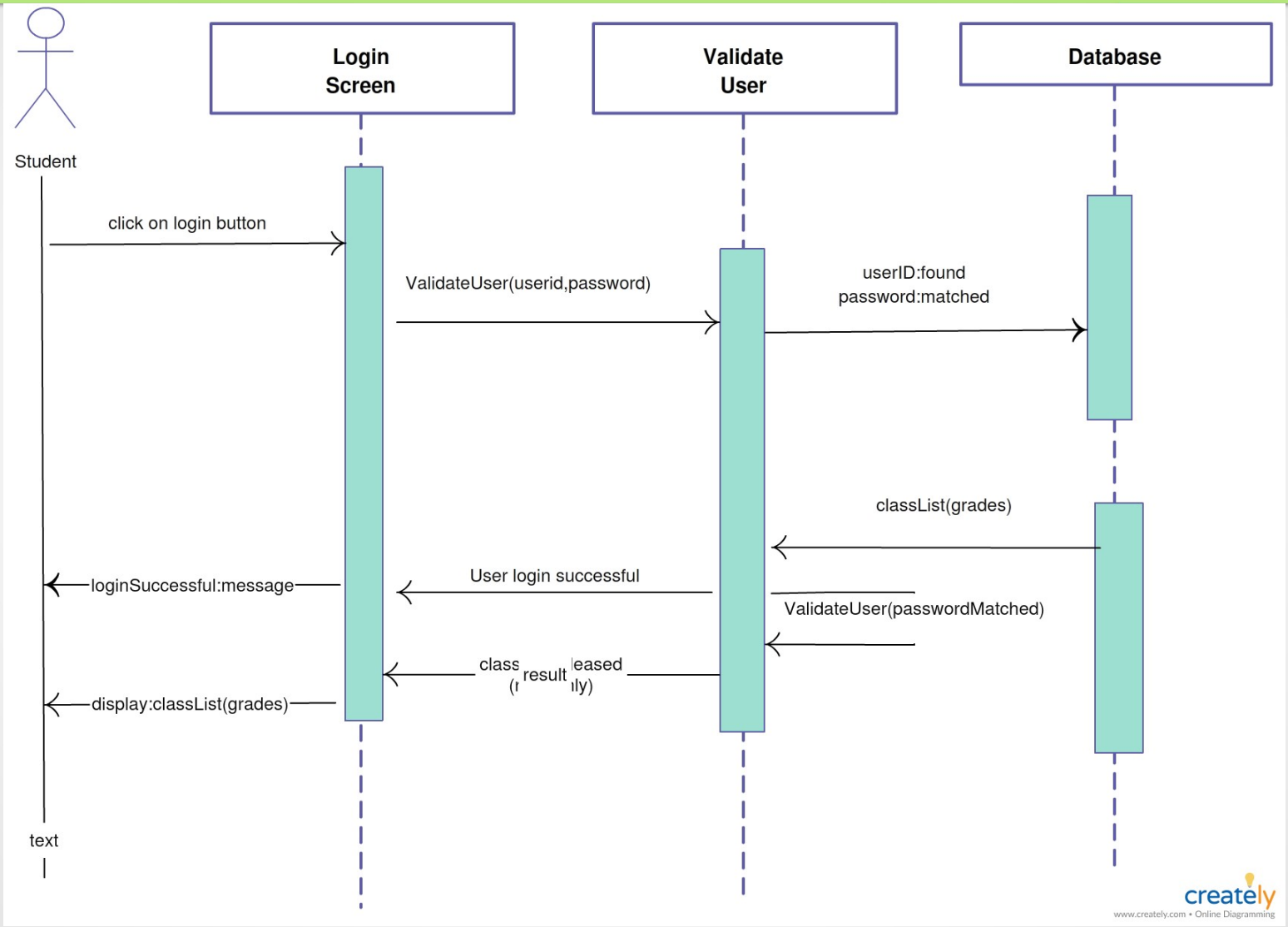
UNIT -5 Sequence Diagram

- **Introduction**
  - A sequence diagram is **used to express use-case realizations .**
  - It shows **how objects interact to perform the behaviour of all or part of a use-case.**
  - A sequence diagram is a type of interaction diagram because **it describes how—and in what order—a group of objects works together.**
  - The sequence diagram is used primarily to show the interactions between objects of a system, in the sequential order of the occurrence of interactions.

UNIT -5 Sequence Diagram



UNIT -5 Sequence Diagram



UNIT -5 Sequence Diagram

- **Introduction**
  - The diagram conveys this information **along the horizontal and vertical dimensions.**
  - **The vertical dimension shows, top-down, the time time sequence of messages / calls as they occure**
  - **The horizontal dimension show, left-right, the object instances that the meesages are sent to.**
  - A dynamic modelling diagram focusing on identifying the behaviour of related objects within the system.

UNIT -5 Sequence Diagram

- **Introduction**
  - Sequence diagrams are **used to give a picture of the design because they provide a way to visually step through invocation of the operations** defined by classes.
  - **Sequence diagrams are used to display the interaction between users, screens, objects and entities within the sysem.**

A sequential map of message passing between objects over time.

UNIT -5 Sequence Diagram

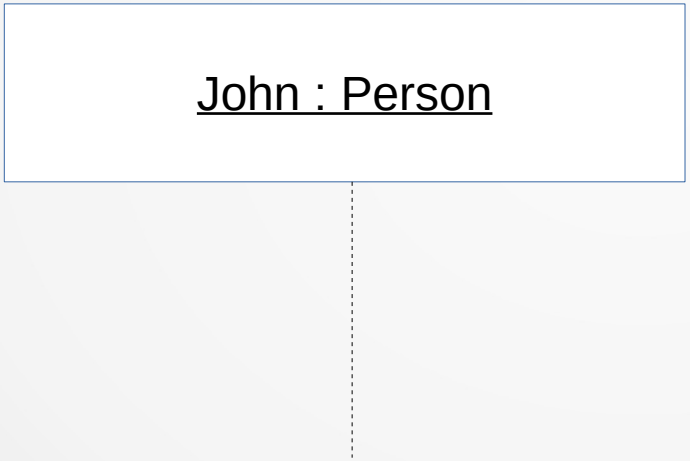
- **Elements of Sequence Diagram**
  - A Sequence diagram is **Two-dimensional** in nature.
  - **Vertical axis**, shows the **life of the object** that it represents.
  - **Horizontal axis,** shows the **sequence of the creation or invocation of these objects.**
  - **Elements**
    - **Life Lines**
    - **Messages**
    - **Activation**
    - **Guards**
    - **Combined Fragments**
    - **Objects**

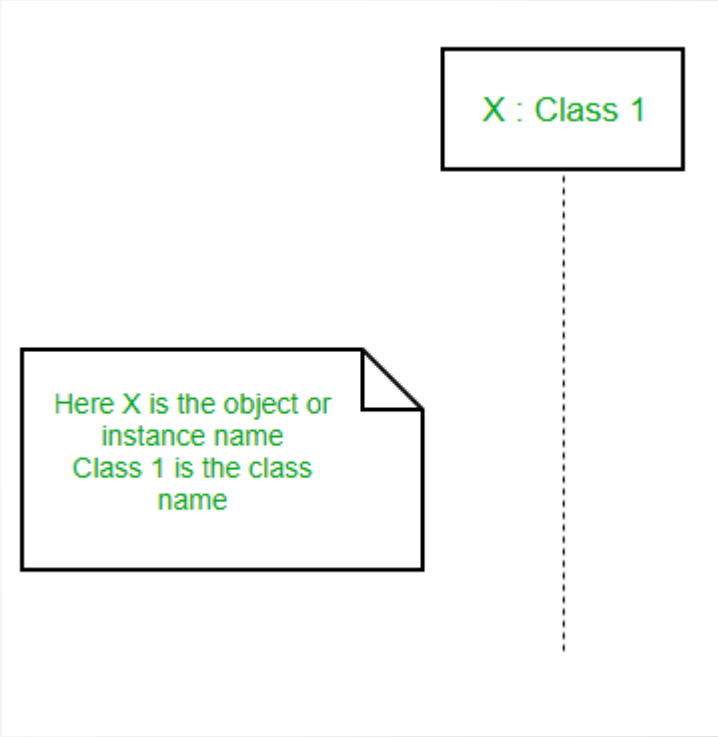
UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Life Lines**
    - **Lifelines** **represent either roles or instances that participate in the sequence** of interaction being modelled.
    - Lifeline notation elements are placed at the top of the diagram.
    - So basically each instance in a sequence diagram is represented by a lifeline.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Life Lines**





