Design Doc

UML Diagrams

What is UML?

- ➔ Visual modeling language used to specify, visualize, construct and document software systems
- ➔ Pictorial language used for making software blueprints
- ➔ Not only made for developers but also business users, common people and anybody interested to understand the system
- ➔ Not a development method, but an accompanying process that helps to build a successful system

UML Diagrams

Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to **visualize** the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

You would need 3 diagrams in your Design Doc-

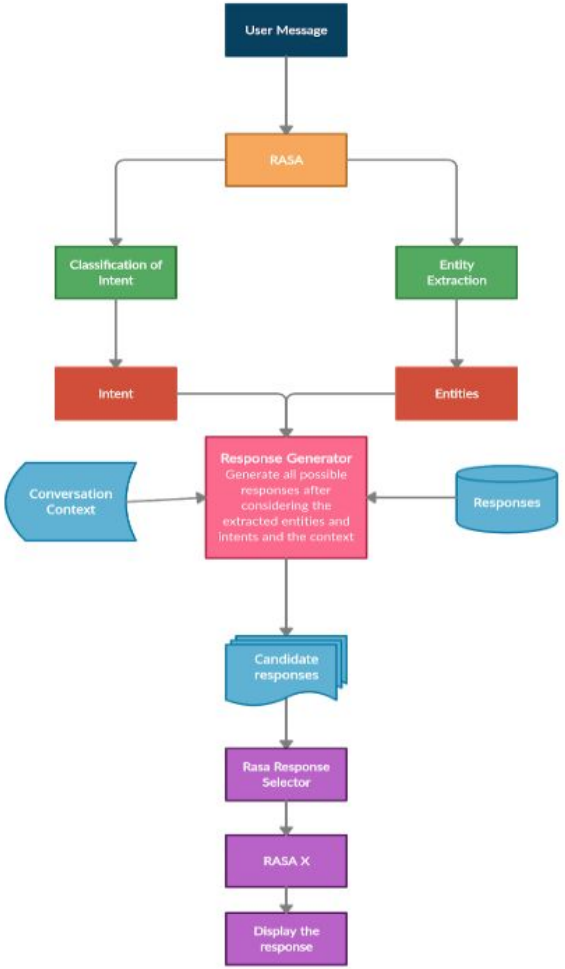
- Architectural Diagram
- UML Class Diagram
- **UML Sequence Diagram for multiple use cases**

UML and Object Oriented Design

- ➔ UML has a direct relation with Object Oriented Analysis and Design
- ➔ OO concepts - objects, class, abstraction, encapsulation, inheritance, polymorphism
- ➔ UML diagrams are a representation of these OO concepts
- ➔ OO analysis and design steps
  - ◆ OO Analysis - Identify objects and their responsibilities / functions
  - ◆ OO Design - Identify their relationships / association based on requirements
  - ◆ OO Implementation - Convert to executables using OO languages
- ➔ UML is used in the Design phase

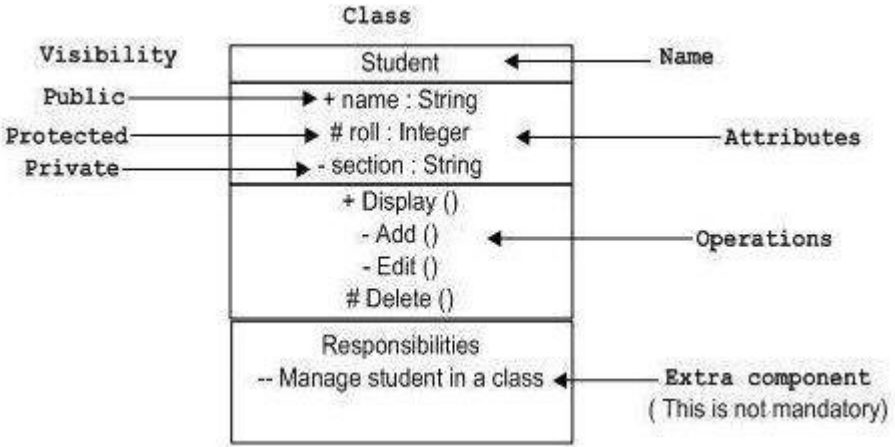
Architectural Diagram

High level block diagram of different subsystems.



