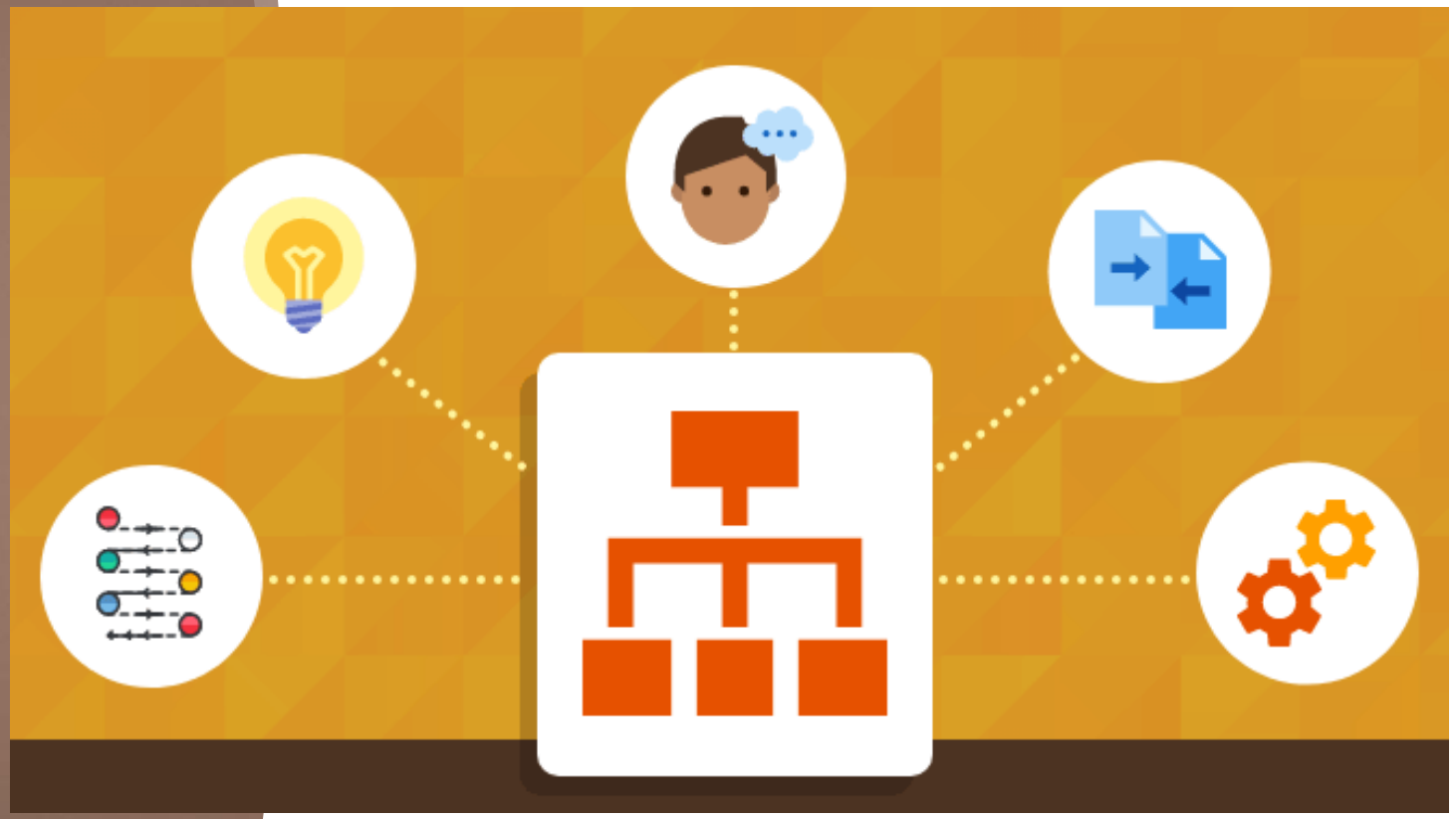
SCS1207
Software Engineering I

H N D THILINI

System Modeling



Objectives

Aim: Introduce system models developed as part of **requirements engineering** and **system design.**

At the end of this lecture you will understand

- how graphical models can be used to represent software systems,
- why several models are required to fully represent the system,
- the fundamental system modeling perspectives,
- The principal diagram types in UML and how these may be used in system modeling.

Topics Covered

- Context models
- Interaction models
- Structural models
- Behavioral models

System Modeling

- The process of developing abstract models of a system
- Each **presenting a different view** or perspective of the system.
- **Represent** a system using some kind of **graphical notation**, mostly based on notations in the Unified Modeling Language (UML).
- Helps the analyst to understand the functionality of the system
- Models are used to communicate with customers.

System Models

- Models of the existing system
  - Used to clarify **what the existing system does** and can be used as a basis for discussing its strengths and weaknesses.
  - Can lead to requirements for the new system.
- Models of the new system
  - Used to help explain the **proposed requirements to other system stakeholders**.
  - Engineers use these models to discuss document the system for implementation.
- In a model-driven engineering process,
  - it is possible to generate a complete or partial system implementation from the system model.

System Modeling – Different Perspectives

- **External** perspective
  - model the *context or environment* of the system.
- **Interaction** perspective
  - model the *interactions between a system and its environment*, or between the components of a system.
- **Structural** perspective
  - model the *organization of a system* or the structure of the data that is processed by the system.
- **Behavioral** perspective
  - model the *dynamic behavior of the system* and how it responds to events.

Unified Modeling Language (UML)

- **Standard language** for object-oriented modeling.
- Introduced by, Booch, Rumbaugh and Jcobson (2004,2005)
- The UML has 13 diagram types
  - supports the creation of many different types of system model.
- Five main UML diagrams can be used to represent the essentials of a system.

UML diagram types

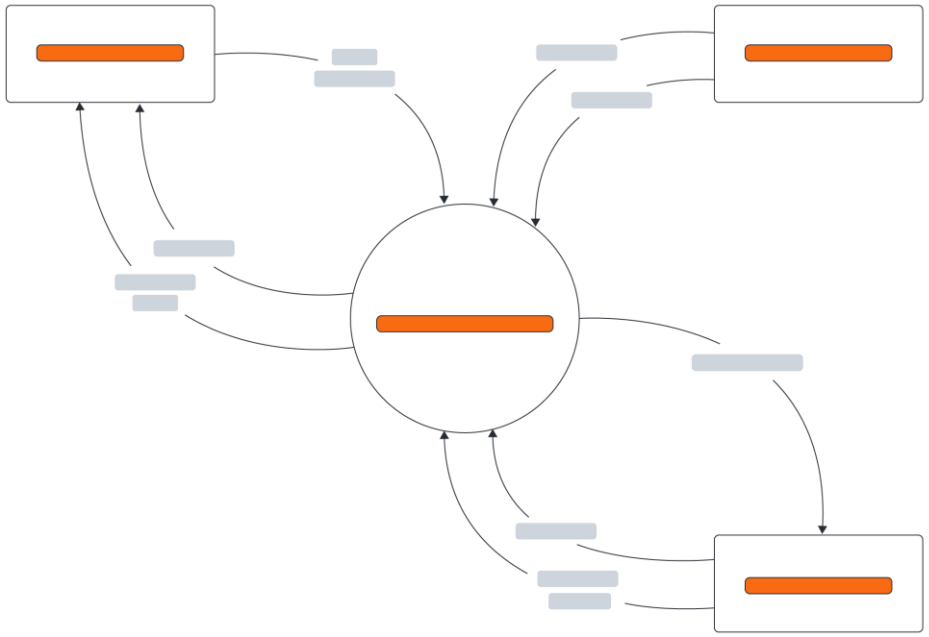
- Use case diagrams
  - show **the interactions** between a **system** and its **environment**.
- Sequence diagrams
  - show **interactions** between **actors** and between system **components**.
- Activity diagrams
  - show the **activities involved in a process** or in data processing
- State diagrams
  - show how the system **reacts** to **internal** and **external events**.
- Class diagrams
  - show the **object classes** in the system and the **associations** between these classes.

CONTEXT MODELS

Decide on the System boundaries- what is and is not part of the system

Context models

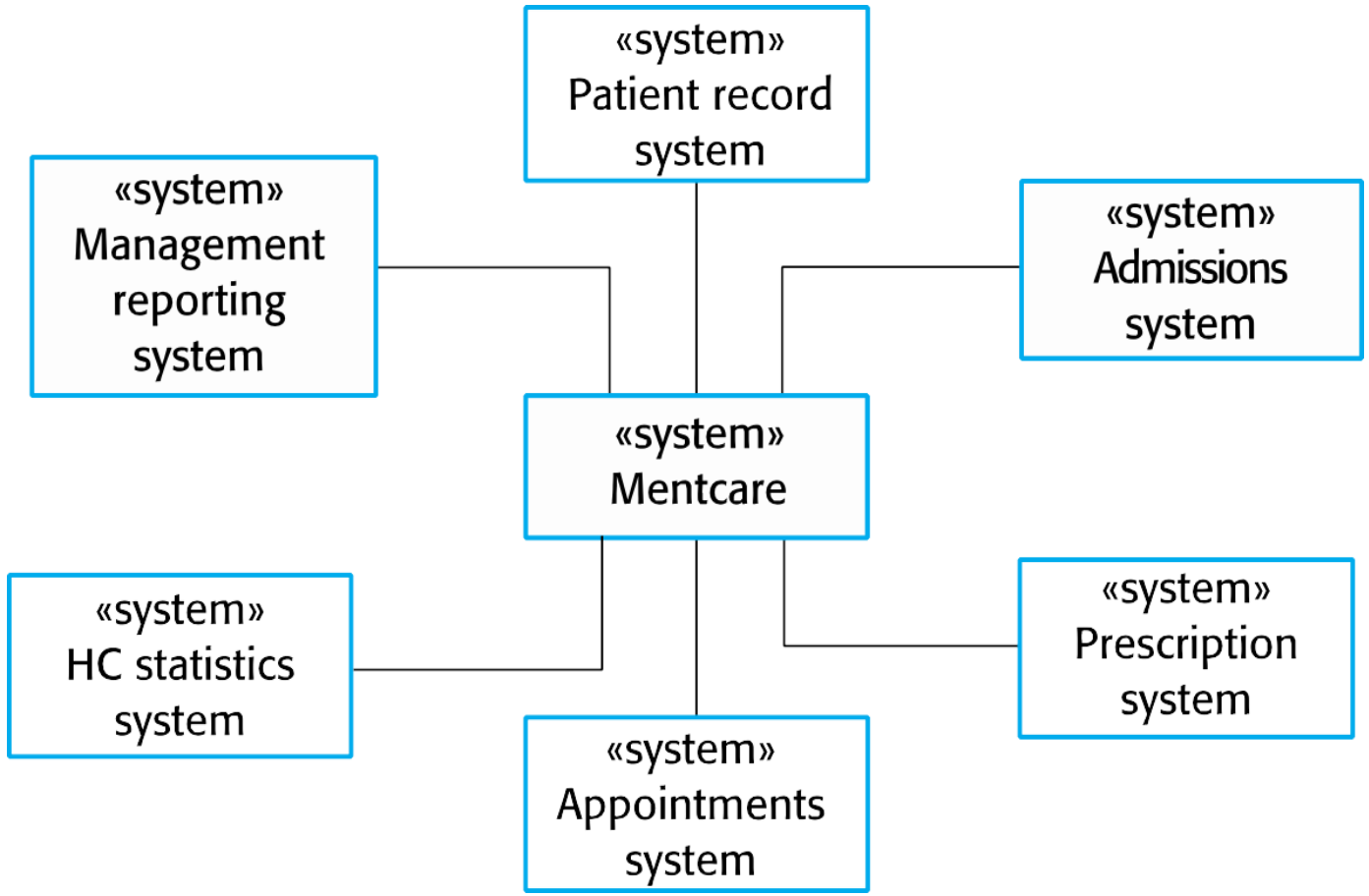
- Used **to illustrate the operational context of a system**
  - show what lies outside the system boundaries.
- **Social and organizational** concerns may affect the decision on where to position system boundaries.