UNIT -5 Sequence Diagram

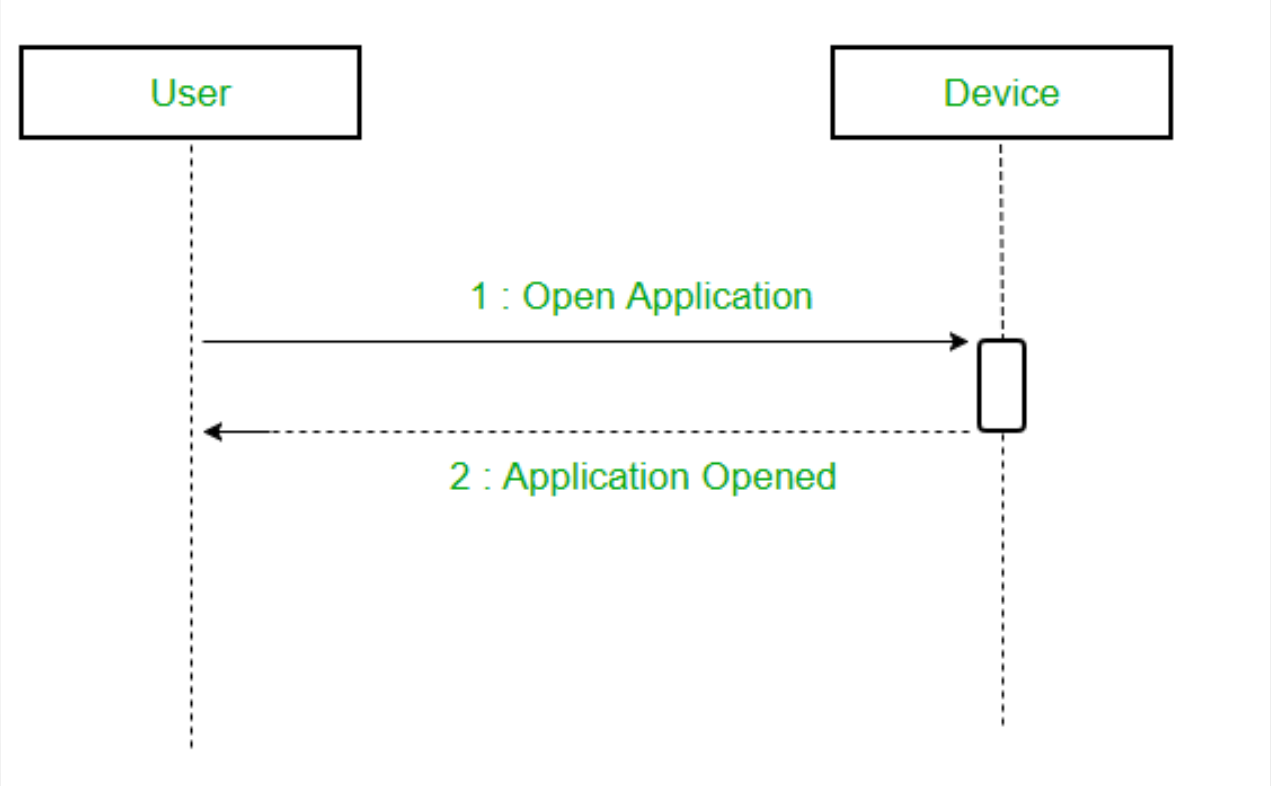
- **Elements of Sequence Diagram**
  - **Messages**
    - A Message defines a **specific kind of communication between instances in an interaction.**
    - **It represents a method call and invocation between the objects.**
    - **A message specifies the sender and the receiver, and defines the kind of communication that occurs between lifelines.**

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Messages**
    - Communnication can invoke or call using
      - **Synchronous Call (synchCall)**
      - **Asynchronous Call (asynchCall)**

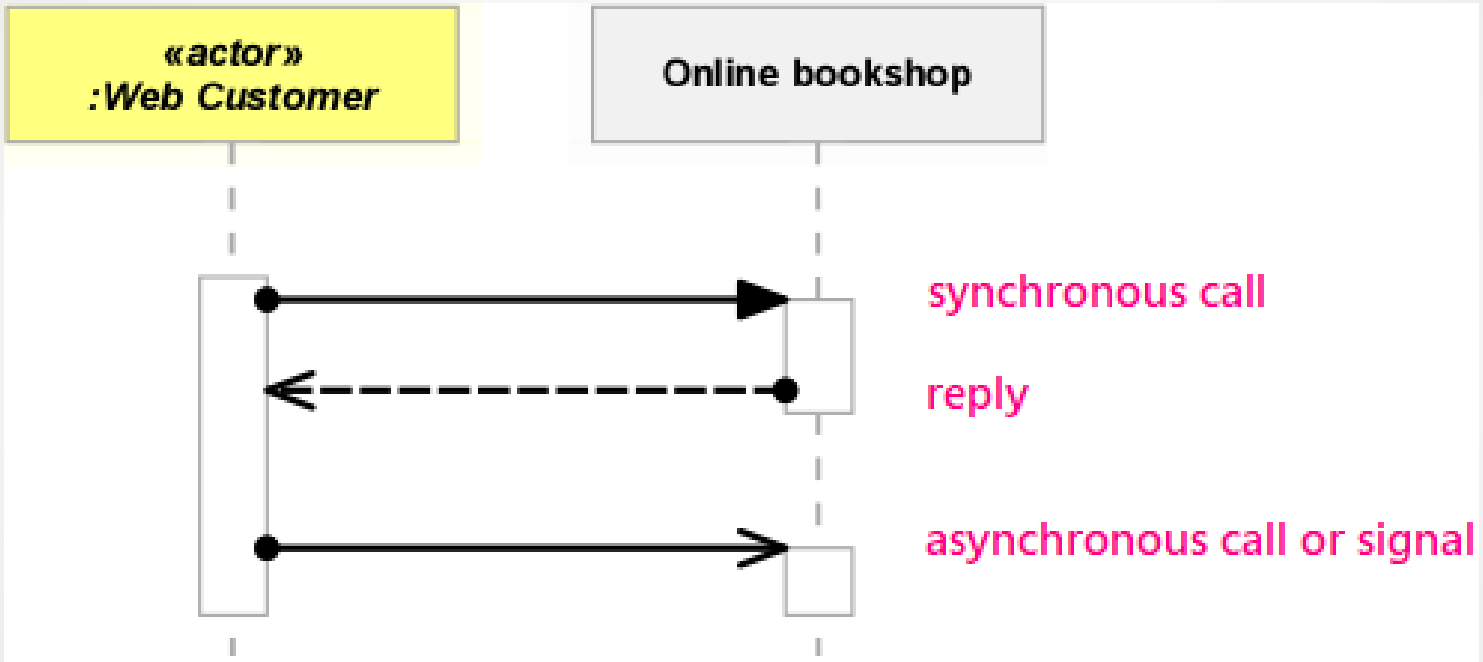
UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Messages**
    - **SynchCall**
      - 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.
      - We use a solid arrow head to represent a synchronous message.


User
Device
1 : Open Application
2 : Application Opened

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Messages**
    - **AsynchCall**
      - 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.


«actor»
:Web Customer
Online bookshop
synchronous call
reply
asynchronous call or signal

