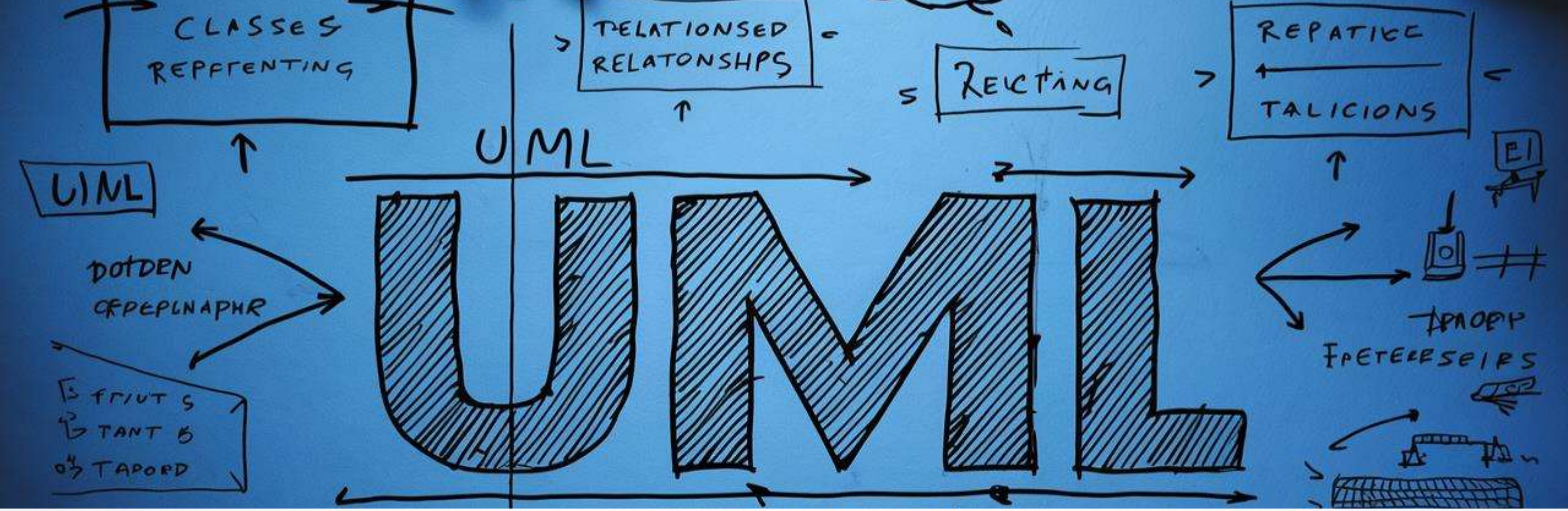


Object Oriented Analysis and Design using Unified Modeling Language

Er. Shiva Kunwar

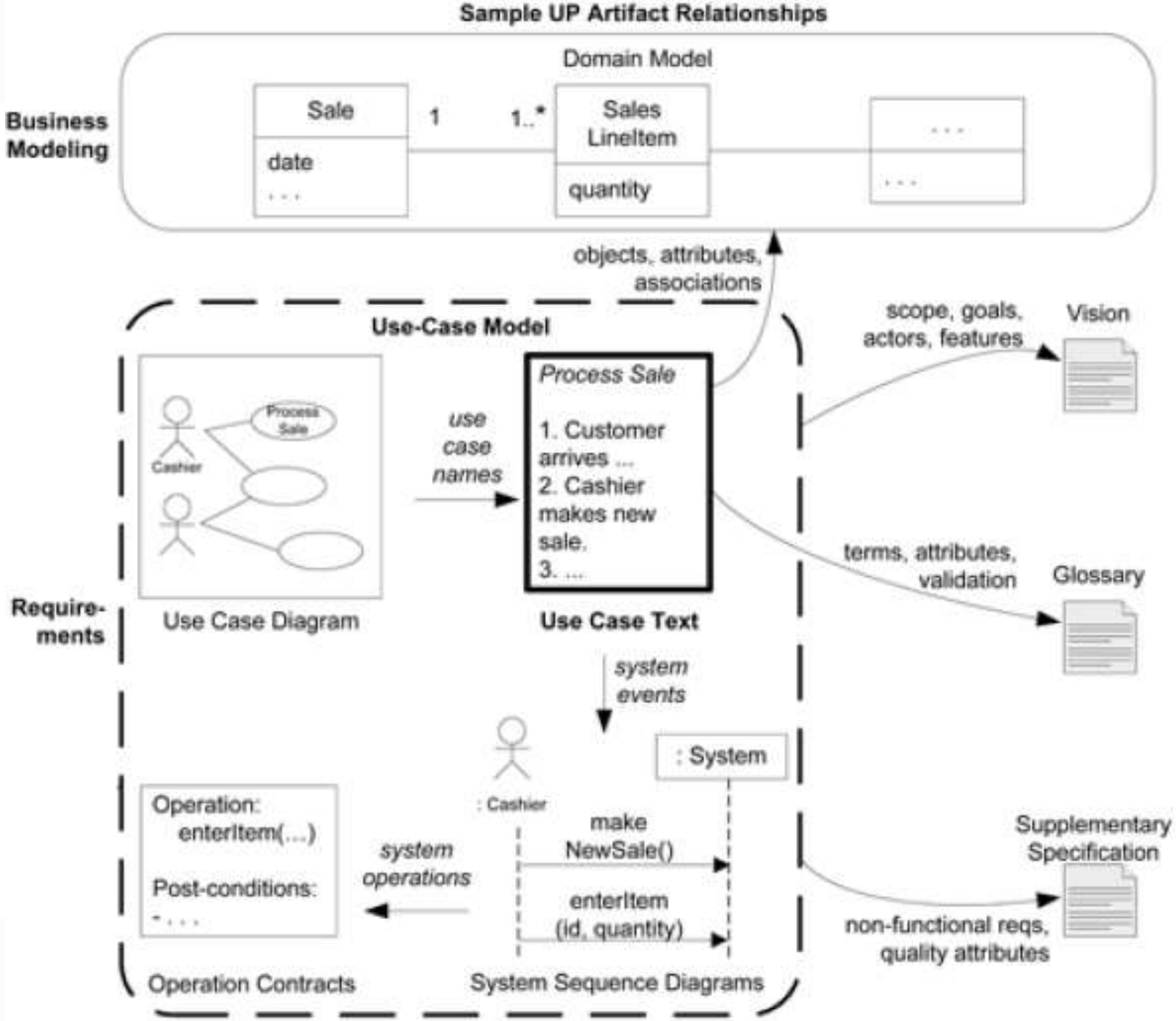
Lecturer, GU

Lesson 1: Object Oriented Fundamentals (10hrs)

- Introduction
- Object Oriented Analysis and Design
- Iterative development and unified process
- Case Study
- Understanding Requirements
- **Use Case modeling, Relating Use cases — include, extend and generalization**
- Overview of the Unified Modeling Language: UML Fundamentals and Notations

USE CASE Exam

- Create a use case diagram for a Library Management System. The system should allow members to search for books, borrow books, and return books. Librarians should be able to add new books, remove old books, and manage member records.
- Create a use case diagram for an Event Management System. The system should allow attendees to register for events, view event schedules, and provide feedback on sessions. Event organizers should be able to create new events, schedule sessions, and manage attendee lists. Vendors should have the ability to apply for booth space and manage their profiles.


Sample UP Artifact Relationships
Business Modeling
Domain Model
Sale
date
...
1
1..*
Sales LineItem
quantity
...
...
objects, attributes, associations
Use-Case Model
scope, goals, actors, features
Vision
Process Sale
Cashier
use case names
Process Sale
1. Customer arrives ...
2. Cashier makes new sale.
3. ...
terms, attributes, validation
Glossary
Require-ments
Use Case Diagram
Use Case Text
system events
: System
: Cashier
Operation:
enterItem(...)
Post-conditions:
- ...
system operations
make NewSale()
enterItem (id, quantity)
Supplementary Specification
non-functional reqs, quality attributes
Operation Contracts
System Sequence Diagrams

USE CASE Modeling

- Use case modeling is a technique in software engineering to visually represent the interactions between actors (users or external systems) and the system under consideration.
- It helps in understanding, documenting, and validating the functional requirements of a system from an end-user perspective.
- Use cases are text stories, widely used to discover and record requirements.

