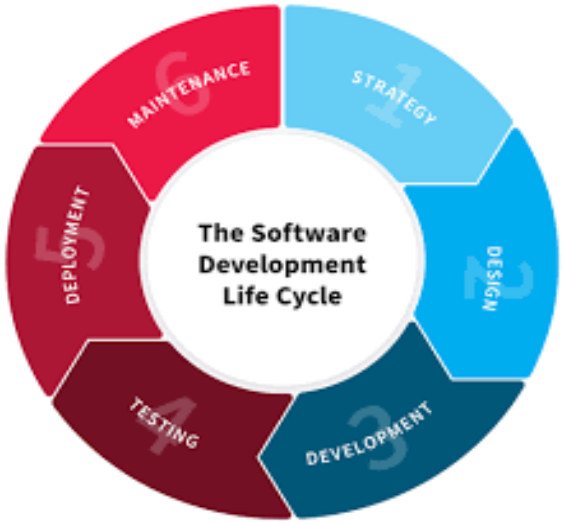
System Analysis Course

Week 03: UML (Use Case)

Ahmed Kord

Rana Khattab



Outline

- UML
  - Use Case
- Practical Part – on Software Program

UML (Unified Modeling language)

UNIFIED
MODELING
LANGUAGE

UML

- The ***UML*** is commonly defined as ‘a ***standard language*** for ***specifying***, ***constructing*** , ***visualizing***, and ***documenting*** the artifacts of a software system’.

- The ***UML*** does *not prescribe any particular methodology*, but instead is flexible and customizable to fit any approach and it can be used in conjunction with a wide range of software lifecycles and development processes.

GOALS

❖ The ***primary goals*** in the design of the UML were to:

1. Provide users with a ***ready-to-use***, expressive visual modeling language so they can develop and exchange meaningful models.

2. ***Be independent*** of particular programming languages and development processes.

3. Provide a formal basis for ***understanding the modeling language***.

UML

- **UML** defines a number of diagrams, of which the main ones can be divided into the following two categories:
- ***Structural diagrams,*** which describe the static relationships between components. These include:
  - class diagrams,
  - object diagrams,
  - component diagrams,
  - deployment diagrams.

UML cont.

- ***Behavioral diagrams,*** which describe the dynamic relationships between components. These include:
  - use case diagrams,
  - sequence diagrams,
  - collaboration diagrams,
  - state chart diagrams
  - Activity diagram

USE CASE

- ***Use case*** diagrams model the ***functionality*** provided by the system (***use cases***), the ***users*** who interact with the system (***actors***), and the ***association*** between the ***users*** and the ***functionality***.

- Use cases are used in the requirements collection and analysis phase of the software development lifecycle to represent the ***high-level requirements of the system.***

- More specifically, a use case specifies a ***sequence of actions***, including variants, that the system can perform.

USE CASE DIAGRAM

- An individual *use case is represented by an ellipse*, an *actor by a stick figure*, and an *association by a line between the actor and the use case*.
- The role of the actor is written *beneath the icon*. Actors are not limited to *humans*.
  - If a system communicates with ***another application***, and expects input or delivers output, then that application can also ***be considered an actor***.
- A use case is typically represented by a verb followed by an object, such as View property, Lease property. An example use case diagram for Client with four use cases is shown in the Figure and a use case diagram for Staff in the Figure . The use case notation is simple and therefore is a very good vehicle for communication.

USE CASE DIAGRAM

- ***Use case:*** **A** set of scenarios that describing an **interaction between a user and a system**.
- ***A use-case*** is always initiated by an actor.
- Directly or indirectly via another use-case.
- ***System boundary:*** indicates the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not.

USE CASE DIAGRAM

- **RELATIONSHIPS :**
- **1.Association**:
- communication between an **actor** and a **use case**; represented by a solid line.

- **2.Generalization:**
- relationship between one general use case and one specific use case.
- Represented by a line with a triangular arrow head toward the parent use case, the more general modeling element.

