CIS*3750 - System Analysis and Design in Applications

Luiza Antonie, Fall 2025, University of Guelph

Week 10 Demo (Nov 3 & 5)

- In person, during your regular lab session
- 10-minute live demo of your high-fidelity prototype
- It must showcase **2-3 key user walkthroughs** focused on the core "intelligent" learning functionality of your ITS
- Simple walkthroughs like account creation or logging in will not be considered a valid main walkthrough
- While a code-based UI is recommended, navigating a Figma prototype is also acceptable
- Provide a shareable link to your prototype (e.g., GitHub or Figma link) for review. The entire group must be present for the demo.

Unified Modeling Language (UML)

- UML is a modeling language in software engineering
- It provides a standard way of visualizing the design of a system

What is UML?

- A software engineering tool
- Designed to help visualize the system
- Created/developed between 1994 & 1996 by Grady Booch, Ivar Jacobson, and James Rumbaugh

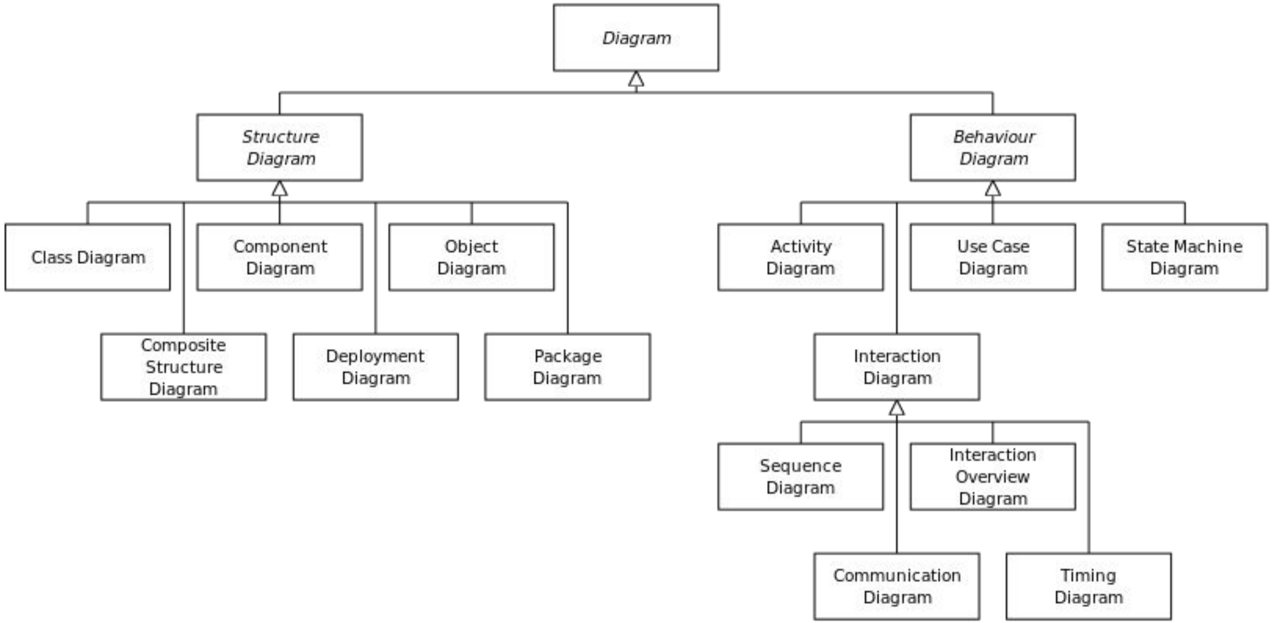
When do you use UML?