System boundaries

- System boundaries are established to define what is inside and what is outside the system.
  - They show other systems that are used or depend on the system being developed.
- The position of the system boundary has a profound effect on the system requirements.
- Defining a system boundary is a political judgment
  - There may be pressures to develop system boundaries that increase / decrease the influence or workload of different parts of an organization.

The context of the Mentcare system



Context Models

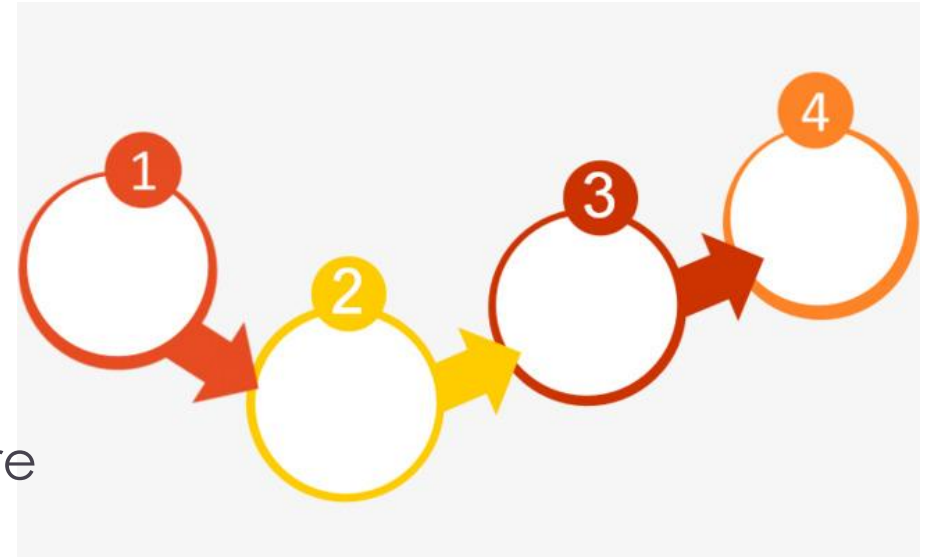
- Show the environment and other automated systems.
- Do NOT show the types of relationships between the systems in the environment.
- External systems may
  - Produce data for or consume data from the system
  - Share data / connected directly
  - Physically co-located or located in separate buildings
- All theses requirements may affect the requirements.
- Model these different models are used.

INTERACTION MODELS

Sequence Diagrams, Collaboration Diagrams

Interaction models

- All systems involve interaction of some kind.
- There are different types of interactions;
  - User interactions-involve user inputs and outputs
  - Interactions between the software and other systems in the environment
  - Interactions between the components of a software system

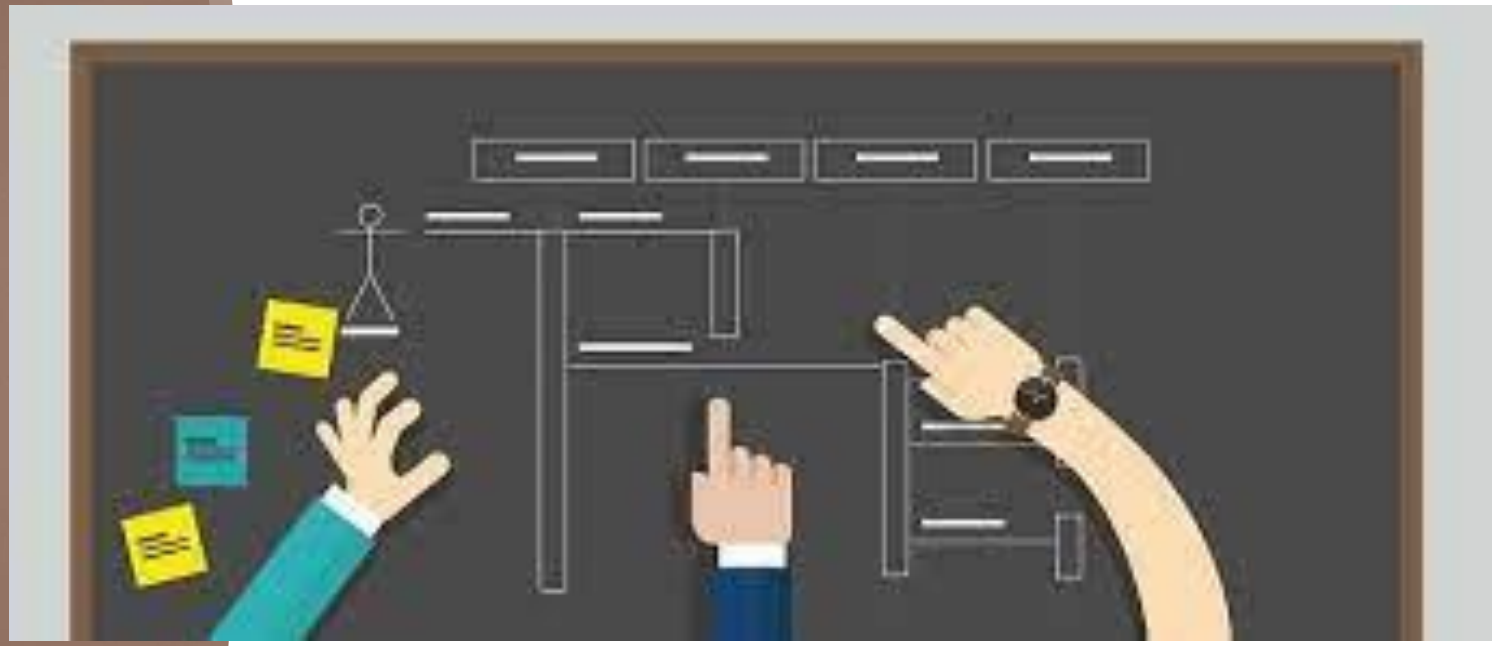


- Show how a set of **actors and objects communicate with each other** to perform the steps of a use case, or of some other piece of functionality.

Interaction model

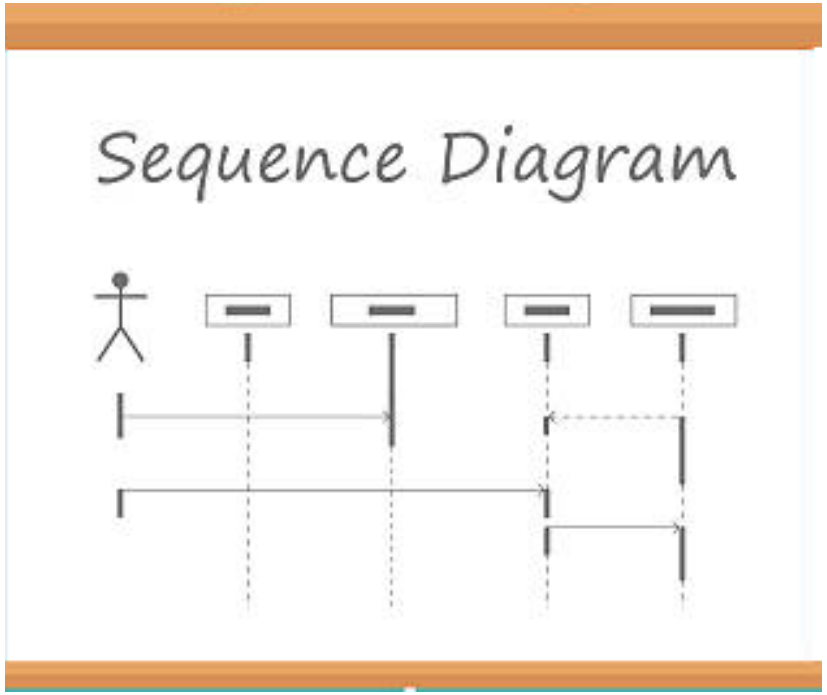
- Elements in an interaction diagram:
  - **Instances of classes or actors:** instances of classes (i.e. objects) are **shown as boxes** with the class and object identifier underlined
  - **Messages: different types of communication;** These are **shown as arrows from actor to object, or from object to object.**
- The main objective
  - **to better understand the sequence of messages.**

SEQUENCE DIAGRAMS



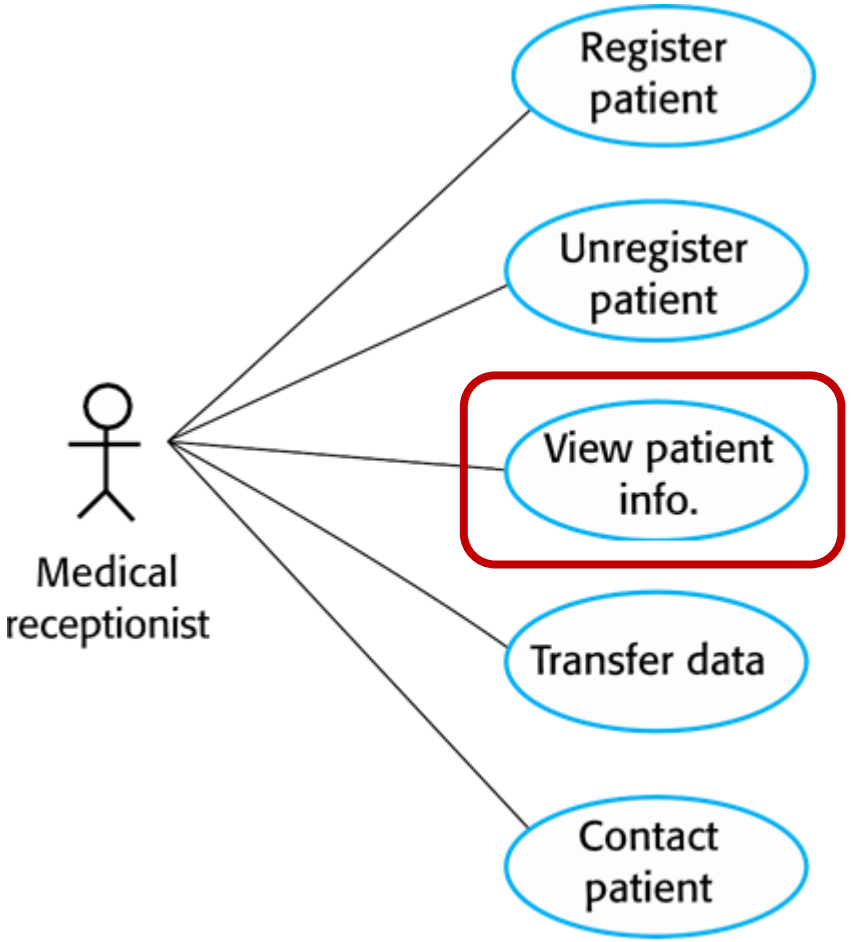
Sequence Diagrams

- Commonly used by developers.
- Model the interactions between objects in a single use case.
- Illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.
- Shows different parts of a system work in a ‘sequence’ to get something done.