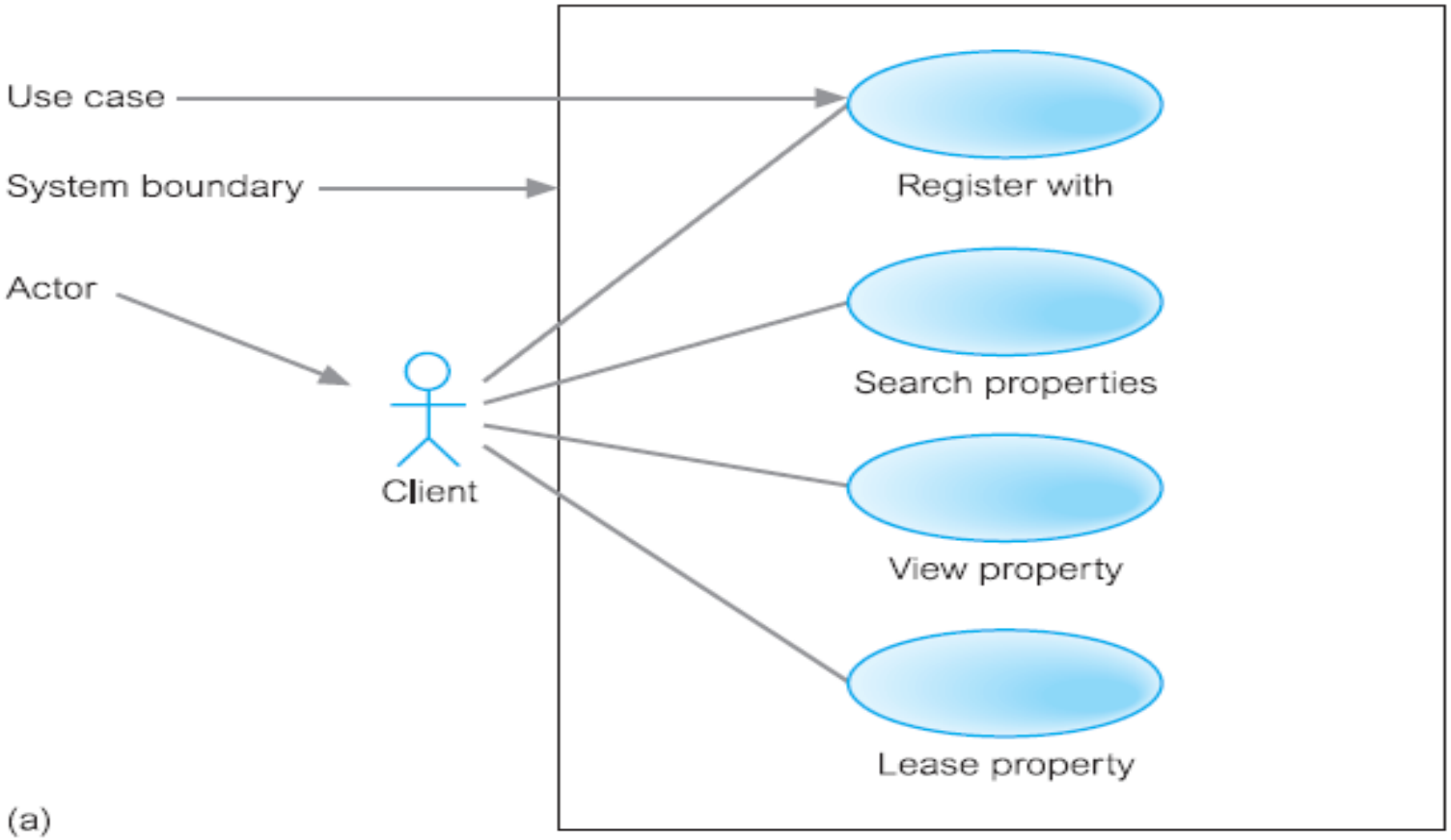
3.Include: 

The including use case embodies some behavior common with the base use case.