- Analysis and design of software
- Communication
- Modeling software (object oriented)

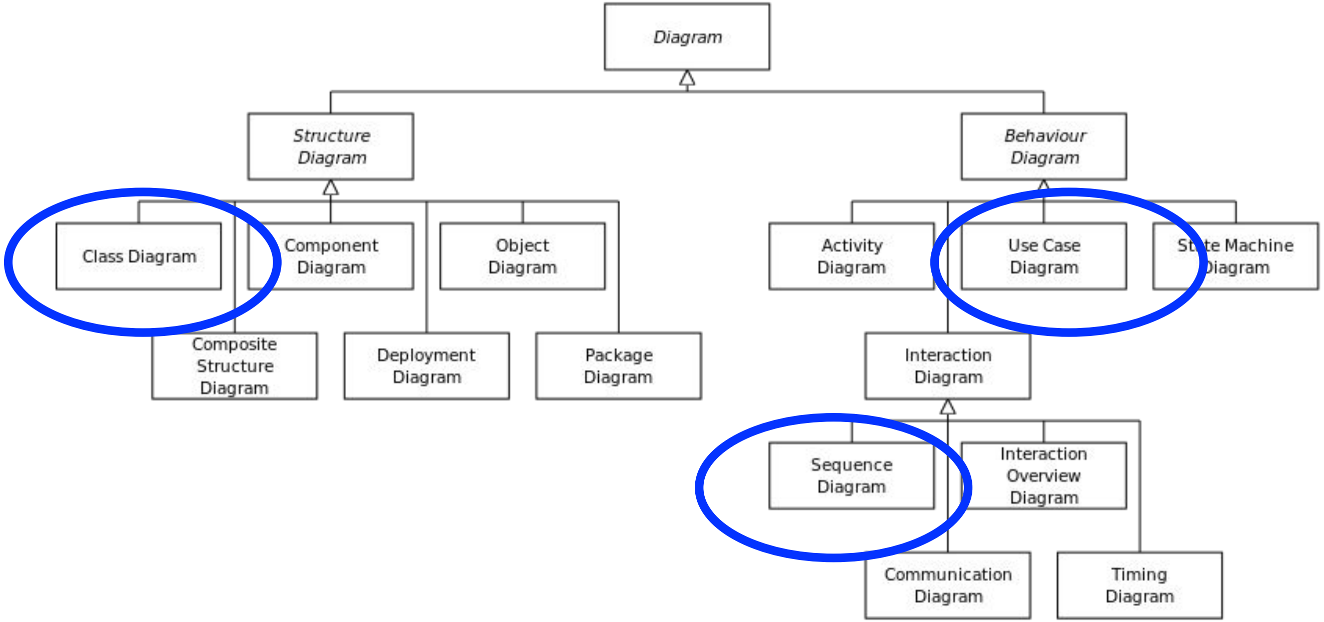
UML

- Visualize 2 views/aspects of the system:
  - Static/Structural - helps visualize the static structures of the system
      - objects
      - attributes
      - operations
      - relations
  - Dynamic/Behavioural - helps visualize the dynamic nature of the system

UML Diagrams



UML Diagrams



Use Cases

- Describe an activity the system performs in response to a request by an actor (e.g., user)
- Are logical models; they describe the activity without implementation details
- Tools to model system requirements
- Focus on users and their goals

Use Cases - Benefits

- Modeling tool used in analysis
- Useful in communication
- Useful in functional decomposition

Steps in Use Case Modeling

- Find actors
- Find use cases
- Describe how actors interact with use cases
- Present and discuss the use case model(s) as diagrams
- Test and evaluate use cases

Use Cases

- Are logical models; they describe the activity

- Functionality or goal
  - described by a verb phrase
  - linked to actors

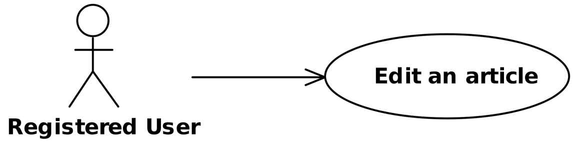


Image source: https://en.wikipedia.org/wiki/Use_case

Use Cases

- Describe an activity the system performs in response to a request by an actor
- Actors ⟷ Think of the their role
  - person
  - company or organization
  - computer system or program (software or hardware)