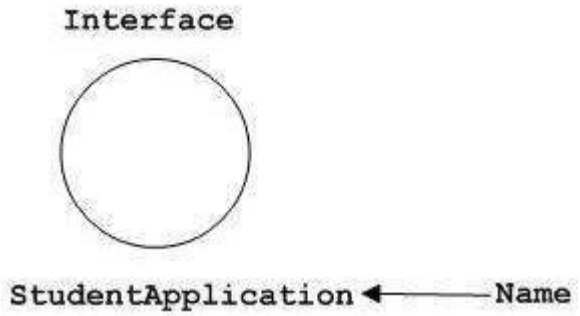
UML Notation

Class - used to represent objects (not the same as software classes)



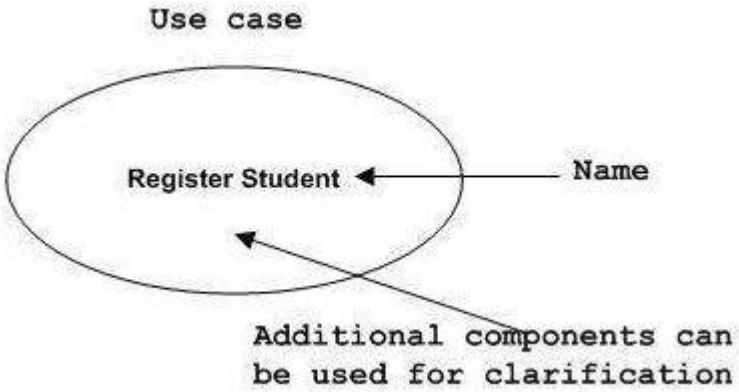
UML Notation

Interface - used to describe functionality (without implementation)



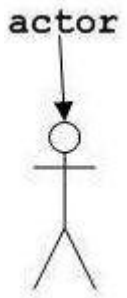
UML Notation

Use Case - used to capture high level functionalities of a system



UML Notation

Actor - used in use case diagrams to describe internal or external entities

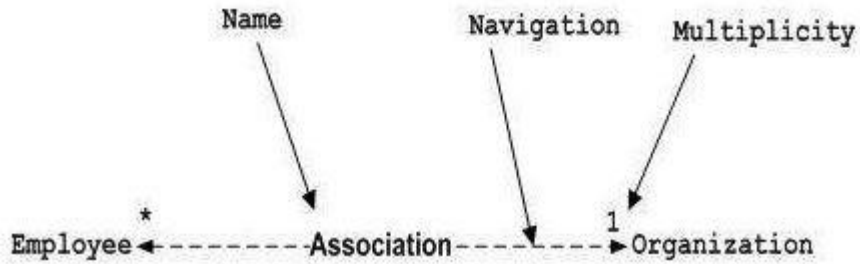


UML Notation

Dependency - used to represent the dependency between two elements

UML Notation

Association - describes how many elements in a UML diagram are associated



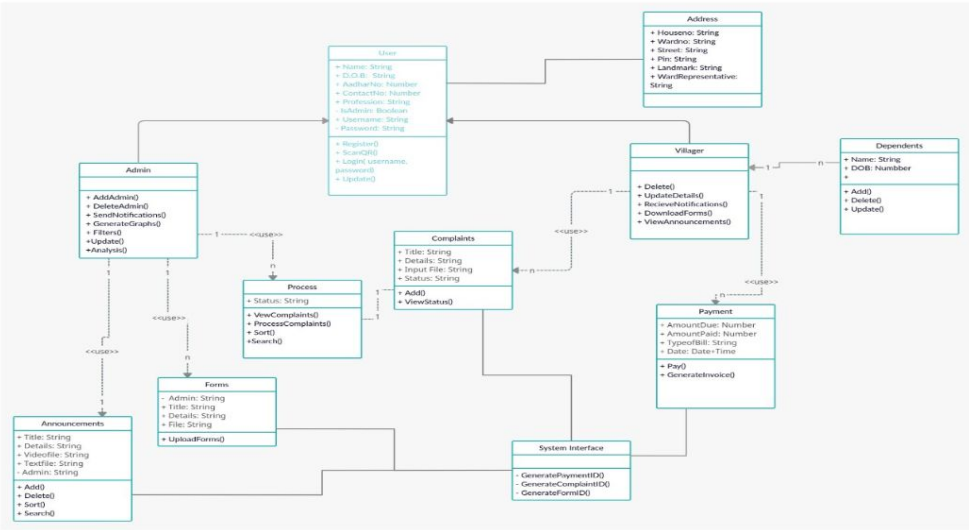
UML Notation

Generalization - describes inheritance relationship (parent-child relationship)

UML Class diagram

- The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.
- Identify the classes, for each class -

<Class No. 1>	eClass state • What information is the class responsible for maintaining? • E.g., a printing subsystem might hold the current status of all the printers it controls as well as the queue of print jobs waiting to be printed. Class behavior • What methods does the class implement? • E.g., classes related to the printing subsystem might support the queuing up of new jobs, estimating the time until a given job completes, or emailing status information at the end of a job.
<Class No. 2>	Class state • What information is the class responsible for maintaining? Class behavior • What methods does the class implement?
<Class No. 3> add more rows as needed.	Class state • Etc. Class behavior • Etc.