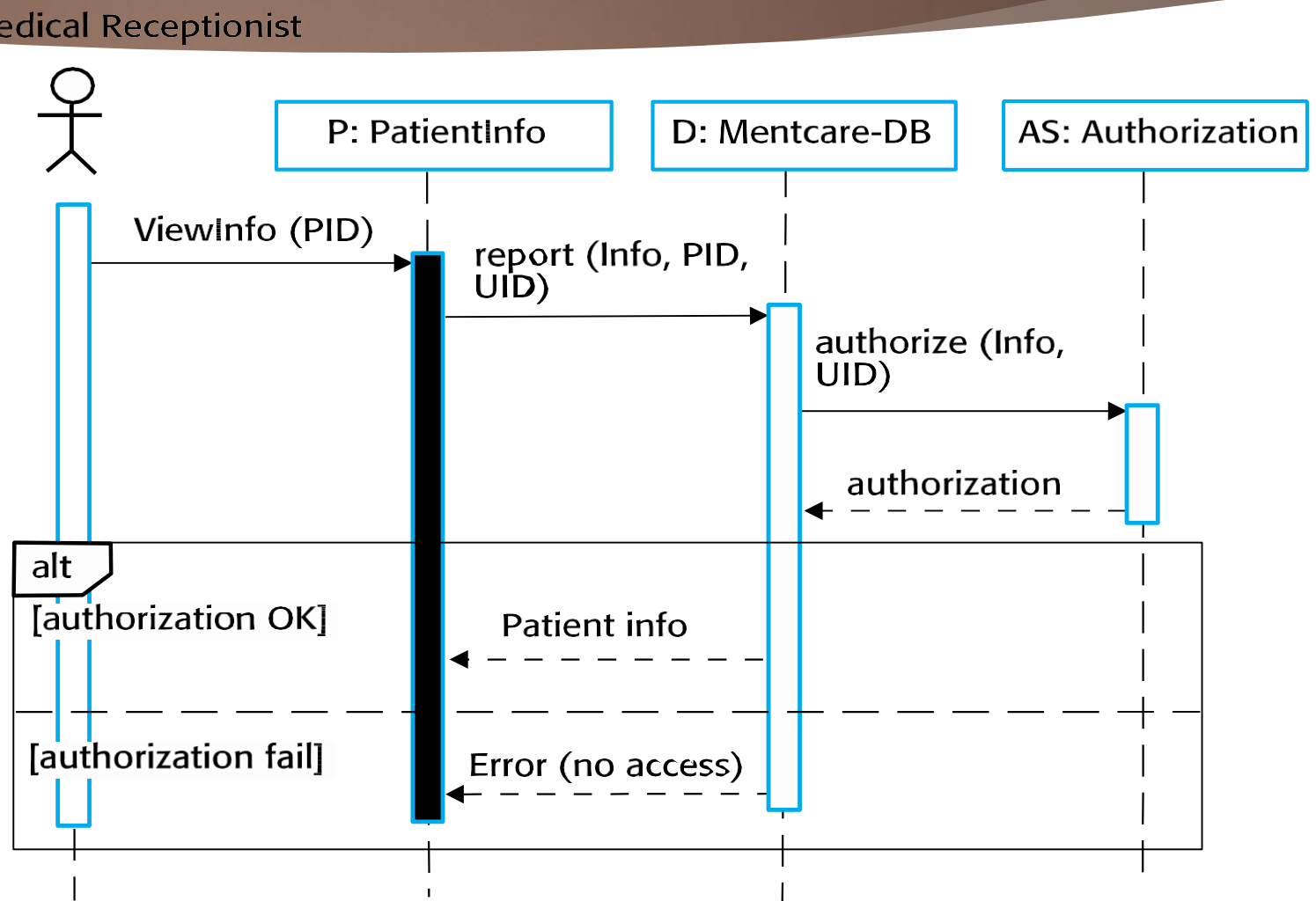
Sequence Diagram - Example

- Use cases involving the role ‘Medical Receptionist’ (Mentcare system)



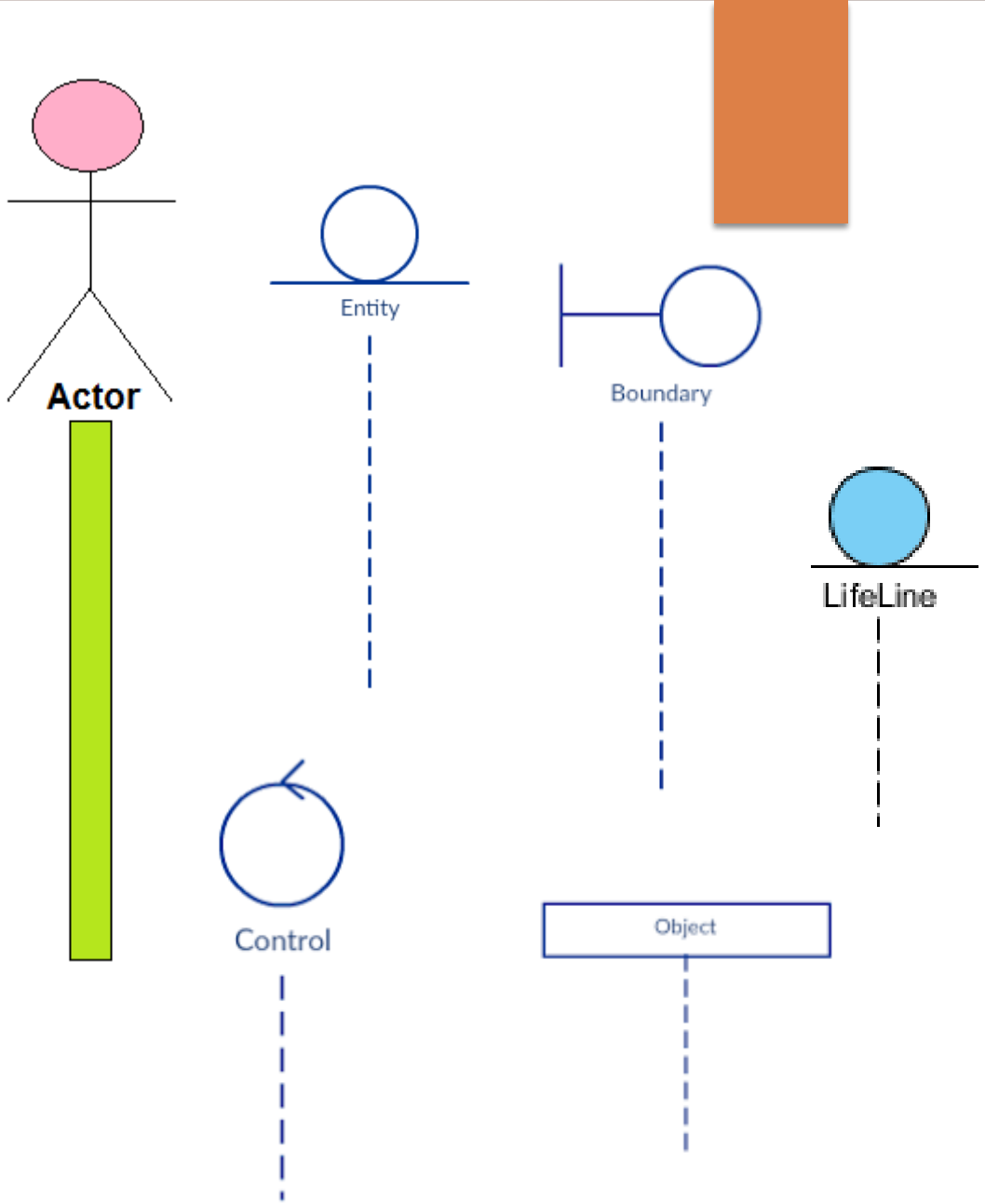
Sequence Diagram - Example

- Sequence diagram for 'View patient information'



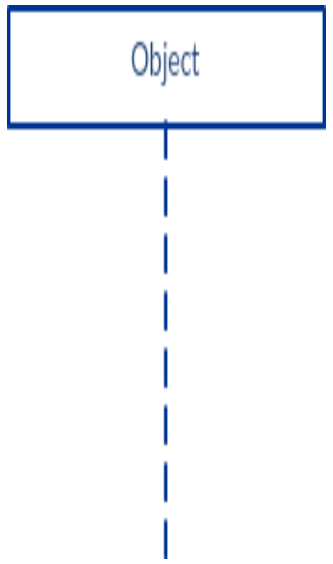
Sequence Diagram
Notations

Ref: https://creately.com/blog/diagrams/sequence-diagram-tutorial/



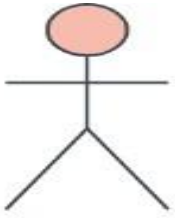
Basic Symbols and Components

- **Lifeline Notation**
  - Made up of several of these lifeline notations
  - Should be arranged horizontally across the top of the diagram.
  - No two lifeline notations should overlap each other.
  - Represent different objects or parts that interact with each other in the system during the sequence.



Basic Symbols and Components

- A lifeline notation with an actor element symbol
  - Used when the particular sequence diagram is owned by a use case



Actor symbol

Shows entities that interact with or are external to the system.

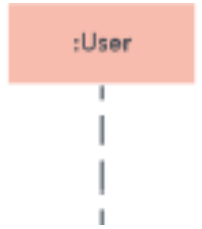
Represents a class or object in UML. The object symbol demonstrates how an object will behave in the context of the system.

P: PatientInfo

Object symbol

Basic Symbols and Components

- Lifelines begin with a labeled rectangle shape or an actor symbol.
- Represents the passage of time as it extends downward.
- This dashed vertical line shows the sequential events that occur to an object during the charted process.