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Messages**

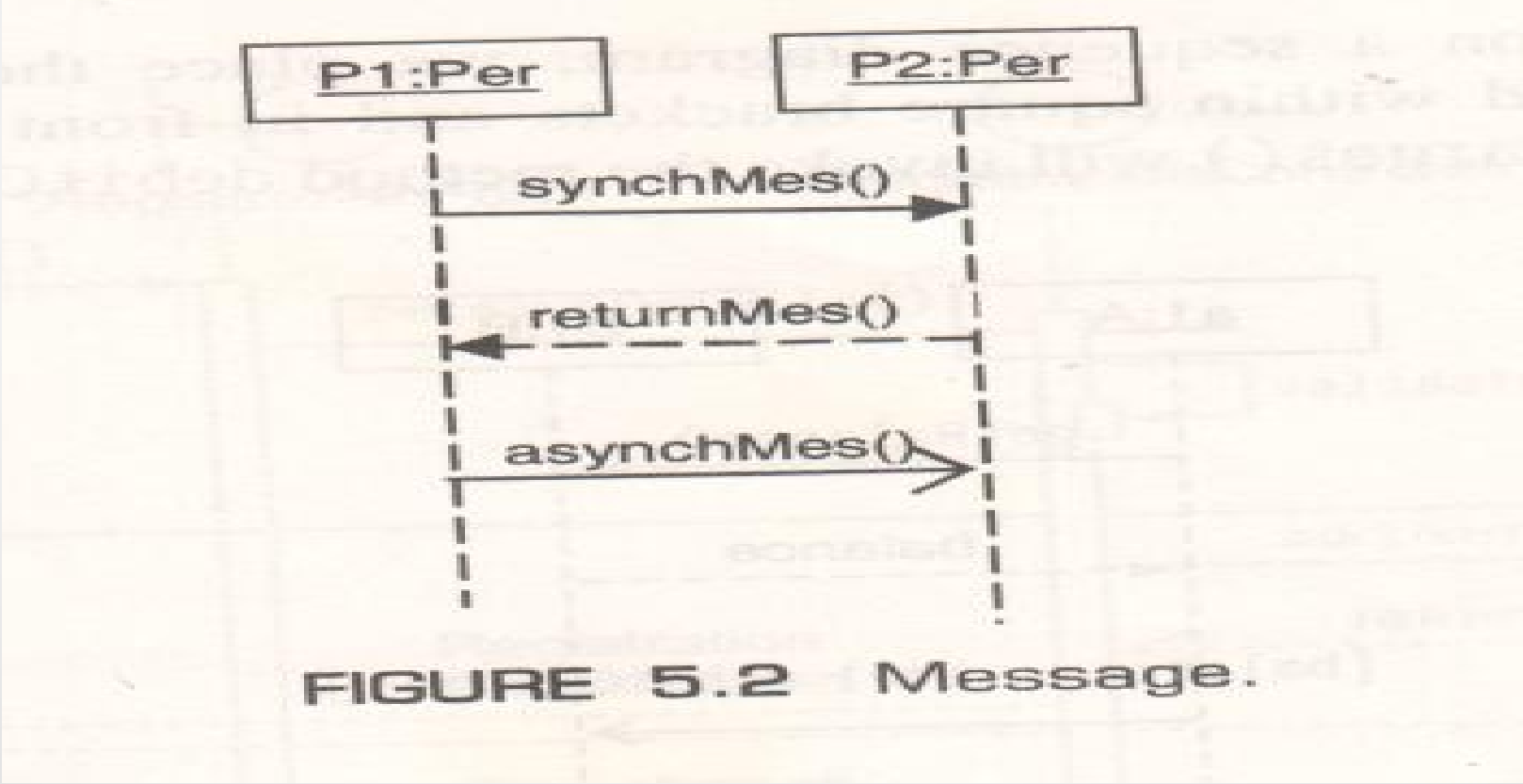


FIGURE 5.2 Message.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Activation**
    - Activation **boxes represent the time an object needs to complete the task.**
    - **Activation is created automatically when we create a synchronous or an asynchronous message.**
    - **Each activation represents an execution in a behaviour.**
    - **Activation boxes also called, *focus of control***

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Activation**

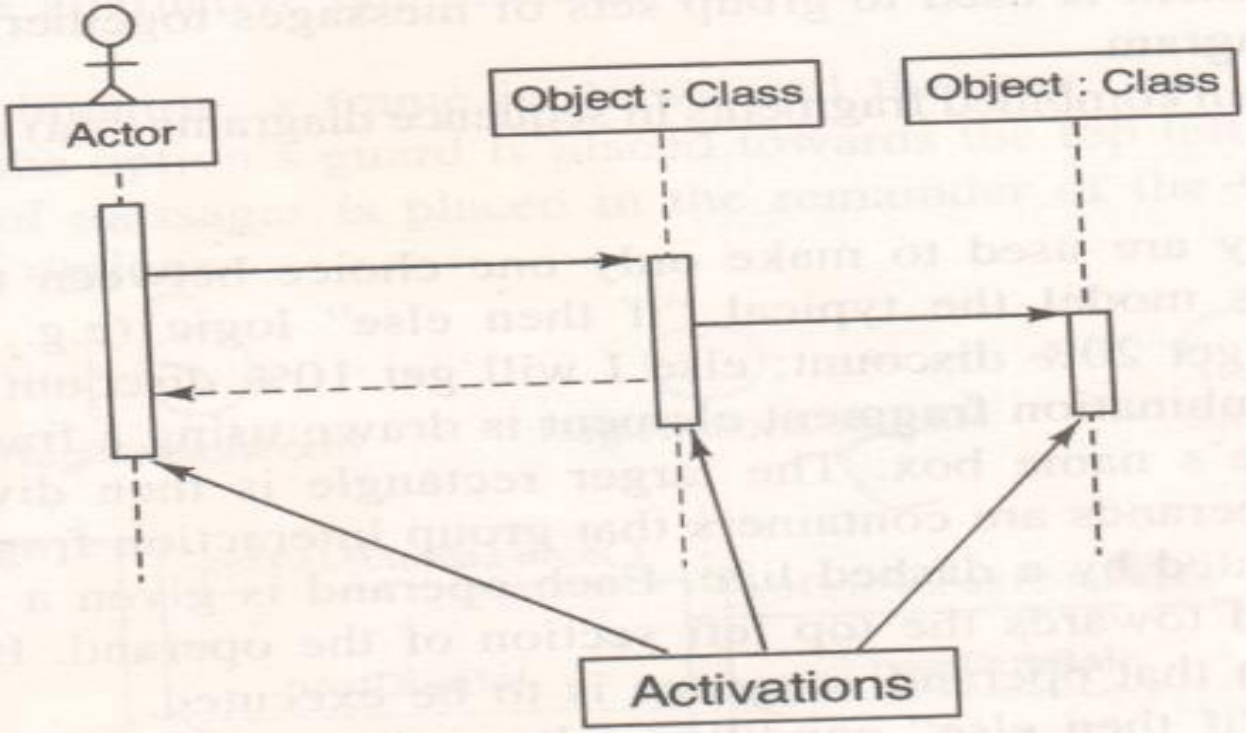


FIGURE 5.3 Activations.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Guards**
    - Guards **are the conditions used throughout UML diagrams to control flow.**
    - When modelling object interactions, there will be time when a conditions must be met for a message to be sent to the object.
    - To display a guard, we place the guard element above the message line being guarded within square brackets and in front of the message name.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Guards**

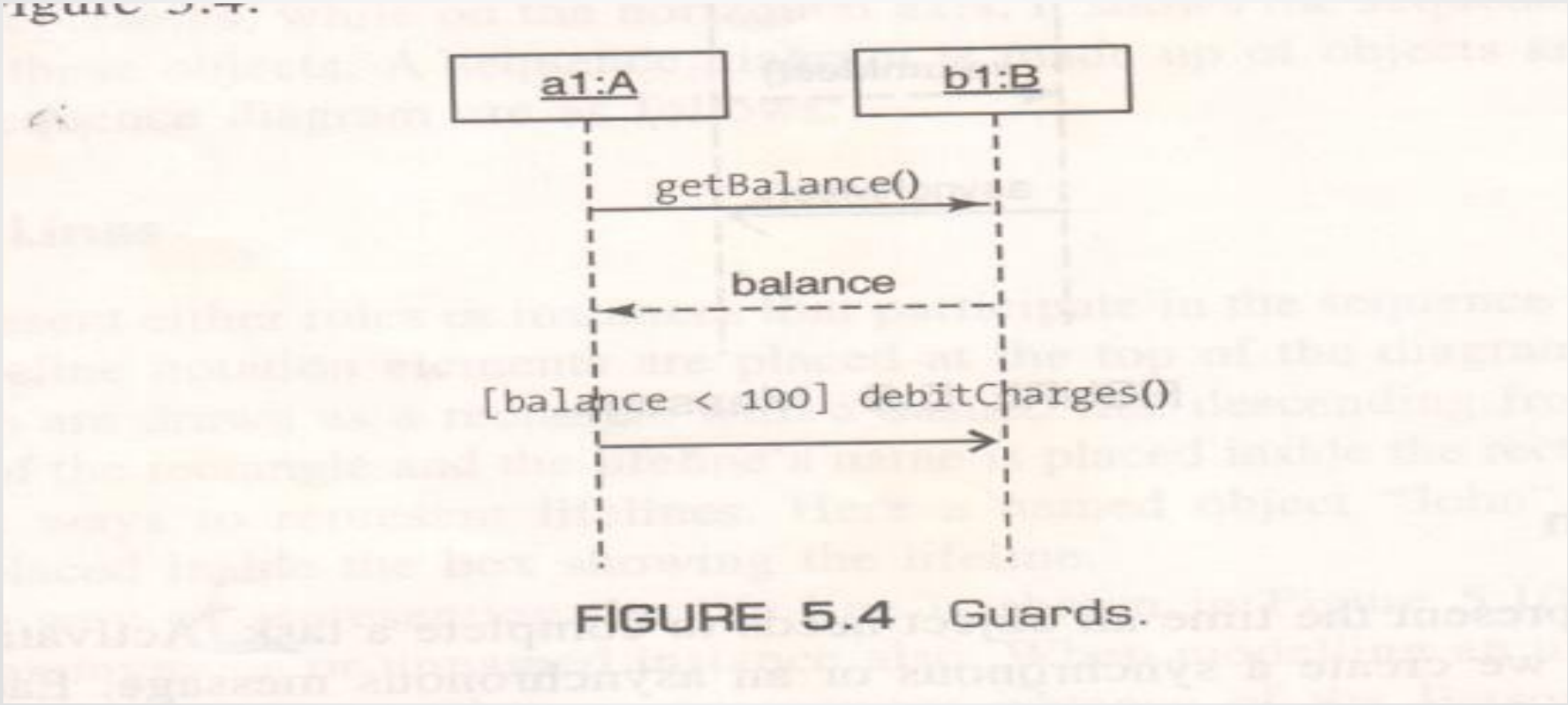


FIGURE 5.4 Guards.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Combined Fragments:**
    - **To add procedural logic to sequence diagrams.**
    - **Three main types:**
      - Alternatives
      - Options
      - Loops