Use Cases - Actors

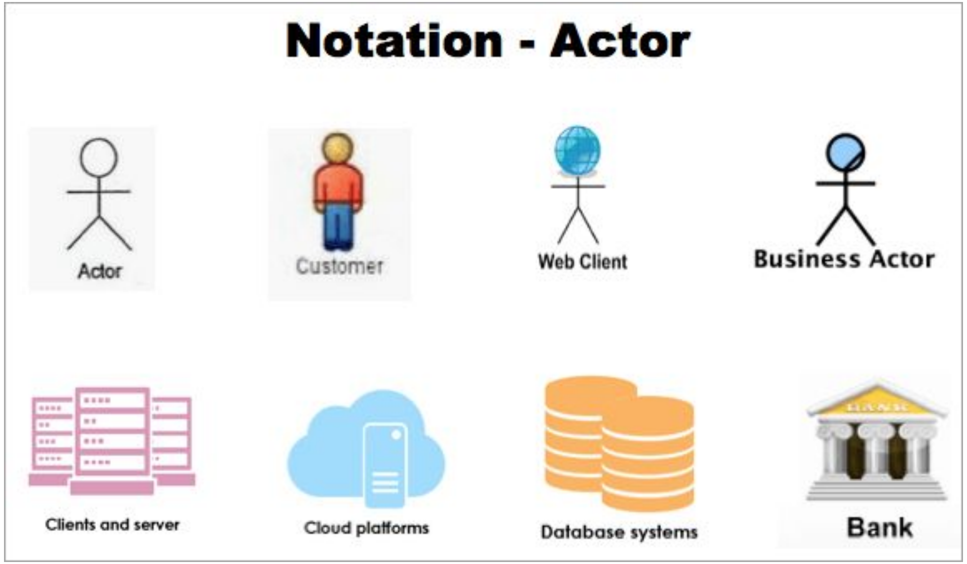
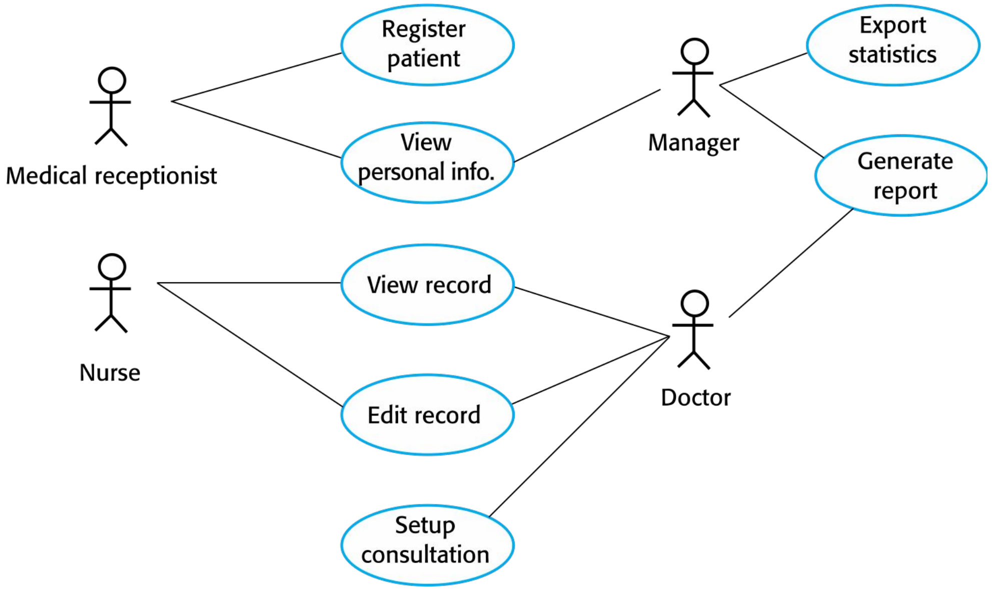


Image source: https://www.softwaretestinghelp.com/use-case-diagram-tutorial/

Use Case Diagram

- Describes the externally observable behaviour of a system
- Describes the main uses of the system
- Includes a set of use cases that describe all possible interactions with the system
- Useful for starting to prototype

