Object Oriented Software Engineering

UML Diagrams: Use Case Diagrams

Presented By:

T.A. **Asmaa Hammad El-sayed**

E-mail: **eng.asmaa134@gmail.com**

Agenda

Use Case Diagram

Case Study

Use Case Diagrams

Use Case Diagram

- **What is the Use Case??**
  - A use case is a methodology used to identify, clarify, and organize system **requirements** (**functionality)**.
- **Functional requirements**
  - Functional requirements are the services that the system should provide, They are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions.

Use Case Diagram

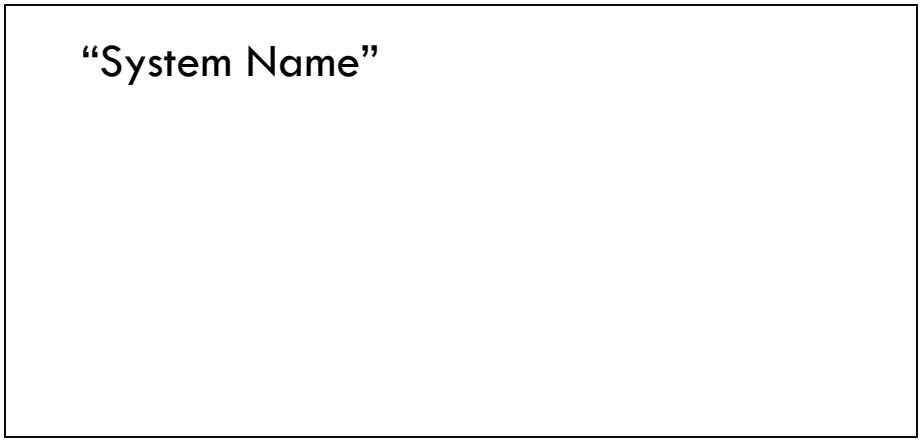
- **Non-functional requirements**
  - Constraints on the services or functions offered by the system such as:
    - Security
    - Reliability
    - Performance
    - Maintainability
    - Availability
    - Scalability
    - Usability

Use Case Diagram (cont.)

- **Use Case Diagram** is a way of visualizing the relationships
  - between actors and use cases (functions)
  - among use cases
- **Use Case Diagram has 4 major elements:**
  - The **system** described
  - The **actors** that the system interacts with
  - The **use-cases**, or services, that the system knows how to perform
  - The **relationships** between the above elements

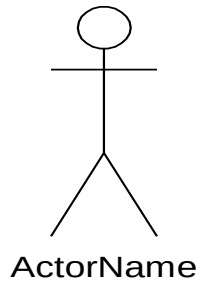
Use Case Diagram :System

- **System boundary:**
  - A boundary box is placed around the perimeter of the **system** to show how the actors communicate with the system.
  - The name of the system appears above or inside the box



Use Case Diagram : Actors

- **Actor** is a someone or something that interacts with the system (exchanges information with the system)
- Example:
  - Policeman – Enters data
  - Supervisor – Allowed to modify/erase data
  - Manager – Allowed to view statistics.
- **An actor** in a Use Case Diagram is represented as



Use Case Diagram : Actors

- **Actors** don't need to be human
  - ✓ May be an external system
  - – Examples:
    - ✓ GPS satellite: Provides the system with GPS coordinates
    - ✓ If actor is a system, in use case diagram it represented by by a rectangle 
- **An actor** has a name that reflects its role

Use Case Diagram :Actors

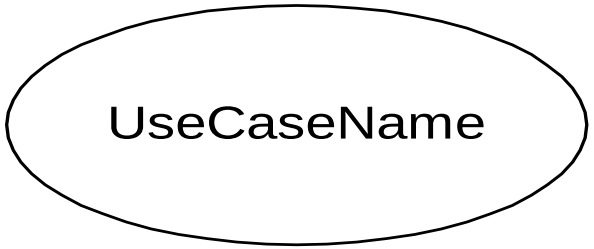
- **Types of actors**
  - **Primary Actors:** The Actor(s) using the system to achieve a goal. The **Use Case** documents the interactions between the system and the actors to achieve the goal of the primary actor.
  - **Secondary Actors:** Actors that the system needs assistance from to achieve the primary actor's goal.
  - A primary actor initiates an interaction with the system.
  - The system initiates interactions with secondary actors.