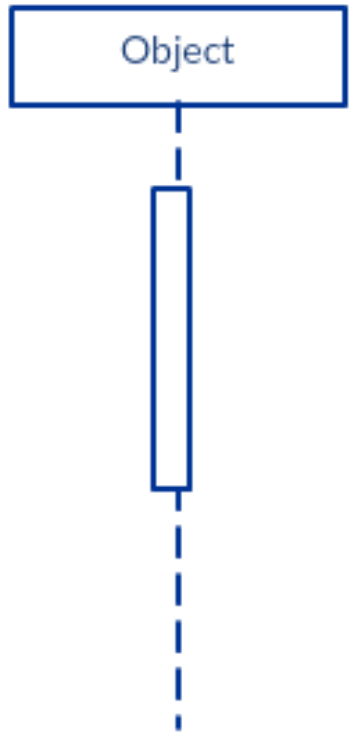
Lifeline symbol

Activation box

Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.

Basic Symbols and Components

- **Activation Bars**
  - A box placed on the lifeline.
  - Used to indicate that an object is active (or instantiated) during an interaction between two objects.
  - The length of the rectangle indicates the duration of the objects staying active.



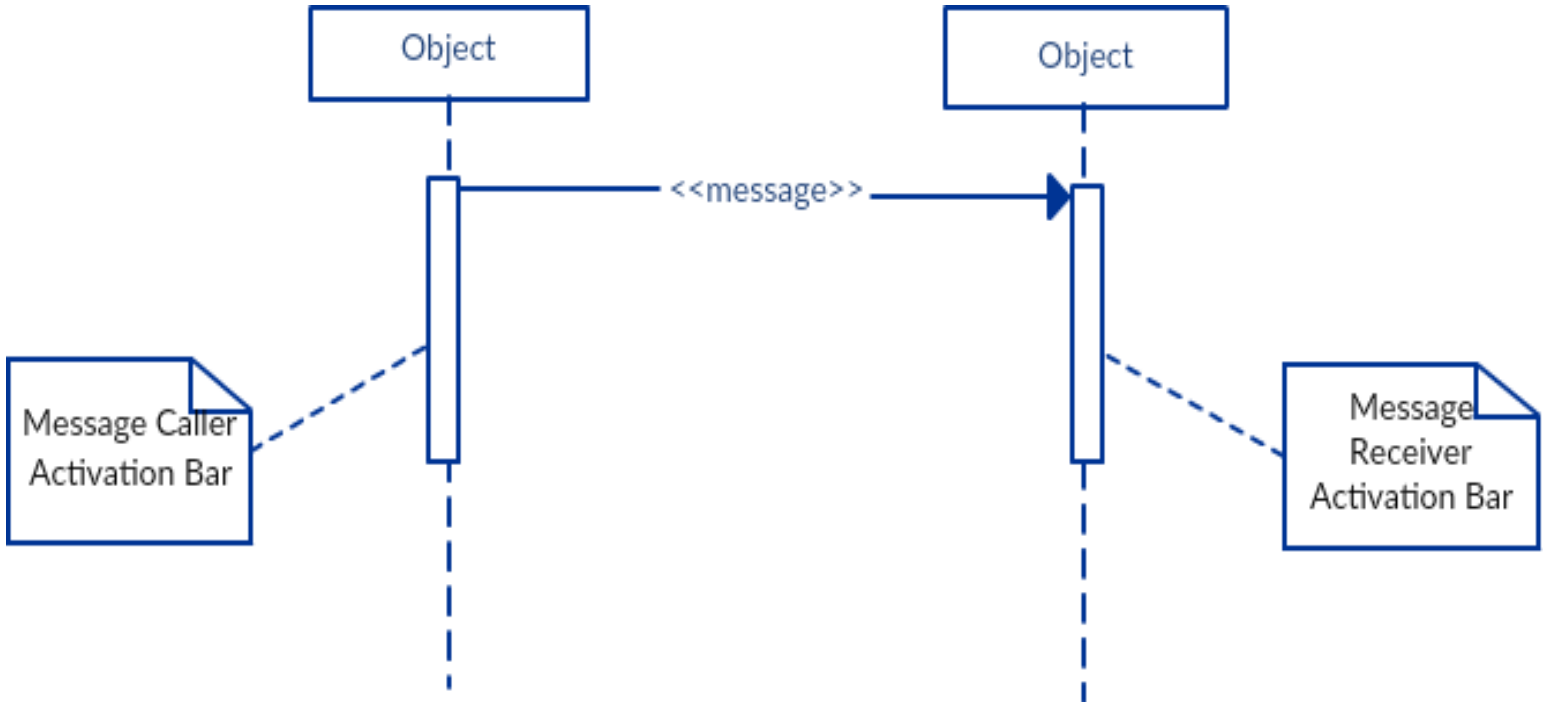
Basic Symbols and Components

- **Interaction between two objects**
  - Occurs when one object sends a message to another.
  - The use of the activation bar on the lifelines of the
    - Message Caller (the object that sends the message) and
    - Message Receiver (the object that receives the message)

  -Indicates that both are active/is instantiated during the exchange of the message

Basic Symbols and Components

- **Activation bar example**



Basic Symbols and Components

- **Message Arrows**
  - An arrow from the *Message Caller* to the *Message Receiver*.
  - A message can flow in any direction;
    - from left to right, right to left or
    - back to the Message Caller itself.
  - Different arrowheads can indicate the type of message being sent or received.
  - A message signature: message description
    - *attribute = message_name (arguments): return_type*

Basic Symbols and Components

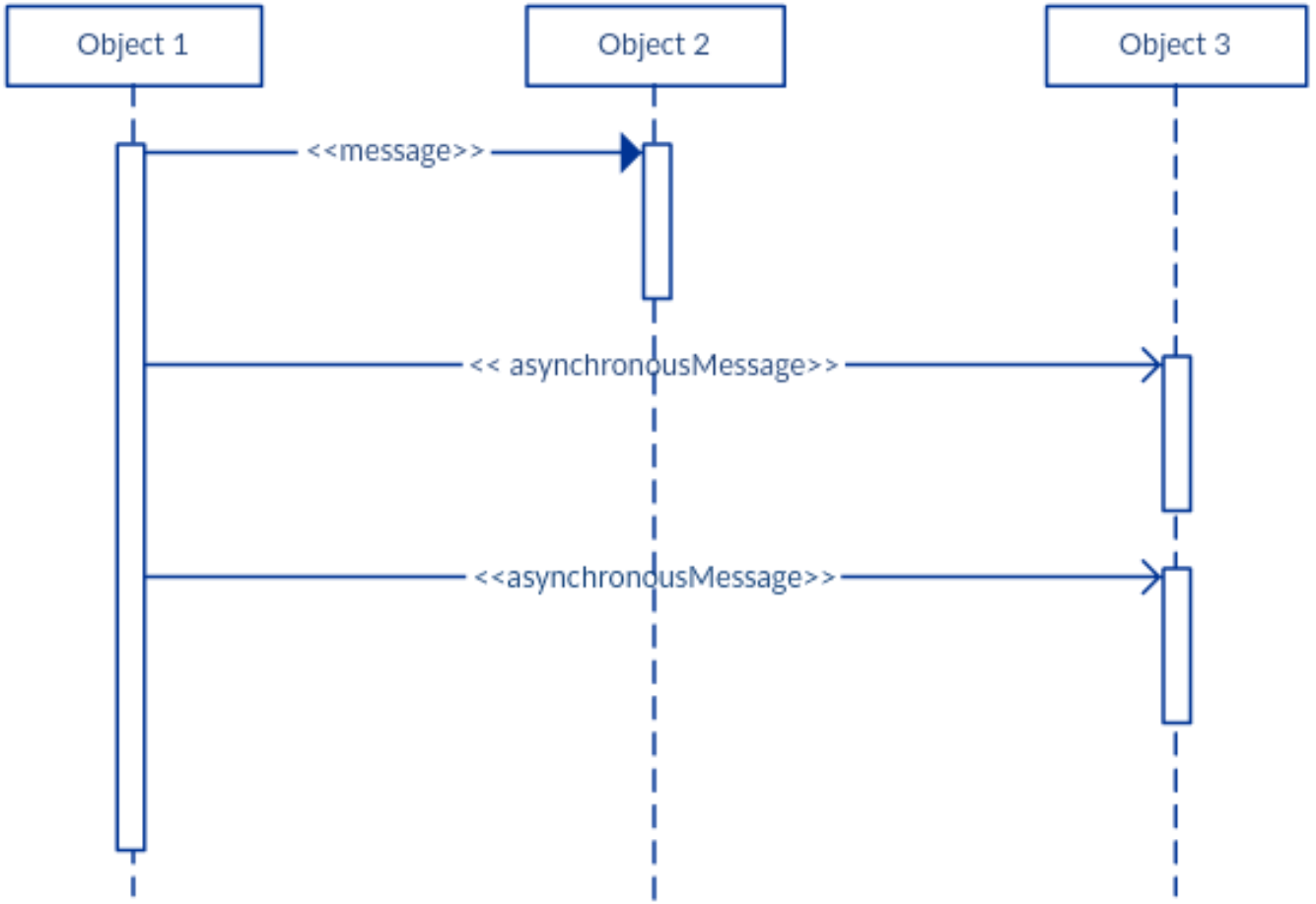
- ***Synchronous message***
  - Represented by a **solid line with a solid arrowhead.**
  - Sender waits for the receiver to process the message and return before carrying on with another message
  - The diagram should show both the call and the reply.

Basic Symbols and Components

- ***Asynchronous message***
  - Represented by a **solid line with a lined arrowhead.**
  - Asynchronous messages **do not require a response before the sender continues**.
  - Only the call should be included in the diagram.

