Rules for Drawing UML Use Case Diagram

1. **Identify the System Boundary**:
   - Draw a **rectangle** to represent the system scope, enclosing all use cases and actors.
   - Label the rectangle with the system name to define what is being modeled.
2. **Define Actors**:
   - Represent actors (users or external systems) as **stick figures** outside the system boundary.
   - Label each actor with a clear, specific name reflecting their role (e.g., “Student”).
   - Ensure actors represent **roles**, not individuals.
3. **Identify Use Cases**:
   - Represent use cases (system functionalities) as **ovals** inside the system boundary.
   - Name each use case with a concise, action-oriented phrase starting with a verb (e.g., “View Timetable”).
   - Ensure use cases reflect **user goals**, not low-level implementation details.
4. **Connect Actors to Use Cases**:
   - Draw **solid lines** (associations) between actors and use cases to show interactions.
   - Ensure each association represents a meaningful interaction.
5. **Use Include Relationships**:
   - Use a **dashed arrow** labeled `<<include>>` from a base use case to a mandatory sub-use case (e.g., “Login” included in “View Timetable”).
   - Ensure the included use case is always performed as part of the base use case.
6. **Use Extend Relationships**:
   - Use a **dashed arrow** labeled `<<extend>>` from an optional or conditional use case to the base use case (e.g., “Receive Notification” extends “View Timetable”).
   - Specify the condition for the extension in a note.
7. **Use Generalization (Optional)**:
   - Represent shared behavior with a **solid arrow with a hollow triangle** pointing to the parent actor or use case.

- Use sparingly to avoid complexity (e.g., “Faculty Member” generalizes to “Lecturer”).

8. **Avoid Overcomplication**:
    - Focus on **high-level user requirements**, excluding internal system processes or technical details.
    - Limit use cases to significant functionalities for clarity.
9. **Ensure Clarity and Consistency**:
    - Use clear, unambiguous names aligned with stakeholder terminology.
    - Avoid crossing lines; use layout tools for organization.
10. **Maintain Simplicity for Stakeholders**:
    - Design the diagram to be understandable by non-technical stakeholders.
    - Focus on external behavior, avoiding UML-specific jargon.Additional Guidelines