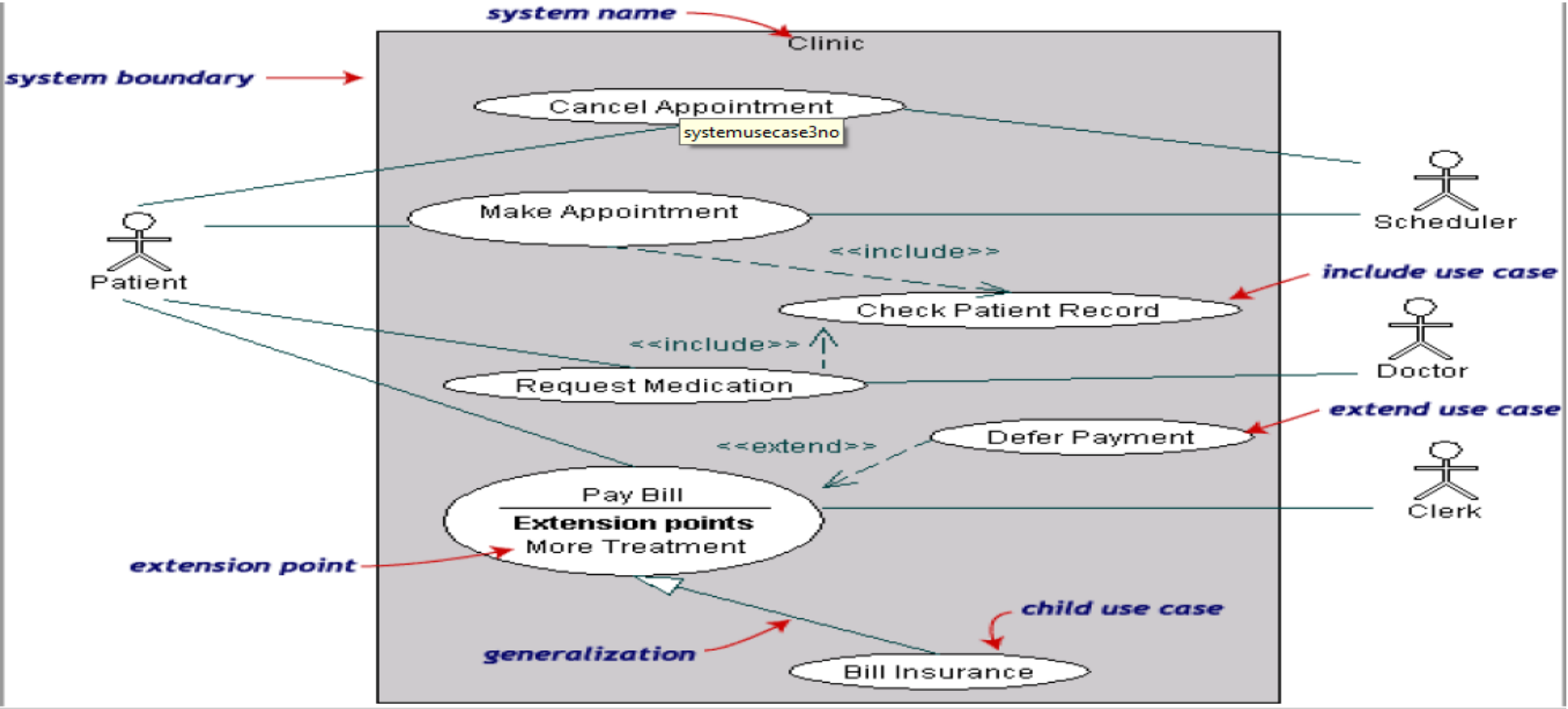
4.Extend: 

The extending use case may add behavior to the base use case. The base class declares “extension points”.