Basic Symbols and Components

- **Example**

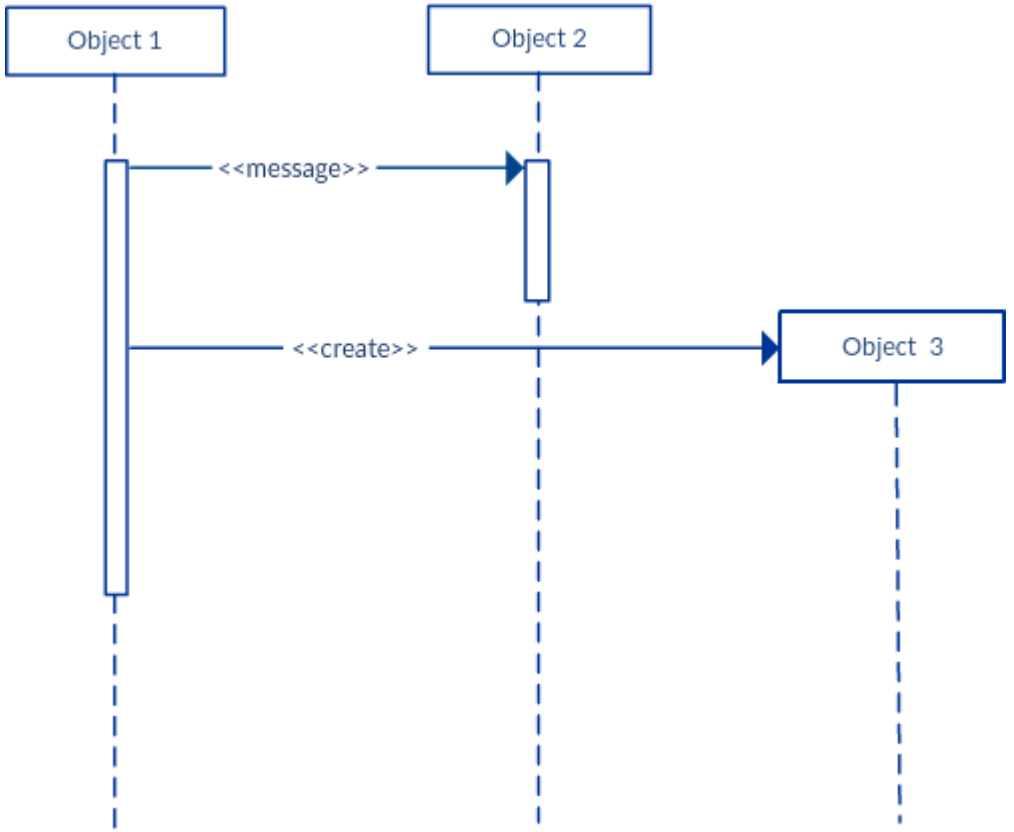


Basic Symbols and Components

- ***Return message***
  - Represented by a **dashed line with a lined arrowhead**.
  - indicate that the message receiver is done processing the message and is returning control over to the message caller

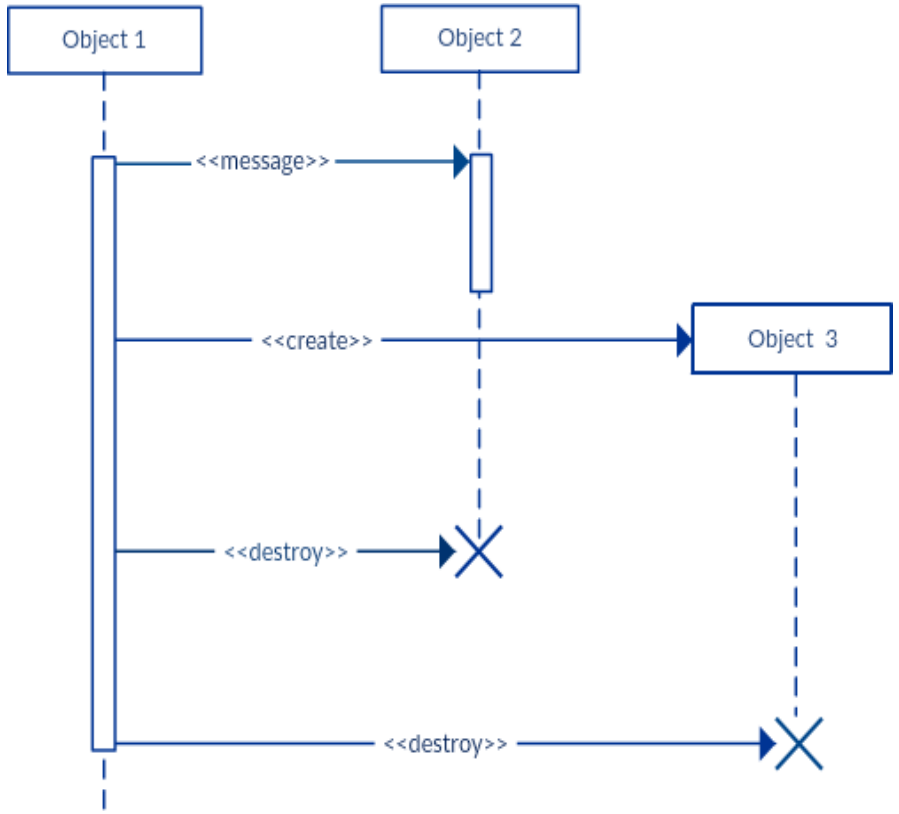
Basic Symbols and Components

- ***Participant creation message***
  - Objects do not necessarily live for the entire duration of the sequence of events.
  - Objects or participants can be created according to the message that is being sent.



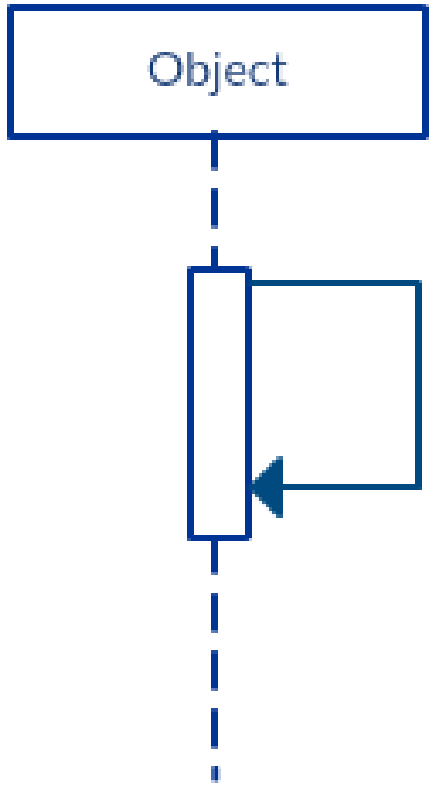
Basic Symbols and Components

- ***Participant destruction message***
  - When no longer needed participants can also be deleted from a sequence diagram.
  - This is done by adding an 'X' at the end of the lifeline of the said participant.



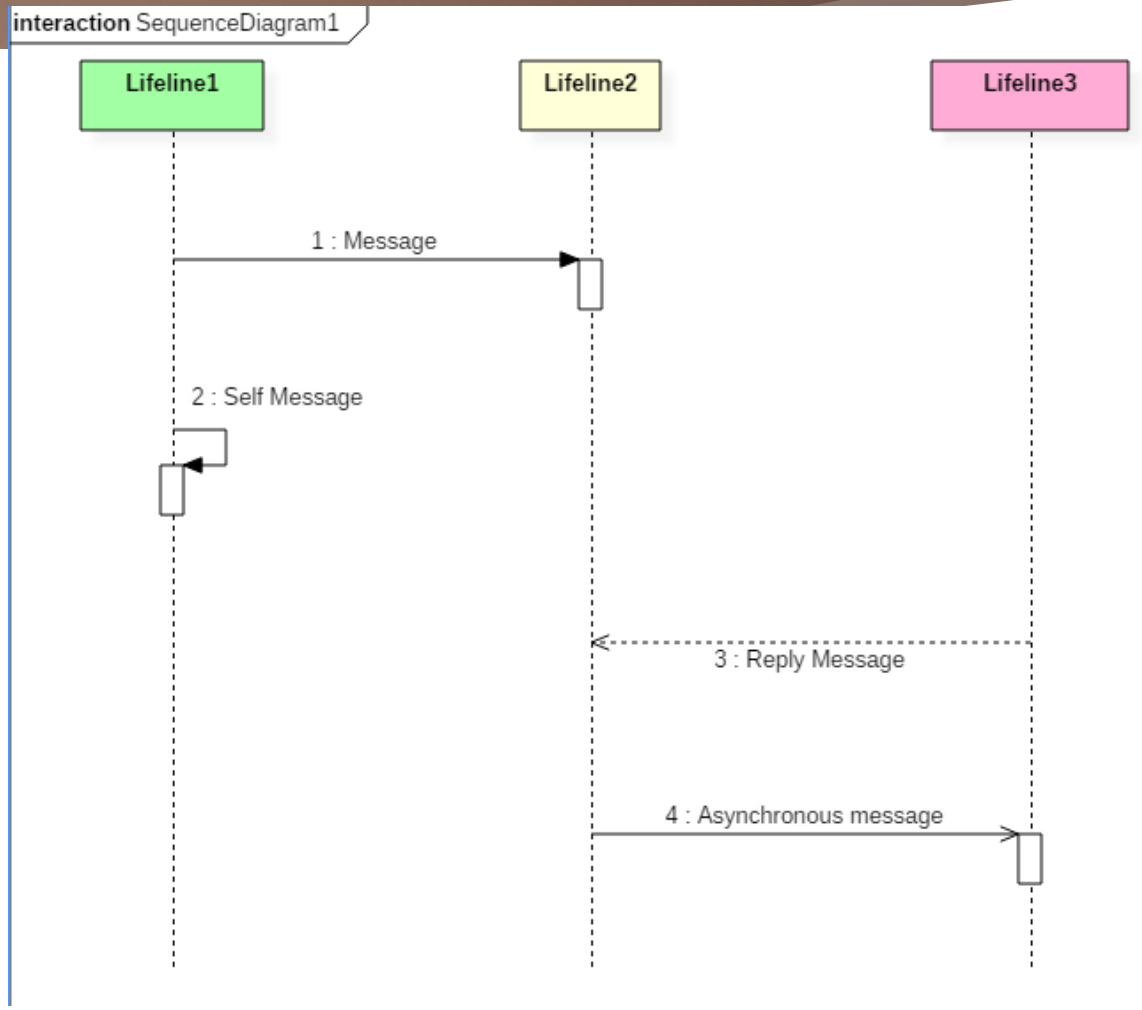
Basic Symbols and Components

- ***Reflexive message***
  - When an object sends a message to itself, it is called a reflexive message.
  - Indicated with a message arrow that starts and ends at the same lifeline.