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Combined Fragments - Alternatives**
    - They are used to make only **choice between two or more message sequence.**
    - **It is "if then else"**
    - Drawn using a frame.
    - An alternatives combination fragment element is drawn using a frame. The word **"alt" is** placed inside the frame's name box.
    - To have one or more alternative, an operand with that sequence's guard and messages must be added.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Combined Fragments -** Alternatives

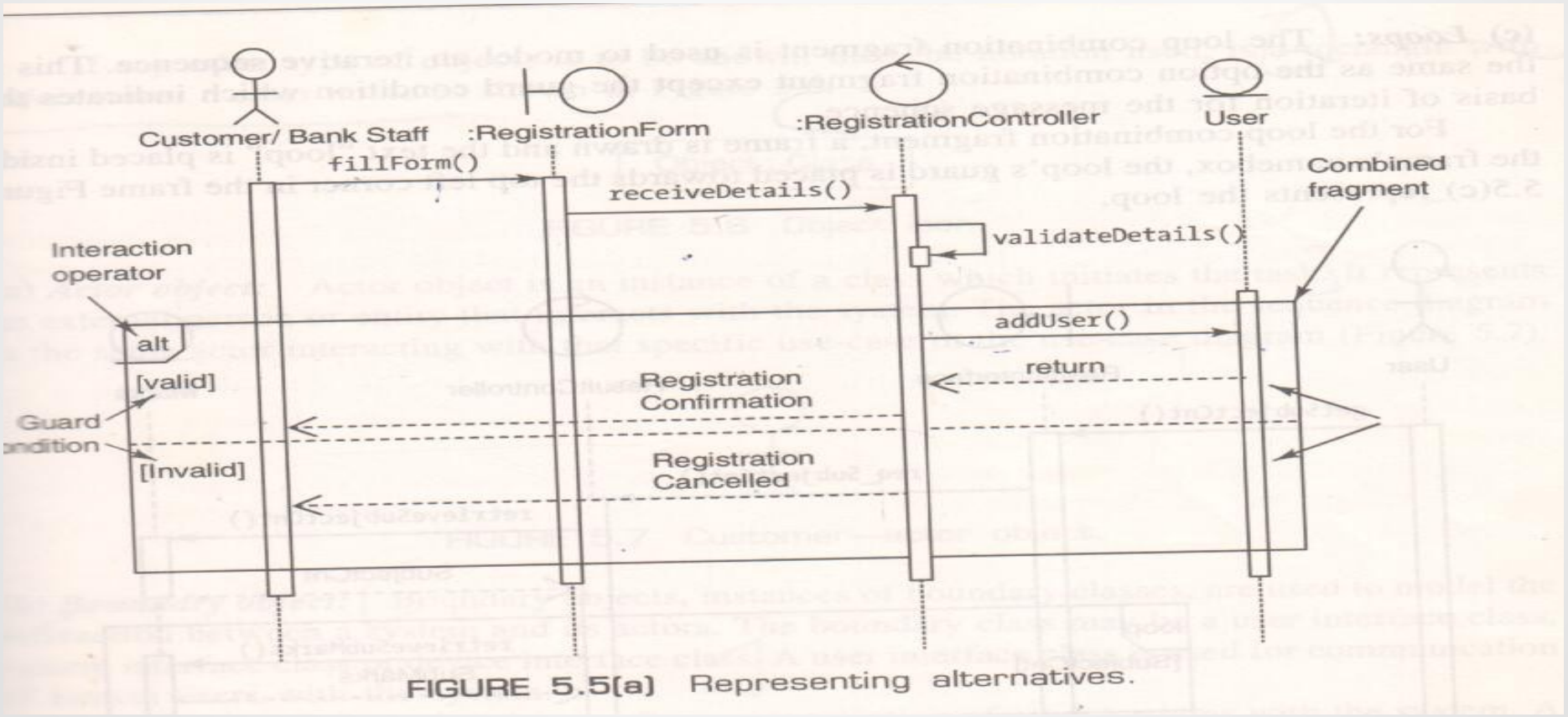


FIGURE 5.5(a) Representing alternatives.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Combined Fragments - Options**
    - The option combination fragment model a **sequence that will occur for a certain condition, other wise, the sequence does not occur.**
    - **It is "if then"**
    - The word **"opt" is** placed inside the frame's name box.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Combined Fragments -** Options

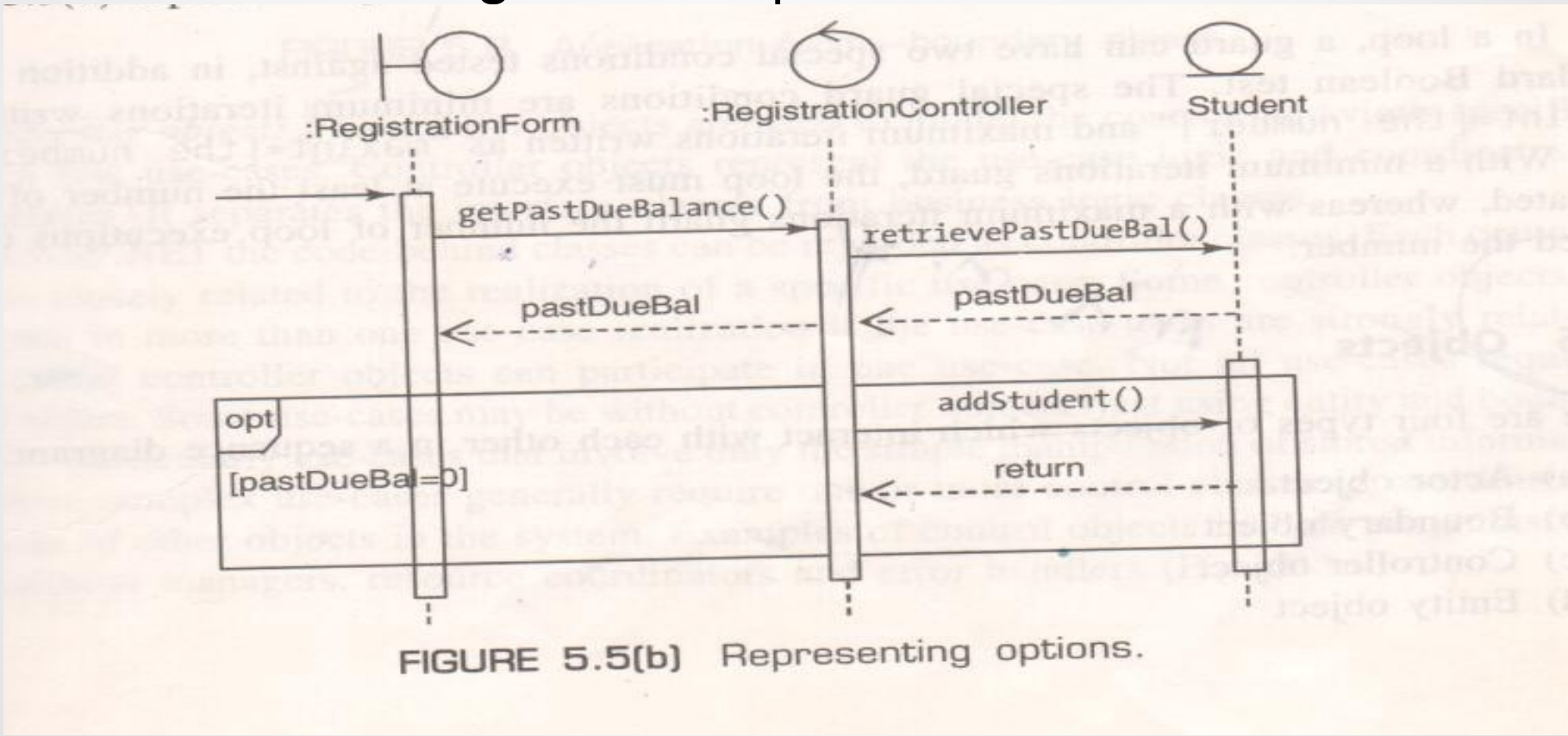


FIGURE 5.5(b) Representing options.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Combined Fragments - Loops**
    - The loop combinaton fragment is used **to model an iterative sequence.**
    - The word **“loop” is** placed inside the frame’s name box.
    - The guard can have two special conditions having
      - Minimum iterations “minint = [the number]”
      - Maximum iteration “maxint” = [the number]”

