Bismillah

In The Name of Allah

PRESENTED BY:

f2023266l124 M Rafay Shah

f2023266l126 Saim Jameel

S2023266095 Feroz

Title:

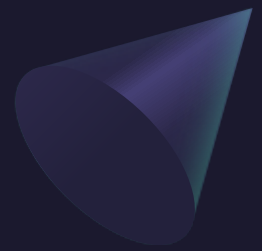
Understanding UML: The Language of Software Design

Subtitle:

A Comprehensive Guide to Unified Modeling Language

Key points:

- What Is UML?
- Purpose of UML
- Importance Of UML In Software Engineering
- Key UML Diagrams



What is UML?

- UML stands for Unified Modeling Language.
- It is a standardized visual language for specifying, constructing, and documenting software systems.

Purpose:

- Make complicated systems easy to understand with diagrams.
- Help people share ideas and work together effectively.

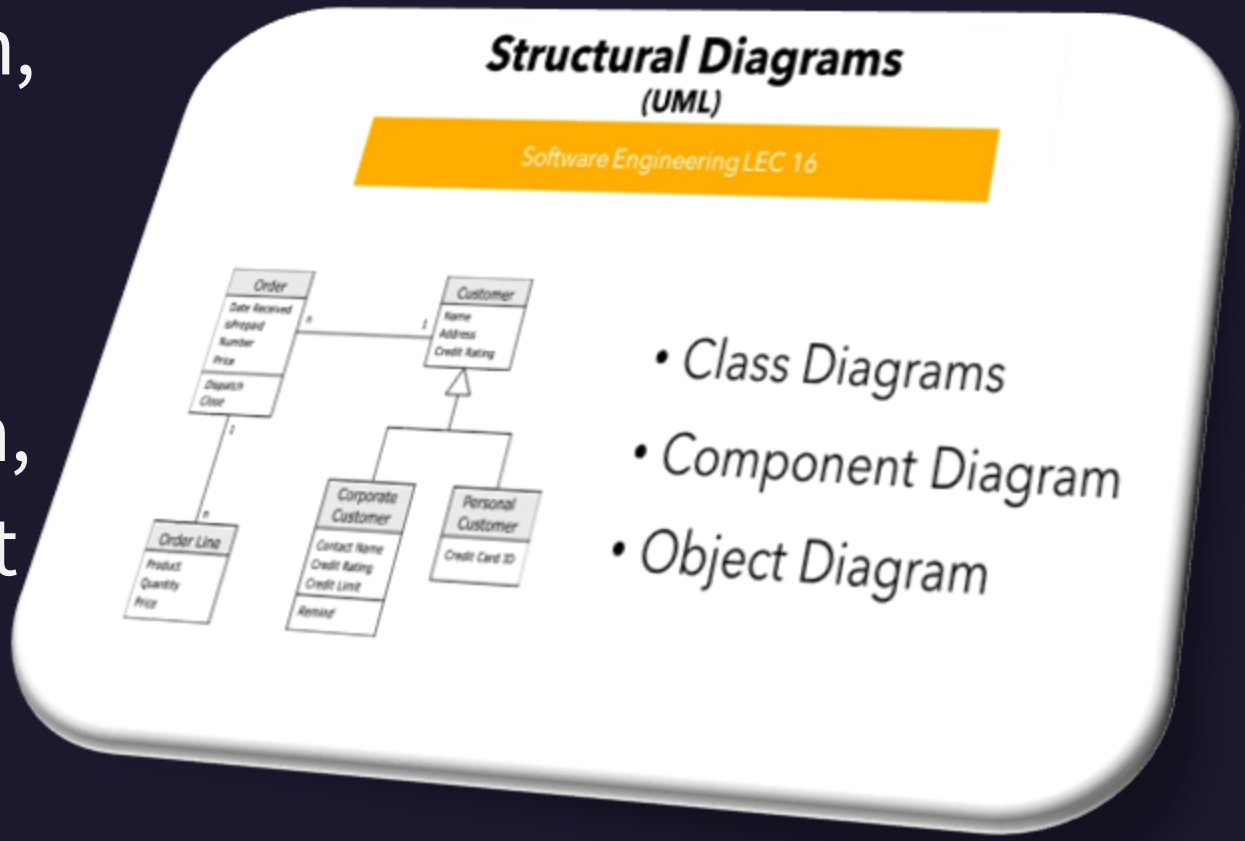
Importance of UML in Software Engineering

- Facilitates better understanding of system requirements.
- Enhances communication between developers, analysts, and clients.
- Provides a blueprint for system architecture and design.
- Encourages reusability and scalability of software components.