Basic Symbols and Components

- **Sequence Modeling example**

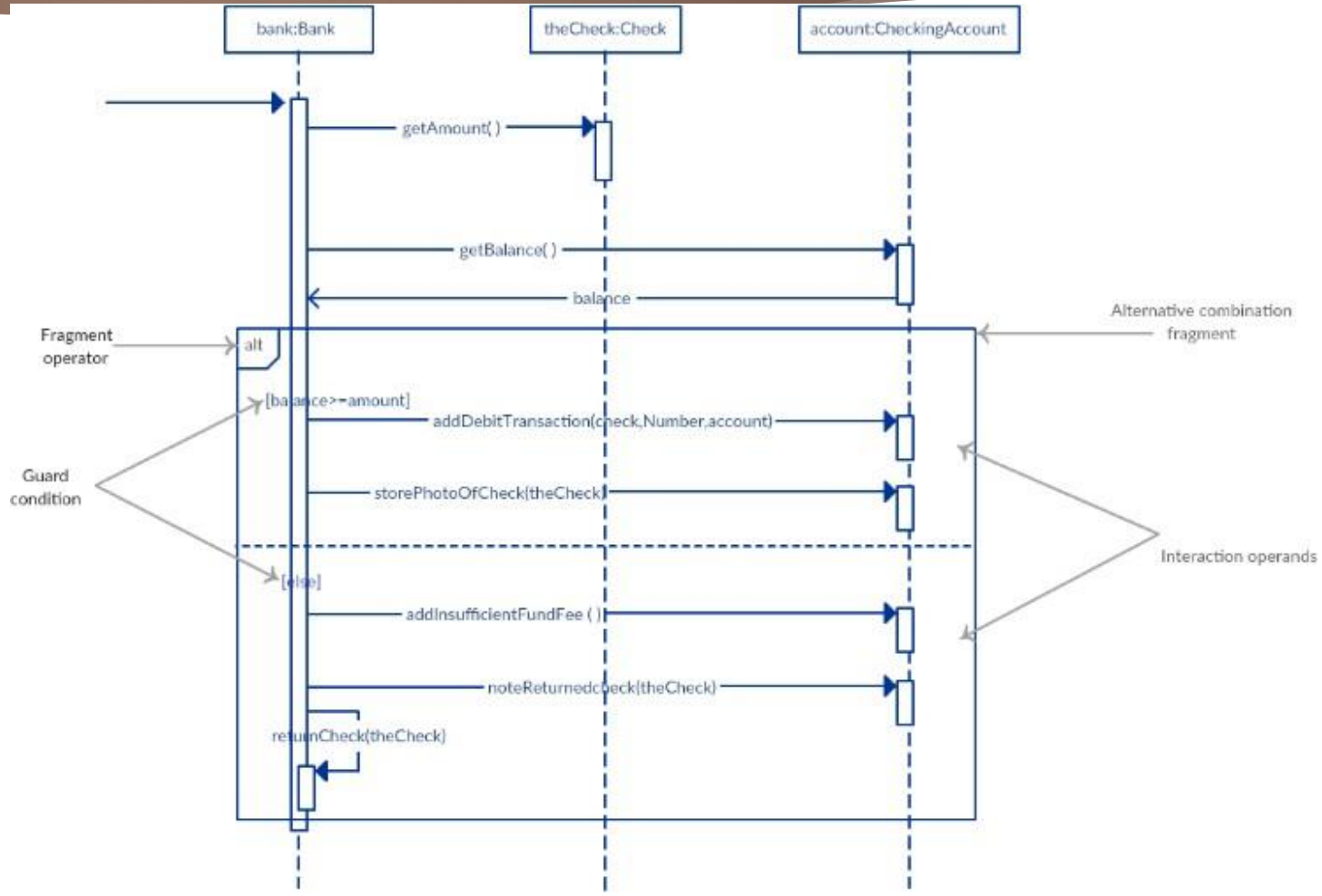


Basic Symbols and Components

- ***Alternatives***
  - Symbolizes a choice (that is usually mutually exclusive) between two or more message sequences.
  - To represent alternatives, use the labeled rectangle shape with a dashed line inside.