Understanding Use Cases

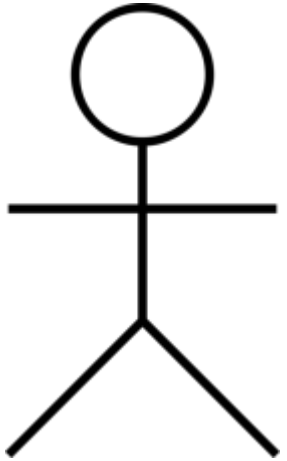
- Use Case Diagrams
- Use Case Descriptions

Use Case Diagram

- Use case is used to model the system or subsystem of an application.
- Use cases represent specific interactions between external actors (users or systems) and the system to achieve a particular goal.
- Use cases capture the functionalities the system provides to its users.
- Each use case should represent a meaningful and cohesive piece of functionality.

Use Case Diagram

- **Actors**
- Definition: Actors are external entities (users or systems) that interact with the system by participating in one or more use cases.
- **Key Points:**
- Actors represent roles played by external entities.
- Actors initiate and participate in use cases.
- Represented by stick figures.

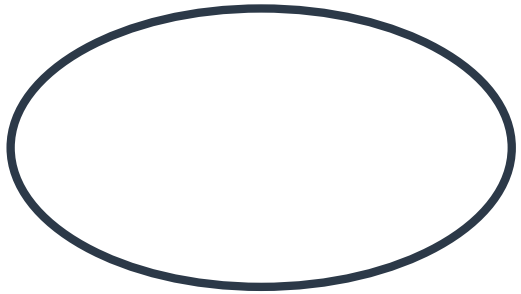


Use Case Diagram

- **Types of Actors**
- **Primary**: Lead actor, direct participation with the system
- **Secondary**: Supporting actor, provides information to the primary actor
- **Off-Stage**: Indirect participation provides support to both primary and secondary actors.
- ATM System: - P: Customer S: Machine O: Bank

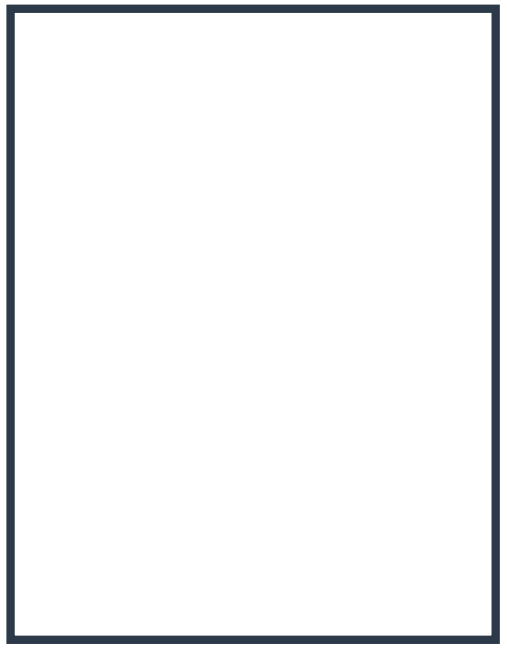
Use Case Diagram

- **Use Case:**
- Definition: Use case describes a set of actions or services the system provides to tits actors to accomplish a specific goal.
- **Key Points:**
- They are the features of the system.
- Actors interact to achieve the objective of the use case.
- Represented by the oval shapes.



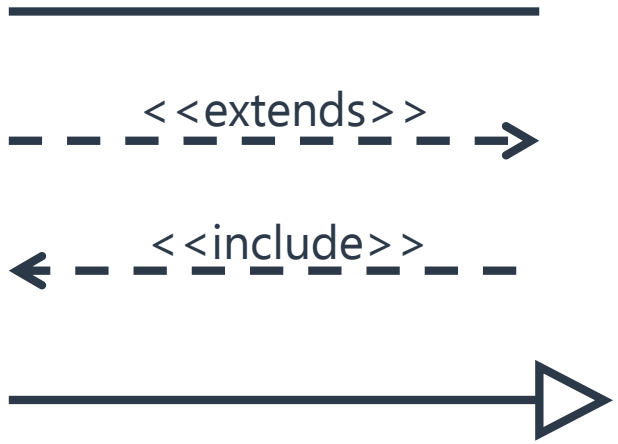
Use Case Diagram

- **System Boundaries:**
- Definition: The system boundary separates the system from its external environment, defining what is inside and outside the system.
- **Key Points:**
- System boundaries help in defining the scope of the system.
- External entities interact with the system at the boundary.
- Represented by the box around the use cases.