Key UML Diagrams

1) Structural Diagrams:

- Show the static parts of a system, like how components or classes are connected.
- Examples include Class Diagram, Object Diagram, and Component Diagram.



2)Behavioral Diagrams:

- Focus on the dynamic actions of a system, such as workflows or interactions over time.
- Examples include Use Case Diagram, Sequence Diagram, and Activity Diagram.

Structural Diagrams:

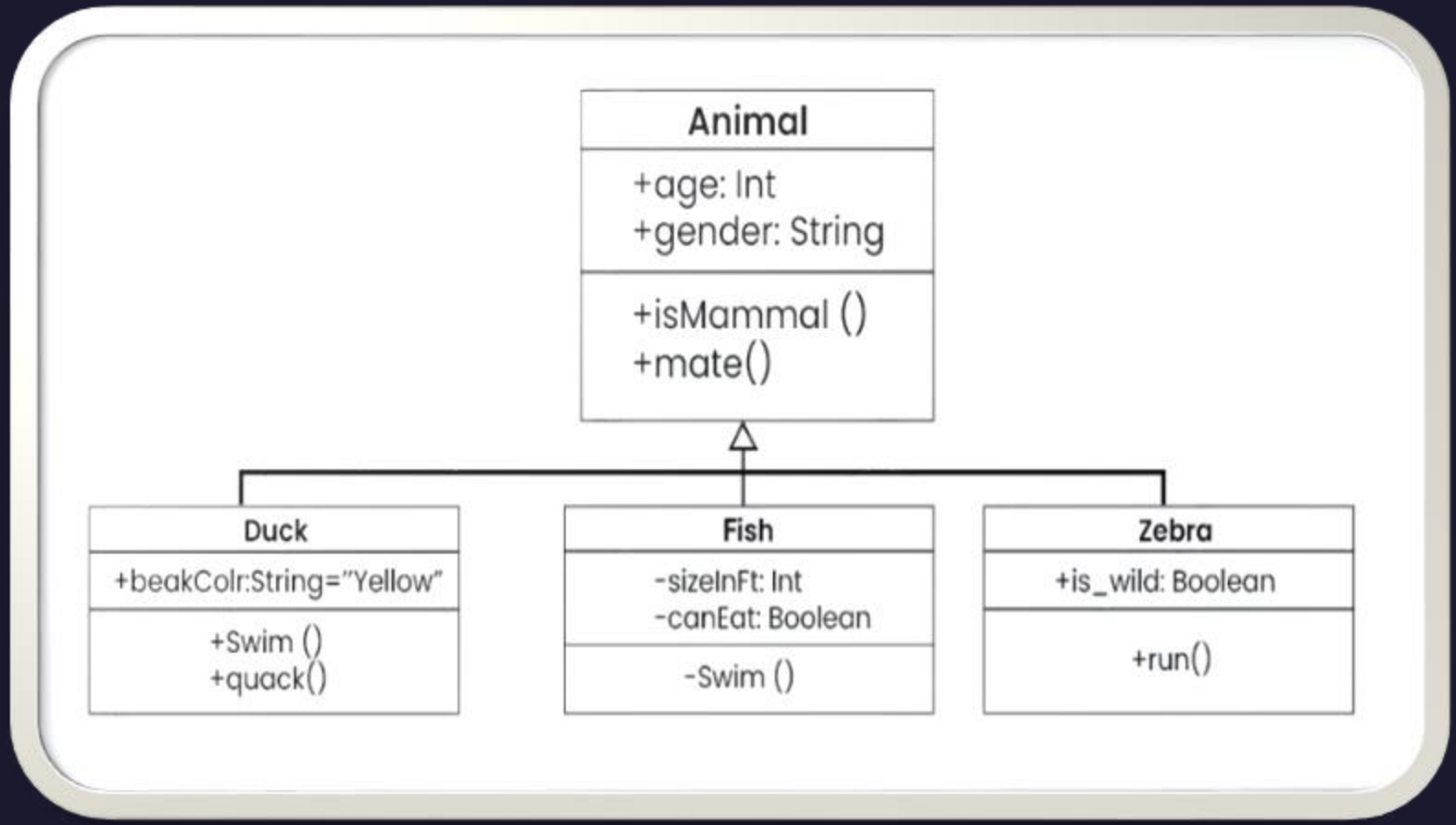
Class Diagram:

- Shows classes , their attributes, and relationships.
- Acts as a blueprint for coding.