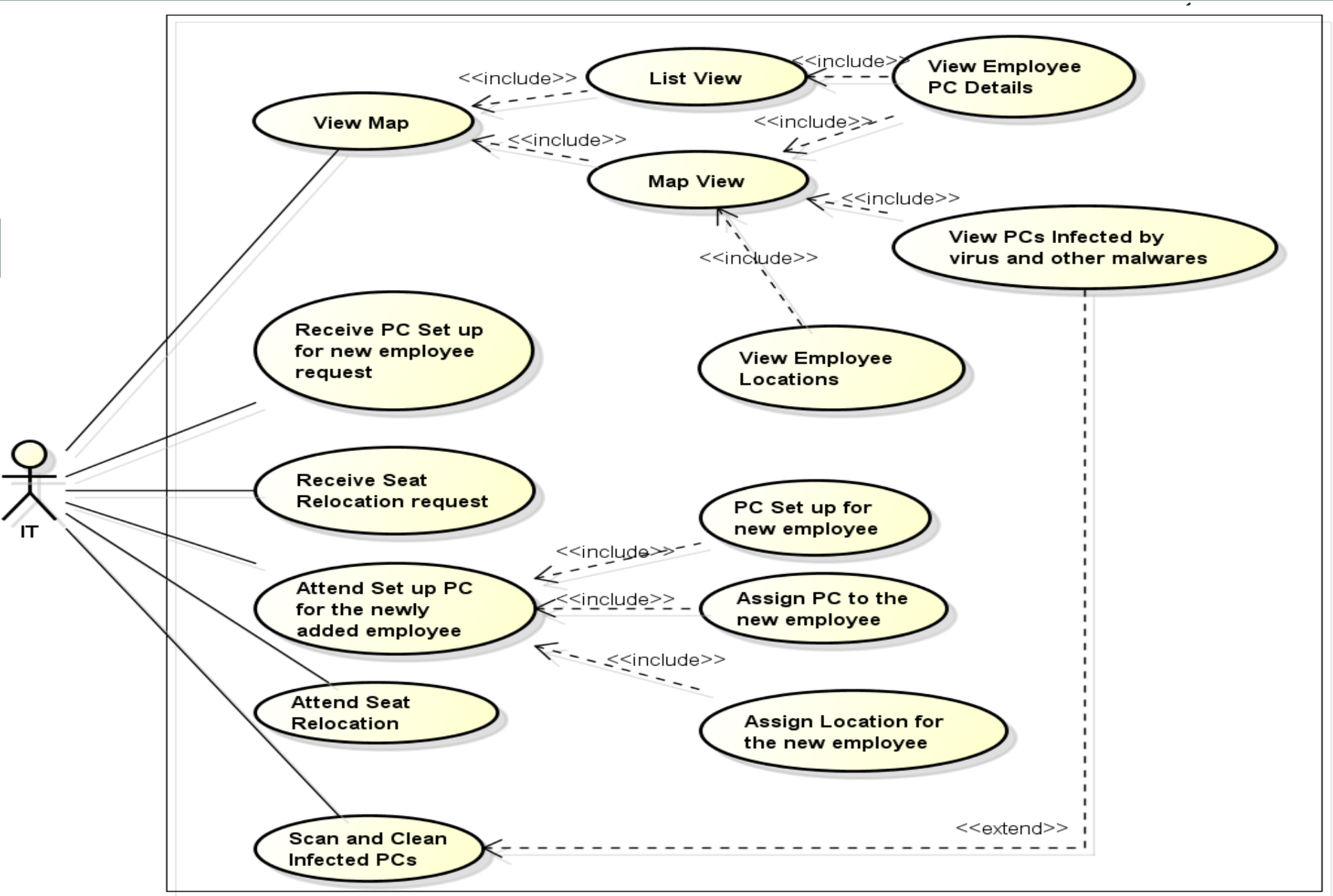
USE CASE DIAGRAM Example 1



USE CASE DIAGRAM Example 2



USE CASE DIAGRAM Example 3



Outline

- UML
  - Use Case
- Practical Part – on Software Program

Thank You