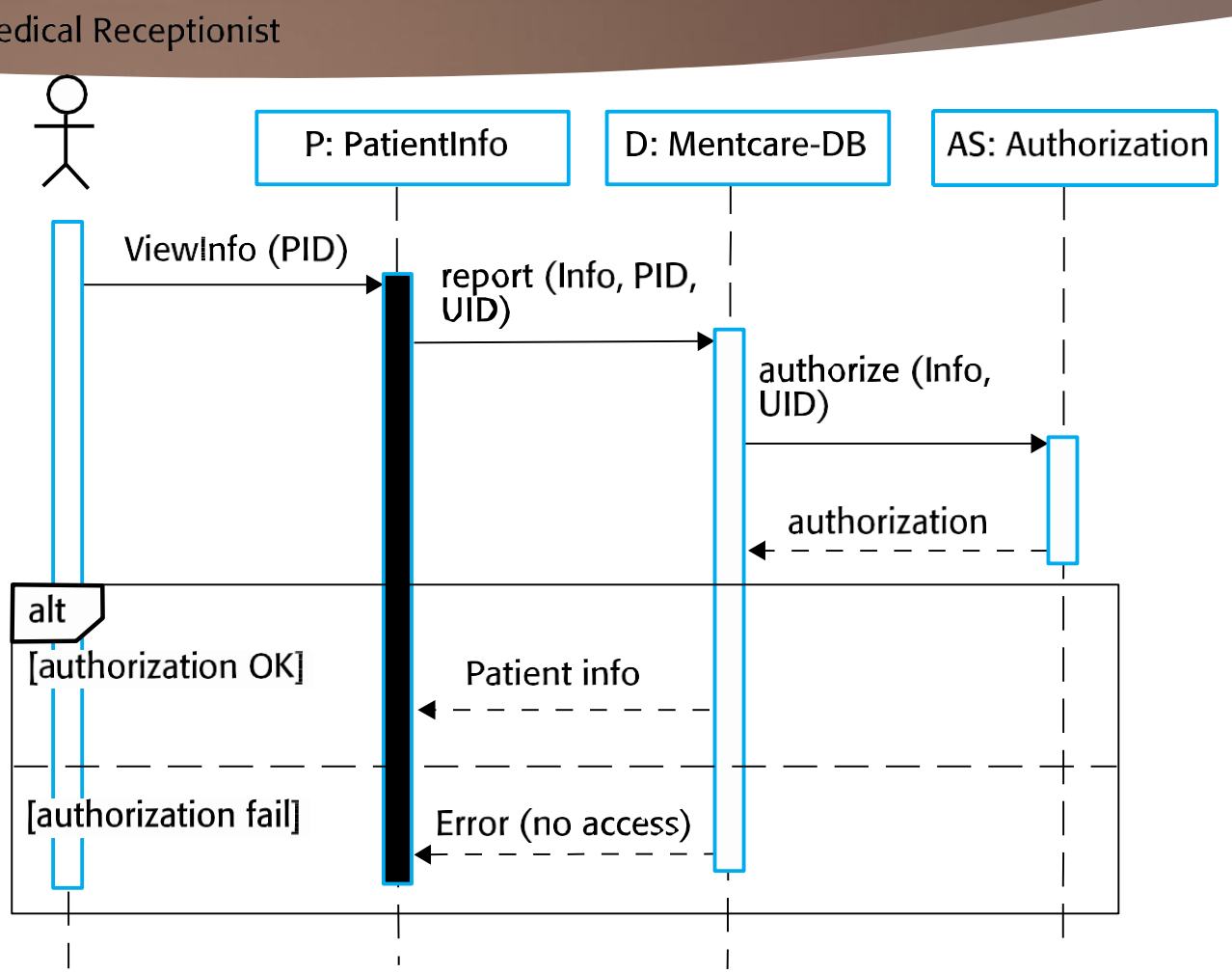
Basic Symbols and Components

- ***Alternatives Example***



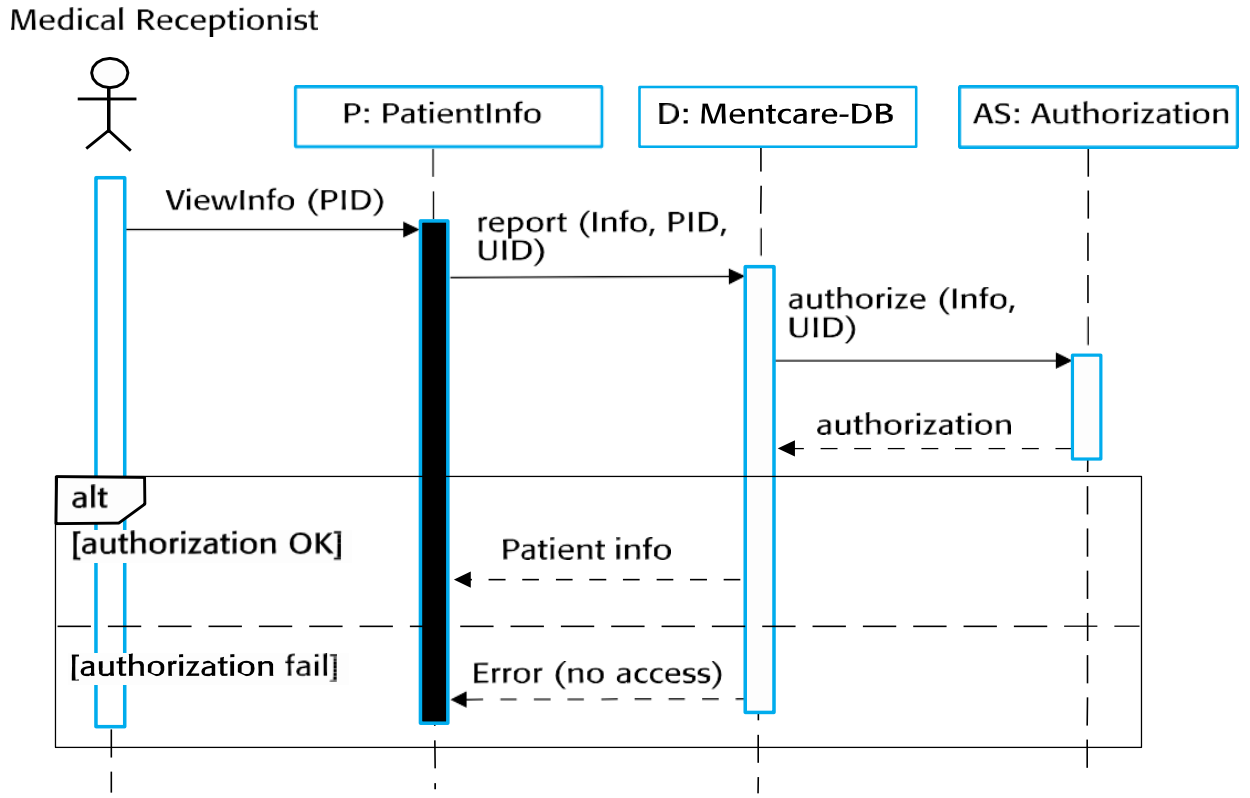
Sequence Diagram

- Sequence diagram for View patient information



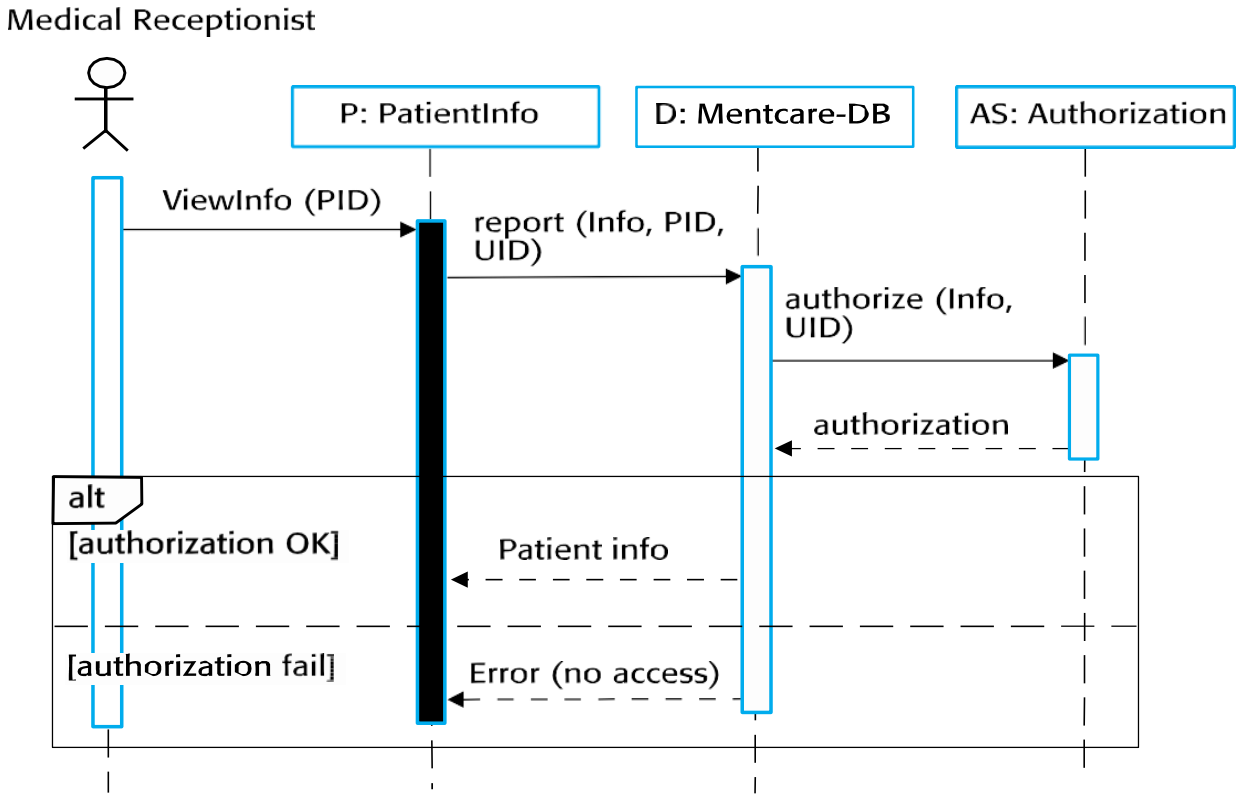
Sequence Diagram

- Sequence diagram for View patient information
  - The medical receptionist triggers the ViewInfo method in **an instance P of the PatientInfo object class**, supplying the patient's identifier, PID to identify the required information. P is a user interface object, which is displayed as a form showing patient information.



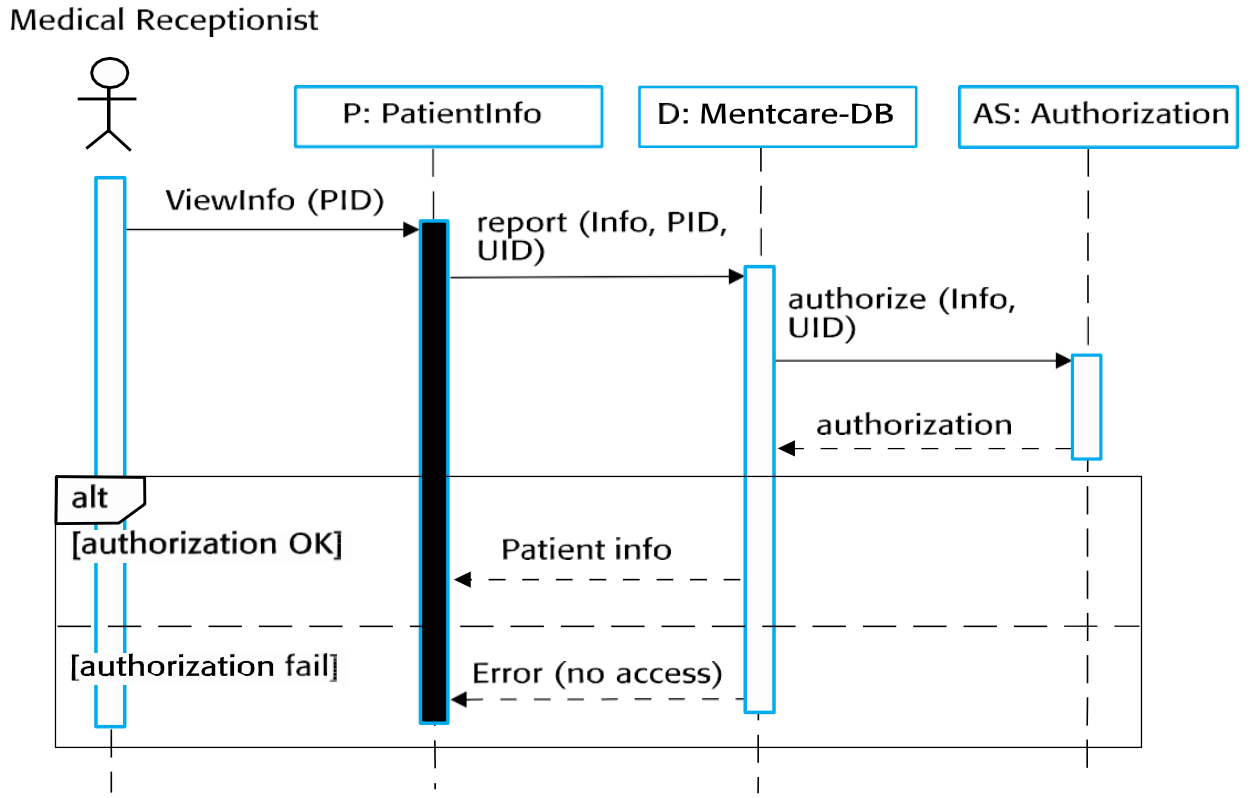
Sequence Diagram

- Sequence diagram for View patient information
  - The instance P calls the database to return the information required, supplying the receptionist's identifier to allow security checking. (At this stage, it is not important where the receptionist's UID comes from.).



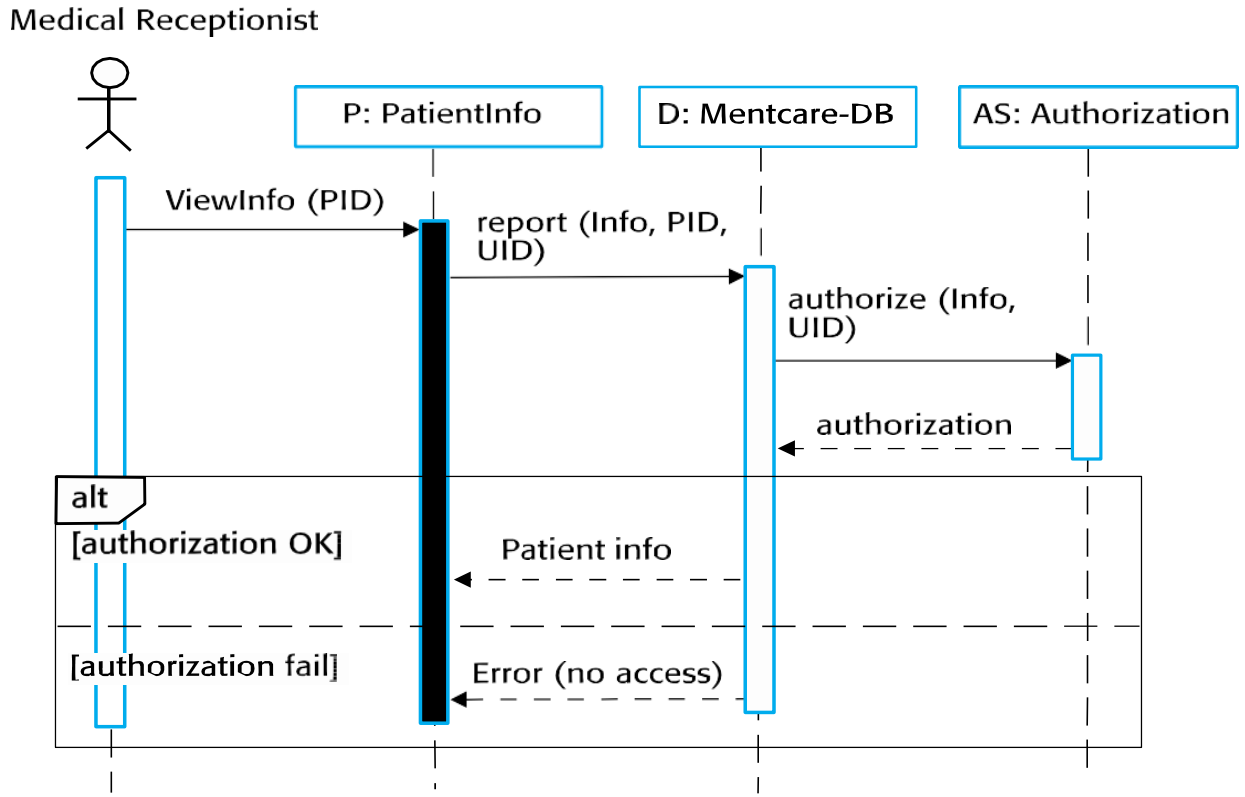
Sequence Diagram

- Sequence diagram for View patient information
  - The database checks with an authorization system that the receptionist is authorized for this action.
  - If authorized, the patient information is returned and is displayed on a form on the user's screen.