Class Diagrams

➔ Graphical representation of the static view of the system

➔ Collection of classes, interfaces, associations, collaboration, constraints

➔ Things to keep in mind -

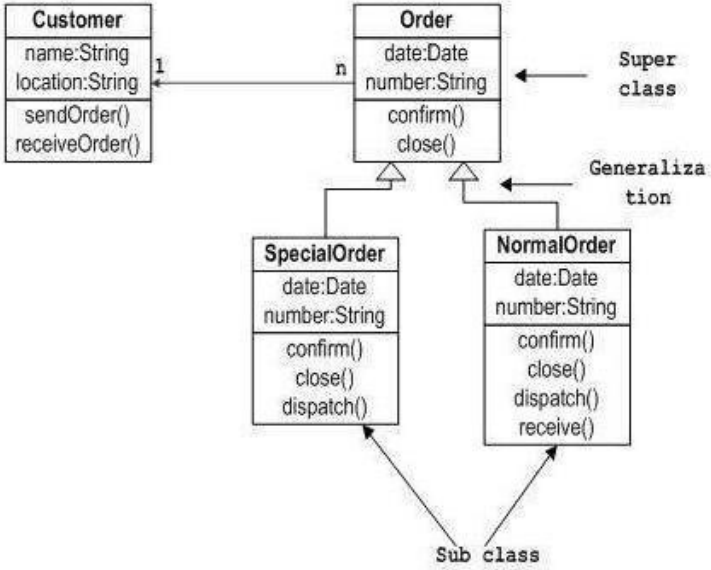
- Identify elements and their relationships
- Identify the attributes and methods of each class
- Use minimum number properties to avoid complication
- Use meaningful names
- Multiple iterations will make it perfect

Class Diagrams - Example

Basic ordering system where a customer can place an order on the application.

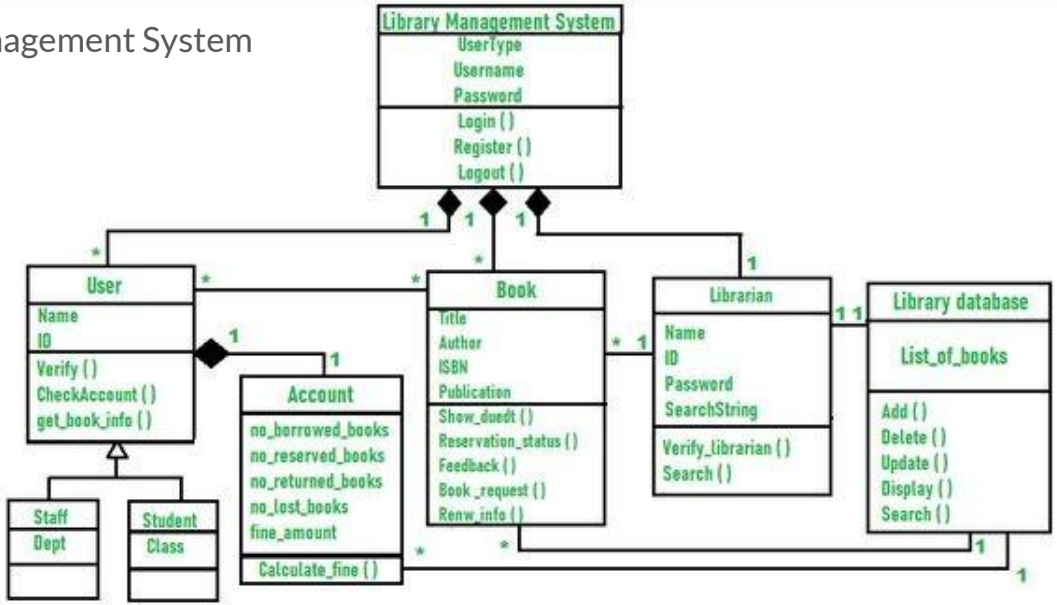
- Elements - order and customer
- One-to-many association relationship since a customer can have multiple orders
- Order class is an abstract class and has two concrete (inherited) classes - SpecialOrder and NormalOrder
- The inherited classes have all properties of the parent class and a few additional functions