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Combined Fragments - Loops**
    - The loop combinaton fragment is used to model an iterative sequence.
    - The word **“loop” is** placed inside the frame’s name box.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Combined Fragments - Loops**

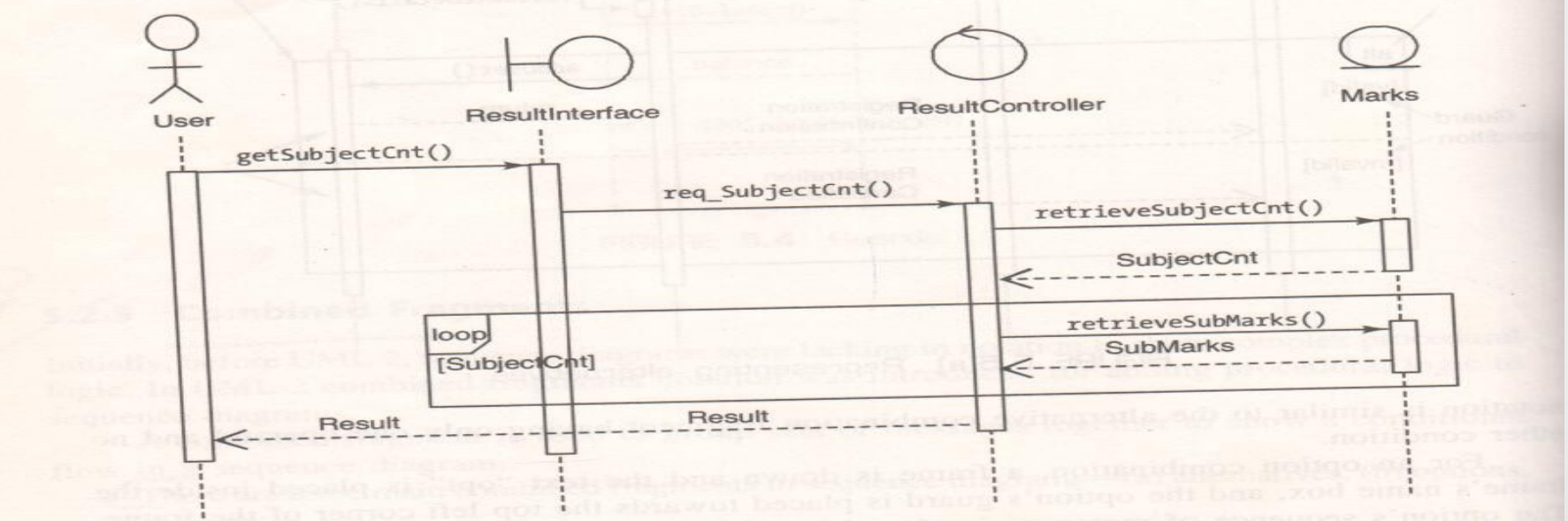
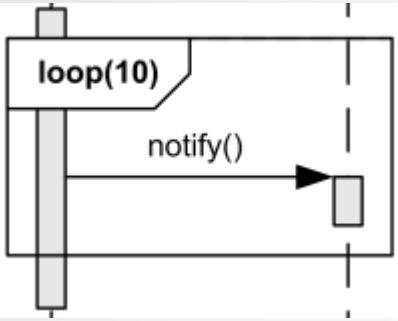
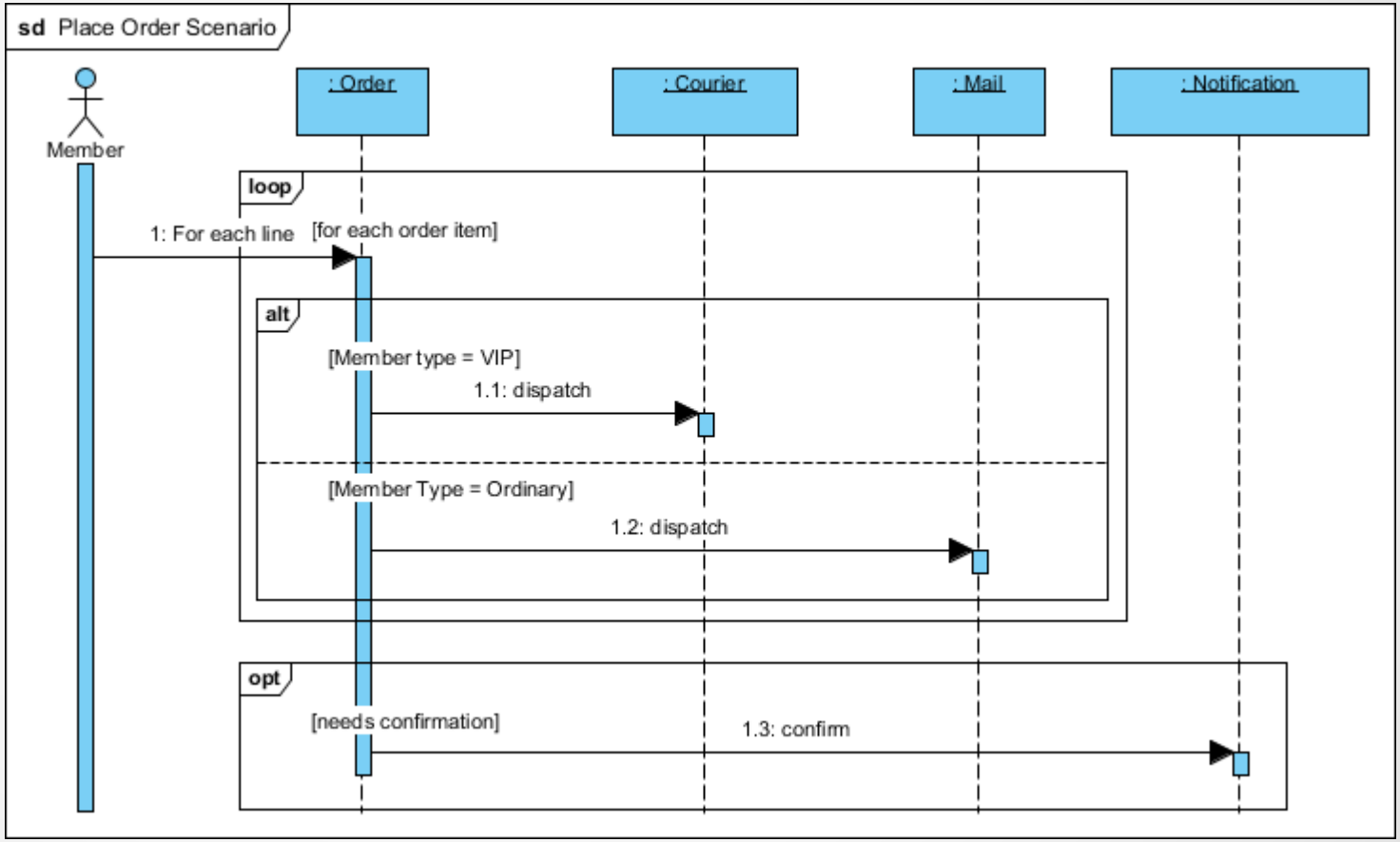


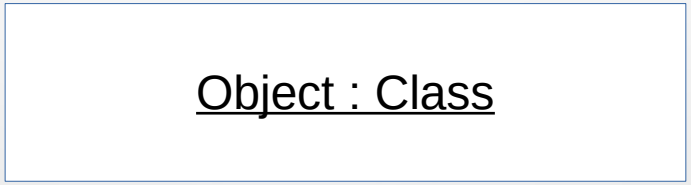
FIGURE 5.5(c) Representing loop.


loop(10)
notify()


sd Place Order Scenario
Member
: Order
: Courier
: Mail
: Notification
loop
1: For each line
[for each order item]
alt
[Member type = VIP]
1.1: dispatch
[Member Type = Ordinary]
1.2: dispatch
opt
[needs confirmation]
1.3: confirm

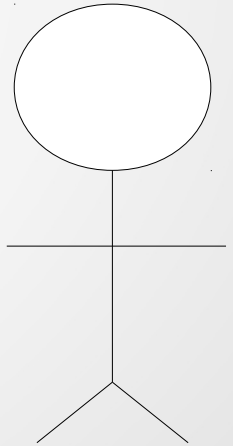
UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Objects**
    - There are four types of objects which interact with each other in a sequence diagram.
      - Actor Object
      - Boundary Object
      - Controller Object
      - Entity Object
    - If any other type of object is to be shown,then the notation used *object name : class name*



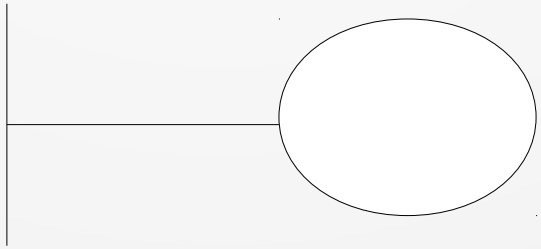
UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Objects – Actor**
    - Actor object is an instance of a class **which initiates the task.**
    - It represents an external person or entity that interacts with the system.



