**Use Case Diagram:** A Use Case diagram illustrates a set of use cases for a system,

**Purpose:**

* To represent the functional requirements of a system.
* To show interactions between users (actors) and the system.
* To visualize high-level system functionality.

**Components:**

1. **Actors:**
   * Represent users or other systems that interact with the system.
   * Can be primary (initiating interaction) or secondary (supporting interaction).
2. **Use Cases:**
   * Represent specific functionalities or tasks the system performs in response to an actor's request.
   * Typically shown as ovals.
3. **Relationships:**
   * **Association:** A line connecting an actor to a use case, indicating interaction.
   * **Include:** Indicates that a use case uses the functionality of another use case.
   * **Extend:** Indicates optional behavior that can extend the base use case under certain conditions.
   * **Generalization:** Shows hierarchy between actors or use cases.

**Notations:**

* Actors are depicted as stick figures.
* Use cases are depicted as ovals.
* Relationships are represented as lines with appropriate markers for include, extend, or generalization.

**Benefits:**

* Helps in understanding user requirements and system functionalities.
* Facilitates communication among stakeholders.
* Aids in identifying system boundaries and scope.

**Sequence Diagram:** The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur.

**Purpose:**

* To model the dynamic behavior of a system.
* To illustrate how objects interact in a particular sequence to accomplish a specific task.

**Components:**

1. **Objects:**
   * Represent entities that participate in the interaction (depicted as rectangles at the top of the diagram).
2. **Lifelines:**
   * Dashed vertical lines extending downward from each object, representing the object's presence over time.
3. **Messages:**
   * Horizontal arrows showing communication between objects.
   * Can represent synchronous messages, asynchronous messages, return messages, etc.
4. **Activation Bars:**
   * Rectangles on lifelines indicating the period an object is active or controlling the flow of the interaction.
5. **Conditions and Loops:**
   * Can depict conditional behavior (if-else) and loops (repeated messages) using frames.

**Notations:**

* Lifelines are dashed lines below the object rectangles.
* Arrows indicate messages sent between objects.
* Activation bars are vertical rectangles on lifelines.

**Benefits:**

* Provides a clear view of object interactions and their timing.
* Helps in understanding complex scenarios with multiple objects.
* Useful for detailing specific use cases.