Use Case Diagram

- **Association:**
- Definition: The line connecting an actor to a use case indicates that the actor performs that particular use case.
- **Key Points:**
- Represented by a line, line with an arrow, or dashed arrow.



Use Case Diagram

- **Multiplicity:**
- Definition: Multiplicity defines the number of instances of one class related to one instance of the other class in an association.
- **Key Points:**
  - Multiplicity indicates the number of actors or use cases associated with each other.
  - It is expressed using numeric ranges.
  - Represented by numbers like 0,1 or 0....* or 1....*.

Use Case Diagram

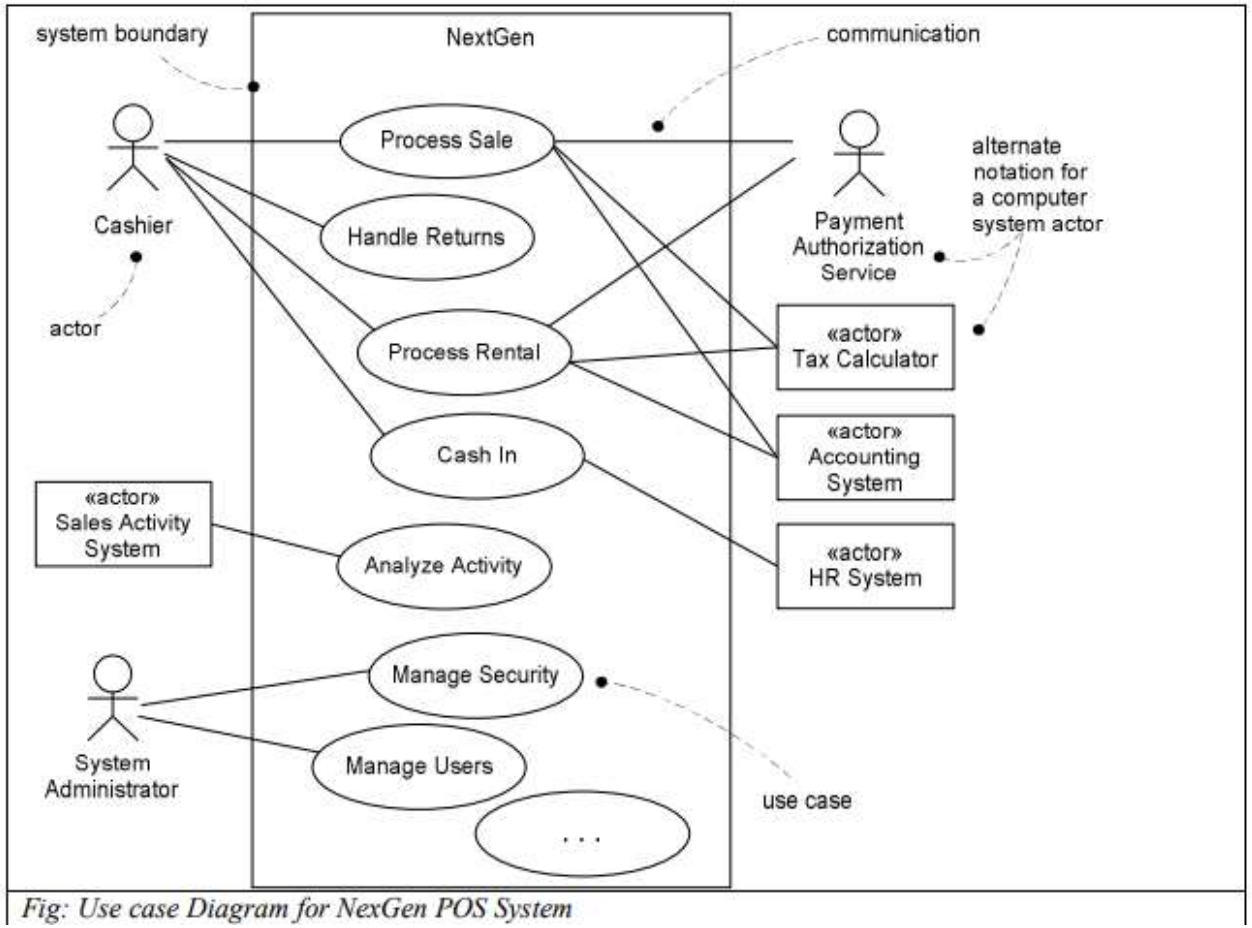


Fig: Use case Diagram for NexGen POS System

Use Case Diagram

- Prepare a Use Case Diagram of your project.
- Guidelines:
  - Choose system boundary. (Identify internal vs external actors)
  - Identify primary actors
  - For each actor, identify user goals.
  - Define use cases that satisfies user goals.

Use Case Description

- **Components of a Use Case Description:**
- Use case descriptions provide detailed information about a specific use case, including its behavior and requirements.
- **Key Components:**
- **Preconditions:**
- Conditions that must be true before the use case is executed.
- **Main Flow:**
- The primary sequence of actions that occur in a use case.

Use Case Description

- **Alternative Flows:**
- Variations or alternative paths that may be taken during the execution of the use case.
- **Postconditions:**
- Conditions that must be true after the use case is executed successfully.

Use Case Description

- **Key Points:**
- Use case descriptions are essential for documenting the expected behavior of the system.
- Preconditions set the context for the use case.
- The main flow outlines the standard or expected course of actions.
- Alternative flows cover deviations or variations in the main flow.
- Postconditions define the expected state or outcome after the use case is completed.

Use Case Description

- For an "Online Purchase" use case:
- **Preconditions**: Customer is logged in and has items in the shopping cart.
- **Main Flow**: Customer selects items, adds them to the cart, proceeds to checkout, provides shipping details, makes payment, and completes the purchase.
- **Alternative Flow**: If the payment fails, the system prompts the customer to retry.