➢Why It's Useful:

➢Makes the system easy to plan and understand.

➢Helps organize code better.



Structural Diagrams:

Object Diagram:

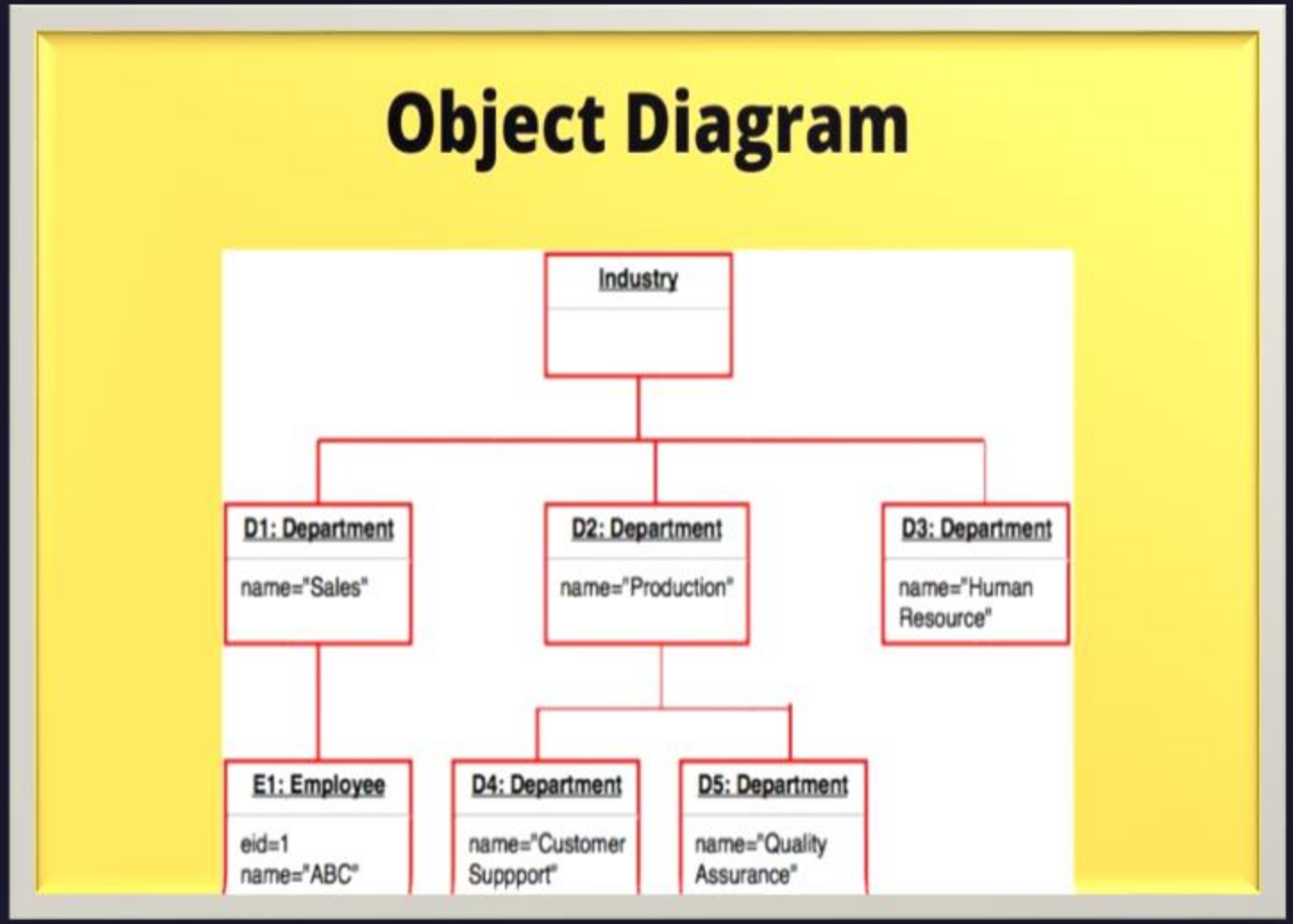
- Focuses on real examples

of classes at a specific time.