UNIT -5 Sequence Diagram

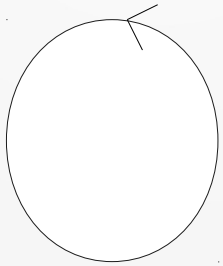
- **Elements of Sequence Diagram**
  - **Objects – Boundary Object**
    - Boundary objects, instance of boundary classes, are **used to model the interaction between a system and its actors.**
    - The **boundary class may be a user interface class, system interface class or device interface class.**



Windows, screens, web pages and menus are examples of boundaries that interface with users.

UNIT -5 Sequence Diagram

- **Elements of Sequence Diagram**
  - **Objects – Controller Object**
    - Controller objects **are used to model the control behaviour specific to one or a few use-cases.**
    - Controller objects **represent the use case logic and coordinates the other classes.**

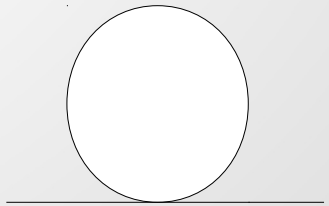


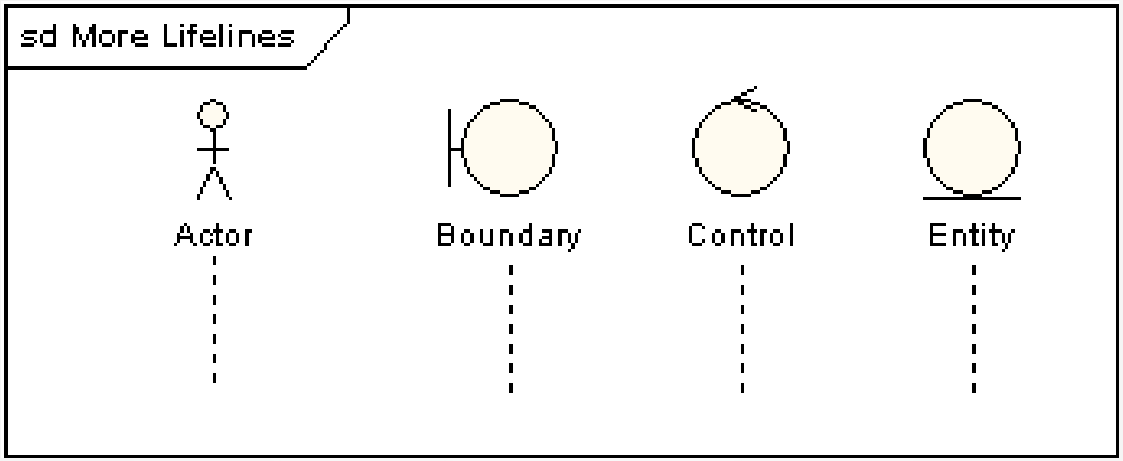
UNIT -5 Sequence Diagram

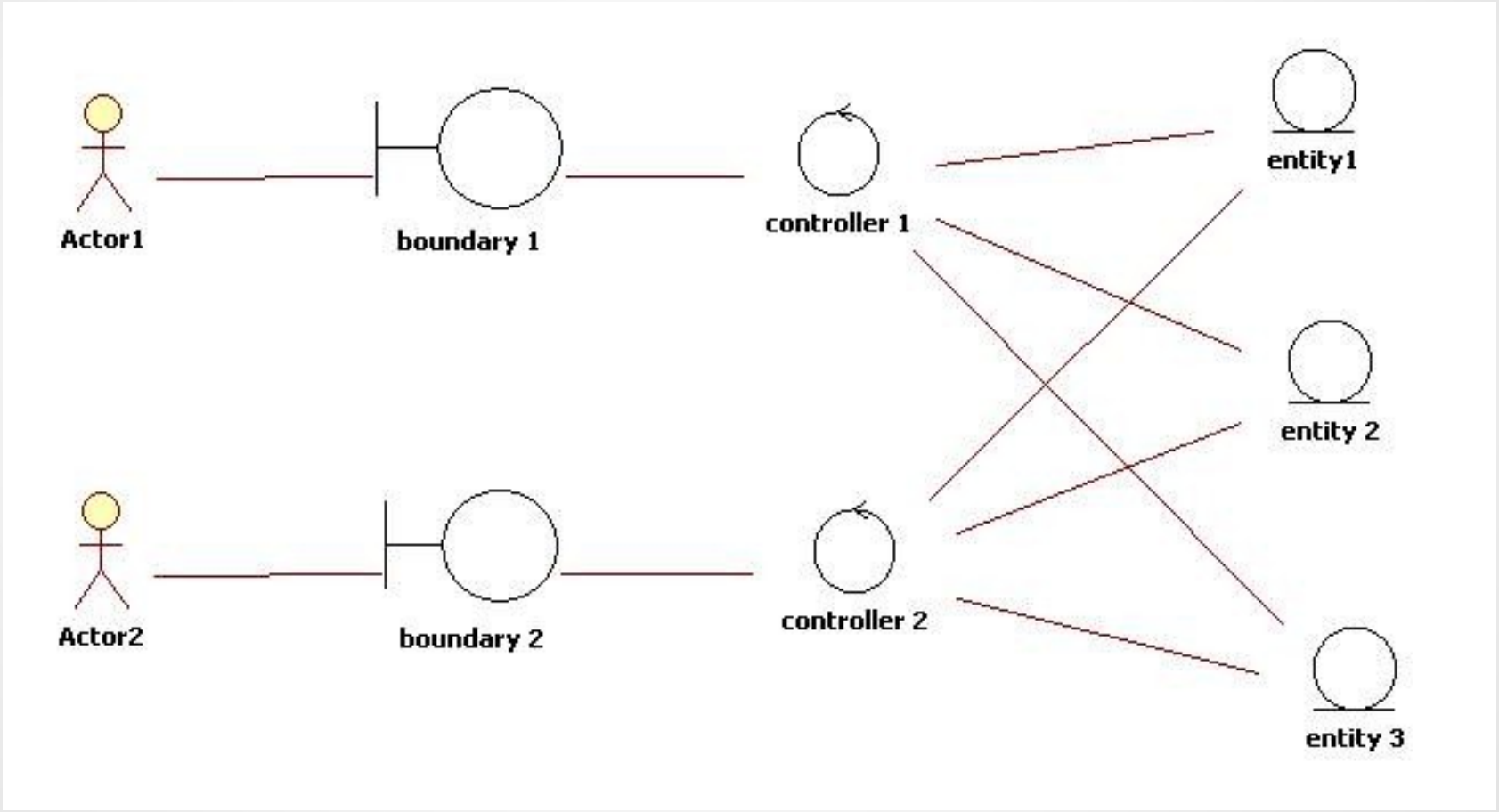
- **Elements of Sequence Diagram**
  - **Objects – Controller Object**
    - Objects that mediate between boundaries and entities.
    - Serve as the glue between boundary elements and entity elements, implementing the logic required to manage the various elements and their interactions.
    - Examples of **control objects include programs, such as transaction managers, resource coordinators and error handlers.**

