SYSTEM ANALYSIS AND DESIGN

USE CASE DIAGRAMS

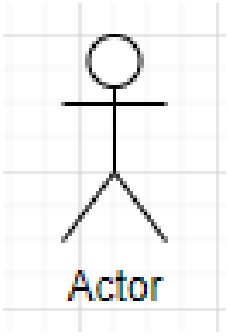
A use case diagram it provides a visual representation of how users interact with a system. It serves as a blueprint for understanding the functional requirements of a system from a user's perspective, aiding in the communication between stakeholders and guiding the development process.

Use case Notations:

UML notations provide a visual language that enables software developers, designers, and other stakeholders to communicate and document system designs, architectures, and behaviors in a consistent and understandable manner.
Some of the notation include:

(i) Actor

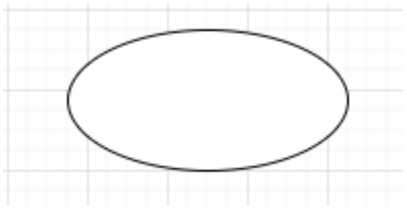
Actors are external entities that interact with the system. These can include users, other systems, or hardware devices.



(ii) Use cases

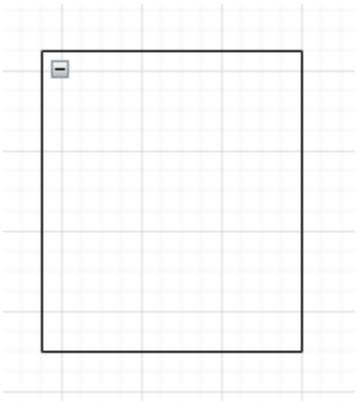
Use cases are like scenes in the play. They represent specific things your system can do. For example logging in, viewing the cart, adding the items to the cart

They are represented as ovals



(iii) System Boundary

The system boundary is a visual representation of the scope or limits of the system you are modeling. It defines what is inside the system and what is outside. The boundary helps to establish a clear distinction between the elements that are part of the system and those that are external to it. It is represented box within which all the use cases are enclosed



Use case Diagram Relationships

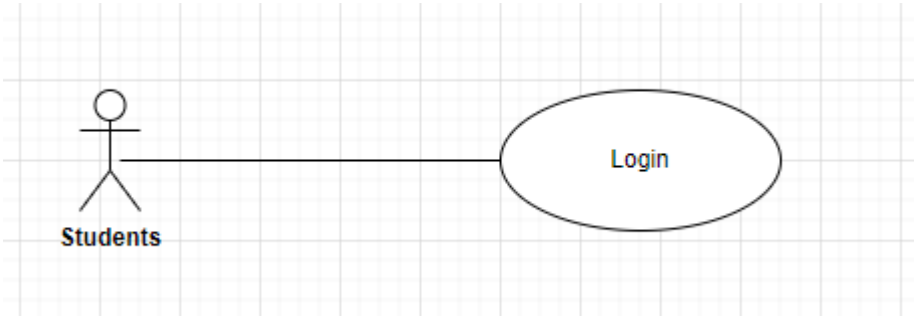
In use case diagrams, relationships play a vital role as in depicting the interactions between actors and use cases. They provide a comprehensive view of the system's functionality and its various scenarios.

Relationships include the following:

Association

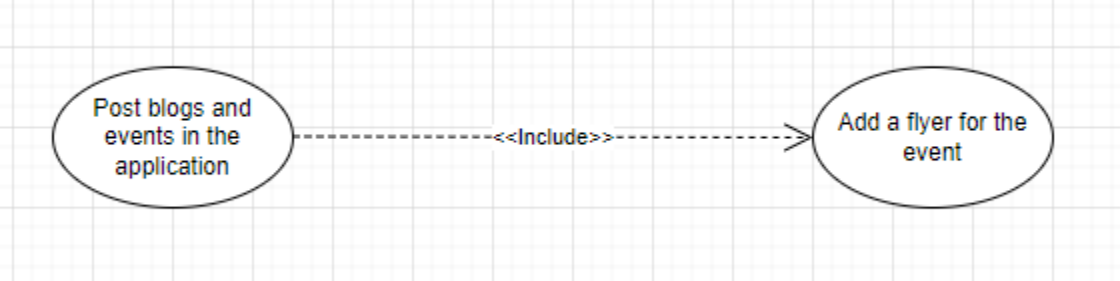
It represents a communication or interaction between an actor and a use case. It is shown by a line connecting the actor to the use case. This relationship signifies that the actor is involved in the functionality described by the use case.

For example as per my project, UniPal, the Student interacts with the application by logging in.