- **Postconditions**: Items are removed from the cart, and the customer receives a confirmation email.

Use Case Description Formats

- **Brief (high Level)**: concise one-paragraph summary, usually of the main success scenario.
- It briefly describes a process, usually in two or three sentences.
- It is useful to create this type of use case during the initial requirements to understand the complexity and functionality of a system quickly.
- They are very terse and vague on design decisions and essential in nature.
- It is useful to start with high-level use cases to obtain some understanding of overall major processes quickly.

Use Case Description Formats

- **Casual**: informal paragraph format.
- It comprises multiple paragraphs that cover various scenarios.
- Addresses main and alternate scenarios and provides backup plan.
- They are real and concrete in nature,

Use Case Description Formats

- **Fully Dressed**: the most elaborate.
- All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees.
- These detailed use cases are written after many use cases have been identified and written in a brief format.

Use Case Description

- Use Case ID
- Use Case Name (Start with a verb)
- Scope (The system under design)
- Level (“user-goal” or “subfunction”)
- Primary Actor (Calls on the system to deliver its services)
- Stakeholders and Interests(Who cares about this use case and what do they want)
- Preconditions (What must be true on start, and worth telling the reader)
- Success Guarantee(Postconditions) (What must be true on successful completion, and worth telling the reader)

Use Case Description

- Main Success Scenario (or Basic Flow) (A typical, unconditional happy path scenario of success)
- Extensions (or Alternative Flows) (Alternate scenarios of success or failure)
- Special Requirements (Related non-functional requirements)
- Technology and Data Variations List (Varying I/O methods and data formats)
- Frequency of Occurrence (Influences investigation, testing, and timing of implementation)
- Open Issues

Use Case Description

Use case #6

Use case name: PauseGame

Participating actors: Player

Entry condition: Player is already playing the game.

Exit condition:

- Player continues to play the game, OR
- Player can choose to return to the main menu, OR
- Player can exit from the whole game, OR
- Player can load another game, if there is already a game saved.