Use Case Diagram :Actors

- **Types of actors**
  - **For example:**
  - A user clicks the search button on an application's user interface. The application sends SQL query to a database system. The database system responds with a result set. The application formats and displays the result set to the user.
    - In this scenario the user is a primary actor because he initiates the interaction with the system (application).
    - The database system is a secondary actor because the application initiates the interaction by sending SQL query.

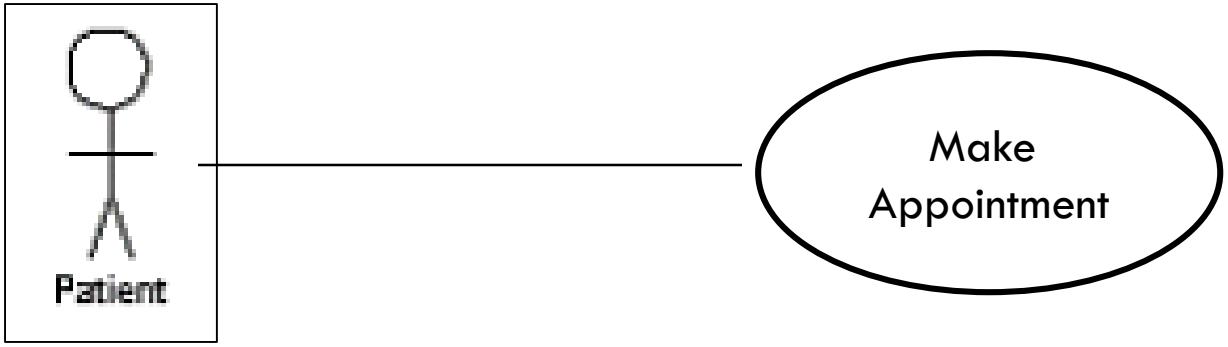
Use Case Diagram :Use case

- **A use case** meets an actor's goal
- Use case name is a verb
- **A** use case in a Use Case Diagram is represented as a oval



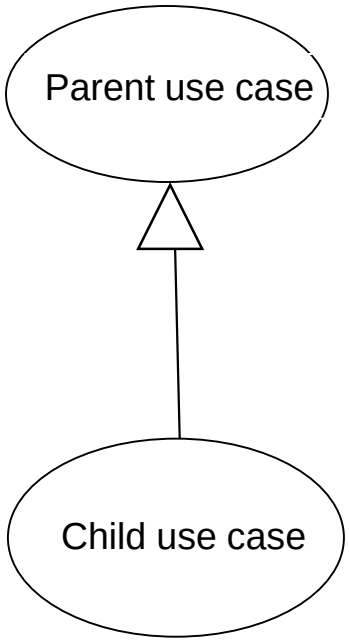
The <<*Association*>> Relationship

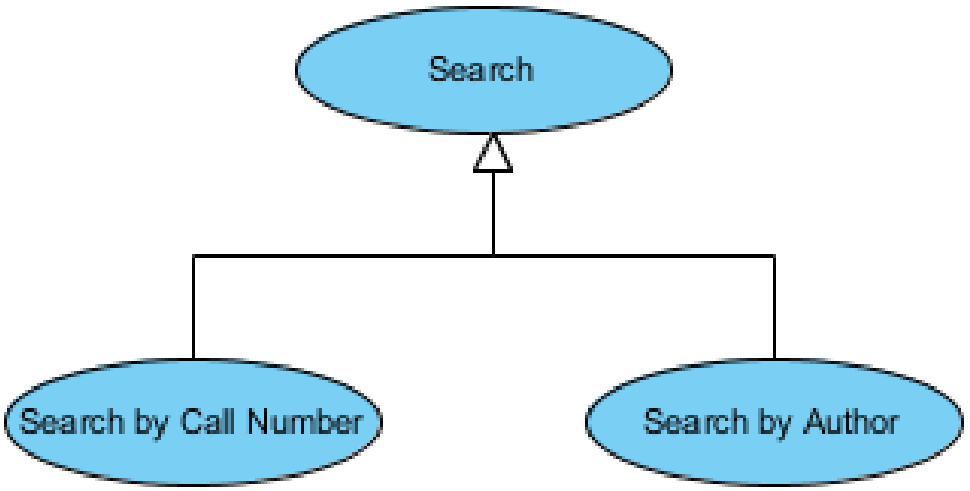
- Association : represent communication between actor and use case
- Depicted by line