Include

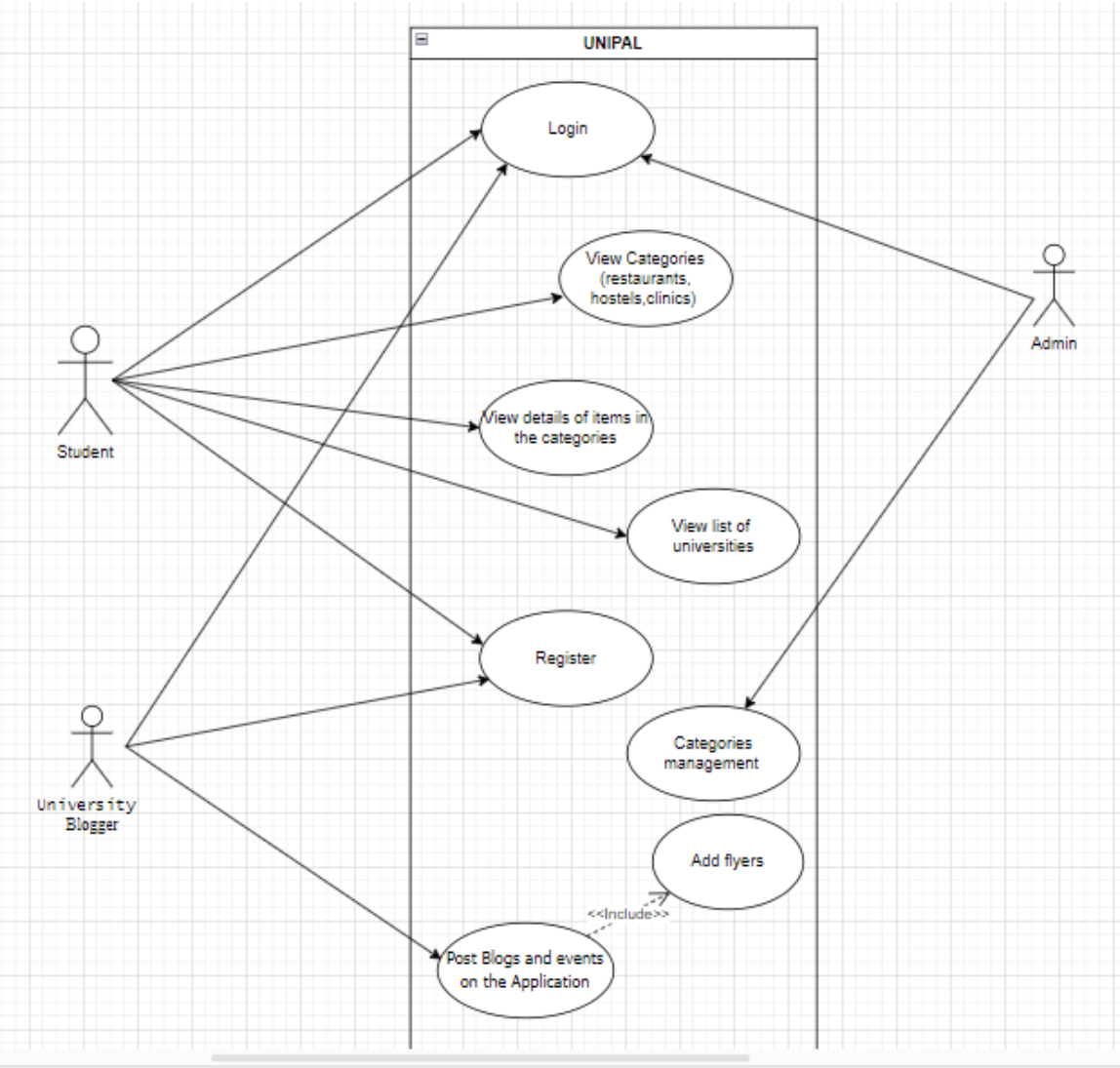
It indicates that a use case includes the functionality of another use case. It is denoted by a dashed arrow pointing from the including use case to the included use case. This relationship promotes modular and reusable design.
For example, In UniPal, when a university blogger is posting an event, he includes a flyer in it as well



Generalization

Generalization Relationship establishes an “is-a” connection between two use cases, indicating that one use case is a specialized version of another. It is represented by an arrow pointing from the specialized use case to the general use case.