Use Case Diagram

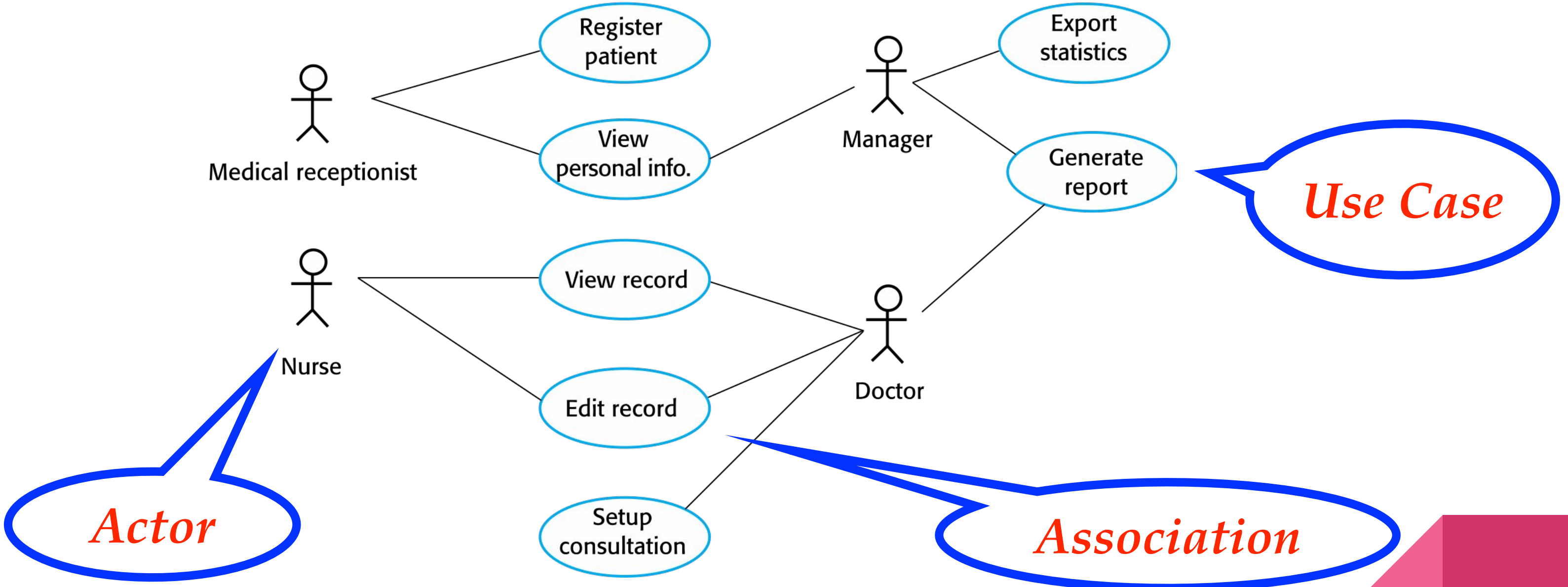


https://iansommerville.com/software-engineering-book/

Use Case Diagram - Components

- **Actors** - who interacts with the system
- **Use cases** - describe what the system does, its functionality
- **Associations** - shows the relationships between actors and use cases

Use Case Diagram - Components



https://iansommerville.com/software-engineering-book/

Use Case Diagram

- High-level description of the system
- Describes all possible interactions with the system
- Provides a summary of the relationships between use cases and actors

How to Create a Use Case Diagram

- Identify actors
  - persons, systems, organizations

- Identify use cases
  - functionality
  - what actors need from the system

Use Case Diagram - Activity (3-5 students)

- Draw a use case diagram for an app to manage a bookstore.

Class Diagram

- A type of <u>static</u> structure diagram
- Describes the static structure of a software system
- It shows the system's classes, their attributes, operations (or methods), and the relationships among the classes.