- Shows how objects interact in a real-world scenario.

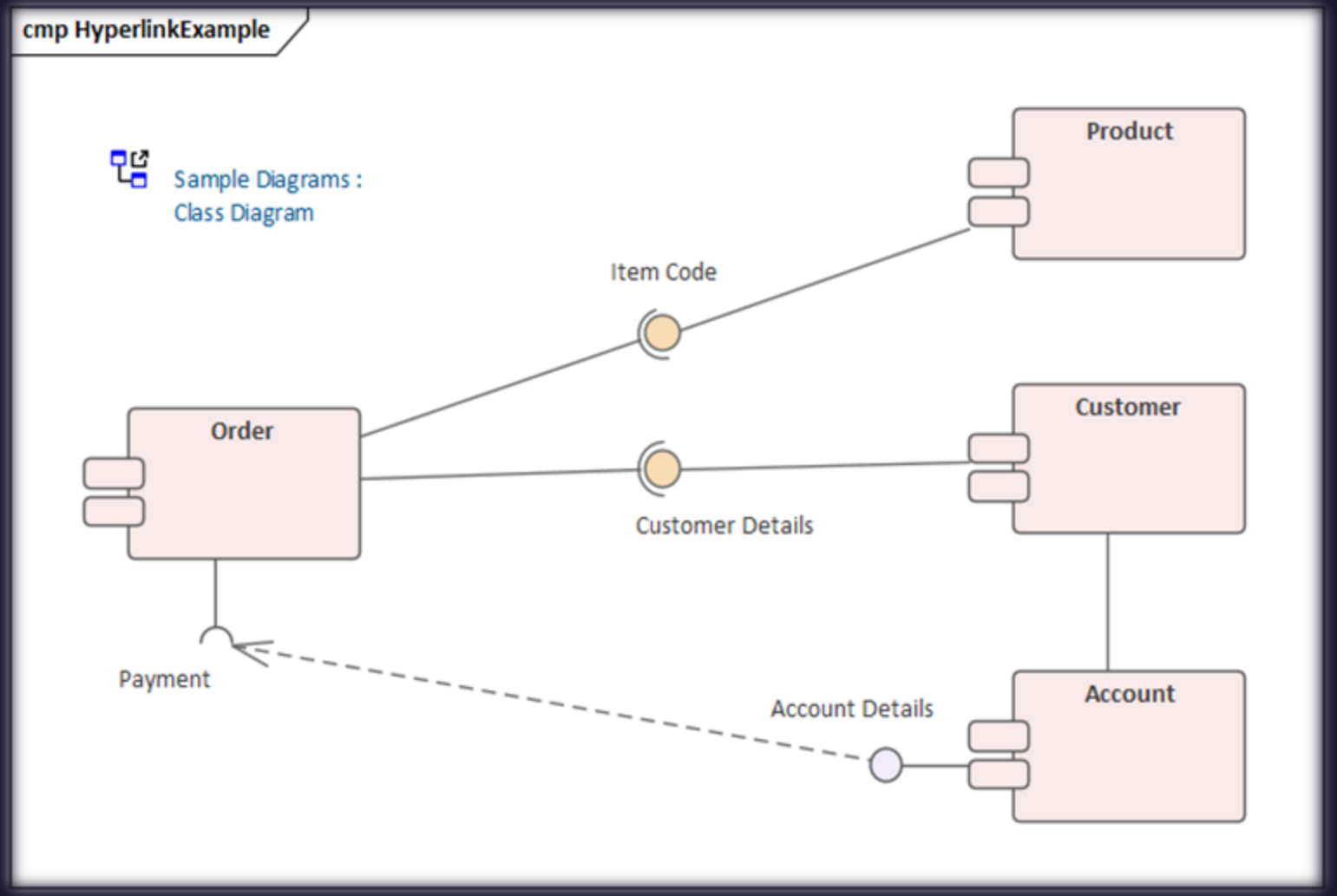
Why It's Useful:

- Helps debug and understand how the system behaves at runtime.