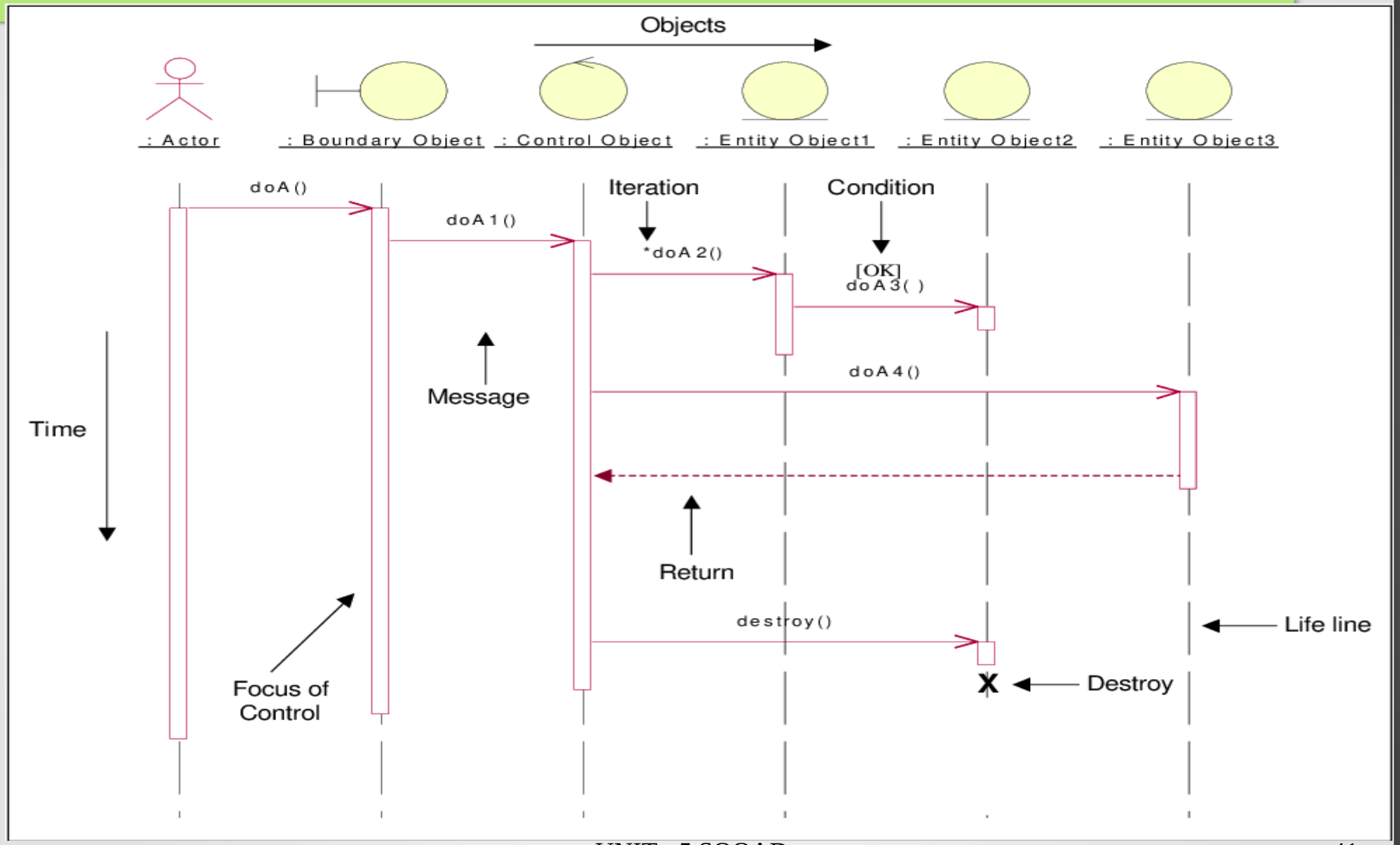
UNIT -5 Sequence Diagram

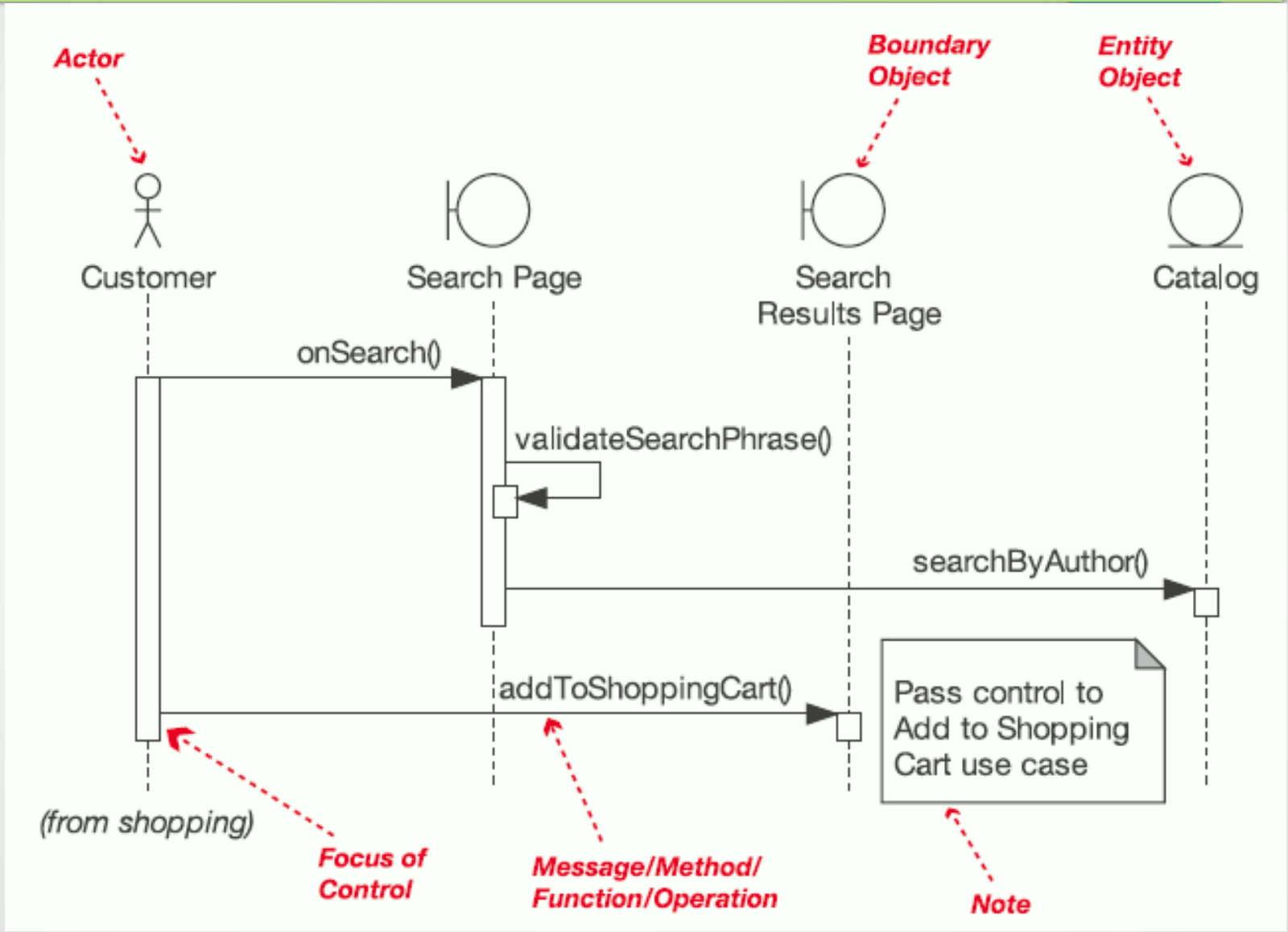
- **Elements of Sequence Diagram**
  - **Objects – Entity Object**
    - An **entity object is used to model information and associated behaviour that must be stored.**
    - It **used to hold and update information** about phenomenon.
    - Such as event, a person or some real-life object
    - Example: Account and cusomter




sd More Lifelines
Actor
Boundary
Control
Entity


Actor1
boundary 1
controller 1
entity1
entity 2
Actor2
boundary 2
controller 2
entity 3


Objects
: Actor
: Boundary Object
: Control Object
: Entity Object1
: Entity Object2
: Entity Object3
doA()
doA1()
Iteration
*doA2()
Condition
[OK]
doA3( )
doA4()
Message
Time
Return
destroy()
Life line
Focus of
Control
X
Destroy


Actor
Boundary Object
Entity Object
Customer
Search Page
Search Results Page
Catalog
onSearch()
validateSearchPhrase()
searchByAuthor()
addToShoppingCart()
Pass control to Add to Shopping Cart use case
(from shopping)
Focus of Control
Message/Method/ Function/Operation
Note

UNIT -5 Sequence Diagram

- **Guidelines for design of Sequence Diagram**
  - For **each use-case, one sequence diagram** is drawn.
  - Always have **one boundary class** per actor/use case pair, **one control class** per use-case.
  - Ordering of message sequence is always shown from **left to right**
  - An **actor name must be the same** as specified in a use case diagram.
  - An actor can have the same name as a class.
  - **Primary actors** must be specified on the **left-most side** of the diagram
  - **Reactive system actors** must be specified on the **right most side of diagram**

UNIT -5 Sequence Diagram

- **Guidelines for design of Sequence Diagram**
  - **Proactive system actors** must be specified on the **left most side** of diagram.
  - An **object can call itself recursively.**

CASE STUDY - 1

Milton Jewels Pvt. Ltd.

Milton Jewels has specialized in online jewellery retail since 1998, selling wonderful ranges of both children's and women's jewellery. A customer can register online so that he/she can check the status of the placed order. A customer can purchase any jewellery item online either by using his/her existing account or as an anonymous user specifying shipping address and contact information. Customer can only check the status of his/her order if he/she creates an account. The customer will pay online through credit card or debit card and the order will be delivered on the shipping address within one week.