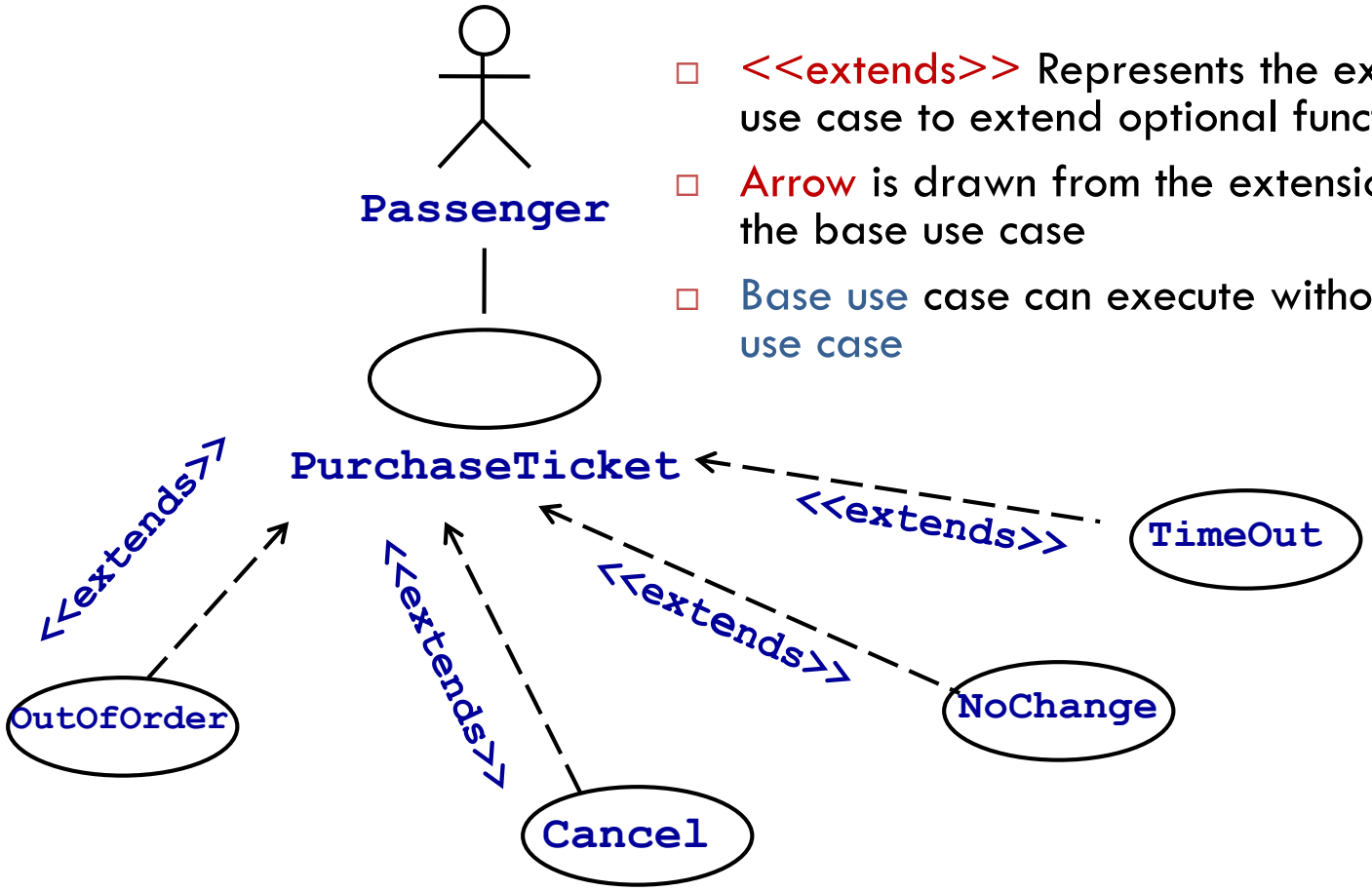
The <<*Generalization*>> Relationship

- Generalization Relationship
  - Represented by a line and a hollow arrow
    - From child to parent
  - The child use case inherits the behavior and meaning of the parent use case.