Structural Diagrams:

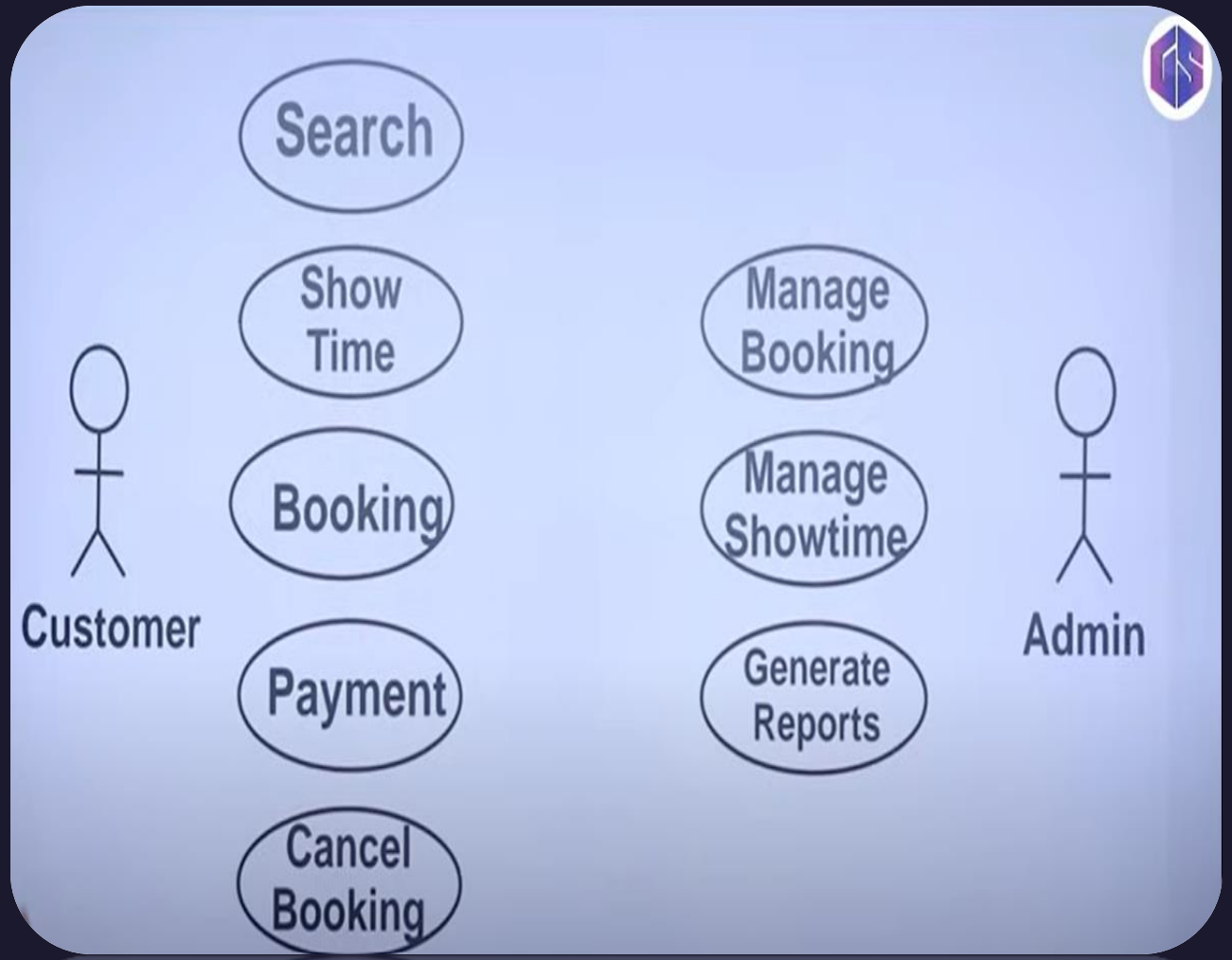
- **Component Diagram:**
- Displays system parts and how they depend on each other.
- Helps understand how the system is divided into parts.
- **Why It's Useful:**
- Makes complex systems easier to manage.
- Helps plan system upgrades or changes.