Class Diagrams - Example



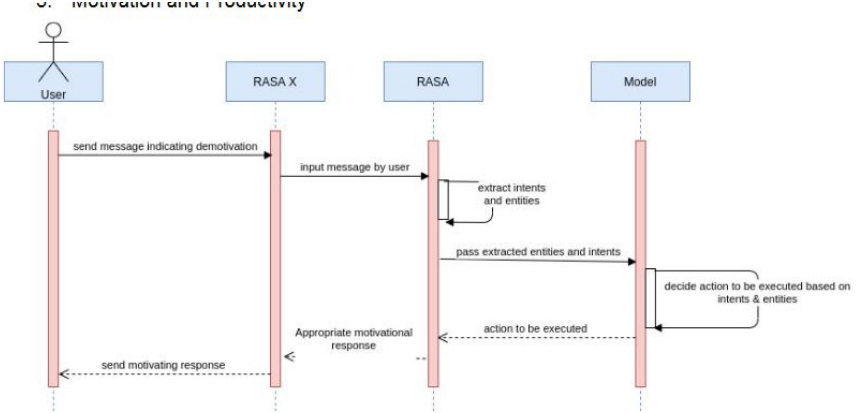
Class Diagrams - Another Example

Library Management System



UML Sequence diagram

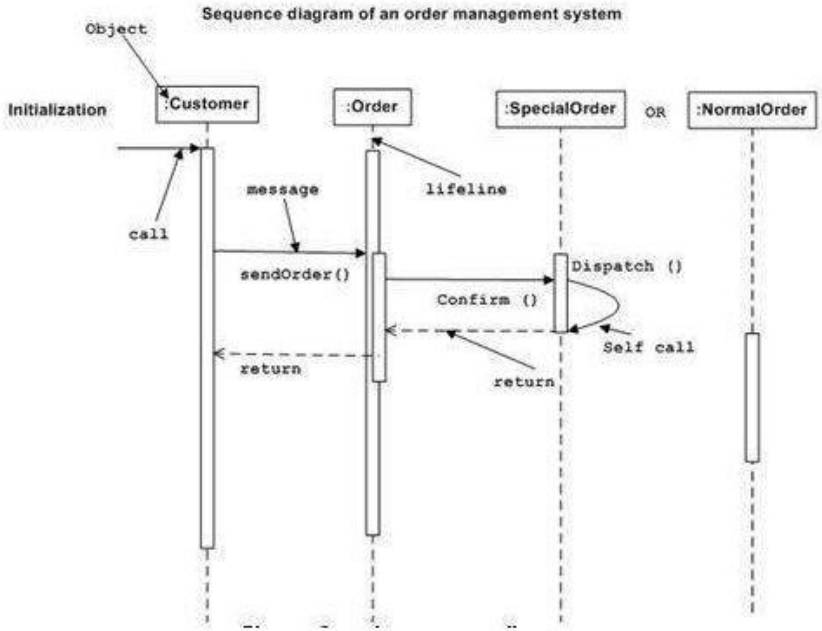
- A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.
- Need to draw atleast 4 diagrams for 4 different use cases.



Sequence/Interaction Diagrams

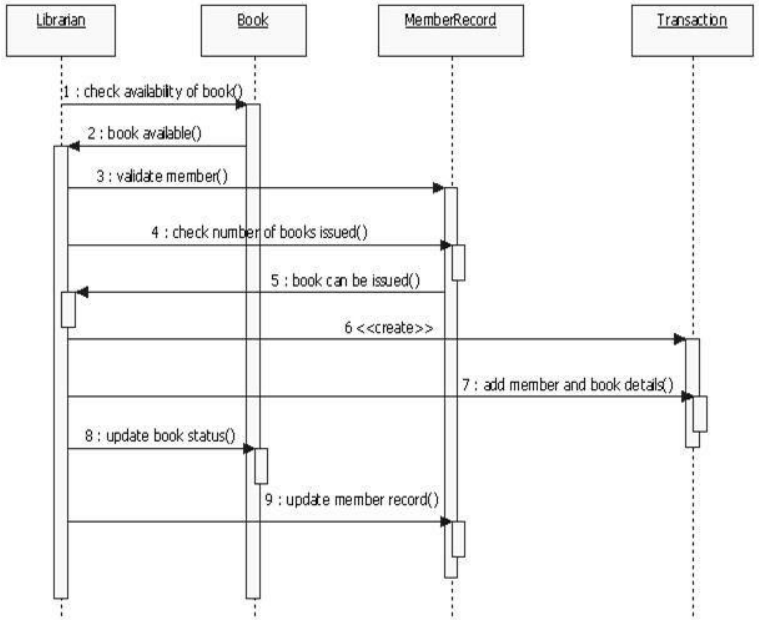
- ➔ Captures the dynamic nature of the system
- ➔ Visualize the interactions in the system
- ➔ Things to keep in mind -
  - ◆ Identify objects taking part in the interactions
  - ◆ Identify message flows among the objects
  - ◆ Identify the sequence in which messages are flowing