Class Diagram

- What you should know from a class diagram:
  - what are the classes
  - what are the properties (attributes) of those classes
  - what are the operations (methods) of those classes
  - what are the relationships among the classes

Class Diagram

- When to use a class diagram:
    - Communication tool (e.g., clarify requirements with stakeholders)
    - Plan prototype
    - Design document
    - Identifies relationships
    - Identifies highly coupled sections of code

Class Diagram

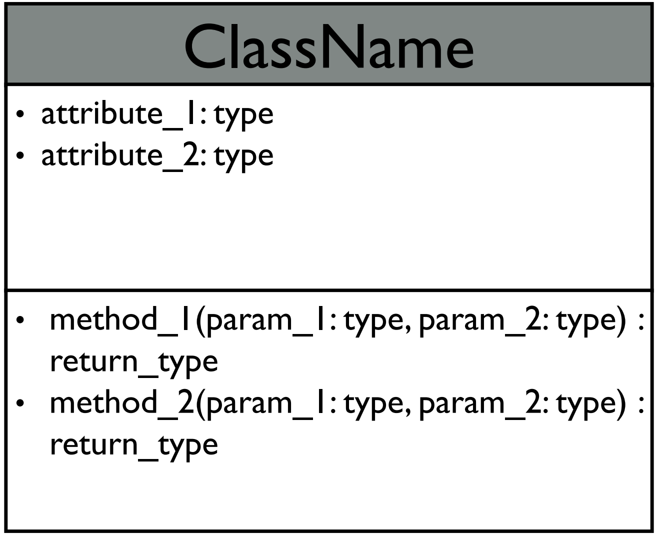
- Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes.
- An object class can be thought of as a general definition of one kind of system object.
- An association is a link between classes that indicates that there is some relationship between these classes.

https://iansommerville.com/software-engineering-book/

Class Diagram

- A class is a description for a group of objects:
  - similar properties (attributes)
  - common behaviour (methods)

Class Diagram

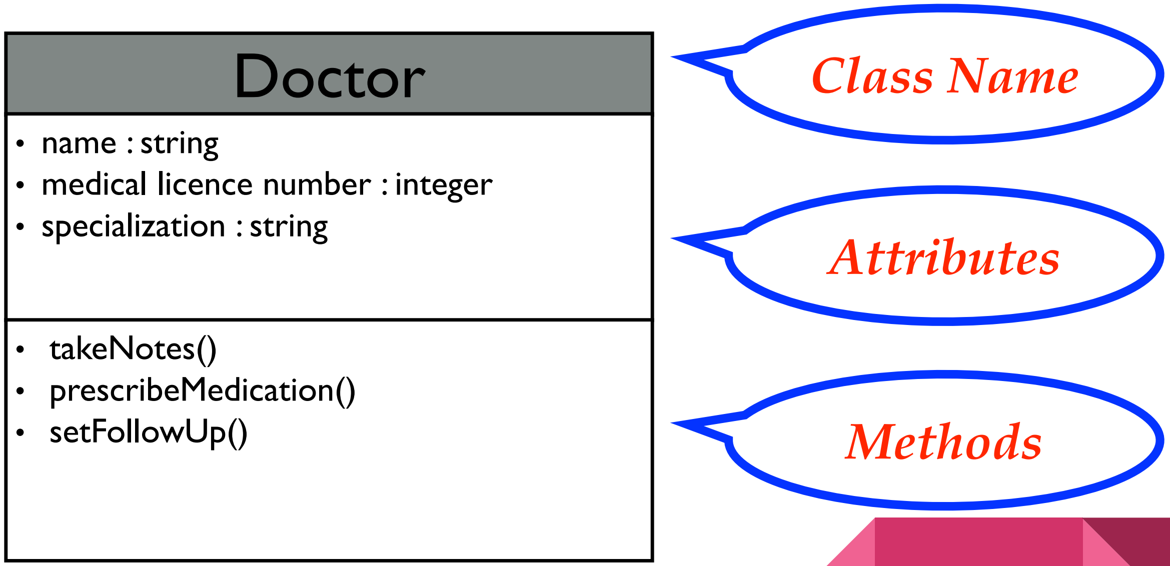


ClassName

- attribute_1: type
- attribute_2: type

- method_1(param_1: type, param_2: type) : return_type
- method_2(param_1: type, param_2: type) : return_type