Use Case Diagram of my Project, UniPal



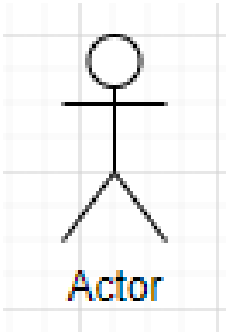
SEQUENCE DIAGRAMS

A sequence diagram is a diagram that visually represent the interactions between objects or components in a system over time. They focus on the order and timing of messages or events exchanged between different system elements. They are used to visualize dynamic behavior, designing system architecture, documenting system behavior and is also used in use case analysis

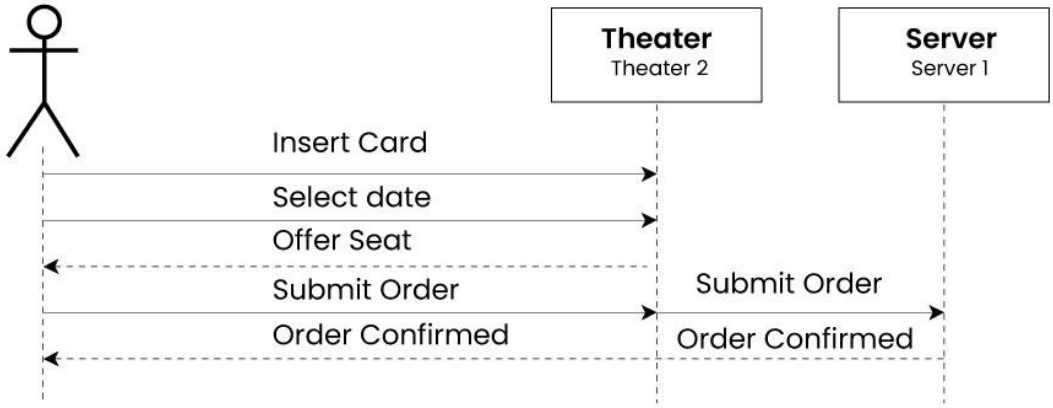
Sequence Diagrams Notations

(i) Actor

An actor in a UML diagram represents a type of role where it interacts with the system and its objects. An actor is always outside the scope of the system we aim to model using the UML diagram.



Example of usage of an actor in sequence diagrams



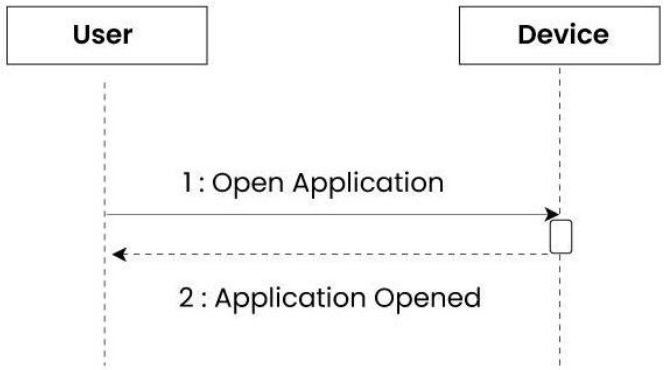
(ii) Lifelines

A lifeline is a named element which shows an individual participant in a sequence diagram. Each instance in a sequence diagram is represented by a lifeline. Lifelines are0 located at the top in a sequence diagram

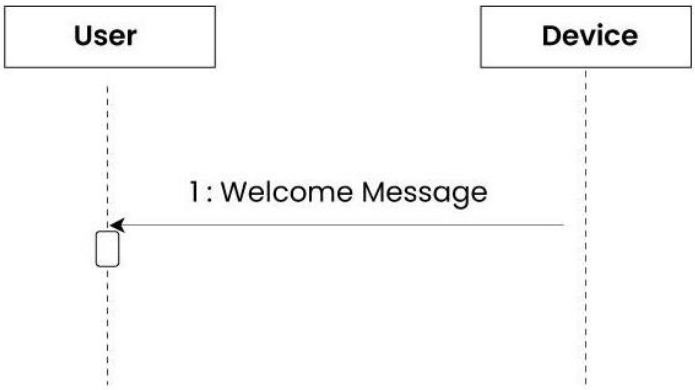
(iii) Messages

Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. Messages are in different categories as illustrated below with their notations

- **Synchronous Messages**
  A synchronous message waits for a reply before the interaction can move forward

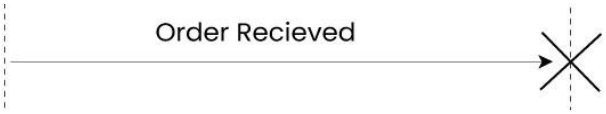


- **Asynchronous Messages**
  An asynchronous message does not wait for a reply from the receiver. Lined arrows head to represent an asynchronous message.



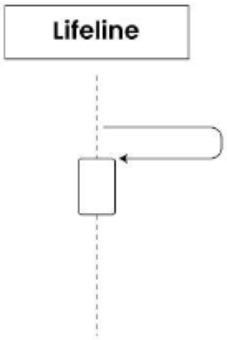
- **Create Message**
  It is used to instantiate a new object in the sequence diagram.

- **Delete Message**
  Delete Message is used to delete an object



- **Reply Message**
  Reply messages are used to show the message being sent from the receiver to the sender.

- Self Message
  This a message sent by an object to itself.



Sequence Diagram for UniPal App