UML:

* Unified Modeling Language is way of represent things in design diagram
* UML makes understanding the feature easier and easy to implement
* It is a graphical way of representing relationship between classes

There are 2 types of UML:

* Structural (concern on the structure of the code)
  + **Class Diagram (imp)**
  + Component Diagram
  + Package Diagram
  + Object Diagram
* Behavioral (concern on how the system works and interacts)
  + **Sequential Diagram (imp)**
  + **Use Case Diagram**
  + Activity Diagram (imp)

Important UML Diagrams in LLD:

**Use Case Diagram:**

* Visual representation of how users interact with the system
* Helpful for understanding functional requirement in user perspective

Notations:

* Actor:
  + External Entities who interacts with the system. They initiates the usecases
* Use Case:
  + These are the actual functionalities of the system
* System Boundary:
  + Limits of the system

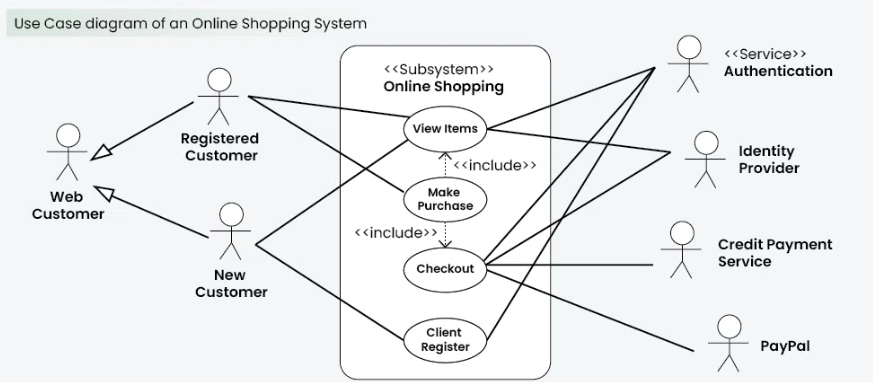
Association Relationships:

* The Association Relationship represents a communication or interaction between an actor and a use case.

Steps to Draw a usecase diagram:

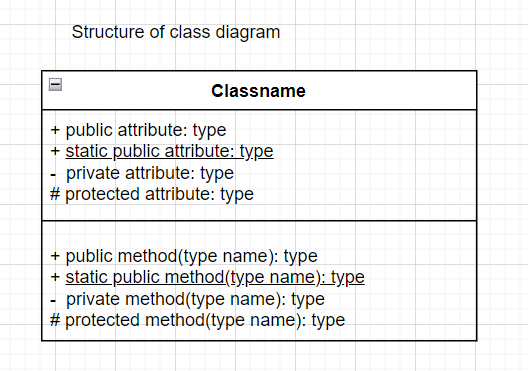
1. Identify Actors
2. Identify Use Cases
3. Connect Actors and Use cases
4. Add System Boundary
5. Define Relationships
6. Review and Refine
7. Validate

Example Online Shopping System



**Class Diagram:**

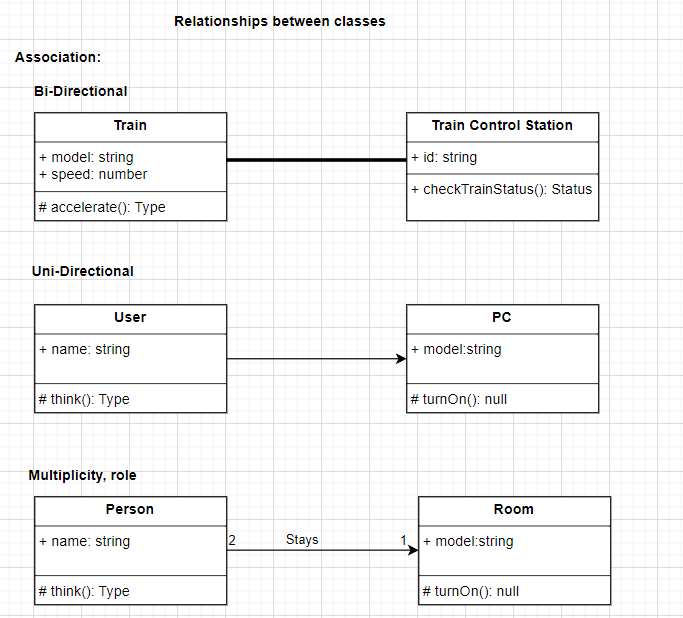
* These diagrams visually represent the structure and relationships of classes in a system.
* Class contains the attributes and methods
* There are access levels for attributes and methods.
  + **Public:** represented with “+”
  + **Private:** represented with “- “
  + **Protected:** represented with “#”
* Static method and properties are represented by underling the attribute/method



**Relationship between classes:**

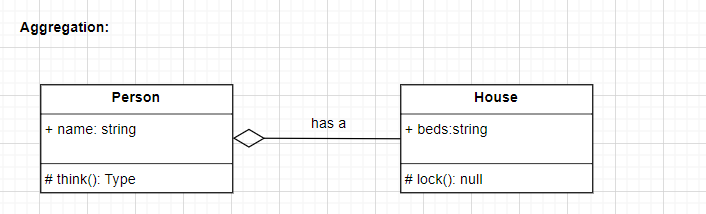
**Association Relationship:**

1. Two classes can called each other.
2. There are different types:
   1. Bi-Directional
   2. Uni-Directional
   3. Multiplicity, Role (represent the no of one class members can access other class)



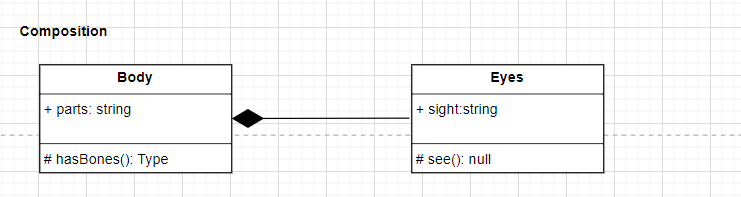
**Aggregation Relationship:**

1. Wherever you see a relationship like Class A **has an relationship** with Class B where B can exist without A
   1. Eg: vehicle has a tyre



**Composition Relationship:**

1. Wherever you see a relationship like Class A **has an relationship(part of)** with Class B where B cannot exist without A
   1. Eg: body has an eye



**Inheritance Relationships:**

1. Wherever you see a relationship like Class A **is a** Class B
   1. Eg: Employee is a Person