Behavioral Diagrams:

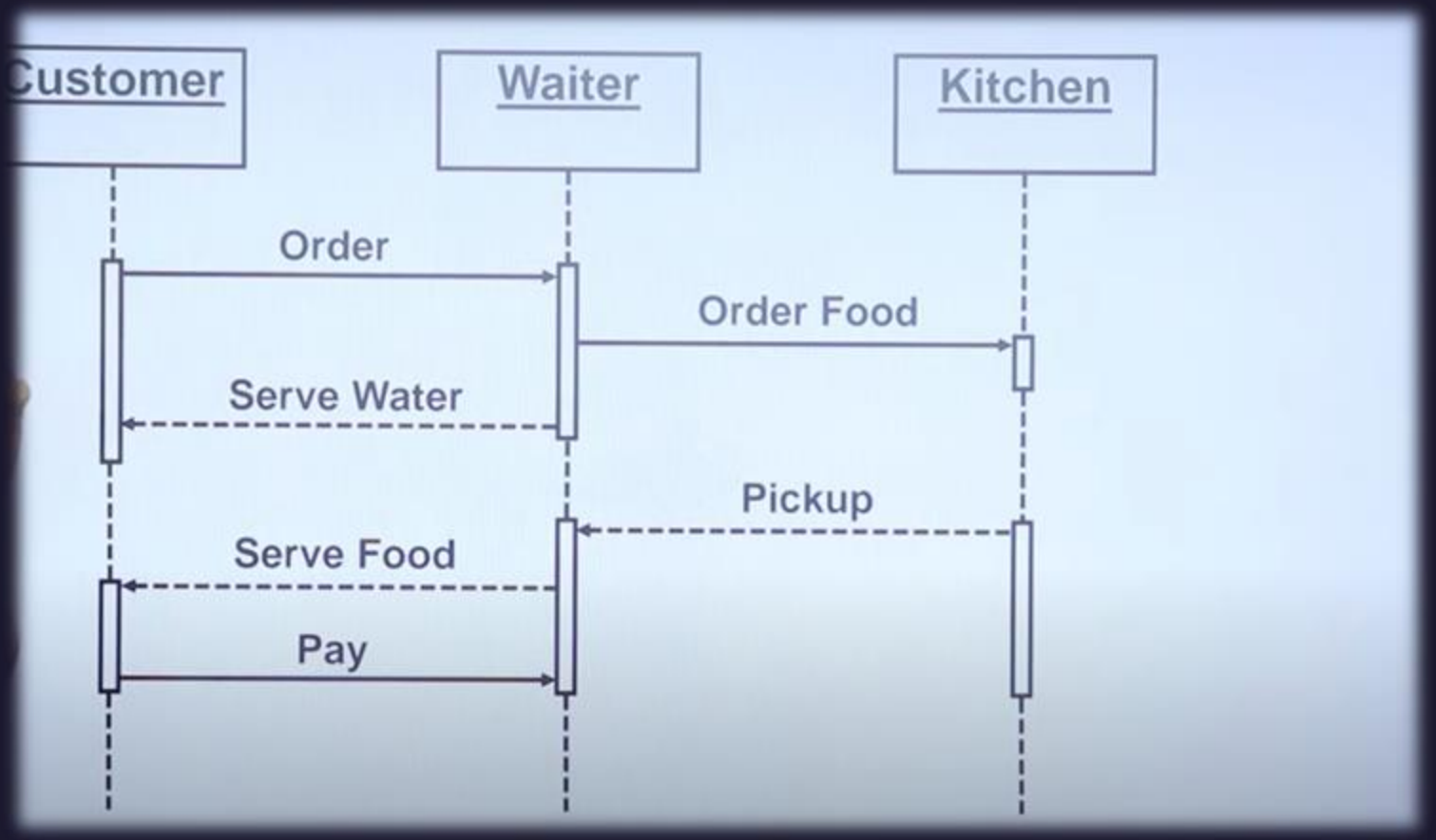
Use Case Diagram:

- Shows what users can do with the system.
- Identify system requirements.
- **Why It's useful:**
- Helps developers and stakeholders communicate clearly.