Milton Jewels Pvt. Ltd.

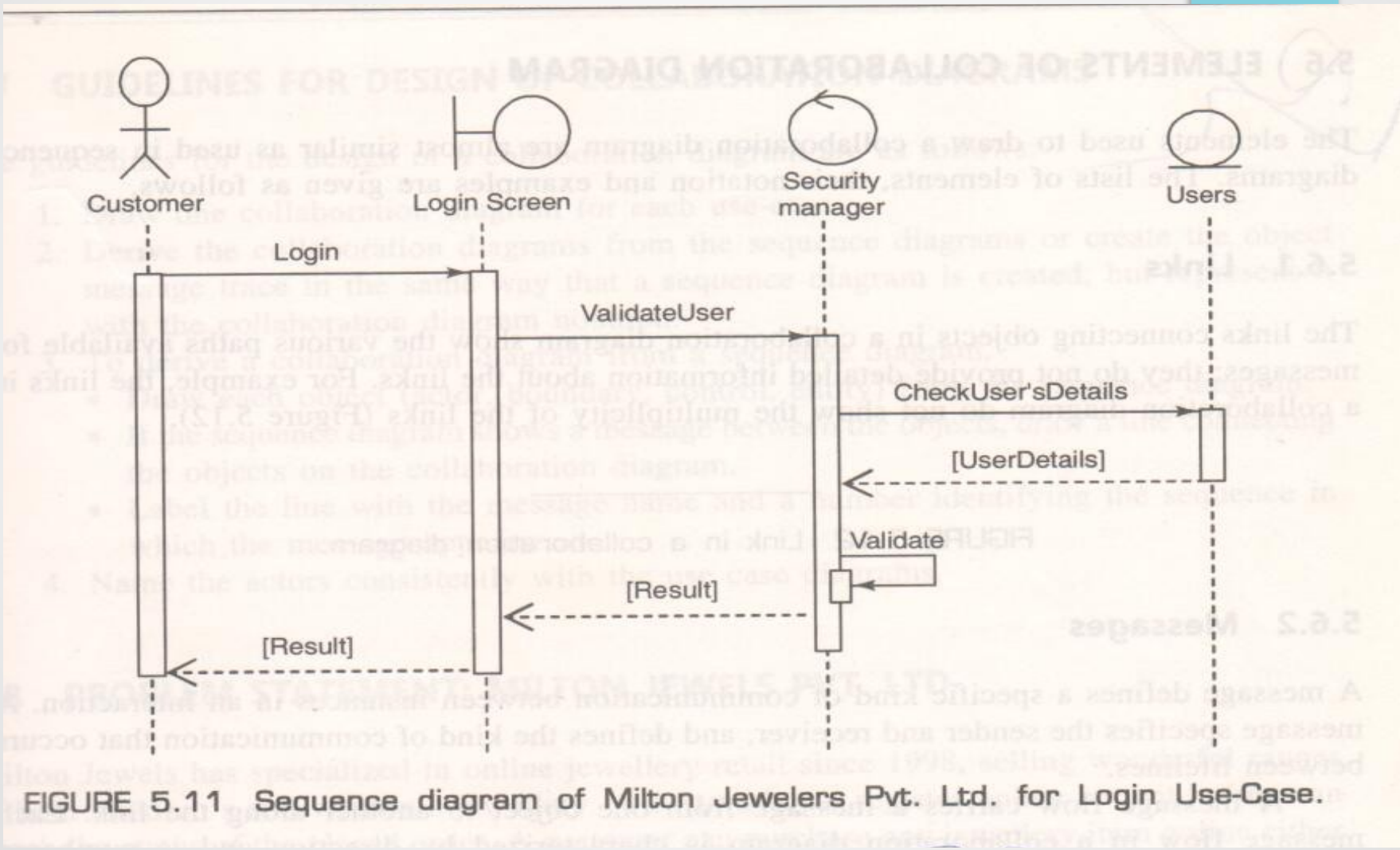
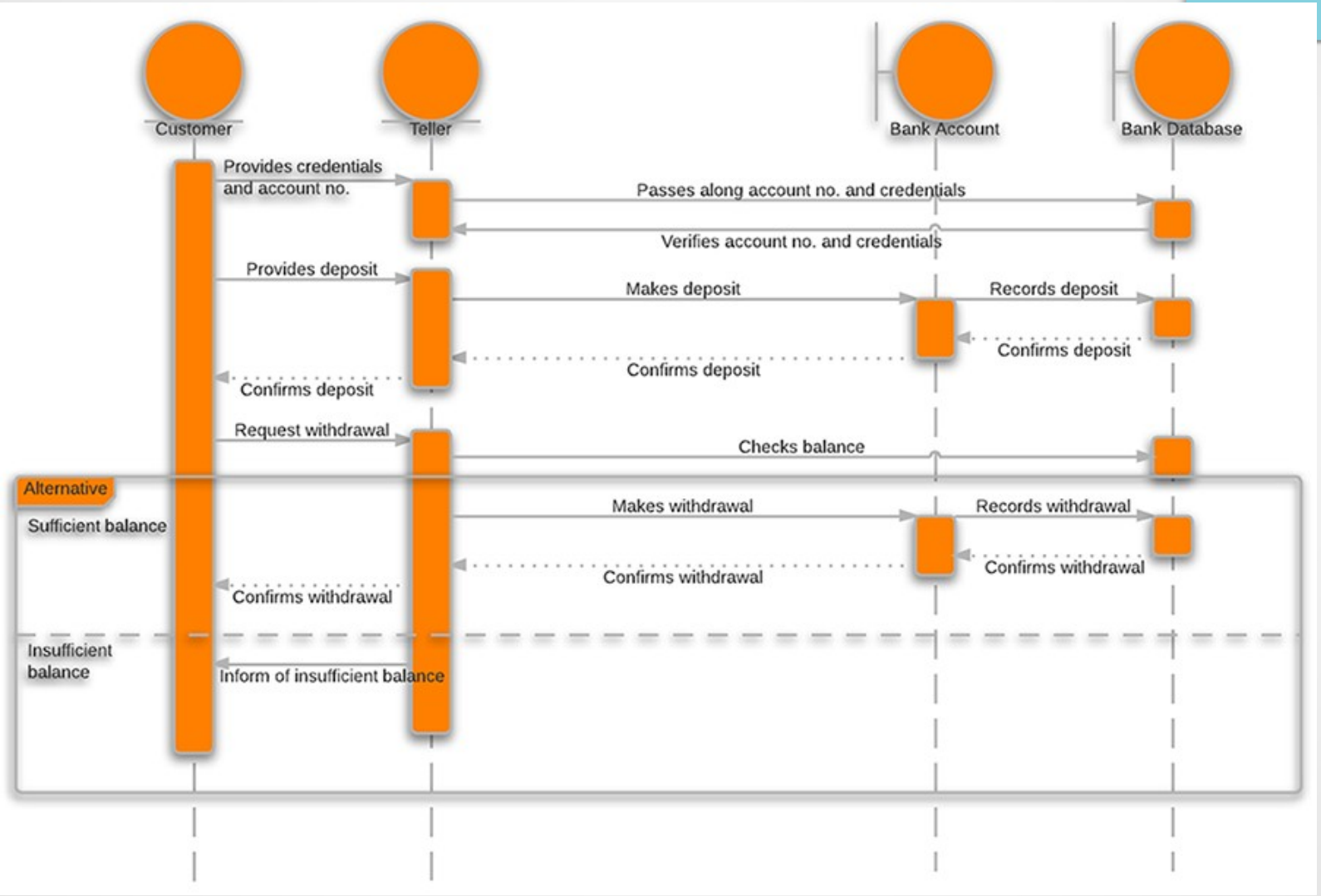


FIGURE 5.11 Sequence diagram of Milton Jewelers Pvt. Ltd. for Login Use-Case.

CASE STUDY 2

BANKING SYSTEM



CASE STUDY 3

STUDENT LOAN SYSTEM

- A University gives loans to students. Before getting a loan, there is an evaluation process after which if the loan is approved, agreement is reached. A transaction records each step of the evaluation process, and another transaction records the overall loan agreement. A student can take any number of loans, but only one can be active at any time. Each loan is initiated by a seperate transaction. Then, the student repays the loan with a series of repayments. Each repayment transaction is recorded. After the complete settlement, finally the loan account is closed.

  Two output functions are desired:

  1. an inquiry function that prints out the loan balance for any student

STUDENT LOAN SYSTEM

2. a repayment acknowledgement sent to each student after payment is received by the university.

The university loan office decides to implement the student loans on a single processor. Inquires should be processed as soon as they are received. However, repayment acknowledgements need only be processed at the end of each day.

For the above application,create appropriate diagrams.

- CHECK PDF OF STUDENT LOAN SYSTEM