Sequence Diagrams - Example



Sequence Diagrams - Another Example

Library Management System



Use Case Diagrams

- Graphical representation of the dynamic view of the system
- Used to gather requirements of a system including internal and external influences
- Functionalities, actors and relationships have to be identified
- Things to keep in mind -
  - Provide appropriate names so that the functionality can be identified
  - Give a suitable name to actors
  - Show relationships and dependencies clearly but do not try to include all types of relationships. The main purpose of the diagram is to identify requirements

Use Case Diagrams - Example

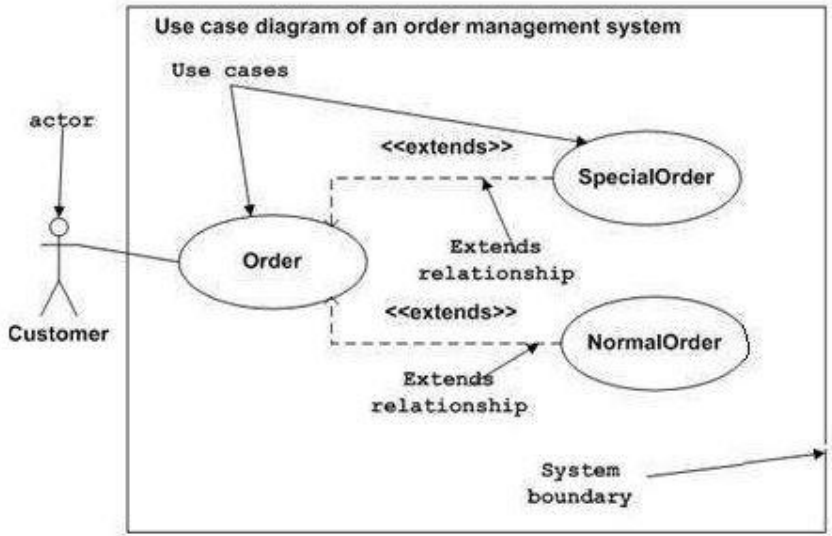
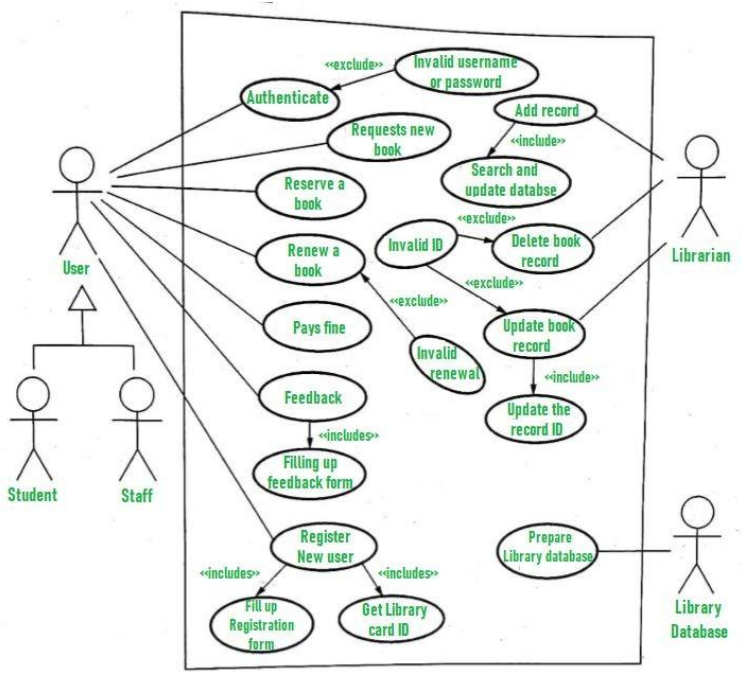


Figure: Sample Use Case diagram

Use Case Diagrams - Another Example

Library Management System



Reference Diagrams

- https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/
- https://www.tutorialspoint.com/uml/uml_class_diagram.htm