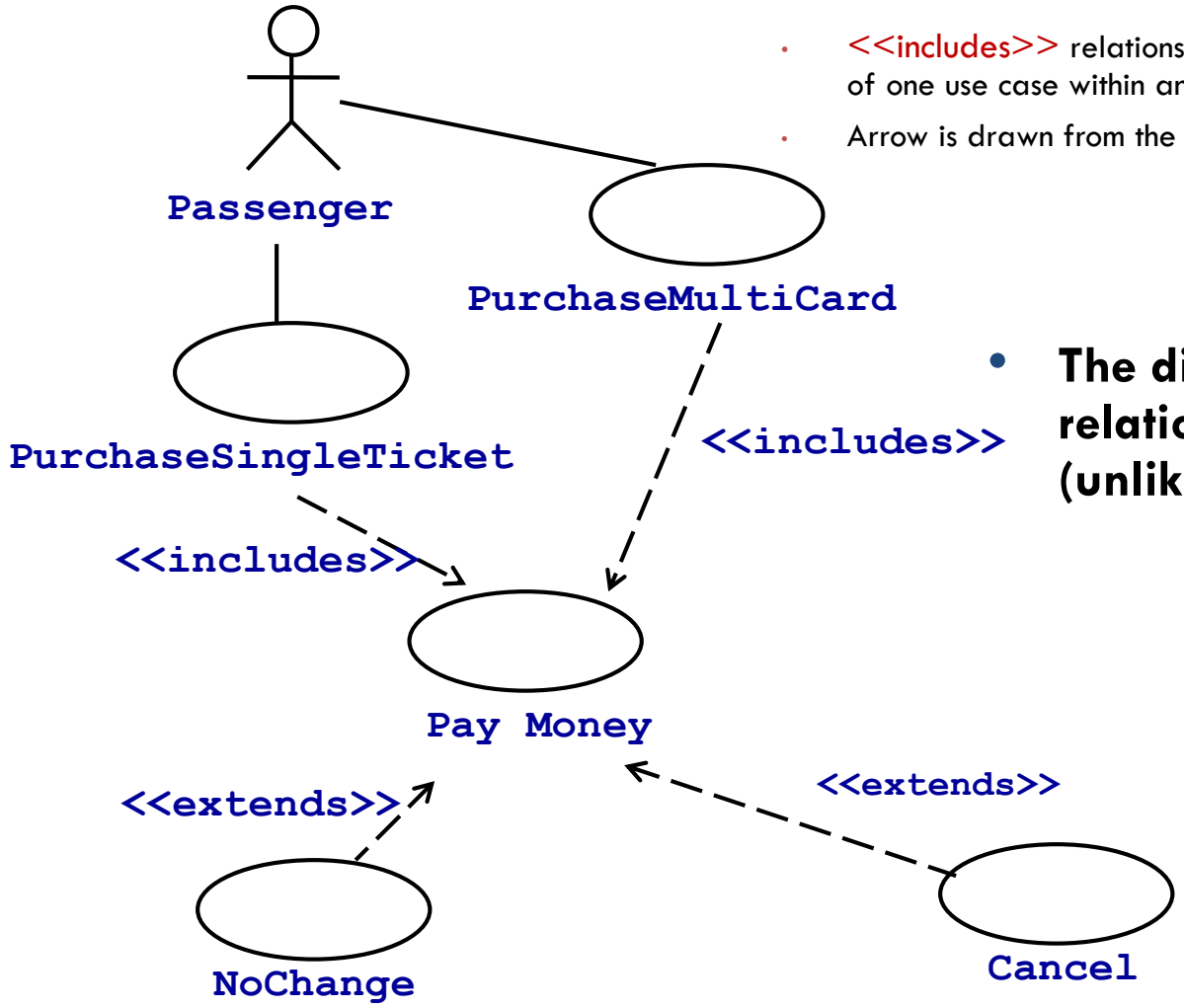


The <<*extends*>> Relationship



- <<extends>> Represents the extension of the use case to extend optional functionality
- Arrow is drawn from the extension use case to the base use case
- Base use case can execute without the extend use case

The *<<includes>>* Relationship



- <<includes>> relationship Represents the inclusion of the functionality of one use case within another
- Arrow is drawn from the base use case to the included use case

- **The direction of an <<includes>> relationship is to the using use case (unlike <<extends>> relationships).**

Use Case formats

•Brief	• one-paragraph usually the main success scenario
•Casual	• Main success &alternative scenarios
•Fully dressed	• All steps and variations in details