Class Diagram



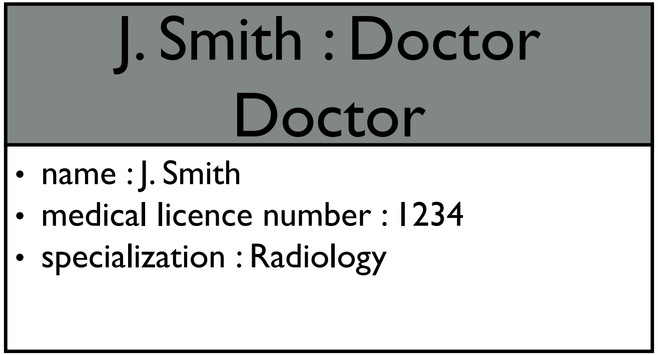
Doctor

- name : string
- medical licence number : integer
- specialization : string

- takeNotes()
- prescribeMedication()
- setFollowUp()

Class Diagram

- The instances of a class are called objects
- What is an instance?
  - An instance is a realization of a class
  - It’s the thing that is built based on the blueprint

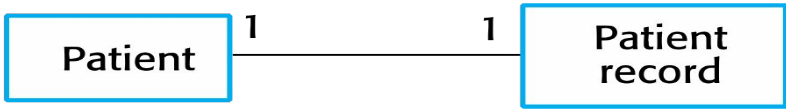


J. Smith : Doctor
Doctor

- name : J. Smith
- medical licence number : 1234
- specialization : Radiology

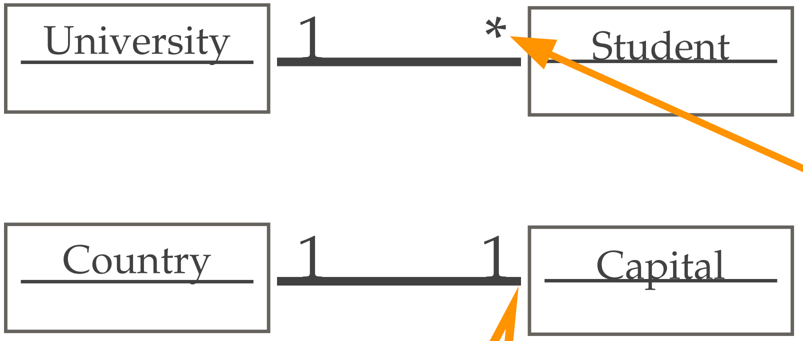
Class Diagram - Relationships

- Association (line)
- Inheritance (open triangle)
- Aggregation (open diamond)
- Composition (solid diamond)



https://iansommerville.com/software-engineering-book/

Class Diagram - Association



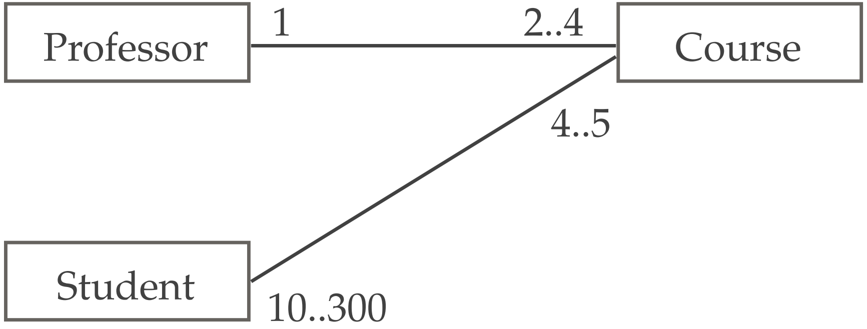
Multiplicity
A university has one or more students (0..* = *)

Multiplicity
A country has one capital

Class Diagram - Multiplicity

- Exactly 1 ——> 1
- Zero or more ——> 0..* or *
- One or more ——> 1..*
- Zero or one ——> 0..1
- A range of values ——> 1..3

Class Diagram - Association


Professor
1
2..4
Course
4..5
Student
10..300