Sequence Diagram

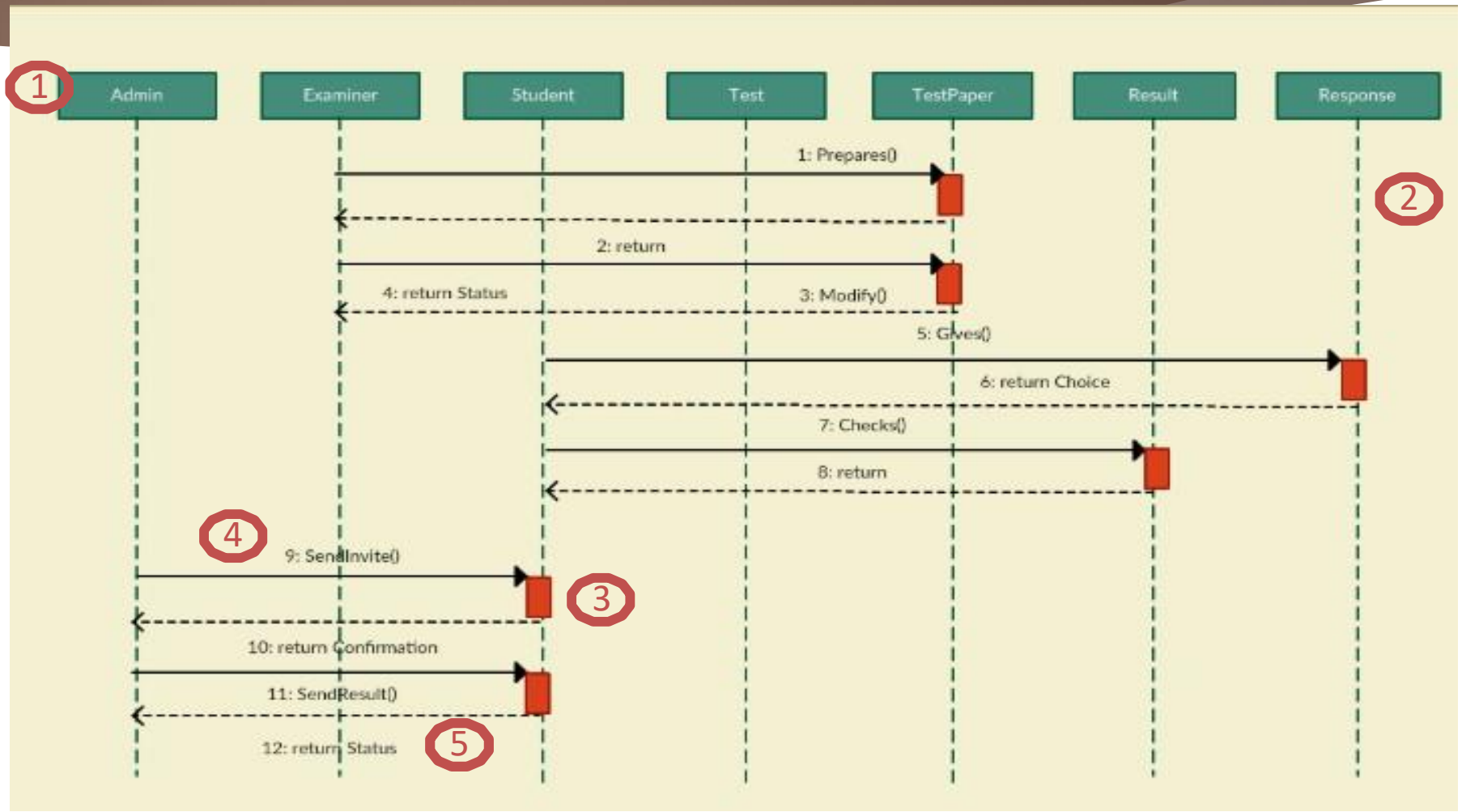
- Sequence diagram for View patient information
  - If authorization fails, then an error message is returned. The box denoted by “alt” in the top-left corner is a choice box indicating that one of the contained interactions will be executed. The condition that selects the choice is shown in square brackets.



Sequence Diagram

- Exercise

Name the numbered elements



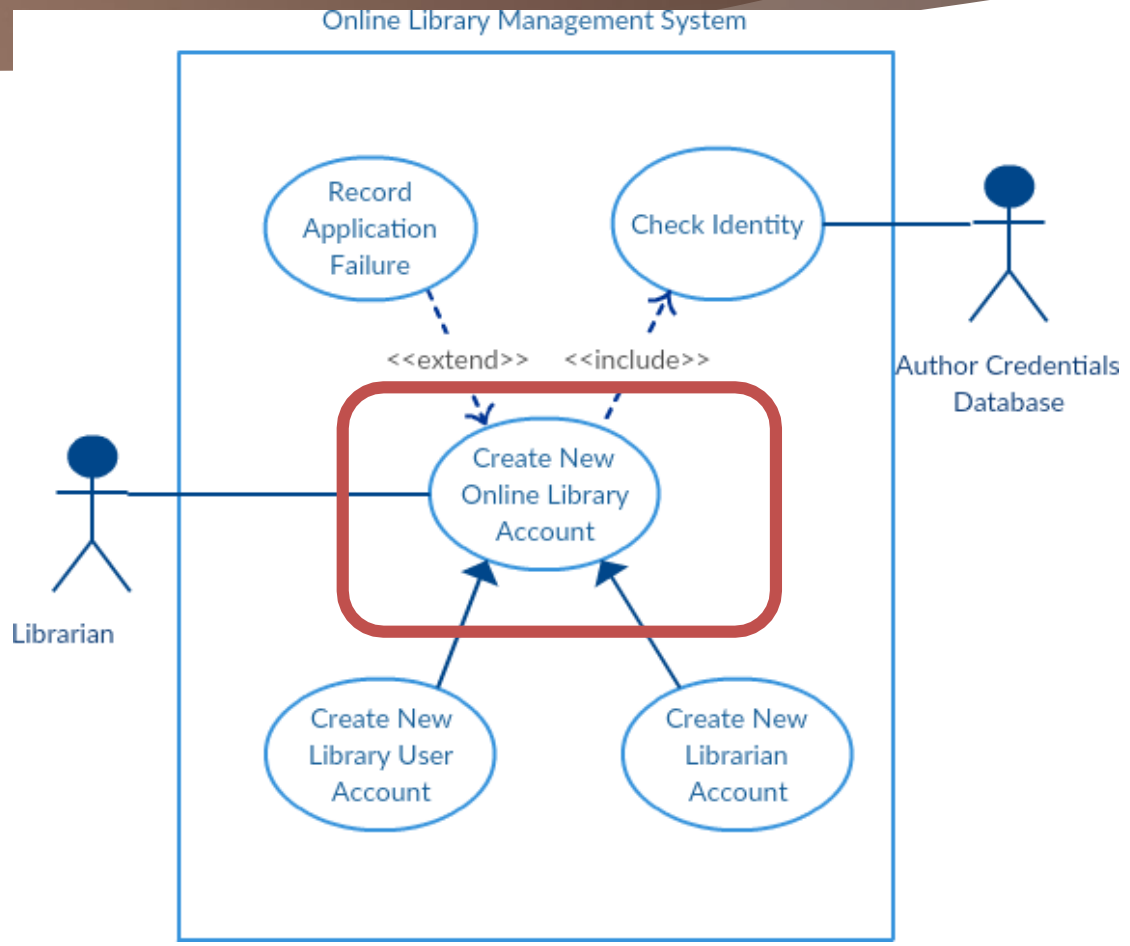
Sequence Diagram

- **A SEQUENCE DIAGRAM REPRESENTS THE SCENARIO OR FLOW OF EVENTS IN ONE SINGLE USE CASE.**
- **THE MESSAGE FLOW OF THE SEQUENCE DIAGRAM IS BASED ON THE NARRATIVE OF THE PARTICULAR USE CASE.**

Sequence Diagram – Exercise 1

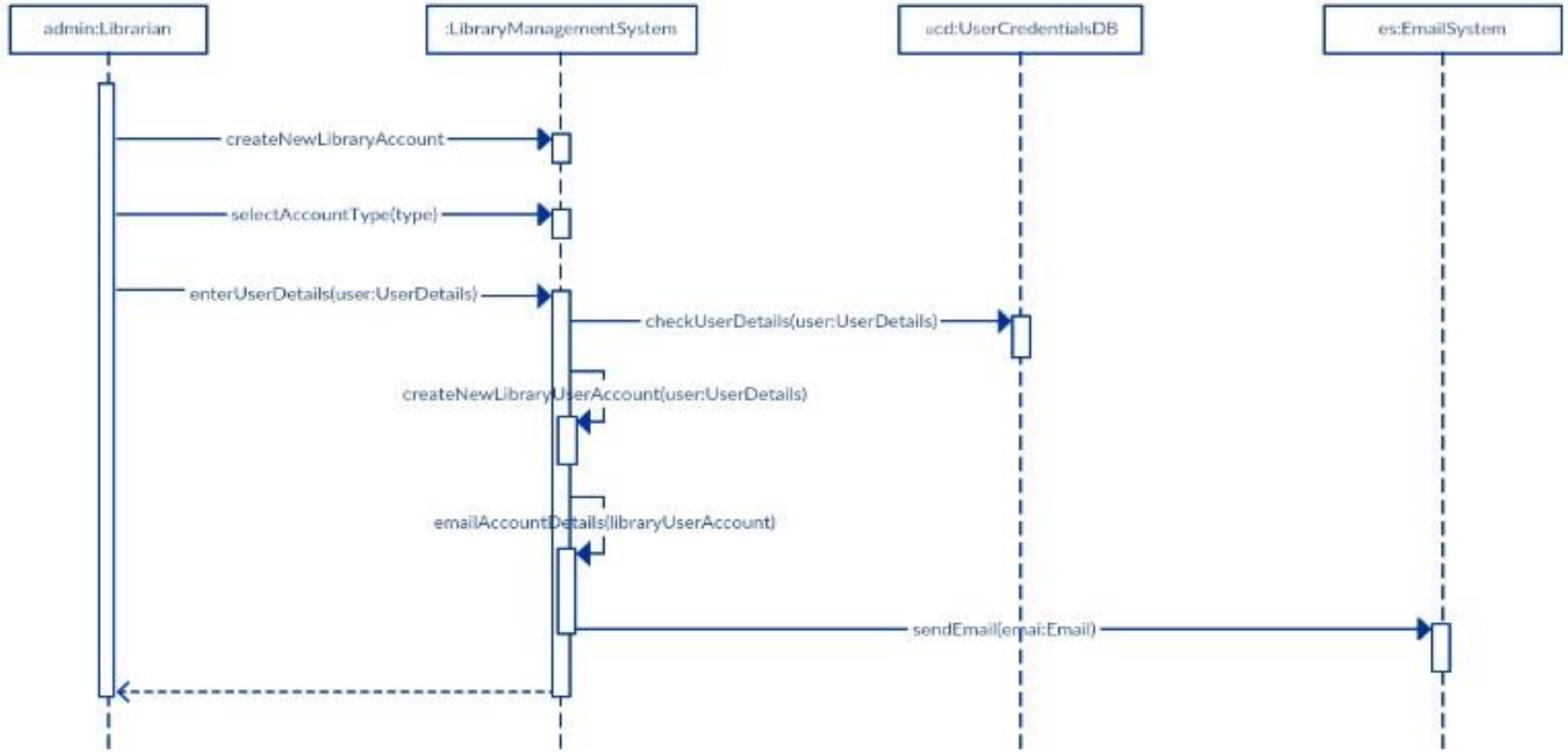
- Create New Library User Account

  - The librarian request the system to create a new online library account
  - The librarian then selects the library user account type
  - The librarian enters the user's details
  - The user's details are checked using the user Credentials Database
  - The new library user account is created
  - A summary of the of the new account's details are the emailed to the user



Sequence Diagram – Exercise 1

- Answer



Sequence Diagram – Exercise 2

- Explain the sequence diagram