Case Study

Patient information system for mental health care

Actor – Goal list

Actor	Goal(s)
Medical Receptionist	1-make appointments 2-Register Patient 3-View Personal Info 4-Unregister patient 5-transfer data
Nurse	1- View Record 2-Edit Record 3-collecting medical history
Doctor	1- View Record 2-Edit Record 3-view patient history 4-Setup Consultation 5-Generate Report

Actor – Goal list

Actor	Goal(s)
Manager, administrative staff	1-View Personal Info 2-record treatments and their costs 3-Export Statistics 4-Generate Report
Medical records staff	maintain the records system
Health visitor	1-View Personal Info 2-Setup Consultation 3-Check on treatment

Use case diagram

Examples of use case scenarios

Brief use case

- **Register Patient** (primary actor : medical receptionist).

When a new patient attends a clinic, the medical receptionist asks the about his/her data which include name, age, address, and next of kin. A new record is created by a medical receptionist and personal information (name, address, age, etc.) is added to it.

Casual use case

- **View personal info** (primary actor : medical receptionist).
  - **Main success scenario:** The medical receptionist logs on to the system to View Info about patient, supplying the receptionist's identifier to allow security checking . BY supplying the patient's identifier, PID , the database return the information required. Through the database, the system checks that the user is authorized for this action. If authorized, the patient information is returned and a form on the user's screen.
  - **Alternative scenario:** If authorization fails, then an error message is returned and the system asks the user to enter another identifier.

Fully dressed use case: Collecting medical history

- **Use case UC1:** Collecting medical history
- **Scope:** MHC-PMS
- **Primary actor:** Nurse
- **Stakeholders and interests:**
  - **Nurses** want to coordinate the consultations with doctors and administer some treatments without any errors.
  - **Patients** want to record their information in the system to help them avoid next consultation errors.
  - **Doctors** want to use patient medical history for assessing and treating patients.
  - **Manager** wants to ensure that the system meets current medical guidelines for patient care.

Fully dressed use case: Collecting medical history

- **Preconditions**
  - The patient has seen a medical receptionist who has created a record in the system and collected the patient's personal information (name, address, age, etc.).
  - A nurse is logged on to the system and is collecting medical history.
- **post-condition**
  - The patient record including medical history is entered in the database
  - A record is added to the system log showing the start and end time of the session and the nurse involved.

Fully dressed use case: Collecting medical history

- **Main success scenario:**

1. The nurse searches for the patient by family name. If there is more than one patient with the same surname, the given name (first name in English) and date of birth are used to identify the patient.
2. The nurse chooses the menu option to add medical history.
3. The nurse then follows a series of prompts from the system to enter information about consultations elsewhere on mental health problems (free text input), existing medical conditions (nurse selects conditions from menu), medication currently taken (selected from menu), allergies (free text), and home life (form).

Fully dressed use case: Collecting medical history

- **Extensions (Alternatives):**

***a. At any time, system fails**

1. Nurse restarts System, logs in, and requests recovery of prior state.
2. System reconstructs prior state.

2a. System detects anomalies preventing recovery:

1.System signals error to the Nurse, records the error, and enters a clean state.

2. Nurse starts from the beginning.

***b. Patient cannot/will not provide information on medical history.**

1.The nurse should enter free text recording the patient's inability /unwillingness to provide information.

2.The system should print the standard exclusion form stating that the lack of information may mean that treatment will be limited or delayed. This should be signed and handed to the patient.

Fully dressed use case: Collecting medical history

- **Extensions (Alternatives):**

1a. The patient's record does not exist or cannot be found.

1. The nurse should create a new record and record personal information.

3a. Patient conditions or medication are not entered in the menu.

1. The nurse should choose the 'other' option and enter free text describing the condition/medication.

Fully dressed use case: Collecting medical history

- **Special requirements:**
  - Touch screen , scanner , papers ,pen, keyboard, …….
- **Frequency of occurrence:**
  - It’s a main function that can continuously implemented.
- **Open Issues:**
  - How to ensure the valid medical history?
  - How to use medical history to improve patient care?
  - can the patient directly enter his history, or does the nurse have to do?

Assignment #3

1- Complete chapter 3 and sections 4-1 and 4-2 in chapter 4 in your SRS.

2- Send it.

Thanks