Behavioral Diagrams:

- **Sequence Diagram:**
  Shows the step-by-step flow of actions between system parts.
- Visualize how tasks or processes happen in a specific order.
- **Example:**



Login Process:

Behavioral Diagrams:

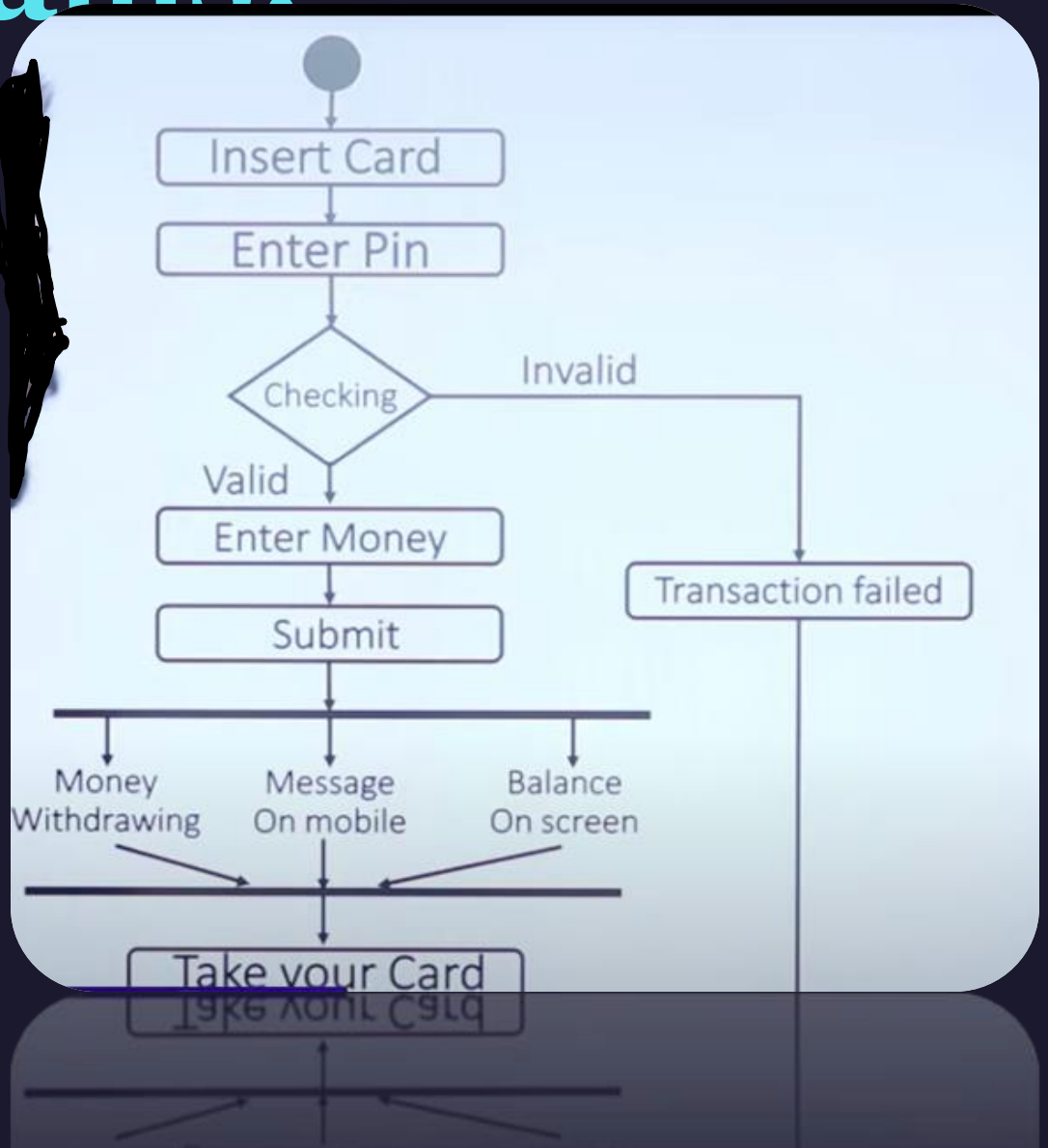
Activity Diagram:

The step-by-step flow of a process or task in the system.

- Helps stakeholders understand how a process works.
- Makes it easier to improve or debug workflows.

Example:

ATM machine




THANK YOU!