Main Flow of Events:

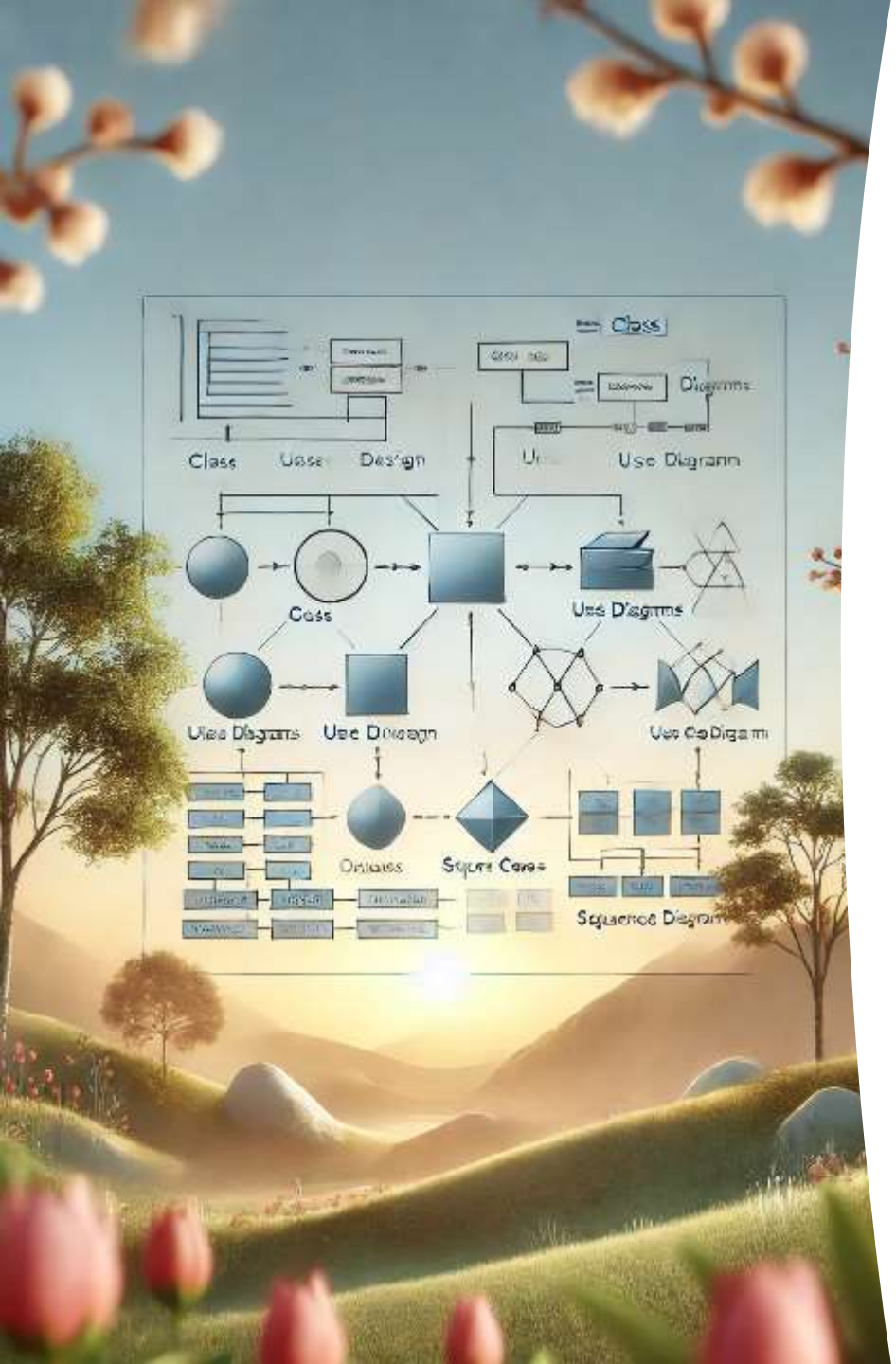
1. Player presses the pause button during the game.
2. The system displays the in-game menu.
3. Player continues the game.

Alternative Flow of Event:

- Player choses to save the game and returns to the in-game menu.
- Player choses to return to the main menu.
- Player directly exits the game instead of continuing the game.
- Player loads another game. (go to step 3) If there is no saved game load button will be disabled and warning massage will be shown.

Use Case Description

- A use case describes three things:
  - An actor (user) that initiates an event
  - An event that triggers a use case
  - The use case that performs the actions triggered by the even



END OF LECTURE

SHIVA.KUNWAR@HOTMAIL.COM
+977-9819123654

PREVIEW FOR LECTURE 5

USE CASE MODELING cont.